INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

RECONOCIMIENTO DE EXPRESIONES FACIALES

TESIS PRESENTADA POR ANDRÉS GERARDO HERNÁNDEZ MATAMOROS PARA OBTENER EL GRADO DE DOCTOR EN COMUNICACIONES Y ELECTRONICA

DIRECTOR DE TESIS

DR. HÉCTOR MANUEL PÉREZ MEANA

DICIEMBRE 2017

Sección de Estudios de Posgrado e Investigación

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 10:30 horas del día 18 del mes de diciembre del 2017 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI-ESIME-CULHUACAN para examinar la tesis titulada:

"Reconocimiento de Expresiones Faciales"

Presentada por el alumno:

Hernández	Matamoros	Andrés Gerardo
Apellido paterno	Apellido materno	Nombre(s)

Con registro: A 1 4 0 8 9 3

aspirante de:

DOCTORADO EN COMUNICACIONES Y ELECTRONICA

Después de intercambiar opiniones, los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesis

Dr. Héctor Manuel Pérez Meana

Dr. Volodymyr Ponomaryov

Dra. Mariko Nakano

S.E.P.
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION
ESIME CULHUACAN

Dr. Gabriel Sánchez Pérez

Dr. Juan Carlos Sánchez García

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Héctor Manuel Pérez Meana

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de Mexico el día 1 del mes Diciembre del año 2017, el que suscribe Andrés Gerardo Hernández Matamors alumno del Programa de Doctorado en Comunicaciones Y Electrónica con número de registro A140893, adscrito a SEPI-ESIME-CULH, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de Dr Héctor Manuel Pérez Meana y cede los derechos del trabajo intitulado Reconocimiento de Expresiones Faciales, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección matamoros.leo@icloud.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Andrés Gerardo Hernández Matamors

Nombre y firma

Agradecimientos

A mis padres por haberme hecho como la persona que soy, muchos de mis logros se los debo a ustedes.

A mi familia por todo el apoyo recibido a lo largo de mi vida.

Al Dr. Héctor Pérez por su gran motivación, dirección y asesoría.

A todos los miembros del Programa JUSST, por haberme brindado la oportunidad de participar en este programa.

Al IPN, particularmente a todos los que laboran en la SEPI de ESIME Culhuacán..

A CONACyT por el apoyo que me brindo para realizar mis estudios de Posgrado.

Resumen

Uno de los temas de investigación mas activos en la ultima década, gracias al poder de procesamiento con el que cuentan las computadoras actualmente, es la visión artificial mas específicamente la capacidad con la que las computadoras nos miran, es decir que ellas sean capaces a través del procesamiento digital de imágenes y el reconocimiento de patrones de predecir o saber nuestro estado de animo, esto nos lleva a realizar el análisis de un rostro que es una de los temas de investigación mas recurrentes en los últimos años con aplicaciones como el reconocimiento de personas, interacción hombre-maquina, video vigilancia, telemedicina, etc. En esta tesis se aborda 3 grandes problemas del reconocimiento de expresiones faciales el primero de ellos es la detección del perfil del rostro, por otra parte tenemos la extracción automática de las regiones de interés del rostro y por ultimo pero no menos importante es el costo computacional del clasificador que será el encargado de aprender y evaluar de manera adecuada las expresiones faciales. Estas dos problemáticas son estudiadas en detalle, para la detección de perfil y la extracción automática se proponen algoritmos basados en las integrales proyectivas, erosión y dilatación de imágenes, para el costo computacional se propone un clasificador basado en clustering y lógica difusa, con esto no solo se sabrá a que clase pertenece una expresión, también será posible saber su grado de pertenencia con cada clase. Los algoritmos propuestos para los diferentes problemas alcanzan siempre resultados iguales o superiores a los reportados en la literatura, siendo comparables con los que constituyen el estado del arte, pero con una mejora muy significativa de la eficiencia computacional.

.

Abstract

One of the most active research topics in the last decade, thanks to the processing power with which computers currently count, is artificial vision, more specifically the ability with which computers look at us, that is to say that they are capable through of the digital processing of images and the recognition of patterns of predicting or knowing our state of mind, this leads us to perform the analysis of a face that is one of the most recurrent research topics in recent years with applications such as recognition of people , man-machine interaction, video surveillance, telemedicine, etc. This thesis addresses 3 major problems of recognition of facial expressions. The first problem is the detection of the face profile, on the other hand, we have the automatic extraction of the regions of interest of the face and, last but not least, the computational cost. of the classifier that will be in charge of learning and evaluating facial expressions appropriately. These two problems are studied in details, for the detection of profile and automatic extraction algorithms are proposed based on the projective integrals, erosion and dilatation of images, to reduce the computational cost, a classifier base don clustering and fuzyy logic is proposed, this classifier gives what class belongs an expression belongs, it will also be possible to know your degree of belonging with each class. The proposed algorithms for different problems always achieve results equal to or higher than those reported in the literature, being comparable with method's state of the art, but with a very significant improvement in computational efficiency.

Índice general

Tabla de contenido

Índice de Figuras

Índice de Tablas

Planteamiento del problema

El reconocimiento de patrones es un tema que ha sido abordado en la literatura a través de los años, empezando por señales unidimensionales, hasta llegar a tener imágenes en 2 y 3 dimensiones. En esta tesis se aborda el tema del reconocimiento de expresiones faciales a través de una imagen en 2 dimensiones, se propone esto debido a que en la mayoría de sistemas de video vigilancia o dispositivos electrónicos cuentan con este tipo de cámaras.

Los sistemas de reconocimiento de expresiones faciales mediante visión artificial han enfrentado algunos obstáculos. Uno de las mayores dificultades proviene del hecho de que la cara es un objeto tridimensional y en el caso de identificación basada en imágenes se reduce a solo dos dimensiones, esto representa perdida de información al usar cámaras normales, además los estudios tienen el inconveniente de que dichas imágenes del rostro al ser adquiridas por la cámara pueden sufrir rotación, traslación, oclusión parcial o total, variación de luminosidad, cambio de fondo, etc.

Por otra parte se tiene una dificultad mas que radica en que es necesario recortar el rostro del sujeto de manera precisa, así como las regiones de interés del mismo, para evitar información no necesaria para el sistema de reconocimiento de expresiones faciales, esto es porque al hacer un gesto felicidad o de tristeza el cabello no se mueve o representa algún cambio importante a diferencia de la boca o los ojos que a través de estudios se han determinado como las regiones importantes al momento de realizar diferentes gestos con el rostro.

Un caso especial en el reconocimiento de expresiones es el caso de las secuencias de vídeo donde no todo el tiempo se proporciona una toma del rostro de manera frontal, por esta razón es primero lograr identificar cuando un rostro se cuando de frente para así después poder realizar el análisis de necesario para determinar la expresión facial de la persona.

Previamente hemos hablado de las regiones de interés del rostro para lograr el reconocimiento de expresiones faciales, pero no es todo el problema que se tiene al hacer el reconocimiento, una parte muy importante del reconocimiento de patrones, es la parte del clasificador , para que esta funcione de manera adecuada necesita recibir vectores que describan de manera adecuada lo que se desea clasificar, en algunas ocasiones esto no siempre es posible y dichos vectores no pueden ser separados de forma lineal o con algunas técnicas que ya existen en la literatura, como las redes neuronales artificiales o las maquinas de soporte vectorial, por esta razón se aborda en esta tesis el desarrollo de un clasificador basado en técnicas de clustering y lógica difusa.

Justificación

En la actualidad dentro de la literatura existen varios sistemas de reconocimiento de expresiones faciales, la mayoría de los algoritmos desarrollados para estos sistemas son fundados con imágenes en 2 esto es porque son las cámaras mas comerciales, desafortunadamente estos sistemas están limitados por los cambios de rotación y traslación del rostro, así como de los cambios en la intensidad de la iluminación, las limitaciones anteriormente descritas son algunas de las dificultades que se pretenden resolver en esta tesis, haciendo énfasis en el problema de la rotación del rostro, extracción automática de las regiones de interés y el costo computacional del clasificador.

Recientemente, para superar estos desafíos se han realizado estudios de que regiones del rostro son las que se mueven con distintas expresiones como lo es alegría, tristeza, ira etc., dichos estudios ayudaran al sistema a ser más robusto frente a variaciones del rostro debido a los diferentes factores antes mencionados. Con sistemas mas robustos es posible salirse de condiciones controladas para aplicar los sistemas de reconocimiento de expresiones faciales en condiciones no controladas, esto se refiere a no siempre tener un fondo fijo, variaciones en la iluminación y diferentes ángulos del rostro. El tener diferentes ángulos del rostro es un problema muy común en ambientes descontrolados ya que no siempre se puede tener de frente a la persona viendo a la cámara.

Objetivo General

El objetivo principal es desarrollar un sistema de reconocimiento de expresiones faciales, que sea capaz de reconocer el perfil de una persona, segmentar de manera automática las regiones de interés y finalmente dar los porcentajes de parentesco con cada una de las expresiones faciales con las que cuenta este sistema

Objetivos Particulares

- Desarrollar un sistema que permita reconocer el perfil de un rostro en una imagen, esto es si la persona esta viendo de frente a la cámara o si se encuentra girada hacia algún lado.

- Desarrollar un sistema que realice la segmentación de manera automática de las regiones de interés del rostro, estas regiones de interés son la boca y la frente/ojos.

- Desarrollar un clasificador basado en técnicas de clustering y lógica difusa, para reducir los tiempos de entrenamiento del sistema.

- Aplicar el sistema desarrollado en una base de datos de videos con ambiente no controlado.

Introducción

En enero de 2000, Oliver, Rosario y Pentland, investigadores del Instituto Tecnológico de Massachusetts perteneciente al laboratorio "Media Lab", publicaron un artículo en el que hablan sobre una emergente rama de la visión computacional, a la que llama como "mirando a la gente" (en ingles, *looking a-people*) [1]. Este documento habla de la trayectoria de dicha área y de su perspectiva a futuro, anticipando tecnologías, aplicaciones y problemas. Hoy en día las computadoras son básicas en el día a día del ser humano y conforme el mundo se vuelve más complejo, debemos de aprender a utilizar nuestros recursos con mayor eficiencia, sin duda para lograr esto necesitamos de la ayuda que nos pueden ofrecer las computadoras.

Desde los inicios de las computadoras el hombre ha buscado con ayuda de estas tener una vida más cómoda, siempre intentando tener lo mejor al menor esfuerzo posible, dejando a estas a cargo de tareas monótonas y simples hasta tareas complejas, es por ello que día a día, en todo el mundo, se están realizando múltiples esfuerzos de universidades y empresas, para poder desarrollar sistemas inteligentes donde la interacción entre el ser humano y las computadoras sea mas natural, para esto necesitamos que las computadoras "miren a la gente".

Por esta situación esta investigación está centrada en el *reconocimiento de expresiones faciales*. Abordando 3 de los principales problemas que presentan el reconocimiento de expresiones faciales, así como una aplicación en video no controlado, esta tesis esta organizada de la siguiente manera: Capítulo 1 se describen los antecedentes del reconocimiento de expresiones faciales y el estado del arte, Por su parte en el capítulo 2 mostramos las 3 principales contribuciones de esta tesis, para en el capítulo 3 mostrar 2 sistemas de reconocimiento de expresiones faciales donde se utiliza las aportaciones realizadas en estas tesis con el objetivo de compararse con otros autores.

Capítulo 1 Antecedentes

Desde el inicio de la computación se ha buscado que las computadoras nos ayuden a realizar diferentes tareas, que sean capaces de interactuar esto nos llevado a forjar un termino que hoy en día es muy usado "Reconocimiento de patrones"[2], esta es la ciencia que se encarga de la descripción y clasificación (reconocimiento) de objetos, personas, señales, representaciones, etc. Esta ciencia trabaja con base en un conjunto previamente establecido de todos los posibles objetos (patrones) individuales a reconocer. El margen de aplicaciones del reconocimiento de patrones es muy amplio, sin embargo las más importantes están relacionadas con la visión y audición por parte de una computadora, de forma análoga a los seres humanos. El esquema de un sistema de reconocimiento de patrones consta de varias etapas relacionadas entre sí (los resultados de una etapa pueden modificar los parámetros de etapas anteriores). La siguiente figura muestra un esquema general de un sistema de reconocimiento de patrones, en el cual el sensor tiene como propósito proporcionar una representación factible de los elementos del universo a ser clasificados. Es un sub-sistema muy importante ya que determina los límites en el rendimiento de todo el sistema. La Extracción de Características es la etapa que se encarga, a partir del patrón de representación, de extraer la información discriminatoria eliminando la información redundante e irrelevante. El Clasificador es la etapa de toma de decisiones en el sistema. Su rol es asignar los patrones de clase desconocida a la categoría apropiada.

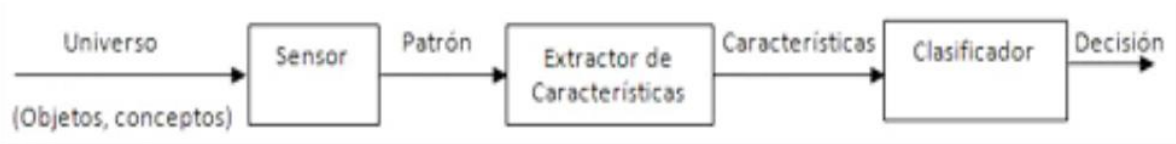


FIGURA 1.1 ESQUEMA GENERAL DE UN SISTEMA DE RECONOCIMIENTO DE PATRONES.

El objetivo de estas etapas es ajustar el sistema para que sea capaz de clasificar señales u objetos de entrada en una de las clases predefinidas. Para ello deberá analizar un cierto número de características y para poder clasificar satisfactoriamente señales de entrada, es necesario un proceso de aprendizaje en el cual el sistema crea un modelo de cada una de las clases a partir de una secuencia de entrenamiento o conjunto de vectores de características de cada una de las clases. El sistema de reconocimiento de patrones debe tener en cuenta las fuentes de variabilidad como son el ruido, rotaciones, cambio de escala y deformaciones, lo cual se logra incluyendo en la secuencia de entrenamiento patrones que hayan experimentado estas modificaciones.

Como se puede observar, el reconocimiento de patrones es la base teórica más importante para la realización de un sistema de reconocimiento de expresiones faciales, retomando la figura 1.1, para nuestro sistema de reconocimiento de expresiones faciales;

- El Universo correspondería a la expresiones faciales por clasificar, el sistema propuesto en esta tesis consta de 7 expresiones faciales (Miedo, asco, tristeza, enojó, felicidad, neutral, serio y sorpresa).
- El sensor seria la cámara de la cual se obtuvieron las fotografías o el video de las personas.

- En el Extractor de características existen en la literatura diferentes métodos como son Filtros de Gabor, Transformada Wavelet, Descriptores Locales de Weber (WLD), etc. Es muy importante señalar que estos métodos de extracción de características se han aplicado en el rostro tanto de manera global (toda la imagen) como de manera local (ciertas regiones del rostro).
- En este punto es importante que en algunas ocasiones es necesario aplicar reducción dimensional, algunos de los algoritmos mas clásicos son el análisis de componentes principales (PCA por sus siglas en ingles), LDA(linear discriminant Analysis), etc.
- En la parte del clasificador, esta es la parte encargada del sistema de aprender y tomar una decisión sobre la pertenencia de un patrón a cierta clase; dentro de la literatura existen diferentes clasificadores como lo son las redes neuronales artificiales y las maquinas de soporte vectorial, desafortunadamente estos clasificadores representan un costo computacional elevado.

La investigación de las expresiones faciales ha sido un tema recurrente, por esta razón la última década se han realizado investigaciones sobre este tema [3]-[8]. Actualmente ya existen algoritmos capaces de distinguir rostros en una imagen y hasta sonrisas, ojos, nariz etc. como es el caso de los algoritmos con los que cuenta OpenCv[9]-[11] basados en Viola-Jones[12], desafortunadamente estos no son suficientes para poder detectar la expresión facial de una persona y con esto lograr saber su estado de ánimo, esto sucede porque al realizar una expresión facial alguna persona puede realizar una mueca y otra no, aunque existen algunos movimientos que se realizan de manera involuntaria cuando realizamos una expresión facial, en [13] es posible observar que músculos del rostro son lo que se mueven a cada expresión facial gracias a esto es posible determinar las regiones de interés del rostro. Otro problema que se presenta al reconocer una expresión facial es si el rostro no está viendo de frente a la cámara, alguna oclusión parcial del rostro o problemas de luminiscencia. Por estas razones, es que propone en esta tesis el detector de perfil de un rostro, la segmentación automática de las regiones de interés y un clasificador con menos costo computacional basados en clustering y lógica difusa

La cantidad de información que los seres humanos podemos extraer de mirar a simple vista, en una imagen donde vemos a una persona realizando una expresión facial es sorprendente, como ejemplo realizaremos un pequeño experimento mostrado en la siguiente figura

Test de Habilidad de percepción Facial

FIGURA 1.2 EXPERIMENTO DE PERCEPCION HUMANA EN UN ROSTRO.

☐ Aparece alguna cara humana en la imagen de la izquierda. En caso afirmativo, señale el centro de los ojos y la boca y elija la mejor respuesta para cada una de las caras:
Sexo: hombre/mujer
Edad aprox.: bebé, niño, joven, adulto, anciano
Raza: caucásico, negro, asiático, hindú, otra
Giro cabeza: izq., der., arriba, abajo, frontal
Mirada hacia: izq., der., arriba, abajo, frontal
Ojos y boca: abiertos, cerrados, entreabiertos
Expresión: neutra, triste, alegre, sorpresa, asco
Tiene: gafas, bigote, barba, pecas, sombrero.
Nombre de la Persona:

Sin duda alguna el lector fue capaz de diferenciar cada una de las cuestiones que le ha preguntado en el experimento anterior, esto es porque al ver una imagen de manera inmediata: contamos el numero de personas presentes,

además de su localización ; es decir, para cada uno, sabemos si se trata de un hombre o de una mujer, su raza y, posiblemente, su grupo étnico especifico; diferenciar si es una persona anciana, adulta, joven o niño, si usa gafas, o lleva sombrero, si tiene barba o bigote, si es calvo, si es rubio o moreno, si tiene pecas, si lleva tatuajes, etc.; por otra parte somos capaces de mencionar la orientación del rostro respecto a la cámara, hacia donde esta viendo, si es una figura publica hasta el nombre podemos recordar. Lo que respecta a las expresiones faciales se han descrito a través de "Unidades de Activación", asociadas a los movimientos de los músculos al realizar cada una de ellas, con este estudio se desarrolla una herramienta llamada "ARTNATOMY"[13] que es una herramienta didáctica diseñada y construida para facilitar, en el ámbito general de los estudios de Bellas Artes, la enseñanza y el aprendizaje de los fundamentos anatómicos y biomecánicos que configuran la morfología de la expresión facial a la hora de representar emociones.

FIGURA 1.2 SOFTWARE ARTANATOMY.

Con ayuda del software mostrado en la figura anterior es posible distinguir a cada expresión facial de las que nos interesan analizar en esta tesis cuales son los movimientos que realizan los músculos como se muestra en la siguiente figura.

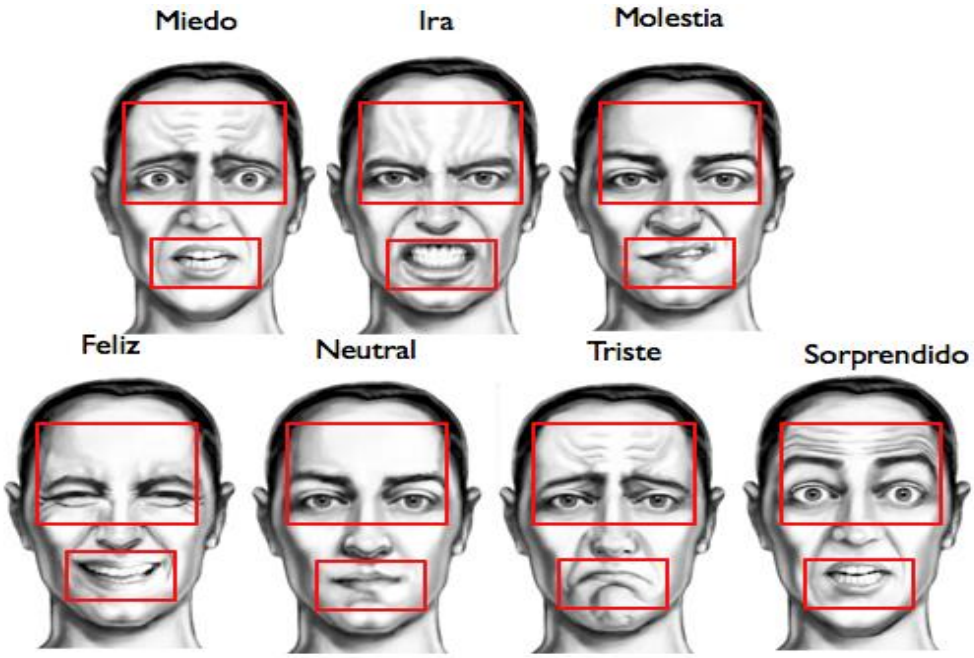


FIGURA 1.3 DIFERENTES MOVIEMIENTOS DE LOS MUSCULOS DEL ROSTRO CON DIFERENTES EXPRESIONES FACIALES.

El resultado del analizar las expresiones faciales nos permite inferir información sobre el estado de animo de la persona, con esto es posible determinar si esta feliz, triste, indiferente, si esta actuando o la expresión es natural. Y todo esto por no hablar de las sensaciones o impresiones subjetivas que una cara provoca en el observador: simpatía, antipatía, enfado, lastima, atracción, etc. Es importante mencionar que para la realización de un sistema de reconocimiento de expresiones faciales se tienen 2 partes importantes como lo muestra la figura 1.1, la parte del sensor y extractor de características en este caso del rostro y la parte del clasificador, por lo que este capitulo lo dividiremos en los siguientes subcapítulos el primero de ellos el 2.1 da pie a poder realizar el análisis de expresiones faciales, siendo el primer paso la detección de un rostro en una imagen y el subcapítulo 2.2 referido a los clasificadores mas usados en la literatura actual.

1.1 Detectores de rostro

Como sucede en todos los ámbitos de la computación donde se intenta reproducir las capacidades que tenemos los seres humanos, nos encontramos con muchas dificultades, algunos algoritmos en el área del reconocimiento de rostros se mencionan a continuación.

- Uno de los detectores de rostros mas populares, el algoritmo de Viola Jones, se explica en la subsección 2.1.1 ya que fue uno de los algoritmos usados para la realización de esta tesis.
- Otro detector clásico esta basado en búsqueda exhaustiva multisecular, ecualización de histogramas y redes neuronales [14].
- El algoritmo de CamShift[15] que se basa en el proceso de seguimiento por color.

Los algoritmos descritos anteriormente llegan a presentar fallos, esta es una de los grandes problemas que se encuentra en el tema de visión artificial y reconocimiento de patrones, en algunas ocasiones se debe a el sobreajuste a los datos de entrenamiento, la escasa capacidad de generalización de los métodos de aprendizaje usados, etc. A pesar de los obstáculos antes mencionados, el beneficio que supondría disponer de técnicas automáticas capaces analizar y trabajar con rostros humanos motiva sobradamente la investigación en este ámbito. Muchos avances en los problemas de rostros pueden ser extendidos, de hecho ya lo han sido en el pasado [16, 17] a otros dominios.

1.1.1 Viola Jones Algoritmo.

El algoritmo de Viola Jones es un método de detección de objetos que destaca por su bajo costo computacional, lo que permite que sea empleado en tiempo real. Su desarrollo fue motivado por el problema de la detección de caras, donde sigue siendo ampliamente utilizado, pero puede aplicarse a otras clases de objetos que, como las caras, estén caracterizados por patrones típicos de iluminación [18][19]. El algoritmo se basa en una serie de clasificadores débiles denominados características Haar-like que se pueden calcular eficientemente a partir de una imagen integral. Estos clasificadores, que por sí mismos tienen una probabilidad de acertar solo ligeramente superior a la del azar, se agrupan en una cascada empleando un algoritmo de aprendizaje basado en AdaBoost para conseguir un alto rendimiento en la detección así como una alta capacidad discriminativa en las primeras etapas, a continuación explicáremos de cada una de las etapas antes mencionadas.

1.1.1.1 Características Haar-like

Las Haar-like son los elementos con los que se realiza la detección. Reciben este nombre por su parentesco a las wavelet de Haar [20]. Estas son características muy simples que se buscan en las imágenes y que consisten en la diferencia de intensidades luminosas entre regiones rectangulares adyacentes. Las características por tanto quedan definidas por unos rectángulos y su posición relativa a la ventana de búsqueda y adquieren un valor numérico resultado de la comparación que evalúan. En el trabajo presentado por Viola-Jones existen tres tipos de características que podemos observar en la figura 2.4, es importante mencionar que la suma de los pixeles en las áreas grises se resta a la de las áreas blancas.

- Características de dos rectángulos cuyo valor es la diferencia entre las sumas de los píxeles contenidos en ambos rectángulos. Las regiones tienen la misma área y forma y son adyacentes.
- Características de tres rectángulos que calculan la diferencia entre los rectángulos exteriores y el interior multiplicado por un peso para compensar la diferencia de áreas.
- Características de cuatro rectángulos que computan la diferencia entre pares diagonales de rectángulos.

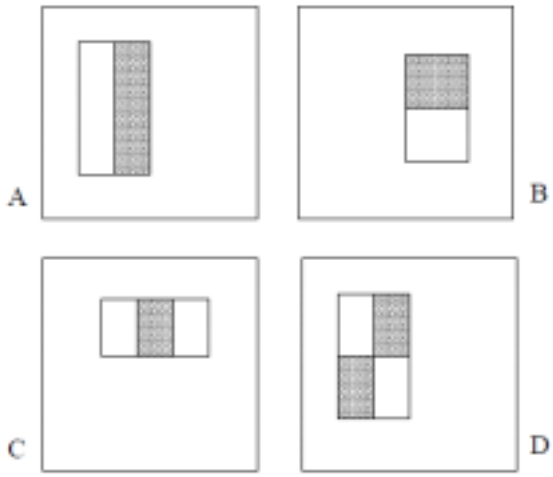


FIGURA 1.4 EJEMPLOS DE CARACTERISITCAS DE 2,3 Y 4 RECTANGULOS.

Para el algoritmo de Viola Jones, las características se definen sobre una ventana de búsqueda básica de 24x24 pixeles.

1.1.1.2 Imagen Integral

La suma de los píxeles de un rectángulo puede ser calculada de manera muy eficiente empleando una representación intermedia denominada imagen integral. La imagen integral en el punto (x, y) contiene la suma de todos los píxeles que están arriba y hacia la izquierda de ese punto en la imagen original, como lo muestra la siguiente ecuación:

$$Im(x,y) = \sum_{x' \leq x, y' \leq y} I(x', y'). \tag{1.1}$$

Donde Im(x,y) es la imagen integral e i(x,y) es la imagen original. La imagen integral se puede calcular en una sola iteración de la imagen empleando las siguientes 2 ecuaciones:

$$s(x,y) = s(x, \mathrm{y} - 1) + i(x,y), \tag{1.2}$$

$$im(x,y) = im(x-1, y) + s(x,y). \tag{1.3}$$

Donde s(x,y) es la suma acumulada de la fila x, con s(x,-1)=0 e im(-1,y)=0.

Usando la imagen integral, cualquier suma rectangular se puede calcular con cuatro referencias a memoria como se muestra en la figura 2.5. Las características de dos rectángulos se pueden computar con 6 referencias , para tres rectángulos se pasa a 8, y a 9 para características con cuatro rectángulos.

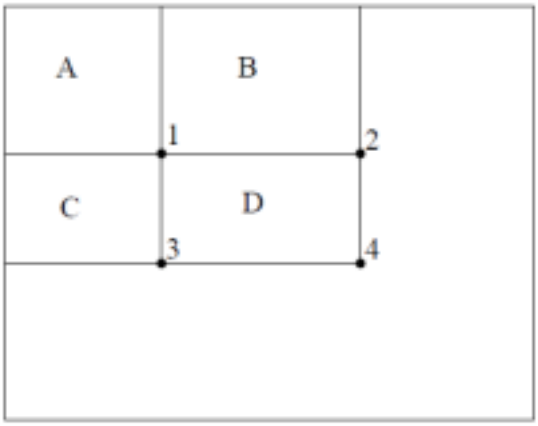


FIGURA 1.5 EJEMPLO DEL CALCULO DE LA SUMA DE PIXELES EN UN RECTANGULO[13].

1.1.1.3 Proceso de Aprendizaje

Para el proceso aprendizaje es necesario realizar un aprendizaje supervisado para crear la cascada de clasificadores. Este proceso se realiza mediante un algoritmo basado en AdaBoost, un meta-algoritmo adaptativo de machine learning cuyo nombre es una abreviatura de adaptative boosting. El boosting consiste en tomar una serie de clasificadores débiles y combinarlos para construir un clasificador fuerte con la precisión deseada. AdaBoost fue introducido por Freund y Schapire en 1995 resolviendo muchas de las dificultades prácticas asociadas al proceso de boosting [21]. En el procedimiento de Viola-Jones, AdaBoost se utiliza tanto para seleccionar un pequeño set de características de las 180000 posibles como para entrenar el clasificador. Para seleccionar las características, se entrenan clasificadores débiles limitados a usar una única característica. Para cada características, el clasificador débil determina el valor umbral que minimiza los ejemplos mal clasificados. Un clasificador débil $hj(x)$ por tanto consiste en una característica fj , un valor umbral θj y un coeficiente pj indicando la dirección del signo de desigualdad.

$$h_j(x) = \begin{cases} 1, & si\ p_j f_j(x) < p_j \theta_j \\ 0, & e.o.c. \end{cases} . \tag{1.4}$$

A continuación el algoritmo AdaBoost empleado es descrito a continuación de manera matemática, junto con representación grafica que se puede observar en la figura 1.6. En cada ronda se selecciona un clasificador débil y por tanto una característica

- Se parte de un conjunto de imágenes (x1, y1),…,(xn, yn) donde $y_i = 0, 1$ para ejemplos negativos y positivos respectivamente.
- Se inicializan los pesos $w_{1,i} = 1/2m$, $1/2l$ para $y_i = 0, 1$ respectivamente donde m es el número de negativos y l el número de positivos.
- Para cada ronda, t = 1,…,T:
  1. Normalizar los pesos como lo muestra la ecuación

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} . \tag{1.5}$$

  2. Para cada característica, j , entrenar un clasificador h_j que solo use una característica. El error se evalúa teniendo en cuenta los pesos W_i como muestra la ecuación

$$e_j = \sum_i W_i \left| h_j(x_i) - y_i \right| . \tag{1.6}$$

  3. Se escoge el clasificador, h_t , con menor error.
  4. Se actualizan los pesos, usando la ecuación

$$W_{t+1,i} = w_{t,i} \beta_t^{1-e_t}. \tag{1.7}$$

- Donde $e_i = 0$ si el ejemplo x_i se clasifica correctamente y 1 en caso contrario y $\beta t = e_t / 1 - e_t$

El clasificador fuerte final se muestra

$$h(x) = \begin{cases} 1, & si \ \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t, \\ 0, & e.o.c. \end{cases} \tag{1.8}$$

donde $\alpha t = 1/\beta_t$.

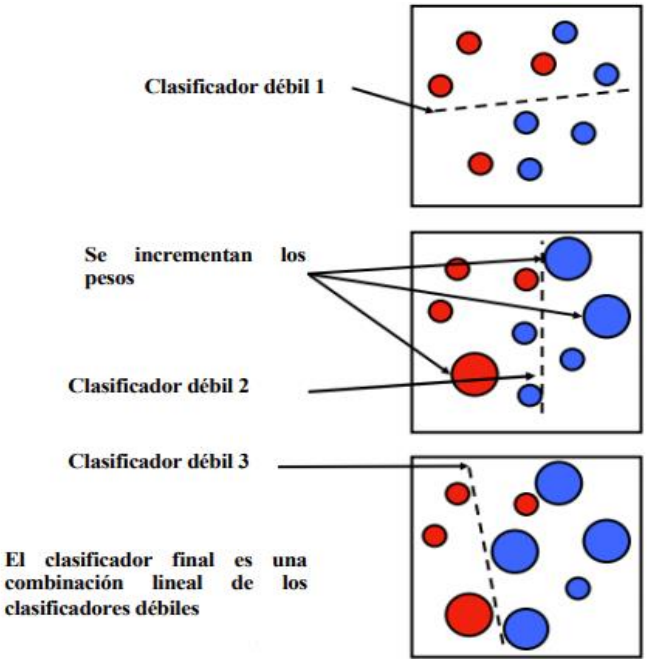


FIGURA 1.6 REPRESENTACION VISUAL DEL PROCESO ADABOOST[15].

En vez de construir un único clasificador mediante el proceso descrito anteriormente, se pueden construir clasificadores más pequeños y eficientes que rechacen muchas ventanas negativas (es decir, aquellas que no incluyan ninguna instancia del objeto buscado) manteniendo casi todas las positivas (es decir, las que contienen una instancia del objeto buscado). Estos clasificadores más simples se utilizan para rechazar la mayoría de las ventanas de búsqueda y solo en aquellas en las que hay mayores probabilidades de encontrar caras se llama a clasificadores más complejos que disminuyan el número de falsos positivos. Este proceso se representa en la figura 1.7. Con esto se obtiene una cascada de clasificadores, cada uno de los cuales es entrenado con AdaBoost y después sus valores umbrales se ajustan para minimizar los falsos negativos. La cascada entrenada por Viola-Jones tiene 38 etapas y más de 6000 características pero de media se evalúan únicamente 10 características por ventana de búsqueda [22].

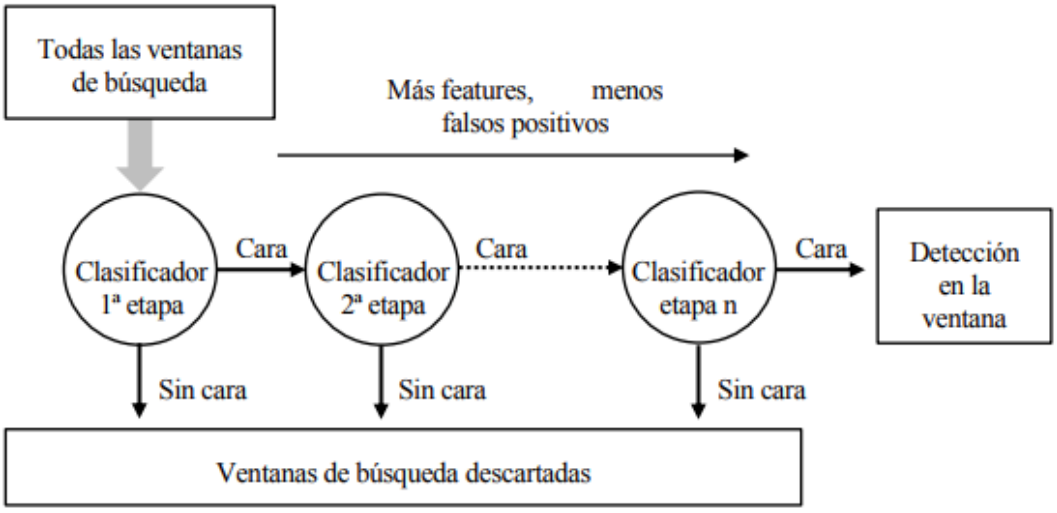


FIGURA 1.7 ESQUEMA DE CLASIFICADORES EN CASCADA.

El siguiente paso es el proceso de detección de un rostro para esto Las imágenes usadas para entrenar al algoritmo fueron normalizadas para minimizar los efectos de diferentes condiciones de iluminación, por tanto, también resulta necesario realizar la normalización en el proceso de detección. Para ello, en vez de normalizar la imagen antes de comenzar el análisis, lo cual implicaría cambiar el valor de todos los píxeles, resulta más sencillo corregir los valores de las características conforme se van calculando. Para normalizar se emplea la varianza:

$$\sigma^2 = m^2 - \frac{1}{\mathrm{N}}\sum_{i=1}^{N} {x_i}^2 \tag{1.9}$$

Donde m es la media del valor de los píxeles, que puede calcularse a partir de la imagen integral. La suma de los píxeles al cuadrado se puede obtener a partir de una imagen integral de la imagen al cuadrado. La cascada de características se evalúa sobre una ventana de búsqueda cuadrada que barre la imagen con incrementos de unos pocos píxeles. La búsqueda se realiza a distintas escalas obtenidas al multiplicar la escala anterior por un factor de escala, normalmente entre 1.1 y 1.3. Puesto que el detector es poco sensible a pequeñas translaciones y diferencias de escala, se suelen producir múltiples detecciones alrededor de cada cara. De hecho, se puede exigir que las detecciones tengan un determinado número mínimo de detecciones vecinas para disminuir el número de falsos positivos. Para combinar las detecciones que se refieren al mismo objeto, se fusionan las detecciones cuyas áreas se solapen más de un determinado valor umbral y el recuadro con la detección final se calcula como la media de todos los recuadros que se han fusionado.

1.1.1.4 Algoritmo de Viola Jones en OpenCv

Antes de hablar sobre la implementación del algoritmo de Viola Jones en OpenCv describiremos de manera muy breve que es OpenCv.

1.1.1.4.1 OpenCv

El 13 de Junio del 2000, Intel® Corporativo anunciaron que estaban trabajando con un grupo de reconocidos investigadores en visión por computador para realizar una nueva librería de estructuras/funciones en lenguaje C. Esta librería proporcionaría un marco de trabajo de nivel medio-alto que ayudaría al personal docente e investigador a desarrollar nuevas formas de interactuar con los ordenadores. Este anuncio tuvo lugar en la apertura del IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). Fue hay cuando nació The Open Computer Vision Library [5], mejor conocido como OpenCv. Las librería de OpenCV están dirigidas fundamentalmente a la visión por computadora en tiempo real. Entre sus muchas áreas de aplicación destacarían: interacción hombre-máquina; segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento; y robots móviles.

1.1.1.4.2 Implementación del Algoritmo de Viola Jones en OpenCv

El algoritmo de detección que implementa OpenCv es una versión del algoritmo de Viola-Jones que permite emplear características inclinadas a 45º grados [16]. Otra particularidad que presenta el método empleado en OpenCv es que las características se definen sobre una ventana de búsqueda básica de 20x20 píxeles en vez de 24x24.

El código relativo a las funciones en cuestión se encuentra en la dirección \OpenCv\sources\modules\objdetect\src dentro de los ficheros de OpenCv. En concreto, en los ficheros “cascadedetect.cpp” de 1300 líneas de código con funciones para cascadas de tipo Haar, Hog y LBP y “haar.cpp” con más de 2500 líneas de código específicas para cascadas de tipo Haar. Este código compilado forma parte de la librería “objectdetect”.

OpenCv también facilita una serie de cascadas ya entrenadas. En este trabajo hemos empleado la cascada “haarcascade_frontalface_alt” que puede encontrarse en la dirección \opencv\sources\data\haarcascades, que fueron las que se usaron en este trabajo para detectar rostro como se vera mas adelante. Las funciones de detección se integran en la clase CascadeClassifier y la función de partida es detectMultiScale cuya definición es la siguiente:

void CascadeClassifier::detectMultiScale(const Mat& image, vector& objects, double scaleFactor=1.1, int minNeighbors=3, int flags=0, Size minSize=Size(), Size maxSize=Size())

Los argumentos de la función son:

- cascade: cascada de clasificadores Haar. Se carga previamente a partir de un fichero XML.
- image: matriz de tipo CV_8U sobre la que se busca el objeto u objetos a detectar.
- objects: vector de rectángulos donde cada rectángulo contendrá un objeto detectado.
- scaleFactor: factor de escala que define los incrementos en las escalas de búsqueda.
- minNeigbours: parámetro que especifica cuántos vecinos debe tener cada rectángulo candidato para que sea aceptado.
- flags: selección de algunas opciones para formatos de cascada antiguos.
- minSize: tamaño mínimo de los objetos a detectar.
- maxSize: tamaño máximo de los objetos a detectar.

En el caso de la cascada empleada, detectMultiScale simplemente llama a la función cvHaarDetectObjectsForROC, esta función se encarga de realizar el escalado y de llamar al resto de funciones necesarias para la detección. En la opción por defecto se escala la cascada de características manteniendo la imagen constante, por tanto, la imagen integral se calcula una sola vez para la imagen original. Con flag= 0|CV_HAAR_SCALE_IMAGE, se escala la imagen por lo que en cada iteración del escalado hay que calcular una nueva imagen integral.

Para cada factor de escala, se llama a HaarDetectObjects_ScaleCascade_Invoker que se encarga de realizar el barrido de la imagen en las direcciones horizontal y vertical. Para cada ventana de búsqueda, se llama a SetImagesForHaarClassiferCascade que inicializa los punteros necesarios para que la cascada se evalúe en dicha ventana de búsqueda y a cvRunHaarClassifierCascadeSum que ejecuta la cascada.

Por último, la función groupRectangles se encarga de combinar las detecciones múltiples en un único rectángulo descartando los rectángulos que tengan menos de un determinado número de vecinos.

1.1.2 Problemas en el procesamiento facial automático.

Los problemas que presenta el procesamiento automático de rostros se puede dividir según el problema a resolver, a continuación mencionaremos los grandes problemas que aquejan este campo:

Detección de rostros. El problema aquí es cuantas rostros podemos ver una imagen, de que tamaño son los rostros. Aquí hay un punto trascendental y que es un requisito para los siguientes problemas que mencionaremos, Donde se encuentra el rostro, es decir las coordenadas del rostro en nuestra imagen.

- **Localización de Componentes faciales.** Ya con un rostro detectado podemos dar el siguiente paso que reconocer y extraer las regiones de interés del rostros, dentro de esta tesis proponemos un algoritmos de extracción automática de las regiones de interés del rostro que es invariante a los cambios de iluminación.
- **Estimación de Perfil.** Al igual que el punto anterior, este es un problemas que se aborda en esta tesis, en la bibliografía actual estiman las pose de imágenes en 3D.

- **Seguimiento de rostros en video.** Su objetivo es encontrar las variaciones de posición, forma y orientación de un rostro.
- **Reconocimiento de personas**. Siendo uno de los temas que mas bibliografía podemos encontrar.
- **Extracción de Información**. Se busca poder determinar de manera automática si el rostro esta sufriendo alguna oclusión ya sea por cabello, lentes, bufanda, etc.
- **Caracterización del rostros**. Uno de los principales problemas que ha sido abordado por muchos autores y de diferentes formas como el an*álisis de componentes principales* (PCA) , desde que Kirby y Sirovich demostraron por primera vez que las caras podían ser codificadas y reconstruidas de forma fiable usando subespacios lineales de muy reducida dimensionalidad [23]. De hecho, la reducción de dimensiones es uno de sus usos mas frecuentes: dada una imagen de una cara, representarla de manera compacta y conservando la mayor parte de la información relevante. Los conceptos de PCA, descomposición en valores y vectores propios, transformada Karhumen-Loeve, y transformada Hotelling son esencialmente equivalentes, otros métodos usados son los filtros de consolación, filtros Wavelet, Haar y Gabor y los modelos de apariencia activa.
- **Análisis de Expresión facial.** Este es el problema que se busca resolver en esta tesis, en el Capitulo 3.

1.1.3 Aplicaciones en el procesamiento facial automático.

Algunos de los posible campos de aplicación asociados a los diversos problemas ya descritos, se describen a continuación.

- **Biometría y Seguridad.** La superioridad de los rostros frente a otras biométricas(Iris, huella, andar etc.) ha sido reconocida en muchos artículos [24,25]. El interés por estas tecnologías ha crecido de manera importante teniendo como principales aplicaciones las siguientes: Control de aduanas, documentos de identidad, sistemas de acceso biométrico, video vigilancia, etc.
- **Interfaces y Entretenimiento.** Esta aplicación es una de las que mas investigaciones en los últimos años ha generado, con el principal objetivo de desarrollar sistemas mas intuitivos y sencillos de manipular teniendo como aplicaciones: interfaces para la navegación, interacción natural con robots, videojuegos basados en la percepción del rostro del jugador, control automáticos de cámaras, etc.
- **Indexación Multimedia.** Esta aplicación tiene como principal objetivo analizar y catalogar toda la información multimedia (videos) que hoy podemos encontrar en internet.

1.2 Clasificadores

Desde el inicio de la humanidad, la sabiduría y la posesión de información han sido sinónimos de supervivencia y poder. Actualmente, la cantidad de información de la que podemos disponer gracias a internet es inmensa haciendo su manejo y tratamientos complicados. La introducción de las nuevas tecnologías han conseguido no solo poder obtener y almacenar grandes cantidades de datos, sino que además surge la pregunta ¿Es posible un uso inteligente de estos datos?, para contestar a esta pregunta se intentan desarrollar métodos de análisis de manera que las computadoras sean capaces de aprender de estos datos, para así mejorar la eficiencia en sus procesos y obtener algún conocimiento sin la necesidad de la intervención humana.

Como consecuencia a estas necesidades surge un nueva área de investigación dentro de la informática y esta es el Aprendizaje Automático que tiene como principal objetivo realizar maquinas de aprendizaje o clasificadores, cuyo objetivo es lograr aprender en base a ciertos datos que el usuario le proporcione. Uno de los objetivos principales de este subcapítulo es presentar el estado del arte del área del aprendizaje automático explicando los diferentes tipos de aprendizaje y sus principales algoritmos, por ultimo mencionaremos un área que en los últimos años se ha estado uniendo con las clasificadores, esta área es la lógica difusa.

1.2.1 Aprendizaje Automático

En Inteligencia Artificial, el aprendizaje o aprendizaje automático se entiende como un proceso por el cual una computadora acrecienta su conocimiento y mejora su habilidad. En él se resaltan dos aspectos complementarios: el refinamiento de la habilidad y la adquisición de conocimiento.

Muchas de las técnicas de aprendizaje automático usadas en la Inteligencia Artificial están basadas en el aprendizaje realizado por los seres vivos. Para ellos la experiencia es muy importante, ya que les permite no volver a cometer los mismos errores una y otra vez. Además, la capacidad de adaptarse a nuevas situaciones y resolver nuevos problemas es una característica fundamental de los seres inteligentes. Por lo tanto, podemos mencionar varias razones de peso para estudiar el aprendizaje: en primer lugar, como método de comprensión del proceso de aprendizaje y, en segundo término, aunque no por ello menos importante, para conseguir programas que aprendan (desde una perspectiva más propia de la Inteligencia Artificial).

Una primera clasificación de las técnicas de aprendizaje automático existentes se puede realizara tendiendo a la filosofía seguida en el proceso de adquisición del conocimiento.

- En el Aprendizaje Supervisado o aprendizaje a partir de ejemplos, con profesor, los ejemplos de entrada van acompañados de una clase o salida correcta.

- En el Aprendizaje No Supervisado o aprendizaje por observación, sin profesor se construyen descripciones, hipótesis o teorías a partir de un conjunto de hechos u observaciones sin que exista una clasificación a priori de los ejemplos. El ejemplo más significativo de este tipo es el de los métodos de agrupamiento o Clustering.

- En el Aprendizaje SemiSupervisado, donde los ejemplos de entrada no van acompañados de una clase o salida correcta.

En todo proceso de aprendizaje se han de distinguir dos etapas: entrenamiento y prueba. Durante el entrenamiento se han de extraer las conclusiones apropiadas a partir de un conjunto de casos de entrenamiento y se obtiene un modelo que sea capaz de mostrar lo aprendido. Posteriormente, durante la fase de prueba se tiene que estudiar la calidad del modelo obtenido en la fase de anterior usando un conjunto de datos de prueba que deben ser distintos a los usados en la fase entrenamiento.

En la etapa de entrenamiento, durante la construcción del modelo adecuado a los ejemplos conocidos, se persiguen los siguientes objetivos:

- Precisión la proporción de ejemplos aprendidos sea la mayor posible.
- Tiempo de aprendizaje razonable.
- Que el ser humano sea capaz de entender el porqué de las cosas.
- Adaptabilidad del sistema ante nuevas situaciones.

En la fase de prueba, se tiene que realizar una estimación del error cometido. Para ello se suele utilizar una matriz (denominada matriz de confusión) en la que se muestran los errores que se han cometido al clasificar los casos de prueba para cada categoría o clase. Hay que tener en cuenta que en algunas aplicaciones un falso positivo no es igual de importante que un falso negativo un ejemplo muy claro de esto es el campo de la Medicina.

En los siguientes puntos se estudian los distintos tipos de aprendizaje automático y se detallan brevemente algunos de los modelos más utilizados por ellos.

1.2.1.1 Aprendizaje Supervisado

El aprendizaje supervisado es un problema de gran interés en el área de la Inteligencia Artificial, donde los datos de entrada denominados conjunto de entrenamiento son instancias de las cuales se vale el sistema para ajustarse perfectamente. El objetivo de la clasificación es obtener un modelo preciso para cada clase utilizando los atributos de los datos de entrada. El modelo obtenido puede servir para clasificar casos cuya clases se desconozcan o, simplemente, para comprender mejor la información de la que disponemos.

Los casos de entrenamiento utilizados en la construcción del modelo suelen expresarse en términos de un conjunto finito de propiedades o atributos con valores discretos o numéricos. Las categorías a las que han de asignarse los distintos casos deben establecerse de antemano, es por esta razón que es un Aprendizaje Supervisado. En general, estas clases serán disjuntas (aunque pueden establecerse jerarquías) y deberán ser discretas (para predecir atributos con valores continuos se suelen definir categorías discretas utilizando términos imprecisos propios del lenguaje natural).

Las técnicas inductivas de clasificación se basan en el descubrimiento de patrones en los datos de entrada, por lo que hemos de disponer de suficientes casos de entrenamiento; es decir, más casos de entrenamiento que el número de clases a diferencia, con esto será posible obtener un modelo de clasificación óptimo.

Son necesarios bastantes datos para poder diferenciar patrones válidos de patrones irregulares o errores. Esta diferenciación se suele realizar utilizando alguna técnica estadística.

Este tipo de aprendizaje es capaz de generar 2 tipos de modelos. Por lo general, se obtiene una función que transforma los patrones de entrada en los resultados deseados, esto con el fin de resolver un problema determinado como podría ser el reconocimiento de la escritura, este tipo de sistema considera los siguientes pasos.

- Determinar los patrones de entrenamiento, esto con el fin de definir qué tipos de datos se van a utilizar para entrenar el modelo.

- Reunir un conjunto de entrenamiento, esto es un conjunto de objetos de entrada que se recopila junto con sus salidas correspondientes.
- Determinar la función de ingreso de la representación de la función aprendida, el patrón de ingreso es un vector de características del objeto de entrada, que contiene una serie de características descriptivas del objeto. El número de características no debe ser demasiado grande, a causa de las propiedades de dimensionalidad, sin embargo deber ser lo suficientemente grande para poder ser capaz de describir el objeto de manera correcta.
- Definir la estructura de la función adecuada para resolver el problema y la técnica de aprendizaje correspondiente, es decir se puede optar por utilizar una red neuronal artificial o un árbol de decisión, para después ejecutar este algoritmo de aprendizaje en el conjunto de la formación obtenida.

La clasificación de objetos es el otro modelo que se puede generar a través del Aprendizaje Supervisado. Una amplia gama de clasificadores están disponibles, donde cada uno de estos tiene sus ventajas y desventajas, es importante hacer mención que el rendimiento de un clasificador depende en una gran medida de las características de los datos que deben clasificarse. Se han realizado diversas pruebas empíricas para comparar el rendimiento del clasificador, esto con el fin de encontrar las características de los datos que determinan el rendimiento del clasificador. Lo clasificadores más utilizados son la redes neuronales artificiales, como el Perceptron multicapa y las maquinas de soporte vectorial.

1.2.1.1.1 Redes Neuronales Artificiales

Las redes de neuronas artificiales son un paradigma de aprendizaje automático y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Estas se pueden ver como sistemas paralelos de computación masiva con un gran número de procesadores con muchas interconexiones. Estas redes se componen de unidades llamadas neuronas o nodos. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida, en la figura 2.1, se muestra un esquema clásico de una Red Neuronal.

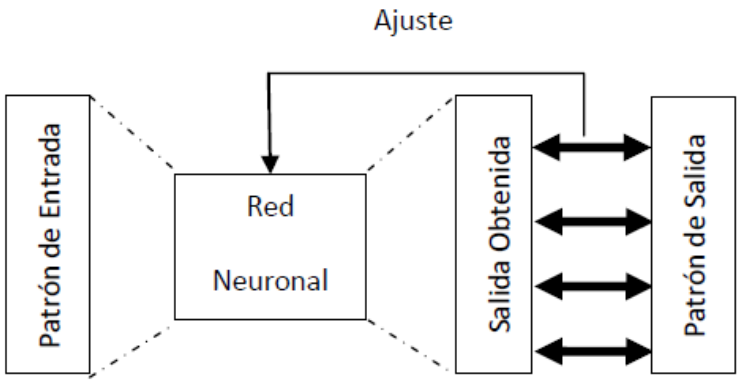


FIGURA 1.9 ESQUEMA TÍPICO DE UNA RED NEURONAL ARTIFICIAL.

Las primeras redes neuronales surgieron a comienzos del siglo XX, pero no fue hasta la década de los 40 cuando empezaron a cobrar fuerza. En 1957 F. Rosenblatt desarrolló el modelo que el día de hoy conocemos como Perceptron, y dos años después, B. Widrow desarrolla la Red Neuronal Adaline. En [26] se puede encontrar un resumen más detallado sobre la historia de las redes neuronales desde su comienzo hasta la actualidad, hoy en día uno de los modelos más utilizados es el Perceptron multicapa.

Este método recibe su nombre debido a la estrecha relación existente entre su estructura matemática y su diseño con el de una red de neuronas biológica. Una red de neuronas biológica se compone de tres partes [26]:

- Los receptores: se encuentran en las células sensoriales y recogen la información en forma de estímulos, estos pueden ser del ambiente o del mismo organismo.
- El sistema nervioso: recibe la información, la elabora y en parte la almacena y se encarga de enviarla a los órganos y a otras zonas del sistema nervioso.
- Órganos diana o efectores: reciben la información y la interpretan en forma de acciones motoras, hormonales, etc.

Del mismo modo que las neuronas biológicas están compuestas en tres partes, las redes neuronales artificiales están formadas por tres niveles de capas:

- Nivel de entrada: tiene una sola capa con ***m*** neuronas de entrada.
- Nivel oculto: puede tener una o más capas, cada una de ellas con ***n*** neuronas, siendo ***n*** un patrón a escoger por el diseñador de la red.
- Nivel de salida: tiene una única capa con ***c*** neuronas, siendo c el número de salidas deseadas.

Un ejemplo de red neuronal se representa en la figura 2.2 y en ella se pueden apreciar los tres niveles antes mencionados, donde en este caso el nivel oculto tiene una sola capa. Este ejemplo es una red neuronal tipo Perceptron Multicapa.

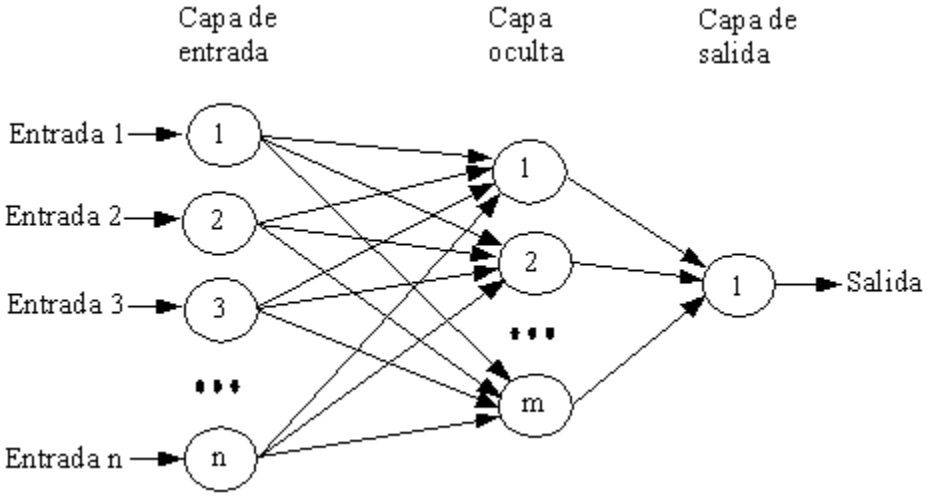


FIGURA 1.8 EJEMPLO DE RED NEURONAL TIPO PERCEPTRON MULTICAPA.

La salida de una red neuronal está representada por una función $f(x)$que es la combinación de otras funciones $g_i(x)$, las cuales a su vez pueden ser combinaciones de otras funciones. Una función muy utilizada para una red neuronal con ***i*** neuronas ocultas es:

$$f(x) = \sum_i \omega_i{}^{(2)} h_i + \theta^{(2)}. \tag{1.10}$$

Siendo, $\omega_i$los pesos asociados a las neuronas, θ una constante, el superíndice es el nivel correspondiente, y $h_i$una función no lineal como por ejemplo:

$$h_i = tanh\left(\sum_j \omega_{ij}{}^{(1)} x_j + \theta_i{}^{(1)}\right). \tag{1.11}$$

La complejidad de la función h_i y, por tanto, de la misma red neuronal, depende del número de capas ocultas de esta. Como sabemos el aprendizaje automático supervisado, consta de 2 fases una de entrenamiento y otra de validación. Durante la primera fase, las redes neuronales artificiales usan un conjunto de datos de entrenamiento para determinar los pesos (parámetros de diseño) que definen a la Red Neuronal. Una vez entrenado este modelo, se pasa a la fase de validación, en la que se procesan el conjunto de datos de validación.

Existen múltiples tipos de redes neuronales que pueden ser clasificados según el tipo de entradas, lo más apropiado al escoger una red neuronal es tener en cuenta las necesidades específicas del problema. La familia de redes neuronales más utilizada para problemas de clasificación es la red de Retro propagación, la cual incluye redes de Perceptron multicapa y redes de funciones de base radiales.

Las redes neuronales tienen un problema común en los métodos de aprendizaje supervisado de nominado sobre-aprendizaje El sobre-aprendizaje ocurre cuando una red neuronal ha aprendido el modelo correctamente en la etapa de entrenamiento, pero no responde adecuadamente a la validación. Esto sucede por diversas causas por ejemplo, porque el número de ciclos de aprendizaje es muy elevado, o porque el número de neuronas de la capa oculta es también muy elevado, etc. El mejor modo de evitar el sobre aprendizaje es lograr un mayor número de casos para el entrenamiento. Cuando esto no sea posible también se puede limitar el número de nodos en la capa oculta, limitar el número de ciclos de aprendizaje o usar la técnica de validación cruzada. La figura 2.3 muestra un caso de red neuronal sobre-entrenada. En la figura, el error de entrenamiento se muestra en azul, mientras que el error de validación se muestra en rojo. Si el error de validación aumenta mientras que el de entrenamiento decrece puede que se esté produciendo una situación de sobre-ajuste.

En [2] hacen una revisión más detallada sobre las redes neuronales, sus tipos, sus aplicaciones, etc.

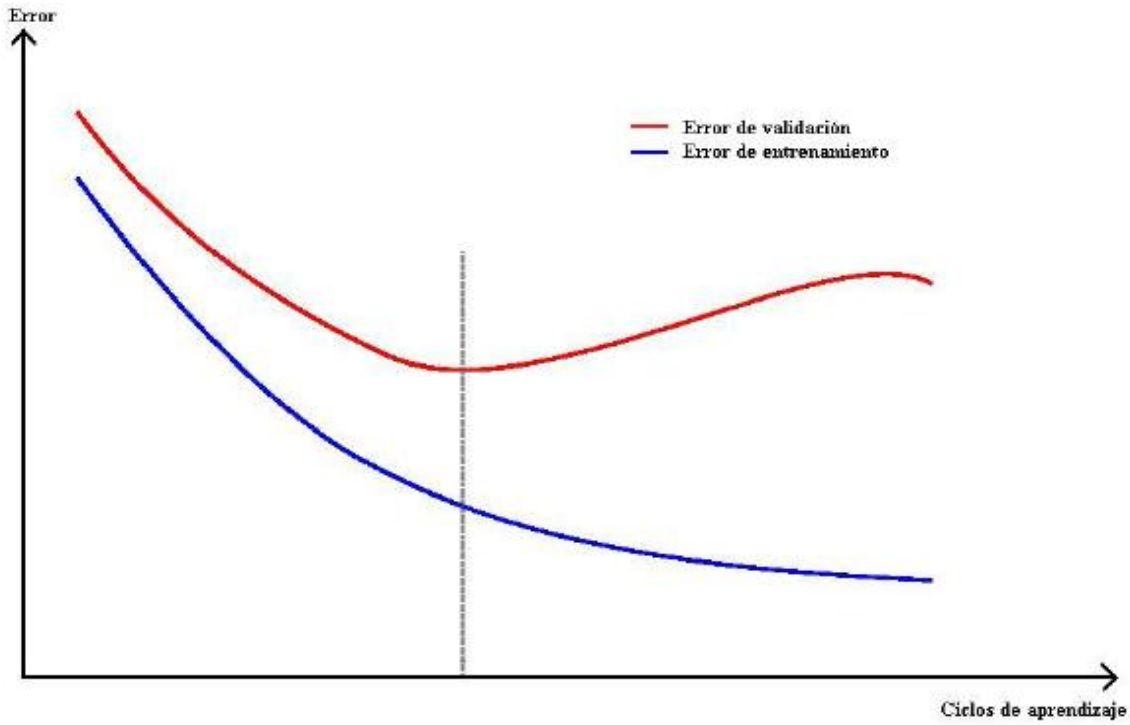


FIGURA 1.9 SOBRE APRENDIZAJE AUTOMÁTICO EN UNA RED NEURONAL.

1.2.1.1.2Maquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial (Support Vector Machine, por sus siglas en inglés SVM) son un nuevo sistema de aprendizaje supervisado que ha tenido un desarrollo muy importante en estos últimos años, tanto en la generación de nuevos algoritmos como en su implementación. Las SVM son sistemas de aprendizaje basados en el uso de un espacio de hipótesis de funciones lineales en un espacio de mayor dimensión inducido por un Kernel. En este nuevo espacio las hipótesis son entrenadas por un algoritmo tomado de la teoría de optimización, que utiliza elementos de la teoría de generalización.

Las SVM son sistemas para entrenar máquinas de aprendizaje lineal de manera eficiente. Tanto para clasificación como para regresión se han encontrado muchas aplicaciones de estas, como por ejemplo en la clasificación de imágenes, en reconocimiento de caracteres, en clasificación de patrones, etc.

Las SVM pueden verse como una solución a los problemas de las redes neuronales. Las máquinas de vectores de soporte se caracterizan por ser problemas de optimización convexos. Uno empieza formulando un problema en un espacio

inicial determinado, pero termina resolviendo el problema en otro espacio de mayor dimensión donde existe una solución lineal. La solución obtenida por las máquinas de vectores de soporte son soluciones dispersas.

Las SVM están basadas en el principio de minimización del riesgo estructural, el cual ha demostrado ser superior al principio de minimización del riesgo empírico, utilizado por las redes neuronales convencionales. Algunas de las razones por las que este método ha tenido éxito es que no padece de mínimos locales y el modelo sólo depende de los datos con más información llamados vectores de soporte (SV por sus siglas en ingles, Support Vector).

La máquina de vectores de soporte más sencilla fue la ideada por Vapnik (para información más detallada de esta técnica ver [2]). Una visión general de las máquinas de vectores es la ofrecida por C.J.C. Burges en [27], aunque existe una bibliografía muy extensa sobre este tema.

1.2.1.2 Aprendizaje No Supervisado

Hasta este punto, sabemos que el aprendizaje supervisado dispone de información en la salida, ahora nos encontramos en el lado contrario con los problemas en los que no se disponemos de ningún tipo de información de salida sobre los datos, pero deseamos organizar los datos para mejorar su comprensión. Este tipo de problemas son los denominados problemas de Aprendizaje No Supervisado. Según [28], existen cinco razones principales por las cuales el uso de técnicas de aprendizaje no supervisado resulta de interés para las aplicaciones:

- La recolección de datos y su posterior etiquetamiento, en un conjunto de datos muy extenso puede suponer un costo muy elevado.
- Otro punto de interés es actuar en sentido inverso; es decir aprender con una gran cantidad de datos sin etiquetar, y sólo usar supervisión para etiquetar los distintos grupos encontrados.

.

- En muchas aplicaciones las características de los patrones cambian con el tiempo. Si estos cambios pueden ser rastreados en un proceso de ejecución sin supervisar, podremos obtener un resultado más idóneo.
- Es posible usar métodos no supervisados para encontrar características que serán útiles en la categorización.
- En el comienzo de la investigación puede ser útil tener una visión general de la naturaleza y la estructura de los datos.

La principal diferencia entre el aprendizaje supervisado y no supervisado recae en que al no disponer de información acerca de la salida, en la fase de entrenamiento no se puede reajustar el modelo en base al error. Pero sigue siendo muy importante separar los datos en datos de entrenamiento y datos de validación para decidir si el método es bueno o no.

La forma de selección del algoritmo y el entrenamiento también se mantienen. En este caso, la posibilidad de validar si los resultados son correctos no es frecuente, puesto que no se dispone de información de la salida. La manera de decidir cuándo se ha aprendido es viendo si el sistema converge o estableciendo un criterio de parada.

1.2.1.2.1 Redes neuronales no supervisadas.

Las redes neuronales explicadas hasta el momento son supervisadas. Esto es, necesitan obtener información acerca de la salida para poder realizar su aprendizaje automático. Sin embargo, en muchas situaciones no es posible tener esta información a priori.

A continuación se mencionan un nuevo tipo de redes neuronales no supervisadas, en la que las propias redes son capaces de modificar sus parámetros internamente sin necesidad de supervisión.

Las redes neuronales no supervisadas, por lo general, tienen una arquitectura sencilla, y se caracterizan por ser más similares a los modelos biológicos que las redes neuronales artificiales supervisadas. La primera aproximación a las redes neuronales no supervisadas fue en 1949, cuando D. Hebb enunció la regla que lleva su mismo nombre, y que se refiere al comportamiento biológico observado en las neuronas. Más tarde, en el año de 1987, Carpenter y Grossberg desarrollaron los modelos ART de redes neuronales no supervisadas basándose en la teoría de resonancia adaptiva.

Uno de los modelos de redes neuronales no supervisadas más utilizados es el desarrollado por Kohonen basado en mapas auto-organizativos. Las redes de Kohonen son redes de dos capas: la capa de entrada que recibe la señal de entrada a la red, y la capa de salida que realiza el cálculo. Cada neurona de la capa de salida debe reflejar las coordenadas que tiene en el espacio que el diseñador de la red decida. Para que las neuronas puedan ser comparadas con la posición de otras neuronas de la red, a cada neurona se le asocia una regla de vecindad. El objetivo es agrupar entradas similares a neuronas de posiciones similares. El conjunto de datos es agrupado, porque es asociado al mismo nodo o al mismo conjunto de nodos.

Existe una amplia bibliografía acerca de este tipo de redes. Ejemplos de aplicaciones de redes neuronales no supervisadas y un estudio más profundo sobre estas puede encontrarse en [26].

1.2.1.2.2 Agrupamiento

Uno de los métodos de clasificación de datos es el agrupamiento o Clustering que constituye un tipo de aprendizaje por descubrimiento muy similar a la inducción, ya que en el aprendizaje inductivo, un algoritmo debe ser capaz de aprender

por si solo a clasificar objetos basándose en objetos previamente etiquetados proporcionados por un profesor esto es un Aprendizaje Supervisado, para el caso de los métodos de agrupamiento no se suministran los datos etiquetados: el algoritmo debe poder descubrir por sí mismo las clases naturales existentes con esto tenemos un tipo de Aprendizaje No Supervisado.

Por ejemplo, el programa AUTOCLASS (Cheeseman, Self, Kelly, Taylor, Freeman y Stutz, "Bayesian Classification", ProceedingsAAAI88, 1988) utilizo razonamiento bayesiano, para un conjunto dado de datos de entrenamiento, sugería un conjunto de clases plausible. Este algoritmo encontró nuevas clases significativas de estrellas a partir de sus datos del espectro infrarrojo, lo que pudo considerarse un descubrimiento por parte de una máquina ya que este hecho era desconocidos para los astrónomos.

Las funciones de densidad de probabilidad suelen tener una moda o un máximo en una región; es decir, las observaciones tienden a agruparse en torno a una región del espacio de patrones cercana a la moda. Las técnicas de agrupamiento analizan el conjunto de observaciones disponibles para determinar la tendencia de los patrones a agruparse. Estas técnicas permiten realizar una clasificación asignando cada observación a un agrupamiento (clúster), de forma que cada agrupamiento sea homogéneo y diferenciable de los demás.

Los agrupamientos naturales obtenidos utilizando una técnica de agrupamiento mediante similitud, resultan muy útiles a la hora de construir clasificadores cuando no están bien definidas las clases, ya que no existe un conocimiento suficiente de las clases en que se pueden distribuir las observaciones, cuando se desea analiza un gran conjunto de datos ("divide y vencerás") o, simplemente, cuando existiendo un conocimiento completo de las clases se desea comprobar la validez del entrenamiento realizado y del conjunto de variables escogido. Los métodos de agrupamiento asocian un patrón a un agrupamiento siguiendo algún criterio de similaridad. Tales medidas de similaridad deben ser aplicables entre pares de patrones, entre un patrón y un agrupamiento y, finalmente, entre pares de agrupamientos. Generalmente, como medidas de similaridad se emplean métricas de distancia. Algunas de las más habituales son la distancia Euclídea, la distancia Euclídea normalizada, la distancia Euclídea ponderada y la distancia de Mahalanobis.

1.2.1.2.2.1 Medidas de Similitud o de Distancia

Las medidas de similitud o de distancia son expresiones matemáticas que permiten resumir en un número el grado de relación entre dos entidades, sobre la base de semejanza o la desigualdad entre la cualidad o la cantidad de sus atributos, o ambas. Estas medidas son fundamentales en la definición de agrupamiento y han sido estudiadas desde la década de 1950 [15], en campos tan variados como la psicología o el tratamiento de imágenes [29].

Según [30] una medida de similitud $\boldsymbol{s}$, en un conjunto de datos $\boldsymbol{X}$, se define como:

$$s : X \times X \to \mathbb{R}, talque \exists s_o \in \mathbb{R}: \ -\infty < s(\boldsymbol{x}, \boldsymbol{y}) \leq s_0 < \ +\infty, \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

Siendo $\mathbb{R}$ el espacio de los números reales, y cumpliéndose

$$s(\boldsymbol{x}, \boldsymbol{x}) = s_0, \forall \boldsymbol{x} \in X$$

$$s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x}), \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

$$s(\boldsymbol{x},\boldsymbol{y}) = s_0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$$

$$s(\boldsymbol{x},\boldsymbol{y})s(\boldsymbol{y},\boldsymbol{z}) \leq [s(\boldsymbol{x},\boldsymbol{y}) + s(\boldsymbol{y},\boldsymbol{z})]s(\boldsymbol{x},\boldsymbol{z})\forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in X$$

No obstante, más común que medir la similitud entre dos puntos o dos objetos, es medir la disimilitud es decir la distancia entre ellos. De acuerdo a [30] podemos definir las medidas de disimilitud o la distancia como:

$$d : X \times X \to \mathbb{R}, talque \exists d_o \in \mathbb{R}: \ -\infty < d_0 \leq d(\boldsymbol{x},\boldsymbol{y}) < \ +\infty, \forall \boldsymbol{x},\boldsymbol{y} \in X$$

Siendo d una métrica, se cumplen:

$$d(\boldsymbol{x},\boldsymbol{x}) = d_0, \forall \boldsymbol{x} \in X$$

$$d(\boldsymbol{x},\boldsymbol{y}) = d(\boldsymbol{y},\boldsymbol{x}), \forall \boldsymbol{x},\boldsymbol{y} \in X$$

$$d(\boldsymbol{x},\boldsymbol{y}) = d_0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$$

$$d(\boldsymbol{x},\boldsymbol{z}) < d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z})\forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in X$$

Las medidas más comunes de distancia o de disimilitud son:

- La distancia de Minkowski: Se define según la ecuación 1.12. Posee la peculiaridad de que aquellas características con valores y varianzas grandes tienden a dominar sobre otras.

$$d_{Mink}(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{l}|x_i - y_i|^n\right)^{1/n}. \quad 1.12$$

- La distancia Euclídea: Es la medida más utilizada siendo esta un caso especial de la distancia Minkowski cuando el valor de n es igual a 2. Esta es invariante a translaciones y rotaciones lineales se define según la ecuación siguiente

$$d_{Euc}(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{l}|x_i - y_i|^2\right)^{1/2}. \quad 1.13$$

.

- La distancia Euclídea cuadrada: Esta es una variación de la distancia Euclídea, se define según la ecuación

$$d_{Euc2}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{l}|x_i - y_i|^2. \quad 1.14$$

- La distancia Promedio: Como su nombre lo indica obtiene el promedio de las diferencias de los valores, se define según la ecuación

$$d_{Prom}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{l}\sum_{i=1}^{l}(x_i - y_i)^2. \quad 1.15$$

- La distancia de Manhattan: Al igual que la distancia Euclídea es un caso especial de la distancia Minkowski pero en este caso el valor de n es igual a 1. Se define según la ecuación

$$d_{Euc}(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{l}|x_i - y_i|\right). \quad 1.16$$

Con lo anterior solo se mencionan algunas de las distancias más importantes que existen dentro de la literatura, siendo estas la base de de los Métodos de Agrupamiento.

1.2.1.2.2.2 Método Adaptativo

El método adaptativo[31] es un algoritmo heurístico de agrupamiento que se puede utilizar cuando no se conoce de antemano el número de clases del problema. Entre sus ventajas se encuentran su simplicidad y eficiencia. Además, las observaciones se procesan secuencialmente. Por desgracia, su comportamiento está sesgado por el orden de presentación de los patrones y presupone agrupamientos compactos separados claramente de los demás.

El primer agrupamiento se escoge arbitrariamente. Se asigna un patrón a un clúster si la distancia del patrón al centroide del clúster no supera un umbral. En caso contrario, se crea un nuevo agrupamiento. Este algoritmo incluye una clase de rechazo a la hora de clasificar. Los patrones se asignan por la regla de mínima distancia. Algunos patrones no son clasificados si el cuadrado de la distancia al agrupamiento más cercano es mayor que el umbral τ.

TABLA 1.1 ALGORITMO DEL MÉTODO ADAPTATIVO

Parámetros		
	τ	Umbral de distancias(al cuadrado)
	θ	Fracción [0,1]
	$\{X_l\}$	Conjunto de Patrones
Variables		
	A	Número actual de agrupamientos
	Z_i	Centroide del agrupamiento i

Algoritmo

Inicialización: $A=1, Z_1=X_1$
Mientras queden patrones por asignar
Obtener el siguiente patrón X y calcular $d(X,Z_i)i=1...A$
Asignar X el al agrupamiento más cercano Z_i si $d(X,Z_i)\leq\theta\tau$
Formar un nuevo agrupamiento con X si $d(X,Z_i)>\tau$: $A++$, $Z_A=X$
Recalcular el centroide Z_i y la varianza de C_l del agrupamiento
Fin

Este método cómo es posible apreciar en el Algoritmo previo, necesita de 2 variables ajenas al conjunto de patrones por agrupar, dentro de cada iteración solo toma un patrón para buscar su centroide correspondiente, por esto se propone más adelante la modificación de este algoritmo.

1.2.1.2.2.3 Algoritmo de Batchelor y Wilkins

Este algoritmo, propuesto por Batchelor y Wilkins [31], es llamado también de máxima distancia. Se trata de un método heurístico incremental que emplea un único parámetro "f". Este si bien introduce una mejora en la clasificación, agrega tiempo de cálculo ya que todos los patrones que no han sido clasificados hasta el momento son inspeccionados para ver cual puede convertirse en el próximo centro, esto implica que la cantidad de consultas sobre cada patrón es del orden del número de elementos a agrupar al cuadrado.

TABLA 1.2 ALGORITMO DE BATCHELOR Y WILKINS.

Parámetro

f Fracción de la distancia media entre agrupamientos

Algoritmo

Primer agrupamiento: Patrón escogido al azar
Segundo agrupamiento: Patrón más alejado del primer agrupamiento

Mientras se creen nuevos agrupamientos
Obtener el patrón más alejado de los agrupamientos existentes
(máximo de las distancias mínimas de los patrones a los agrupamientos)

Si la distancia del patrón escogido al conjunto de agrupamientos
es mayor que una fracción de la distancia media entre los agrupamientos,
crear un agrupamiento con el patrón seleccionado.

Asignar cada patrón a su agrupamiento más cercano

Fin

Es posible apreciar del Algoritmo anterior que este método solo utiliza una variable ajena al conjunto de patrones por agrupar y a diferencia del Método Adaptativo este utiliza todos los patrones por agrupar en cada iteración.

1.2.1.3 Aprendizaje SemiSupervisado

Hasta ahora, hemos visto que el aprendizaje se puede realizar teniendo datos acerca de la salida o sin ellos. El aprendizaje semisupervisado es una combinación del aprendizaje supervisado y no supervisado. Puesto que asignar etiquetas o clases a los datos puede ser muy costoso, se puede recurrir a la opción de usar a la vez un conjunto de datos etiquetados de tamaño pequeño y un conjunto más extenso de datos no etiquetados, mejorando así la construcción de los modelos. Esta técnica es la usada por el aprendizaje semisupervisado. En este método, se ha de tener en cuenta que no siempre los datos no etiquetados son de ayuda en el proceso de aprendizaje. Por lo general, se asume que los datos no etiquetados siguen la misma distribución que los etiquetados para que el uso de datos sin etiquetar sea útil.

Existen numerosos métodos de aprendizaje semisupervisado, los cuales ofrecen unas características determinadas. La manera de escoger el método más adecuado es viendo cuál de ellos se ajusta mejor a las necesidades del problema específico, algunos de estos métodos son el Coentrenamiento, K-means. Se puede encontrar información más detallada sobre estos y otros métodos de aprendizaje semisupervisado en [32].

1.2.1.3.1 Coentrenamiento

Un método de aprendizaje semisupervisado es el Coentrenamiento [33-34], el cual asume que el espacio de características se puede dividir en dos subconjuntos independientes de atributos. Cada subconjunto de atributos es suficiente para aprender un clasificador adecuado. Inicialmente, dos clasificadores separados son entrenados con los datos etiquetados de los dos subconjuntos respectivamente. Posteriormente, cada clasificador clasifica los datos sin etiquetar, y enseña al otro clasificador con algunos de los datos sin etiquetar de mayor fiabilidad o confianza. Cada clasificador es entonces re-entrenado con los nuevos ejemplos entregados por el otro clasificador. El proceso se repite hasta que los dos clasificadores estén de acuerdo, tanto en los datos sin etiquetar, como en los etiquetados.

1.2.1.3.2 K-means

El método de K-means [35-36], es uno de los más usados en aplicaciones científicas e industriales. El nombre viene porque representa cada uno de los grupos por la media de sus puntos, es decir por su centroide, usa como función criterio una función de error cuadrático. En el algoritmo siguiente se muestra el pseudocódigo de este método.

TABLA 1.3 ALGORITMO DE AGRUPAMIENTO K-MEANS.

1.-Definir el numero de conjuntos
2.-Asignar aleatoriamente los k conjuntos iniciales
3.-Calcular los centroides para cada uno de los conjuntos
4.-Asignar a cada punto el conjunto cuyo centroide se encuentre más cerca.
5.-Regresar al paso 2 hasta que ya no sufran cambios los centroides.
6.-Fin

A lo largo de esta sección se han mencionado los diferente tipos de aprendizaje automático que existen, así como algunas de las técnicas más usadas dentro de este, cada una de estas con diferentes características, las cuales ayudaran a resolver problemas específicos.

1.2.2 Aplicaciones del aprendizaje automático.

Hasta el momento se ha estudiado el significado del aprendizaje automático y se han visto algunas técnicas de este dentro de sus distintos tipos. El objetivo final del aprendizaje automático es poder realizar nuevas aplicaciones que ayuden y complementen las aplicaciones que existen al día de hoy.

El aprendizaje automático es una rama descendiente de la estadística y de la informática. Por tanto, son numerosas sus posibles aplicaciones, por ejemplo en el área de la informática, el aprendizaje automático está relacionado con aplicaciones tan diversas como el desarrollo de robots humanoides o Internet, un ejemplo más claro que todos hemos usado alguna vez, es el PageRank usado por Google, que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de las páginas web indexadas por un motor de búsqueda [37].

Por otro lado, en el campo de la estadística, el aprendizaje automático busca obtener conclusiones en el análisis de los conjuntos de datos. Estos métodos de aprendizaje automático no sólo se utilizan en el campo de la estadística y en la informática para desarrollar nuevas aplicaciones, se han abierto distintas aplicaciones en el área de la psicología, la neurociencia y áreas similares, donde el objetivo de estas, es en este caso estudiar el aprendizaje y las conductas en humanos y animales.

La biología y la medicina son también áreas de aplicación de este ya que es posible utilizarlo en la clasificación de secuencias de ADN o en el estudio de enfermedades y su posible predicción.

Por último, el aprendizaje automático también puede ser aplicado en los campos de la economía, donde tendría la posibilidad de adaptarse u optimizar automáticamente su entorno. Así, una posible aplicación en la economía seria el estudio

de los mercados, la búsqueda de clientes o la detección de fraudes; por otro lado en los sistemas de control, el aprendizaje automático puede ser aplicado en sistemas, los cuales pueden mejorar su estrategia a través de la experiencia.

1.3 Lógica Difusa

La lógica Difusa[38] tiene como objetivo la manipulación de información imprecisa, esta será tratada a través de un conjunto difuso, esta nueva manipulación de información puede ser aplicada problemas similares de redes neuronales, resultando especialmente interesante para problemas no lineales o bien no definidos. De la misma manera, los sistemas difusos permiten modelar cualquier proceso no lineal, y aprender de los datos haciendo uso de determinados algoritmos de aprendizaje. No obstante, a diferencia de las redes neuronales, los basados en lógica borrosa permiten utilizar fácilmente el conocimiento de los expertos en un tema, bien directamente, bien como punto de partida para optimización automática, al formalizar el conocimiento a veces ambiguo de un experto de forma realizable. Además, gracias a la simplicidad de los cálculos necesarios, normalmente pueden realizarse en sistemas baratos y rápidos.

1.3.1 Historia de la logica difusa

En la universidad de Berkeley en California, el Ingeniero Zadeh escribió el siguiente principio de incompatibilidad "Conforme a la complejidad de un sistema aumenta, nuestra capacidad nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral mas allá del cual, la precisión y el significado son características excluyentes", ese momento fue cuando se hablo por primera vez de conjuntos difusos. Este nuevo concepto no es mas que la idea de que los elementos sobre los que se basa el pensamiento humano no son numero sino etiquetas lingüísticas. Esta idea es la que permite que se pueda representar el conocimiento, que es principalmente lingüístico de tipo cualitativo y no tanto cuantitativo, en un lenguaje matemático mediante los conjuntos difusos y funciones características asociadas a ello. Esto no quiere decir que exclusivamente se trabaje con números, este lenguaje nos permite trabajar con datos numéricos pero también con términos lingüísticos que aunque son más imprecisos que los números, muchas veces son más fáciles de entender para el razonamiento humano.

Después en la década de los 70's se dio otro paso importante para el desarrollo de la lógica difusa, en universidades de Japón se crearon varios grupos de investigación que hicieron grandes contribuciones sobre las aplicaciones que podía tener este tipo de lógica. De esta forma se consiguió crear el primer controlador difuso para una máquina de vapor o crear un controlador de inyección de química en depuradoras de agua. Es importante señalar que actualmente múltiples productos de consumo cotidiano usan esta lógica como lo muestra la siguiente figura.

FIGURA 1.10 EJEMPLOS DE ELEMENTOS DE LA VIDA DIARIA QUE UTILIZAN LOGICA DIFUSA.

En las siguientes décadas esta teoría fue teniendo más éxito con el descubrimiento de nuevas aplicaciones[39]. En la década de los ochenta, la investigación se orientó hacia las redes neuronales y su similitud con los sistemas difusos. Estos sistemas difusos utilizan métodos de aprendizaje basados en redes neuronales para identificar y optimizar sus parámetros. Para la década de los noventa, la investigación de las redes neuronales y los sistemas difusos da origen a los algoritmos genéticos, dando pie a una herramienta de trabajo muy potente en sistemas de control.

1.3.2 Conjuntos Difusos

Como lógica multivaluada, en la definición de grados de pertenencia, la lógica difusa emplea valores continuos entre 0 (que representa hechos totalmente falsos) y 1 (totalmente ciertos). Así, la lógica binaria clásica puede verse como un caso particular de la lógica difusa. Zadeh propone en 1965 por primera vez la noción de Conjunto Difuso [40]. Este hecho marca el principio de una nueva teoría denominada Teoría de Conjuntos Difusos[41-42]. Los conceptos se asocian a conjuntos difusos (asociando los valores de pertenencia) en un proceso llamado fuzzificación. Una vez que tenemos los valores fuzzificados podemos trabajar con reglas lingüísticas y obtener una salida, que podrá seguir siendo difusa o defuzzificada para obtener un valor discreto (VD). De este modo, a diferencia de la teoría clásica de conjuntos que se basa en el principio básico de la lógica de forma que un individuo pertenece o no pertenece a un conjunto, la idea básica de un conjunto difuso es que un elemento forma parte de un conjunto con un determinado grado de pertenencia. Con esto una proposición no es totalmente (sino parcialmente) cierta o falsa. Este grado se expresa mediante un entero en el intervalo [0, 1]. Un ejemplo claro es la representación de la altura de una población de individuos.

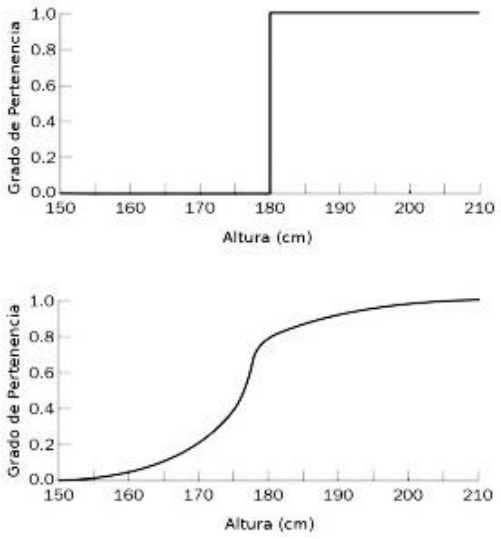


FIGURA 1.11 DESCRIPCION DE CONJUNTOS DISCRETOS(ARRIBA) Y CONJUNTOS DISCRETO(ABAJO).

En la figura anterior en la parte de arriba que corresponde a los valores discretos, podemos apreciar una línea en la altura de 180 cm, esta separa de manera clara a las personas altas de las que no la son, con esto se asocia un valor de pertenecía estricto como se puede apreciar en la tabla 1. Por otro lado en la parte de abajo vemos que no tenemos una separación de manera clara, este conjunto difuso nos permite expresar la altura de una persona, con un grado de pertenencia al conjunto de las personas altas, como lo podemos apreciar también en la tabla 1.

TABLA 1.4 EVALUACION DE ALTURA PARA CONJUNTOS DISCRETO Y DIFUSO.

Persona	Altura	Valor Discreto	Valor difuso(Grado de pertenencia)
1	2.05	1	1
2	1.95	1	1
3	1.87	1	0.95
4	1.80	1	0.82
5	1.79	0	0.71
6	1.6	0	0.36

Con la figura 1.12 y las evaluaciones que muestra la tabla 1, podemos notar que los conjuntos difusos generan una transición suave entre los limites de un conjunto discreto, a la par que salta un nuevo termino conocido como el "Universo del discurso" que se refiere a los posibles valores que puede tomar una variable, para nuestro ejemplo de la altura seria desde 150 cm hasta 210 cm.

Como pudimos apreciar en el ejemplo anterior la teoría de conjuntos difusos intenta desarrollar una serie de conceptos para tratar de un modo sistemático el tipo de imprecisión que aparece cuando los límites de las clases de objetos no están claramente definidos. Un conjunto difuso puede definirse como una clase en la que hay una progresión gradual desde la pertenencia al conjunto hasta la no pertenencia; o visto de otra forma, en la que un objeto puede tener un grado de pertenencia definido entre la pertenencia total (valor uno) o no pertenencia (valor cero). A continuación vamos a definir a conjunto difuso de manera mas seria.

Un conjunto difuso puede definirse de forma general como un conjunto con límites difusos. Sea X el Universo del discurso, y sus elementos se denotan como x. En la teoría clásica de conjuntos discretos se define un conjunto C, se define sobre X mediante la función característica de C como f_c.

$$f_c(x) = \begin{cases} 1 \; cuando \; x \in C \\ 0 \; cuando \; x \in C \end{cases} \tag{1.16}$$

Este conjunto mapea el universo X en un conjunto de dos elementos, donde la función f_C (x) es 1 si el elemento x pertenece al conjunto C y 0 si el elemento x no pertenece al conjunto C. Si generalizamos esta función para que los valores asignados a los elementos del conjunto se encuentren en un rango particular y así indicar el grado de pertenencia de los elementos a ese conjunto, tendremos una función de pertenencia de un determinado conjunto difuso. La función de pertenencia μ_A por la que se define un conjunto difuso A, sería:

$$\mu_A = x \rightarrow [0,1] \tag{1.17}$$

Donde $\mu_A(x) = 1$ si x está totalmente en A, $\mu A(x) = 0$ si x no está en A y $0 < \mu A(x) < 1$ si x está parcialmente en A. Este valor entre 0 y 1 representa el grado de pertenencia de un elemento x a un conjunto A. Así, el intervalo de la ecuación anterior es de números reales e incluye los extremos. Aunque [0, 1] es el rango de valores más utilizado para representar funciones de pertenencia, cualquier conjunto arbitrario con alguna ordenación total o parcial podría ser utilizado.

1.3.3 Operaciones con Conjuntos Difusos

Al igual que en los conjuntos discretos las tres operaciones básicas son: complemento, unión e intersección, pueden generalizarse de varias formas en conjuntos difusos. No obstante, existe una generalización particular que tiene especial importancia. Cuando se restringe el rango de pertenencia al conjunto [0, 1], estas operaciones "estándar" sobre conjuntos difusos se comportan de igual modo que las operaciones sobre conjuntos discretos, la descripción grafica de cada una de las operaciones anteriormente mencionadas se muestra la siguiente figura:

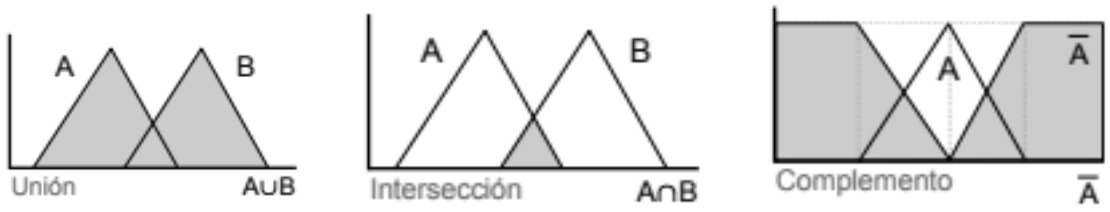


FIGURA 1.12 REPRESENTACION GRAFICA DE LAS OPERACIONES EN CONJUNTOS DIFUSOS.

1.3.4 Propiedades de los conjuntos difusos

Los conjuntos difusos y los conjuntos discretos tienen las mismas propiedades que son las siguientes:

- Conmutativa $A \cap B = B \cap A$
- Asociativa $A \cup (B \cup C) = (A \cup B) \cup A$
- Distributiva $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- Idempotencia $A \cup A = A \; y \; A \cap A = A$

1.3.5 Variables lingüísticas

Una variable lingüística [4] es aquella cuyos valores son palabras o sentencias en un lenguaje natural. De esta forma, una variable lingüística sirve para representar cualquier elemento que sea demasiado complejo, o del cual no tengamos una definición concreta; es decir, lo que no podemos describir en términos numéricos. Así, una variable lingüística está caracterizada por una quíntupla (X, T(X), U, G, M)

- X es el nombre de la variable.
- T(X) es el conjunto de términos de X; es decir, la colección de sus valores lingüísticos (o etiquetas lingüísticas).
- U es el universo del discurso (o dominio subyacente). Por ejemplo, si la hablamos de temperatura "Cálida" o "Aproximadamente 25o ", el dominio subyacente es un dominio numérico (los grados centígrados).
- G es una gramática libre de contexto mediante la que se generan los términos en T(X), como podrían ser "muy alto", "no muy bajo", etc.
- M es una regla semántica que asocia a cada valor lingüístico de X su significado M(X) (M(X) denota un subconjunto difuso en U).

Los símbolos terminales de las gramáticas incluyen:

- Términos primarios: "bajo", "alto", etc.
- Modificadores: "Muy", "más", "menos", "cerca de", etc.
- Conectores lógicos: Normalmente NOT, AND y OR.

Normalmente se definen los conjuntos difusos de los términos primarios y, a partir de éstos, se calculan los conjuntos difusos de los términos compuestos (por ejemplo, con "muy" y "alto" construimos el término compuesto "muy alto"). Una etiqueta lingüística se forma como una sucesión de los símbolos terminales de la gramática: "Muy alto, no muy bajo...". Un uso habitual de las variables lingüísticas es en reglas difusas.

Ejemplo: IF duración-examen IS larga THEN probabilidad-aprobar IS small. Por ejemplo, la variable lingüística velocidad podrías incluir conjuntos difusos como muy lento, lento, medio, rápido, muy-rápido. Naturalmente cada uno de estos conjuntos

representan un valor lingüístico que puede tomar la variable, a continuación se muestran las graficas de algunos modificadores con sus respectivas ecuaciones matemáticas.

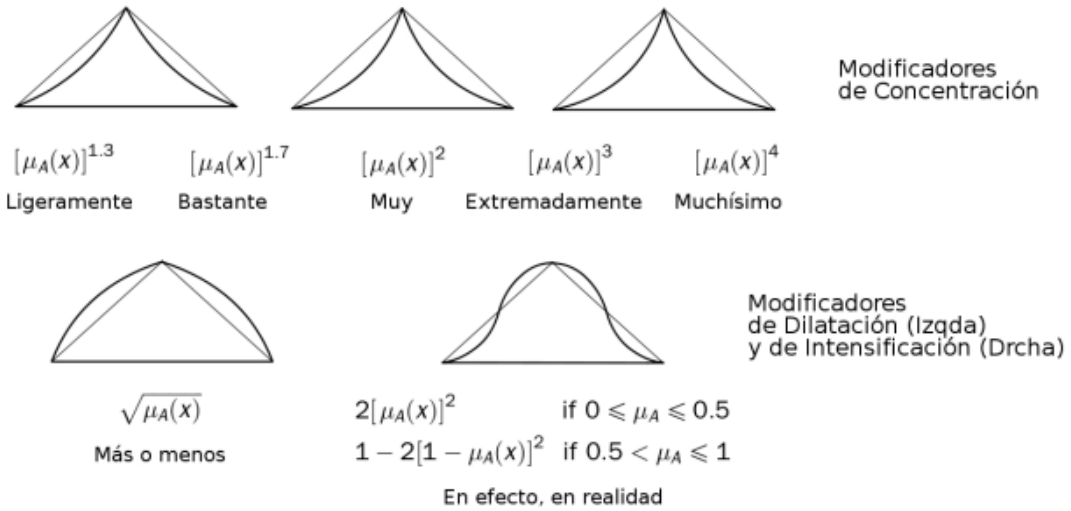


FIGURA 1.13 MODIFICADORES CON REPRESENTACION GRAFICA Y SUS ECUACIONES..

1.4 Bases de Datos

Dentro de la literatura existen muchas bases de datos para trabajar con expresiones faciales se decidió utilizar 2 bases de datos, ambas son a color lo cual es un requisito primordial para los algoritmos planteados en esta tesis, a continuación se dan las especificaciones de cada una de ellas.

1.4.1 Base de Datos KDEF

The Karolinska Directed Emotional Faces (KDEF) [43] es un conjunto de 4900 imágenes de expresiones faciales humanas. El material fue desarrollado en 1998 por Daniel Lundqvist y Jan-Eric Litton en el Instituto Karolinska, Departamento de Clinical Neuroscience, Section of Psychology, Estocolmo, Suecia.

La base de datos [10] consta de 70 sujetos (35 mujeres/35 hombres) que realizan 7 expresiones faciales (Miedo, Enojo, Molestia, Feliz, Neutral, Triste y Sorpresa), con 5 ángulos distintos (-90°,-45°, 0°, 45°, 90°). Para la realización de esta base de datos todos los sujetos usaron camisas grises, el fondo es uniforme y fueron sentados a una distancia aproximada de 3 metros de la cámara, aunque la distancia fue adaptada para cada sujeto ajustando la posición de la cámara con el objetivo de que los ojos y la boca se encontraran en una posición vertical y horizontal definida con anterioridad, la iluminación consistió en una luz suave indirecta distribuida de manera uniforme a lo largo del rostro.

1.4.2 Base de Datos HOHA

Esta base de datos "Hollywood Human Actions" [44] es de videos donde se tiene 8 clases principalmente que son acciones que realiza el protagonista del video, estas acciones son contestar el teléfono, salir de un auto, estrechar la mano, abrazar, besar, sentarse, pararse y levantarse .

Cuenta con 430 videos con una resolución entre 400x300 pixeles y 300x200 pixeles, la cámara no esta estática, los videos no presentan un fondo fijo y hay oclusión de las personas en algunos casos.

1.5 Conclusiones

En este capitulo se presentan el estado del arte de esta tesis, puedo concluir que aunque el reconocimiento de expresiones faciales no es algo reciente en la literatura, desde su inicio se han presentado varios problemas a resolver, como los diferentes fondos, la iluminación etc. Con esto podemos identificar que problemas son los que se buscan resolver en esta tesis.De la misma forma se presentan los diferentes tipos de Aprendizaje que existen, haciendo énfasis en 2 métodos Clustering(Ordenamiento) y Lógica Difusa, con lo cual podemos concluir que la unión de estos 2 métodos nos dará como resultado la propuesta de un nuevo clasificador, aprovechando las ventajas del clustering.

Capítulo 2 Sistemas Propuestos

En la sección anterior se han descrito los problemas con los que se enfrenta el reconocimiento de expresiones faciales, a continuación en los siguientes subcapítulos se explicara los aportes que esta tesis ha realizado a este campo de estudios de la siguiente forma, 3.1 Detector de perfil del rostro, 3.2 Segmentación automática de las regiones de interés del rostro y por ultimo 3.3 la propuesta de un clasificador basado en técnicas de clustering y lógica difusa.

2.1 Detector de perfil del rostro

Para realizar la detección del perfil del rostro seguiremos el diagrama a bloques que se muestra en la siguiente figura.

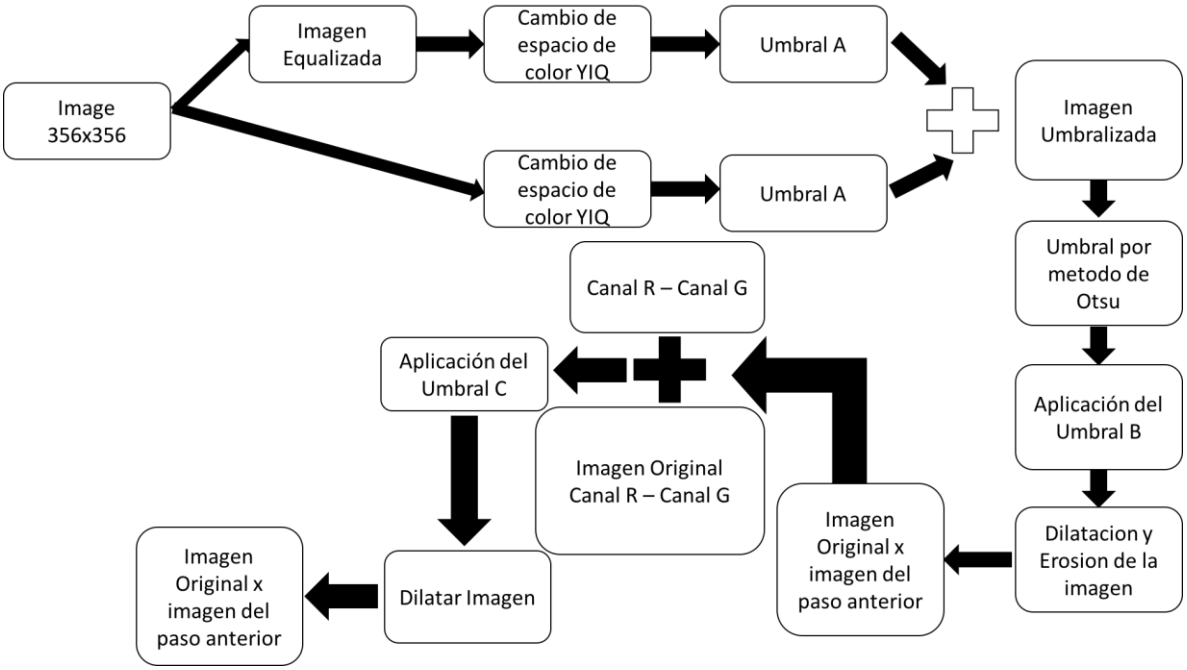


FIGURA 2.1 DIAGRAMA A BLOQUES DEL DETECTOR DE PERFIL.

Es muy importante señalar que en este punto nosotros ya tenemos una imagen de un rostro después de haber aplicado el algoritmo de Viola jones como lo muestra la figura (2.2-A), lo primero que necesitamos es redimensionar la imagen a el tamaño de 356x356 pixeles esto se debe a que en imágenes mas pequeñas el reconocimiento se dificulta. Realizaremos un cambio de espacio de color de la imagen usando a YIQ[45] usando las ecuaciones (2.1-2.3)

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B, \quad (2.1)$$

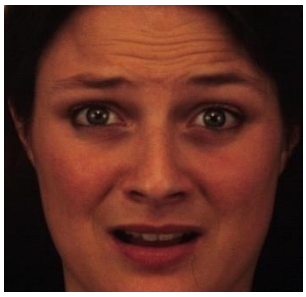
$$I = 0.5960 * R - 0.2740 * G - 0.3220 * B, \quad (2.2)$$

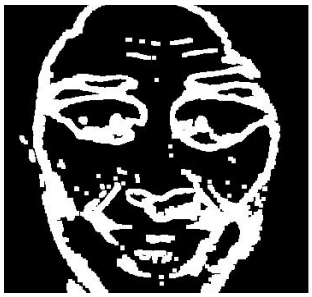
$$Q = 0.2110 * R * 0.5230 * G + 0.3210 * B. \quad (2.3)$$

El siguiente paso es aplicar el umbral A[46], a la imagen en RGB y en YIQ, dicho umbral esta definido en la ecuación

$$(60 < Y < 200) AND (20 < I < 50). \quad (2.4)$$

Después ambas imagines umbralizadas son sumadas dando como resultado la figura (2.2-B), con esto una vez mas solo tenemos una imagen, a esta imagen le aplicaremos el Método de Otsu[47], este método se encarga de buscar el umbral optimo para lograr desarrollar una binarización de la imagen, dando como resultado la figura (2.3-A) para obtener un nuevo umbral que le aplicaremos a dicha imagen





(A) (B)

FIGURA 2.2 (A) IMAGEN ORIGINAL, (B) IMAGEN RESULTANTE.

Ahora, la imagen original es multiplicada por la imagen dilatada mostrada en la figura (3.3-B), dando como resultado la imagen mostrada en la figura (3.3-C), esta ultima es descompuesta en sus 3 canales RGB, con el objetivo de realizar la resta del canal R y el canal G, dando como resultado la imagen $I_T(x,y)$ mostrada en la figura (3.4-A), a continuación se realiza la resta de los canales R y G de la imagen Original dando como resultado la figura(3.4-B), esta imagen es sumada a la obtenida en el paso anterior dando como resultado la imagen mostrada en la figure (3.5-c).

En el siguiente paso nosotros usaremos un valor de 75 para el umbral C, esto se debe a que si usamos otro umbral podemos llegar a perder información importante como lo es los ojos o la boca, a parte de que el ancho del rostro disminuye, en la figura 3.5 podemos ver una comparativa entro un umbral menor a 75 y 75

(A) (B) (C)

FIGURA 2.3 A) IMAGEN DESPUÉS DE UMBRALIZAR CON EL MÉTODO DE OTSU, (B) IMAGEN DILATADA (C) RESULTADO DEL PRODUCTOR ENTRE LA IMAGEN ORIGINAL Y LA IMAGEN DILATADA.

(A) (B) (C)

FIGURA 2.4 (A) RESTA DE LOS CANALES R Y G DE LA IMAGEN UMBRALIZADA (B) RESTA DE LOS CANALES R Y G DE LA IMAGEN ORIGINAL(C) RESULTADO DE LA SUMA DE A Y B.

(A) (B)

FIGURA 2.5(A) IMAGEN CON UMBRAL APLICADO DE 75, (B) IMAGEN CON UMBRAL APLICADO DE 30.

Después de haber aplicado el umbral C, la imagen es dilatada[48] 4 veces con el objetivo de llenar los espacios en blanco, esto nos da como resultado una imagen mostrada en la figura (3.6-A) que será una mascara que aplicaremos a la figura original con el objetivo de eliminar las partes de la imagen que no son piel.

(a) (b)

FIGURA 2.6 (A) MASCARA FINAL, (B) IMAGEN FINAL.

A continuación la tabla 2.1 muestra algunos resultados después de haber evaluado el algoritmo previamente explicado.

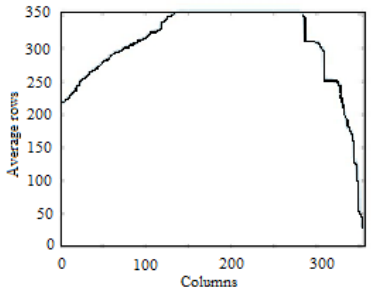
TABLA 2.1 RESULTADOS DEL ALGORITMO DE DETECTOR DE PERFIL.

Imagen Original	Imagen Final	Imagen Original	Imagen Final	Imagen Original	Imagen Final

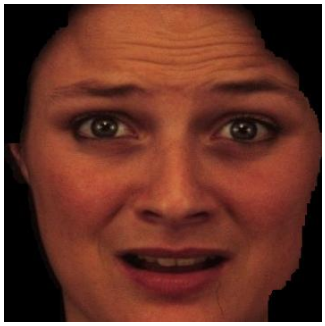
Con los resultados mostrados en la tabla 2.1 notamos que el algoritmo propuesto anteriormente es capaz de eliminar la mayoría del fondo de la imagen, ahora aplicaremos una integral proyectiva[49] para poder caracterizar las imágenes obtenidas. Una integral proyectiva es la suma de los valores de cada pixel de una imagen a lo largo de las filas o columnas[16]. Para nuestro caso de columnas, usaremos esta técnica con el fin de realizar la detección del perfil del rostros.

TABLA 2.2 (A) PERFIL IZQUIERDO, (B) INTEGRAL PROYECTIVA DEL PERFIL IZQUIERDO (C) PERFIL FRONTAL, (D) INTEGRAL PROYECTIVA DEL PERFIL FRONTAL (E) PERFIL DERECHO, (F) INTEGRAL PROYECTIVA DEL PERFIL DERECHO.

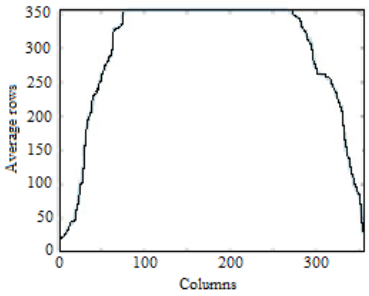
(A)



(B)

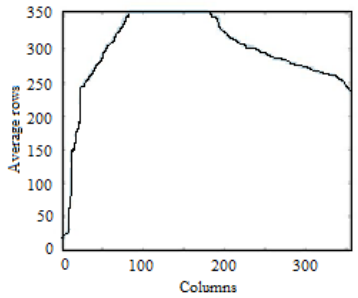


(C)



(D)

(E)



(F)

Para entrenar el sistema detector de perfil, se utilizo la base de datos KDEF descrita en la sección 2.4, con esta base de datos se creo una matriz que contiene los vectores característicos obtenidos a través de las integrales proyectivas de sus imágenes, a esta matriz se le aplico el algoritmo de PCA[50] para reducir sus dimensiones (este método se explicara en la siguiente sección), usamos un clasificador basado en clustering [51-52] se usaron 260 de entrenamiento y 230 de prueba dando como resultado los valores mostrado en la tabla 3.2.

TABLA 2.3 RESULTADOS DEL DETECTOR DE PERFIL.

Izquierda	Frontal	Derecha	Promedio
92.85	98.57	90	93.8
			

2.2 Segmentación Automática de las regiones de interes del rostros

2.2.1 Ajuste de las dimensiones de rostro

El rostro extraído de la imagen a través del Algoritmo de Viola Jones, contiene fondo en algunas ocasiones para la mujeres puede ser cabello o en los hombres el fondo con el que se tomo la fotografía, así como las orejas que no contienen información relevante para el reconocimiento de las expresiones faciales, con el objetivo de eliminar este problema que más adelante en la extracción de características puede convertirse en ruido y disminuir el porcentaje de reconocimiento del sistema propuesto, se realizo lo siguiente; se divide la imagen en sus 3 canales Rojo, Verde y Azul, después se realiza la resta de los canales Rojo y Verde con el objetivo de realzar la piel como se puede ver el Fig. 3.7.b, el siguiente paso es binarizar la imagen obtenida de la resta de los canales a partir de la ecuación 1

$$I(x,y) = \begin{cases} \mathbf{0}, & I(x,y) < 1 \\ \mathbf{255}, & I(x,y) \geq \mathbf{1} \end{cases}. \tag{2.5}$$

FIGURA 2.7.DE IZQUIERDA A DERECHA, IMAGEN ORIGINAL (A), RESTA DE CANALES (B) E IMAGEN BINARIZADA(C)

Ya que se tiene la imagen binarizada como se muestra en la Fig. 3.7.c, se utilizaran los momentos de una imagen [49] que se definen a partir de la ecuación 2.

$$M_{p,q} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^p y^q I(x,y). \quad (2.6)$$

Donde I(x, y) es la intensidad de la imagen en la posición (x, y), N es el número de columnas y M el número de filas de la imagen, siendo p y q quienes definen el momento de la imagen, es decir si se quisiera obtener $M_{2,1}$ la ecuación quedaría como se muestra a continuación.

$$M_{2,1} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^2 y^1 I(x,y). \quad (2.7)$$

Ya que sabemos cómo se calculan los momentos de una imagen, podemos calcular su centro de gravedad [53] basados en dichos momentos como lo muestra las ecuaciones 2.8 y 2.9.

$$X_c = \frac{M_{1,0}}{M_{0,0}}, \quad (2.8)$$

$$Y_c = \frac{M_{0,1}}{M_{0,0}}. \quad (2.9)$$

A continuación definiremos 3 variables intermedias a, b y c, en las ecuaciones 3.10, 3.11 y 3.12 respectivamente.

$$a = \frac{M_{2,0}}{M_{0,0}} - {X_c}^2 \quad (3.10)$$

$$b = 2\left(\frac{M_{1,1}}{M_{0,0}} - X_c Y_c\right) \quad (3.11)$$

$$c = \frac{M_{0,2}}{M_{0,0}} - {Y_c}^2 \quad (3.12)$$

Ya que hemos definido las variables intermedias, encontraremos el ancho del rostro basándonos en la ecuación 2.13 [44].

$$W = 2\sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}. \quad (2.13)$$

Para encontrar el límite izquierdo de la imagen usamos la ecuación descrita en 3.14.

$$\lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil \quad (2.14)$$

Nota:$\lceil a \rceil$esta operación redondea “a” hacia el entero superior.

Para encontrar el límite derecho solo es necesario sumar el ancho en tal forma

$$[x_c] - \frac{[W]}{2} + [w] \tag{2.15}$$

Para encontrar el límite superior , usaremos la ecuación descrita en 2.16.

$$[y_c] - 0.84\frac{[W]}{2} \tag{2.16}$$

A partir de las ecuaciones anteriores es posible realizar el recorte de rostro como lo muestra la figura 3.

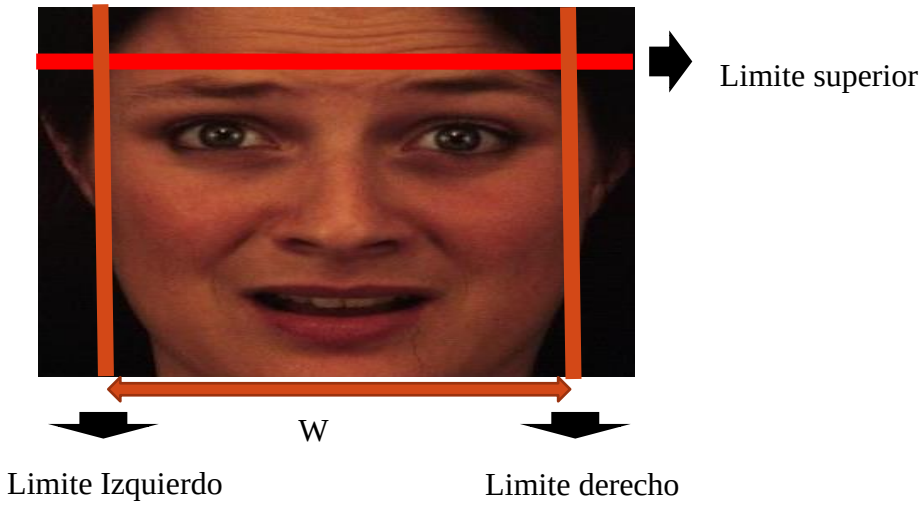


FIGURA 2.8. REGIÓN A SEGMENTAR.

2.2.2 Segmentación de la regiones de interés del rostro

Las regiones de interés se definieron tomando en cuenta el movimiento de los músculos, para cada expresión facial como se puede observar en la Fig. 3.9, por esto se toma la decisión de segmentar la región de la boca y Frente/ojos, fue posible realizar este análisis gracias al software de **ARTNATOMY** [13] . En ambos casos para realizar la segmentación de las regiones de interés propuestas, partiremos del hecho que los rostros mantienen una relación simétrica entre el ancho y largo del rostro, así como la posición de la boca y ojos como lo muestra la Fig. 3.10.

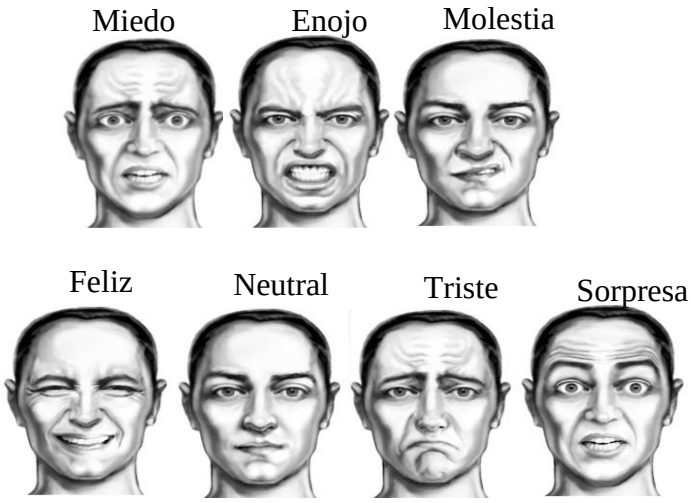


FIGURA 2.9. MOVIMIENTOS DE LOS MÚSCULOS A DIFERENTES EXPRESIONES FACIALES.

FIGURA 2.10. RELACIÓN SIMÉTRICA DEL ROSTRO.

2.2.2.1 Segmentación de la región de Frente/ojos

Para realizar la segmentación de región de Frente/ojos, tomaremos la imagen y será divida en 3 regiones a lo alto (A, B y C) como lo muestra la Fig. 3.10. , tomando la región A de la Fig. 5 como nuestra región de interés,

2.2.2.2 Segmentación de la región de la boca

Para realizar la segmentación de región de la boca, tomaremos la imagen y será divida en 3 regiones a lo alto, tomando la región C de la Fig. 3.10 como nuestra región de interés, pero a diferencia de la región de Frente/ojos donde a todo lo

ancho tenemos región de interés, debido a la frente, aquí es necesario realizar una segmentación para obtener solo la región de la boca.

Al igual que anteriormente para reducir la región del rostro tomaremos los canales Rojo y Verde de la imagen, para restarlos entre ellos, después se realizo un ecualización del histograma [54-55] de dicha imagen obtenida anteriormente, dando como resultado una imagen como la que muestra la Fig. 6.

FIGURA 2.11. IMAGEN ECUALIZADA DE LA RESTA DE LOS CANALES R Y G.

El siguiente paso para la segmentación automática de la boca es obtener la integral proyectiva [49], esta consiste en la suma de los valores de los pixeles, para nuestro caso será a lo ancho; es decir por cada columna obtendremos el promedio de sus filas, esta es una integral proyectiva horizontal, con esto obtendremos un vector de lo ancho de la imagen que se analiza. En la figura 3.13 se muestra la grafica del vector obtenido

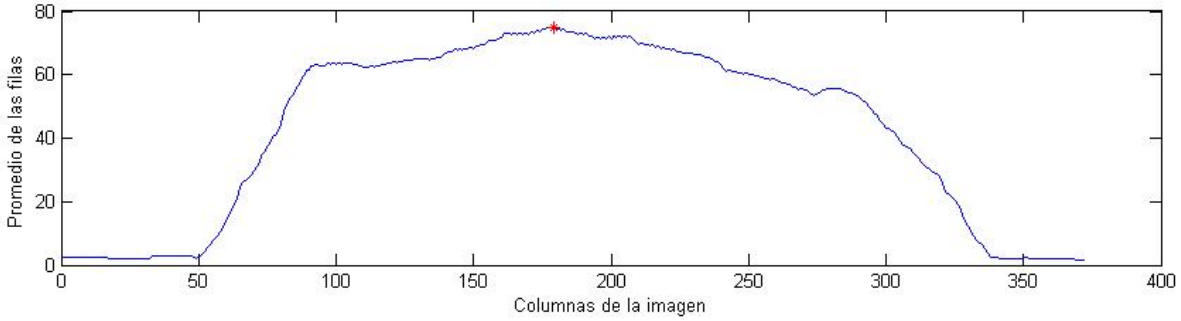


FIGURA 2.12. GRAFICA DE LA INTEGRAL PROYECTIVA HORIZONTAL.

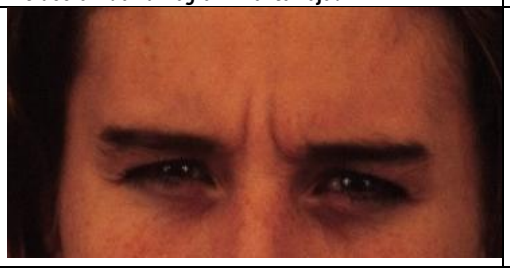
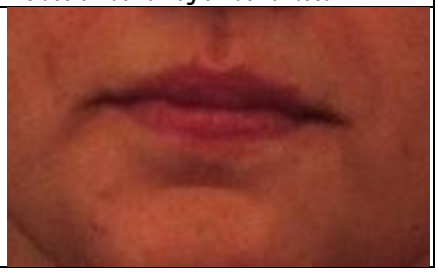
Ya que se tiene el vector de la integral proyectiva, se buscara el valor máximo de este, que llamaremos a partir de este momento "D", Después a partir del centro de la imagen, se cortara la imagen "D" pixeles a su izquierda y "D" pixeles a su derecha, respetando la altura original de esta, como lo muestra la figura 3.13, con esto se extrae la región de interés de manera automática.

FIGURA 2.13. EXTRACCIÓN FINAL DE LA REGIÓN DE LA BOCA.

2.2.3 Ejemplos de extracción de las regiones de importancia

En la base de datos que se uso, en algunas ocasiones cambia la iluminación pero esto no afecta al sistema para la extracción automática de las regiones del rostro como se puedo apreciar en la tabla 3.4.

TABLA 2.4 ALGUNOS RESULTADOS OBTENIDOS CON LA EXTRACCIÓN AUTOMÁTICA DE LAS REGIONES DE INTERÉS DEL ROSTRO.

Imagen Original	Extraccion de la region Frente/ojos	Extraccion de la region de la boca
		

.

2.3 Propuesta de un clasificador basadoo en tecnicas de clustering y logica difusa

El clasificador propuesto opera de manera supervisada para generar varios clúster para cada clase. Se supone que se conoce el número de clases, I y el número de patrones de entrenamiento, K, para cada clase, aunque el número de clúster que se generarán para cada clase se desconoce de antemano. También es importante mencionar que en el clasificador propuesto los clúster de cada clase son entrenados independientemente de las otras clases. Por lo tanto, si se agrega una nueva clase, los clúster ya entrenados permanecen sin cambios, y es necesario estimar solo los clúster para la nueva clase.

2.3.1 Etapa de entrenamiento

Para desarrollar el algoritmo de entrenamiento, considere la siguiente matriz:

$$\mathbf{\Gamma} = \left[\mathbf{\Gamma}_0, \mathbf{\Gamma}_1, \mathbf{\Gamma}_2, ..., \mathbf{\Gamma}_i, ..., \mathbf{\Gamma}_I\right], \tag{2.17}$$

donde

$$\mathbf{\Gamma}_{\mathrm{i}} = \begin{bmatrix} \Gamma_{i,0,0} & \Gamma_{i,0,1} & \cdots & \Gamma_{i,0,B} \\ \Gamma_{i,1,0} & \Gamma_{i,1,1} & \cdots & \Gamma_{i,1,B} \\ \cdots & \cdots & \cdots & \cdots \\ \Gamma_{i,K,0} & \Gamma_{i,K,1} & \cdots & \Gamma_{i,K,B} \end{bmatrix}. \tag{2.18}$$

Indica las características de los vectores de la *i-sima* clases, $\Gamma_{i,n,m}$ corresponde a la *m-sima* componente del *n-simo* vector característicos de la *i-sima* clase, y *B=NM*. Después del entrenamiento del clasificador, el vector **V**(i) contendrá la cantidad de clústeres creados para cada clase, la variable, γ denotara el numero total de clúster en el sistema y la *i-sima* clase será pondrá con valor cero. Debido a que los clúster perteneciente a cada clase se estiman de manera independiente, el entrenamiento del sistema empieza inicializando los clúster pertenecientes a la clase bajo análisis.

Paso 1. Inicializando el proceso

Considerando $\mathbf{Y} = \mathbf{\Gamma}_0$ e $y_{n,m} = \Gamma_{n,m}$, que es,

$$\mathbf{Y} = \begin{bmatrix} y_{0,0} & y_{o,1} & \cdots & y_{0,B} \\ y_{1,0} & y_{1,1} & \cdots & y_{1,B} \\ \cdots & \cdots & \cdots & \cdots \\ y_{L,0} & y_{L,1} & \cdots & y_{L,B} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_0^T \\ \mathbf{Y}_0^T \\ . \\ \mathbf{Y}_L^T \end{bmatrix}. \tag{2.19}$$

Después, asumimos que el índice del primer clúster es *p=0*, a continuación, construimos un vector N_c que tiene el mismo numero de elementos que los clúster inicializados con ceros, obteniendo lo siguiente,

$$N_C = \left[0, 0, 0, 0, ...0\right]. \tag{2.20}$$

A continuación construimos un vector $\mathbf{S}_{\mathrm{i}}$ que contendrá la varianza para cada clúster, también se inicializa con ceros.

$$\mathbf{S}_i = \left[0, 0, 0, 0, ...0\right], \tag{2.21}$$

Asignamos la primera fila de la matriz **Y** al valor del primer centroide:

$$\mathbf{C}(0, j) = \mathbf{Y}(0, j), \quad j = 0, 1, 2, 3..., B \tag{2.22}$$

Siguiente, se establece la matriz $\mathbf{Y}^{(1)}$ eliminando la primera fila de $\mathbf{Y}$, este fue usado para calcular el centroide del primer clúster como muestra la ecuación 3.21. Obteniendo,

$$\mathbf{Y}^{(1)} = \begin{bmatrix} y^{(1)}_{0,0} & y^{(1)}_{0,1} & .... & y^{(1)}_{0,B} \\ y^{(1)}_{1,0} & y^{(1)}_{1,1} & .... & y^{(1)}_{1,B} \\ .... & .... & .... & .... \\ y^{(1)}_{L_N,0} & y^{(1)}_{L_N,1} & .... & y^{(1)}_{L_N,B} \end{bmatrix}, \tag{2.23}$$

donde $\mathbf{Y}^{(1)}$ es una matriz de $L_N \times B$ y $L_N = L\text{-}1$.Despues, determine las filas $\mathbf{Y}^{(1)}$ que tiene la distancia máxima y mínimo respecto a C(0,j), que es

$$d_M = \max(d_k), \quad k = 0, 1, 2, ..., L_N, \tag{2.24}$$

y

$$d_m = \min(d_k), \quad k = 0, 1, 2, ..., L_N, \tag{2.25}$$

donde

$$d_k = \sum_{j=0}^{B} \left(\mathbf{Y}^{(1)}(k, j) - \mathbf{C}(0, j) \right)^2, \quad k = 0, 1, 2, ..., L_N \tag{2.26}$$

.

Luego, usando la m-sima fila de $Y^{(1)}$ (que corresponde a la distancia mínima), modifique $N_c(0)$, el centroide C(0,j), y la varianza $S_i(0,j)$ como se muestra a continuación:

$$Nc(0) = Nc(0) + 1, \tag{2.27}$$

$$\mathbf{C}(0, j) = \frac{Nc(0)-1}{Nc(0)}\mathbf{C}(0, j) + \frac{1}{Nc(0)}\mathbf{Y}^{(1)}(m, j) \quad (2.28)$$

,

$$\mathbf{S}(0, j) = \frac{Nc(0)-1}{Nc(0)}\mathbf{S}(0, j) + \frac{1}{Nc(0)}\left(\mathbf{Y}^{(1)}(m, j) - \mathbf{C}(0, j)\right)^2 \quad (2.29)$$

.

A continuación, usándola M-sima fila, el Segundo clúster de la primera clase se calcula como muestra la siguiente ecuación:

$$\mathbf{C}(1, j) = \mathbf{Y}^{(1)}(M, j), \quad j = 0, 1, 2, 3..., B \quad (2.30)$$

,

En la segunda iteración, primero se calcula la matriz $\mathbf{Y}^{(2)}$, borrando la m-sima and M-sima filas de la matriz $\mathbf{Y}^{(1)}$. De la siguiente forma,

$$\mathbf{Y}^{(2)} = \begin{bmatrix} y^{(2)}_{0,0} & y^{(2)}_{0,1} & .... & y^{(2)}_{0,B} \\ y^{(2)}_{1,0} & y^{(2)}_{1,1} & .... & y^{(2)}_{1,B} \\ .... & .... & .... & .... \\ y^{(2)}_{L_N,0} & y^{(2)}_{L_N,1} & .... & y^{(2)}_{L_N,B} \end{bmatrix}, \quad (2.31)$$

Donde $\mathbf{Y}^{(2)}$ es una matriz de $L_N \times B$ y $L_N = L-3$. Entonces, calculamos las filas de $\mathbf{Y}^{(2)}$ que tiene la distancia máxima y mínimo respecto a C(p,j) , p=0, 1 usando

$$d_{M,N} = \max(d_{k,r}), \quad k = 0, 1, 2,..., L_N; \quad r = 0, 1 \quad (2.32)$$

,

y

$$d_{m,n} = \min(d_{k,r}), \quad k = 0, 1, 2,..., L_N; \quad r = 0, 1 \quad (2.33)$$

,

donde m se refiere al índice de la fila $\mathbf{Y}^{(2)}$ y n es el índice de los centroides con la menor instancia entre ellos, mientas que M es el índice de la fila de $\mathbf{Y}^{(2)}$, y N es el índice del clúster con la distancia máxima entre ellos, y

$$d_{k,r} = \sum_{j=0}^{B}\left(\mathbf{Y}^{(2)}(k, j) - \mathbf{C}(r, j)\right)^2, \quad k = 0, 1, 2,..., L_N; \quad r = 1, 2 \quad (2.34)$$

.

A continuación, usando la m-sima fila, que corresponde a la mínima distancia, modificando $N_c(n)$, el centroide $\mathbf{C}(n,j)$, y la varianza $\mathbf{S}_i(n,j)$, donde n y m se calculan como muestra las siguientes ecuaciones:

$$Nc(n) = Nc(n) + 1, \tag{2.35}$$

$$\mathbf{C}(n, j) = \frac{Nc(n) - 1}{Nc(n)}\mathbf{C}(n, j) + \frac{1}{Nc(n)}\mathbf{Y}^{(2)}(m, j), \tag{2.36}$$

$$\mathbf{S}(n, j) = \frac{Nc(n) - 1}{Nc(n)}\mathbf{S}(n, j) + \frac{1}{Nc(n)}\left(\mathbf{Y}^{(2)}(m, j) - \mathbf{C}(n, j)\right)^2. \tag{2.37}$$

Después, incrementamos el numero de clúster en uno (que es, p=p+1) y usando la fila de $\mathbf{Y}^{(2)}$ cuya distancia con respecto a todos los clústeres es la mas grande, obtenga el valor del p-simo clúster:

$$\mathbf{C}(p, j) = \mathbf{Y}^{(2)}(M, j), \quad j = 0, 1, 2, 3..., B. \tag{2.38}$$

En la tercera iteración, se calcula la matriz $\mathbf{Y}^{(3)}$, borrando la *m-sima* y *M-sima* filas de $\mathbf{Y}^{(2)}$. obteniendo

$$\mathbf{Y}^{(3)} = \begin{bmatrix} y^{(3)}_{0,0} & y^{(3)}_{0,1} & .... & y^{(3)}_{0,B} \\ y^{(3)}_{1,0} & y^{(3)}_{1,1} & .... & y^{(3)}_{1,B} \\ .... & .... & .... & .... \\ y^{(3)}_{L_N,0} & y^{(3)}_{L_N,1} & .... & y^{(3)}_{L_N,B} \end{bmatrix}, \tag{2.39}$$

donde $\mathbf{Y}^{(3)}$ es una matriz de $L_N \times B$ y $L_N = L-5$.Despues, calculamos las filas de $\mathbf{Y}^{(3)}$ que son las distancias con C(p,j) , p=0, 1, 2, usando:

$$d_{M,N} = \max(d_{k,r}), \quad k = 0, 1, 2,..., L_N; \quad r = 0, 1, 2 \tag{2.40}$$

y

$$d_{m,n} = \min(d_{k,r}), \quad k = 0, 1, 2, ..., L_N; \quad r = 0, 1, 2 \tag{2.41}$$

donde m se refiere al índice de las filas $\mathbf{Y}^{(3)}$, y n denota el índice del centroide con la distancias mas pequeña, mientras M es el índice de $\mathbf{Y}^{(3)}$, y N el índice del clúster con la distancias mayor, y

$$d_{k,r} = \sum_{j=0}^{B} \left(\mathbf{Y}^{(3)}(k, j) - \mathbf{C}(r, j) \right)^2, \quad k = 0, 1, 2, ..., L_N; \quad r = 0, 1, 2 \tag{2.42}$$

Después, usando m-sima fila, que corresponde a la distancia mínima, modificar $N_c(n)$, el centroide C(n,j), y la varianza S(n,j), donde n y m están determinados como muestran las siguientes ecuaciones

$$Nc(n) = Nc(n) + 1, \tag{2.43}$$

$$\mathbf{C}(n, j) = \frac{Nc(n) - 1}{Nc(n)} \mathbf{C}(n, j) + \frac{1}{Nc(n)} \mathbf{Y}^{(3)}(m, j), \tag{2.44}$$

y

$$\mathbf{S}(n, j) = \frac{Nc(n) - 1}{Nc(n)} \mathbf{S}(n, j) + \frac{1}{Nc(n)} \left(\mathbf{Y}^{(3)}(m, j) - \mathbf{C}(n, j) \right)^2 . \tag{2.45}$$

Después, incrementamos el numero de clúster en uno (p=p+1), usando la fila de $\mathbf{Y}^{(3)}$ cuya distancia con todos los clúster en la *i-sima* clase es el mas grande, obteniendo el p-simo clúster como se muestra a continuación:

$$\mathbf{C}(p, j) = \mathbf{Y}^{(3)}(M, j), \quad j = 0, 1, 2, 3..., B. \tag{2.46}$$

Paso 2.

En general, Calculamos la matriz $\mathbf{Y}^{(t+1)}$, eliminando de $\mathbf{Y}^{(t)}$ el m-sima y M-sima filas obtenidos en el paso anterior y establecemos $L_N=L_N-2$. obteniendo

$$\mathbf{Y}^{(t+1)}=\begin{bmatrix} y_{0,0}^{(t+1)} & y_{0,1}^{(t+1)} & .... & y_{0,B}^{(t+1)} \\ y_{1,0}^{(t+1)} & y_{1,1}^{(t+1)} & .... & y_{1,B}^{(t+1)} \\ .... & .... & .... & .... \\ y_{L_N,0}^{(t+1)} & y_{L_N,1}^{(t+1)} & .... & y_{L_N,B}^{(t+1)} \end{bmatrix}, \tag{2.47}$$

Después, calculamos las filas de $\mathbf{Y}^{(t+1)}$ que tiene los valores mínimos y máximos respecto a C(p,j) , p=0, 1, …t+1 como se muestra:

$$d_{M,N}=\max(d_{k,r}),\quad k=0,1,2,...,L_N;\quad r=0,1,...,2t \tag{2.48}$$

y

$$d_{m,n}=\min(d_{k,r}),\quad k=0,1,2,...,L_N;\quad r=0,1,,...,2t, \tag{2.49}$$

donde m denota el índice de la fila $\mathbf{Y}^{(t+1)}$, y n es el índice del centroide con la distancia mas pequeña, mientras M el índice de $\mathbf{Y}^{(t+1)}$, y N el índice del clúster con la distancia máxima con todos los clúster pertenecientes a la i-sima clase:

$$d_{k,r}=\sum_{j=0}^{B}\left(\mathbf{Y}^{(t+1)}(k,j)-\mathbf{C}(r,j)\right)^2,\quad k=0,1,2,...,L_N;\quad r=1,2,...,2t \tag{2.50}$$

.

A continuación, usando la m-sima fila, que corresponde a la distancia mínima, modificamos $N_c(n)$, el centroide C(n,j), y la varianza S(n,j), se calculan:

$$Nc(n)=Nc(n)+1, \tag{2.51}$$

$$\mathbf{C}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{C}(n,j)+\frac{1}{Nc(n)}\mathbf{Y}^{(t+1)}(m,j) \tag{2.52}$$

,

y

$$\mathbf{S}(n, j) = \frac{Nc(n)-1}{Nc(n)}\mathbf{S}(n, j) + \frac{1}{Nc(n)}\left(\mathbf{Y}^{(t+1)}(m, j) - \mathbf{C}(n, j)\right)^2 . \tag{2.53}$$

Después usando la fila $\mathbf{Y}^{(t+1)}$ cuyas distancia con respecto a los clúster es la mas grande, obteniendo el valor del clúster p=p+1 como se muestra:

$$\mathbf{C}(p, j) = \mathbf{Y}^{(t+1)}(M, j), \quad j = 0, 1, 2, 3..., B . \tag{2.54}$$

si L_N>2, entonces ir al paso 2.

Después de que se han analizado los vectores pertenecientes a la *i-sima* clase, los centroides C_i y las varianza S_i de cada uno de los grupos de p se organizan como se muestra.

$$\mathbf{C}_i = \left[C^T_{N_{i-1}}, C^T_{N_{i-1}+1}, C^T_{N_{i-1}+2}, ..., C^T_{N_i} \right]^T \tag{2.55}$$

y

$$\mathbf{S}_i = \left[S^T_{N_{i-1}}, S^T_{N_{i-1}+1}, S^T_{N_{i-1}+2}, ..., S^T_{N_i} \right]^T , \tag{2.56}$$

donde $N_i = N_{i-1} + p$ se refiere al numero de clúster de *i-sima* etapa. A continuación, establecemos W(γ)=p, $\lambda = \lambda + p$ y $\gamma = \gamma + 1$. Si $\gamma < I$, donde *I* es el numero total de clases, volvemos al paso 1; por otro lado, los centros y las varianzas de todos los clúster pertenecientes a cualquiera de las clases I se organizan de la siguiente manera:

$$\mathbf{C} = \left[C^T_0, ..., C^T_{N_1}, C^T_{N_1+1}, ..., C_{N_2}, C^T_{N_2+1}, ..., C^T_{N_{\lambda-1}}, ..., C^T_{N_\lambda} \right]^T \tag{2.57}$$

y

$$\mathbf{S} = \left[S^T_0, ..., S^T_{N_1}, S^T_{N_1+1}, ..., S_{N_2}, S^T_{N_2+1}, ..., S^T_{N_{\lambda-1}}, ..., S^T_{N_\lambda} \right]^T , \tag{2.58}$$

donde N_{i+1}-N_i se refiere al numero de clúster de *(i+1)-sima* clase.

2.3.2 Etapa de evaluacion

Durante el entrenamiento del sistema se estima N_i-N_{i-1} clúster pertenecientes a la *i-sima* clase, donde $i=1,\ldots,\lambda_k$. Estos serán usados en la etapa de evaluación. En este punto cualquier se clasifica a través de los N_i-N_{i-1} clúster, el primer paso es que el Sistema evalúa si el patrón de entrada pertenece a alguno de los clúster. Después, utiliza esa información para determinar si el patrón de entrada perteneces a *j-sima* clase. Con esto, Los vectores característicos se estiman como muestra la siguiente ecuación.:

$$\mathbf{\Gamma} = \begin{bmatrix} \phi_0^{(0)}, \phi_1^{(0)}, \phi_2^{(0)} & ..\phi_r^{(0)}.. & . & \phi_{NM-1}^{(0)} \\ \phi_0^{(1)}, \phi_1^{(1)}, \phi_2^{(1)} & ..\phi_r^{(1)}.. & . & \phi_{NM-1}^{(1)} \\ \ldots & .. & . & .. \\ \phi_0^{(L)}, \phi_1^{(L)}, \phi_2^{(L)} & ..\phi_r^{(L)}.. & . & \phi_{NM-1}^{(L)} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ . \\ . \\ w_{NM-1} \end{bmatrix}, \tag{2.59}$$

A continuación la expresión es evaluada:

$$IF\ \mathbf{\Gamma} \in \mathbf{C}_{i,1}\ \text{OR}\ \mathbf{\Gamma} \in \mathbf{C}_{i,2}\ \text{OR}\ \mathbf{\Gamma} \in \mathbf{C}_{i,3}\ \text{OR}.....\mathbf{\Gamma} \in \mathbf{C}_{i,(N_i - N_{i-1})}\ \text{THEN}\ \mathbf{\Gamma} \in \rho_{c_i}, \tag{2.60}$$

donde ρ_{c_i} se refiera a la clase c_i . usando la ecuación 3.59 teniendo en cuenta que cada grupo se caracteriza por su centro y varianza, podemos evaluar el grado de pertenencia del patrón de entrada a la clase ρ_{c_i} utilizando el vector Γ dado por (3.58) de la siguiente manera:

$$\psi_{c_i} = \prod_{j=0}^{L} \left(\exp\left(\frac{(\Gamma(j) - C(i,j))^2}{2S^2(i,j)} \right) \right) \quad 1 \le i \le (N_i - N_{i-1}) \tag{2.61}$$

.

Entonces, el grado de membresía del patrón de entrada esta dado por:

$$\psi_{c_k} = \max\{\psi_{c_i}\}, \quad 1 \le i \le (N_i - N_{i-1});\ k = 1,2,\ldots,\rho_{N_{cl}}. \tag{2.62}$$

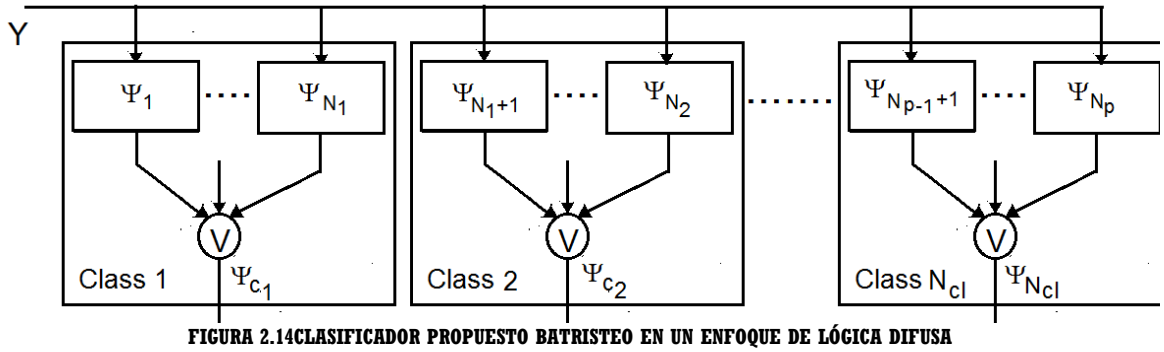


FIGURA 2.14CLASIFICADOR PROPUESTO BATRISTEO EN UN ENFOQUE DE LÓGICA DIFUSA

Los centroides estimados **G** y **F** y l varianza **S** y **V** de los clúster de cada ROI se dan a través de

$$\mathbf{G}=\left[\mathbf{C}_1^T,...,\mathbf{C}_{N_1+1}^T,\mathbf{C}_{N_1+2}^T,...,\mathbf{C}_{N_2}^T,\mathbf{C}_{N_2+1}^T,...,\mathbf{C}_{N_3}^T,...,\mathbf{C}_{N_{\lambda-1}}^T,...,\mathbf{C}_{N_\lambda}^T\right]^T, \tag{2.63}$$

$$\mathbf{F}=\left[\mathbf{F}_1^T,...,\mathbf{F}_{N_1+1}^T,\mathbf{F}_{N_1+2}^T,...,\mathbf{F}_{N_2}^T,\mathbf{F}_{N_2+1}^T,...,\mathbf{F}_{N_3}^T,...,\mathbf{F}_{N_{\lambda-1}}^T,...,\mathbf{F}_{N\lambda}^T\right]^T, \tag{2.64}$$

$$\mathbf{S}=\left[\mathbf{S}_1^T,...,\mathbf{S}_{N_1+1}^T,\mathbf{S}_{N_1+2}^T,...,\mathbf{S}_{N_2}^T,\mathbf{S}_{N_2+1}^T,...,\mathbf{S}_{N_3}^T,...,\mathbf{S}_{N_{\lambda-1}}^T,...,\mathbf{S}_{N_\gamma}^T\right]^T, \tag{2.65}$$

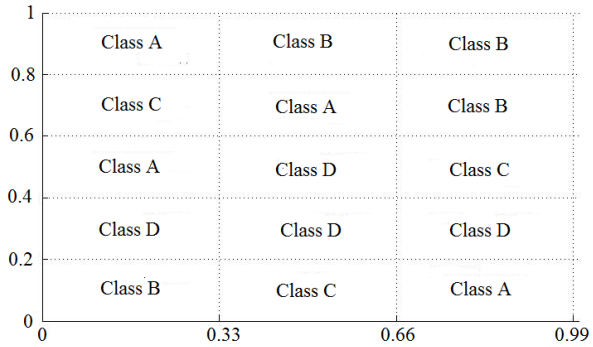
and

$$\mathbf{V}=\left[\mathbf{V}_1^T,...,\mathbf{S}_{N_1+1}^T,\mathbf{V}_{N_1+2}^T,...,\mathbf{V}_{N_2}^T,\mathbf{V}_{N_2+1}^T,...,\mathbf{V}_{N_3}^T,...,\mathbf{V}_{N_{\lambda-1}}^T,...,\mathbf{V}_{N_\lambda}^T\right]^T. \tag{2.66}$$

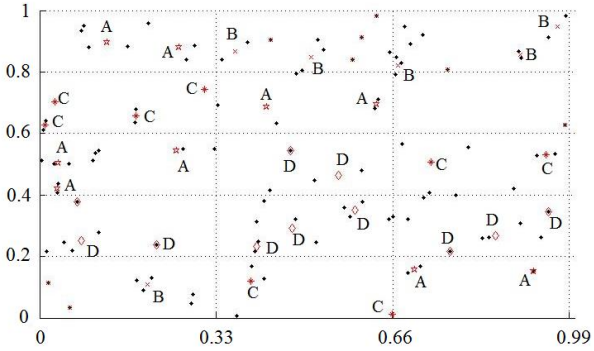
2.3.3 Evaluación de Resultados

Para evaluar la capacidad de estimación de clústers del clasificador propuesto y compararlo con el algoritmo K-means convencional, se requirieron ambos esquemas para clasificar 100 números. Estos números se distribuyeron aleatoriamente

entre 0 y 1 y pertenecían a 4 clases diferentes distribuidas como se muestra en la figura 9a. Después de entrenar tanto el clasificador propuesto como el algoritmo K-means[35-36], los esquemas se evaluaron usando un conjunto diferente de números distribuidos aleatoriamente entre 0 y 1. Las Figuras (3.15 B) y (3.15 C) muestran los datos de prueba junto con los centroides generados. A partir de estas cifras, es evidente que el esquema propuesto puede clasificar correctamente los datos de prueba incluso cuando las clases están desarticuladas, pero que un esquema convencional no puede.



(A)



(B)

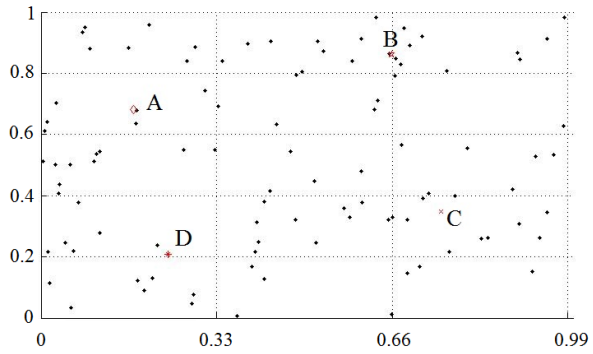


FIGURA 2.15 REPRESENTACION DEL CLASIFICADOR PROPUESTO CON 4 CLASES SEPARADAS ENTRE SI. (A) CLASES (B) CLUSTERSGENERADOS POR EL ESQUEMA PROPUESTO, (C) CLUSTERS GENERADOS POR KMEANS.

2.4 Conclusiones

Este capitulo se presentan tres sistemas que evaluamos por separado el primero de ellos es el reconocimiento del perfil del rostro, como podemos observar los resultados en la tabla 2.3 los resultados son superiores al 90% para cualquier perfil, por otra parte se esta realizando un recorte automático de las regiones de interés del rostro, para lograr esto, lo primero que se propone es un pre recorte del rostro de la imagen obtenida a través del algoritmo de Viola-Jones, para después en base a la simetría del rostro y con ayuda de las integrales proyectivas extraer de manera automática las 2 regiones de interés, la primera de ellas la región de Frente/Ojos y la segunda la de la Boca, en este punto es importante mencionar que se logra la extracción adecuada de las regiones aún con diferente luminiscencia como se puede apreciar en la tabla 2.4, por ultimo se propone un clasificador de bajo costo computacional que es capaz de relacionar clases que se encuentran muy disjuntas como se puede apreciar en la figura 2.15

Capítulo 3 Sistemas de reconocimiento de expresiones faciales

En este capitulo se presentaran 2 aplicaciones en las que se usaron las aportaciones descrita en el capitulo anterior, la primera de ellas se entrena y se prueba con las imágenes de la base de datos **KDEF** y la segunda se entrena con la base de datos **KDEF** y se prueba con los videos de la base datos de **HOHA**, descritas en la sección 2. Es muy importante señalar que los objetivos entre ambos sistemas son diferentes, para el sistema explicado en 4.1 es reconocer la expresión facial que realiza una persona a través de una imagen y para el sistema 4.2 el objetivo es a través del reconocimiento de expresiones faciales aplicado a video, crear una relación entre el estado de animo de la personas cuando realizan diferentes acciones usando el reconocimiento de expresiones faciales para crear este relación.

3.1 Sistema de reconocimiento de expresiones faciales en imágenes

Como ya se menciono anteriormente, este sistema se entrena y se prueba con la base datos **KDEF**, el diagrama de bloques del sistema propuesto es mostrado en la Fig. 4.1 la imagen original de la base de datos **KDEF** se le aplica el algoritmo de Viola-Jones explicado en la sección 2.1.1. para la extracción de un rostro de una escena, después en el bloque de segmentación automático se obtienen las regiones de interés de la boca y Frente/Ojos a través de los momentos de una imagen e integrales proyectivas como se explico en la sección 3.2, por otra parte el bloque del clasificador y decisión de salida se explicaron previamente en la sección 3.3. Con esto solo falta explicar el bloque de la extracción de características, a cada una de las regiones de interés, se le analizo por separado para encontrar el tamaño óptimo de la ventana de los filtros de Gabor, en total se le aplicó a 3 vectores distintos(boca, frente/ojos, boca+frente/ojos) el **PCA**, para después estimar el vector característico.

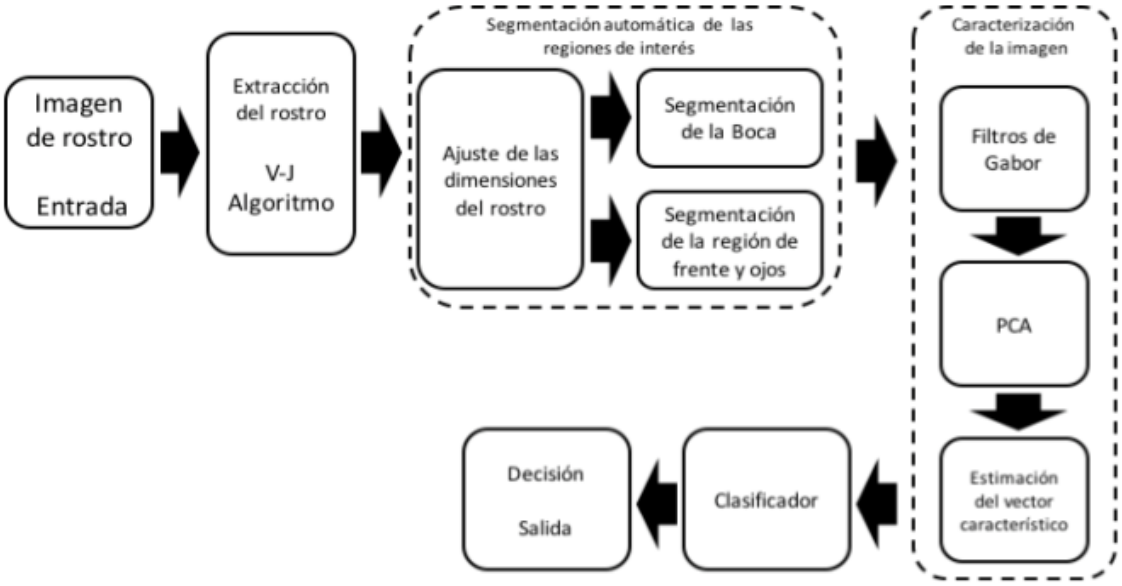


FIGURA 3.1 DIAGRAMA A BLOQUES DEL SISTEMA PROPUESTO.

3.1.1 CARACTERIZACIÓN DE LA IMAGEN

El bloque de caracterización de la imagen esta dividido en 3 subloques que serán explicados a continuación.

3.1.1.1 Filtros de Gabor

Los filtros de Gabor[56] son un ejemplo de filtros utilizados en muchas aplicaciones de procesamiento de imágenes, tales como el análisis de la textura [14]-[15] ya que son invariantes a los que cambios de luminiscencia. Estos son filtros pasa banda, que tienen tanto una orientación, propiedades selectivas de frecuencia y una resolución óptima conjunta en dominios tanto espacial como de frecuencia. Las funciones de Gabor 2D quedan determinadas por la ecuación

$$\boldsymbol{h(x,y) = g(x',y')e^{2\pi jFx'}}. \tag{3.1}$$

Donde los parámetros (x, y) expresan su localización en el dominio espacial, el parámetro F expresa la frecuencia espacial. Como se aprecia en la ecuación 4.1 la señal elemental de Gabor bidimensional espacial, está en función de la respuesta Gaussiana bidimensional , la frecuencia espacial (F) y la rotación aplicada (ϕ), para aplicar esta rotación se utilizará la ecuación 4.2. La respuesta Gaussiana bidimensional, puede expresarse mediante la ecuación

$$\boldsymbol{(x',y') = (xcos\phi + ysen\phi, -xsen\phi + ycos\phi)}. \tag{3.2}$$

$$g(x', y') = \frac{1}{2\pi\sigma^2} e^{-(\frac{x^2+y^2}{2\sigma^2})}. \quad (3.3)$$

La ecuación opera en el conjunto de números complejos, donde la parte real es la función de Gabor simétrica y la parte imaginaria es la función de Gabor asimétrica, retomando de la ecuación 4.1, $e^{2\pi jFx'}$ y usando la Fórmula de Euler, obtenemos lo siguiente $cos2\pi Fx' + jsen2\pi Fx'$ por consiguiente es posible obtener la señal elemental de Gabor con componentes reales e imaginarios a través de la ecuaciones 3.4 y 3.5.

$$h_c = g(x', y')cos(2\pi Fx'), \quad (3.4)$$
$$h_s = g(x', y')sen(2\pi Fx'). \quad (3.5)$$

Donde h_c es la señal elemental de Gabor con componentes reales (simetría par) y h_s es la señal elemental de Gabor con componentes imaginarios (simetría impar). Podemos concebir la información apartada por este par de funciones, como un vector bidimensional cuya magnitud forma el contraste de energía en un punto dado y cuya dirección es especificada a través de la fase de la energía. El contraste de energía es llamado también amplitud de la señal, el cual al representarlo en niveles de gris, muestra la respuesta de la imagen en función de la función espacial, que es independiente de la fase. La información aportada por este par de funciones corresponde al contraste de la energía en un punto dado. El contraste de energía $M(x, y)$ se obtiene mediante la ecuación:

$$M(x, y) = \sqrt[2]{{h_c}^2 + {h_s}^2}. \quad (3.6)$$

Al promediar cada una de estas amplitudes de la señal resultante, se obtiene los vectores característicos de la respuesta de la imagen:

$$M = \frac{\sum_{p=1}^{B} M_p(x, y)}{B}. \quad (3.7)$$

Donde B es el número de bancos de filtros de Gabor [16].

Se uso una $\sigma = \frac{x_0}{2}$, donde x_0 es el número de bloques en lo que se divide la imagen a lo ancho, las orientaciones que se usaron tienen una diferencia de 20 grados $\left(0, \frac{\pi}{9}, \frac{2*\pi}{9}, \frac{\pi}{3}, \frac{4*\pi}{9}, \frac{5*\pi}{9}, \frac{2*\pi}{3}, \frac{7*\pi}{9}, \frac{8*\pi}{9}\right)$, también se utilizaron 6 frecuencias espaciales $\left(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32}, \frac{\pi}{64}\right)$, la figura 9 muestra la disposición del banco de filtros.

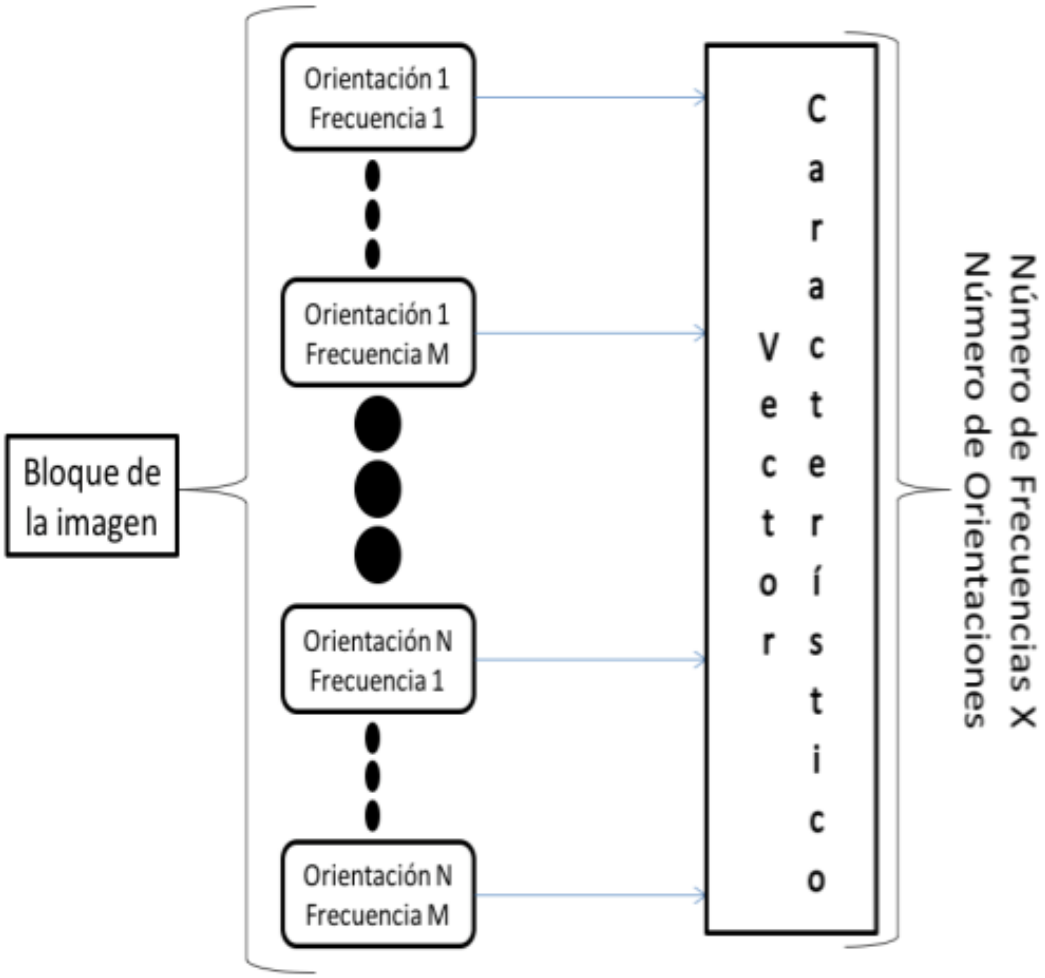


FIGURA 3.2.ESQUEMA DEL BANCO DE FILTROS DE LA FUNCIÓN BIDIMENSIONAL DE GABOR.

Por cada bloque de la imagen a extraer su vector característico, se obtuvieron 54 bancos de filtros de Gabor (6 frecuencias y 9 orientaciones) como lo muestra la figura 10, por último se obtiene un vector característico del número de bloques en que se dividió la imagen, a través de un promedio de los 54 filtros de Gabor para cada bloque de la imagen.

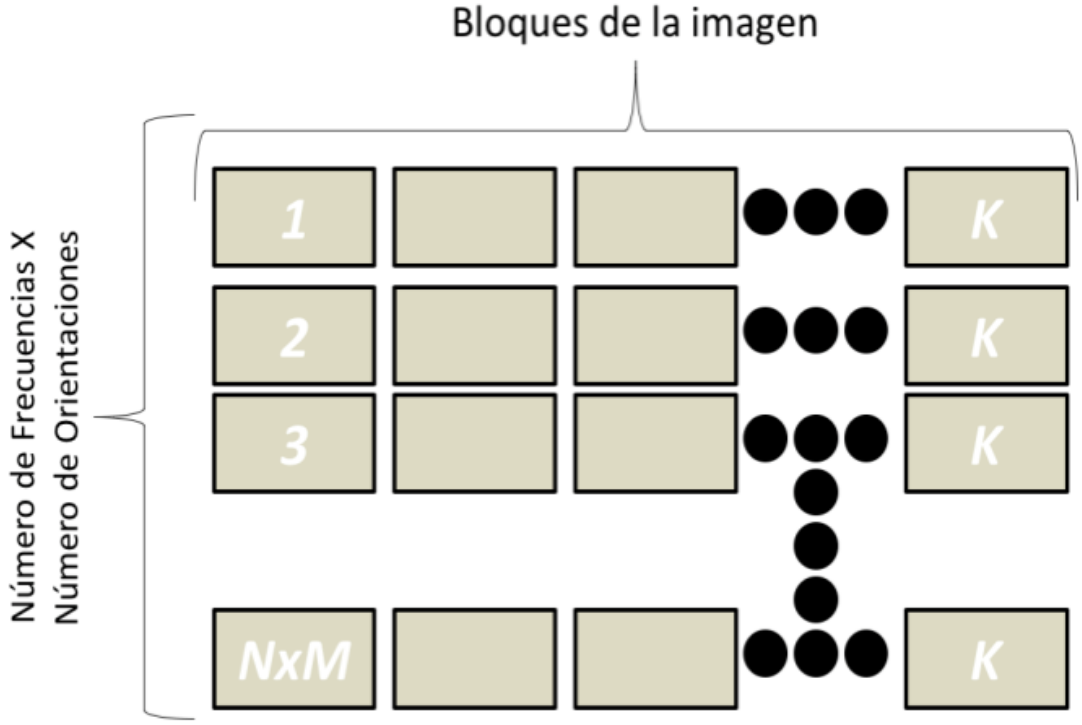


FIGURA 3.3. BANCOS DE FILTRO DE GABOR Y BLOQUES DE LA IMAGEN.

Por lo tanto para la región de la boca el vector característico tendrá de elementos la multiplicación del número de bloques a lo ancho por el número de bloques a lo largo y para la región de la Frente/ojos es similar.

3.1.1.2 PCA

El análisis de componente principales (PCA) [50] permite reducir la dimensión de los datos, transformando el conjunto de p variables originales en otro conjunto de q variables correlacionadas, llamadas componentes principales. Las p variables son medidas sobre cada uno de los **n** individuos, obteniéndose una matriz de datos de orden $n_x p$. Con la aplicación del algoritmo del PCA se obtendrá la matriz PM que contiene las q nuevas variables ó componentes principales a través de combinaciones lineales de las variables originales. Es importante mencionar que obtuvimos tres matrices PM para los resultados mostrados en este artículo; la primera de esta es para los vectores extraídos de la región de la boca (PM_B), después se obtuvo para los vectores extraídos de la región de la Frente/ojos($PM_{F/O}$) y por último la concatenación de ambos vectores extraídos de cada una de las regiones de interés ($PM_{BF/O}$).

3.1.1.3 Estimación del vector característico

Cada vector característico es el producto de las diferentes matrices PM, obtenidas a través del PCA por cada vector característico de las imágenes, como muestra la figura 11, con esto vectores característicos obtenidos ya es posible formar una matriz de entrenamiento y una de prueba, siendo el siguiente paso clasificar dichos datos.

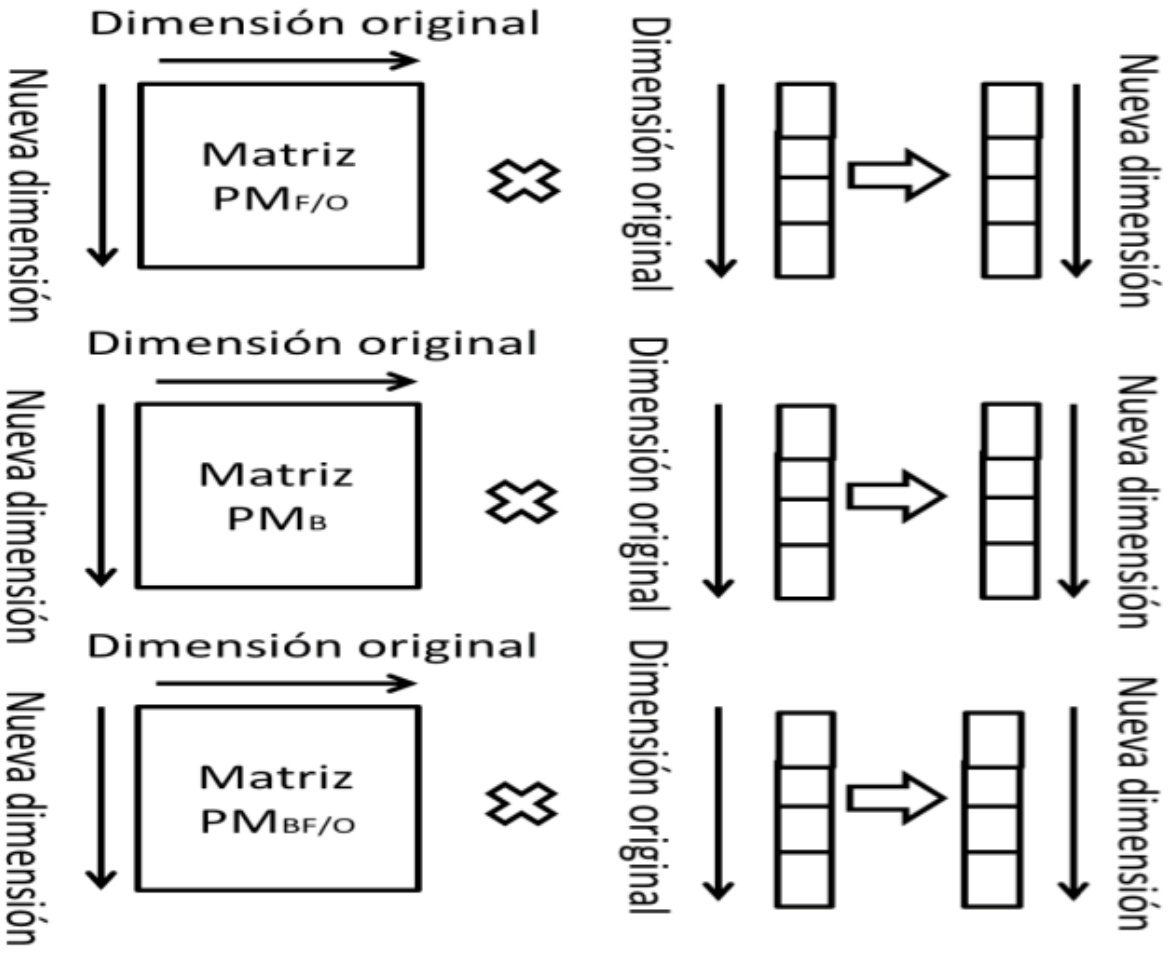


FIGURA 3.4. EJEMPLO DE LA ESTIMACIÓN DEL VECTOR CARACTERÍSTICO.

3.1.2 Resultados

En esta tesis, las imágenes de la región de la boca se redimensionaron a 180 pixeles de ancho y 90 de alto, para la región de la Frente/ojos se redimensionaron a 300 pixeles de ancho y 150 de alto. Como inicio se usaron diferentes ventanas cuadradas para los filtros de Gabor, en ambas regiones del rostro, tomando 50 imágenes de entrenamiento y 20 de prueba.

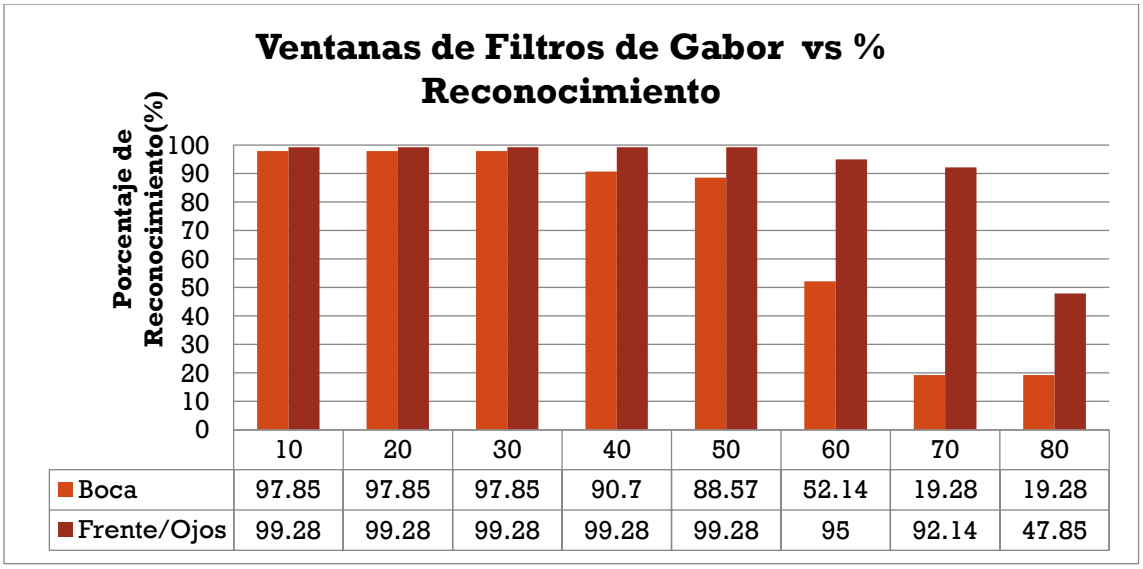


FIGURA 3.5 PORCENTAJE DE RECONOCIMIENTO USANDO DIFERENTES MEDIDAS DE PARA LAS VENTANAS DE LOS FILTROS DE GABOR.

Como muestra la figura 4.5, para la región de la boca el porcentaje de reconocimiento del %97.85 se mantiene si se usan ventanas de Gabor de 10-30, por lo tanto para futuros experimentos usaremos la ventana de 30X30, para la región Frente/Ojos el porcentaje de reconocimiento del %99.28 se mantiene si se usan ventanas de Gabor de 10-50, por lo tanto para futuros experimentos usaremos la ventana de 50X50, el uso de estas ventanas en ambos casos se debe a que el costo computacional es menor. Se realizó el mismo experimento con una red neuronal artificial como lo muestra la tabla 4.1 teniendo como resultados para la boca un %97.14 de reconocimiento entrenando con 50 imágenes y usando ventanas para los filtros de Gabor de 30x30, para la región de la Frente/Ojos se obtuvo un %91.42 de reconocimiento entrenando con 50 imágenes y usando ventanas para los filtros de Gabor de 50x50, en ambos casos el porcentaje de reconocimiento es menor al obtenido utilizando el clasificador propuesto.

TABLA 3.1 . COMPARACIÓN DE LOS PORCENTAJES DE ENTRENAMIENTO Y TIEMPO DEL MISMO ENTRE LOS CLASIFICADORES

Región	Reconocimiento		Tiempo de entrenamiento	
	Propuesto	ANN	Propuesto	ANN
Frente/Ojos	99.28%	91.42%	1.51s	2.4s
Boca	97.85%	97.14%	3.4s	4.4s

La tabla 3.2 y 3.3 muestran las matrices de confusión para la región de Boca y Frente/Ojos respectivamente, podemos ver que el método propuesto para la región de la boca reconoce de manera adecuada Miedo, Enojo, Molestia, Feliz, Triste y Sorpresa con un porcentaje del 100%, solo en neutral tiene un porcentaje del 85% confundiéndose con Triste, para la región de la Frente/Ojos reconoce de manera adecuada Neutral, Enojo, Molestia, Feliz, Triste y Sorpresa con un porcentaje del 100%, en Miedo reconoce el 95% confundiéndose con Triste.

Matriz de confusión, región Boca

	Miedo	Molestia	Triste	Enojo	Feliz	Neutral	Sorpresa
Miedo	**1.00**	-	-	-	-	-	-
Molestia	-	**1.00**	-	-	-	-	-
Triste	-	-	**1.00**	-	-	-	-
Enojo	-	-	-	**1.00**	-	-	-
Feliz	-	-	-	-	**1.00**	-	-
Neutral	-	-	0.15	-	-	**0.85**	-
Sorpresa	-	-	-	-	-	-	**1.00**

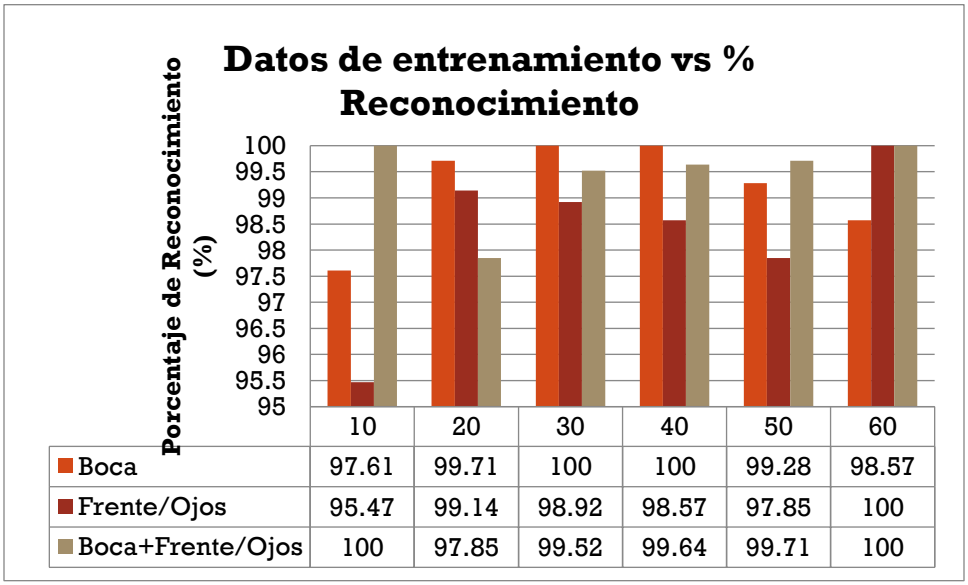
TABLA 3.2. MATRIZ DE CONFUSIÓN, USANDO VENTANAS DE 30X30.

Matriz de Confusión, Región Frente/Ojos

	Miedo	Molestia	Triste	Enojo	Feliz	Neutral	Sorpresa
Miedo	**0.95**	-	0.05	-	-	-	-
Molestia	-	**1.00**	-	-	-	-	-
Triste	-	-	**1.00**	-	-	-	-
Enojo	-	-	-	**1.00**	-	-	-
Feliz	-	-	-	-	**1.00**	-	-
Neutral	-	-	-	-	-	**1.00**	-
Sorpresa	-	-	-	-	-	-	**1.00**

TABLA 3.3. MATRIZ DE CONFUSIÓN, USANDO VENTANAS DE 50X50

En la figura 3.5 tenemos los resultados de haber entrenado al sistema propuesto con 50 imágenes, la figura 4.6 muestra el porcentaje de reconocimiento obtenido al variar el número de patrones con los que entrenamos al sistema, es importante mencionar que en cualquiera de los 3 casos (Región de la boca usando ventanas de 30x30, Región de la Frente/Ojos usando ventanas de 50x50, Región de la boca usando ventanas de 30x30+Región de la Frente/Ojos usando ventanas de 50x50) los porcentajes de reconocimiento son mayores al 95%, cuando se concatenan las regiones de Boca y Frente/Ojos, el mayor porcentaje de reconocimiento es de 100 % este se obtiene cuando entramos al sistema con 10 y 60 imágenes, por lo tanto el mejor resultado, si contamos con el rostro completo para reconocer una expresión facial, es cuando se entrena al sistema con 10 patrones. Si solo se contara con la región de la boca para definir la expresión facial lo mejor es entrenar al sistema con 30 imágenes como lo muestra la figura 4.6, por último si tuviéramos la región de la Frente/Ojos lo óptimo seria entrenar al sistema con 20 imágenes, aunque existe un porcentaje mayor de reconocimiento (100%), pero este está muy cercano a usar todas los patrones de la base de datos para entrenar este sistema, por esta razón es preferible entrenar al sistema con 20 imágenes teniendo un porcentaje de reconocimiento del 99.14%



	10	20	30	40	50	60
Boca	97.61	99.71	100	100	99.28	98.57
Frente/Ojos	95.47	99.14	98.92	98.57	97.85	100
Boca+Frente/Ojos	100	97.85	99.52	99.64	99.71	100

FIGURA 3.6. PORCENTAJE DE RECONOCIMIENTO VARIANDO EL NÚMERO DE DATOS DE ENTRENAMIENTO.

Realizando una comparación con la literatura como lo muestra la tabla 3.4, notamos que el sistema propuesto es superior a otros sistemas donde obtienen el reconocimiento de las expresiones faciales (en ambos sistemas solo utilizan 6 expresiones) a través de las distintas regiones del rostro.

Referencia	Caracterización	Clasificador	Frente/Ojos	Boca	Boca+Frente/Ojos	Base de Datos
Propuesta	Filtros de Gabor	Propuesto	99.14	100	100	KDEF
[57](2014)(1)	Eigenphases+MVA	SVM	60	79.3	82	CK
[58](2015)(1)	Gabor Template	SVM	95.1(2)	90.8	N/A	CK

(1) Solo reconocen 6 expresiones faciales (Miedo, Enojo, Molestia, Feliz, Triste y Sorpresa).
(2) Solo utilizan la regiones de los ojos para el reconocimiento.

TABLA 3.4. COMPARACIÓN DE PORCENTAJES DE CLASIFICACIÓN QUE UTILIZAN REGIONES ESPECIFICAS DEL ROSTRO.

Por otra parte en la tabla 3.5 se realiza una comparación con otros sistemas donde utilizan todo el rostro para el reconocimiento de expresiones faciales.

Referencia	Caracterización	Clasificador	Clases	% Reconocimiento	Base de datos
Propuesta	Filtros de Gabor	Propuesto	7	100	KDEF
[18](2015)	EMD + PCA + LDA,EMD + PCA + LFDA,EMD + KLFDA	KNN, SVM, ELM-RBF	7	100	JAFFE
[18](2015)	EMD + PCA + LDA,EMD + PCA + LFDA,EMD + KLFDA	KNN, SVM, ELM-RBF	7)	99.75	CK

TABLA 3.5. COMPARACIÓN DE PORCENTAJES DE CLASIFICACIÓN QUE UTILIZAN TODO EL ROSTRO.

De las comparaciones realizadas en la tabla 4.4 observamos que el sistema propuesto es mejor en su porcentaje de reconocimiento, además en [57] realizan la extracción de características de manera manual, algo que este sistema propone de manera automática. En la tabla 4.5 observamos porcentajes de reconocimiento similares a los obtenidos en este trabajo, pero es importante mencionar que en [58] utilizan todo el rostro para el reconocimiento de la expresión facial a diferencia de esta propuesta que solo utiliza las regiones de interés propuestas.

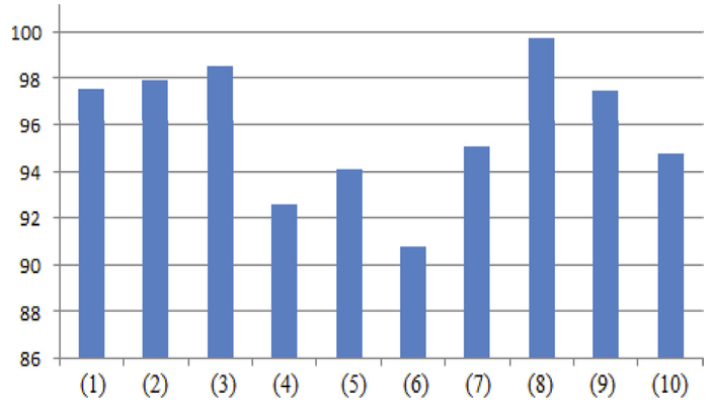


FIGURA 4.1 1 PORCENTAJE DERECONOCIMEINTO (1) PROPUESTA USANDO LA REGION DE LA BOCA, (2)PROPUESTA USANDO LA REGION DE LA FRENTE-OJOS, (3) PROPUESTA USANDO AMBAS REGIONES. PORCENTAJE DE ALGORITMOS PROPUESTOS (4) BENITEZ ET AL. [57], (5) ZHANG ET AL. USA LA REGION DE LA BOCA [58], (6) ZHANG ET AL. USA FRENTE-OJOS [58], (7) ZHANG ET AL. USA AMBAS REGIONES [58], (8) ALI ET AL. [59], (9) WANG AND ZHANG [60] AND (10) BUCIU AND PITAS [61] TAMBIEN SE MUESTRAN PARA SU COMPARACION.

3.2 Sistema de reconocimiento de expresiones faciales en video

3.2.1 Sistema Propuesto

Como ya se menciono anteriormente, este sistema se entrena con la base datos KDEF y se prueba con la base datos HOHA.Este sistema usa video los bloques son similares la mayoría ya han sido previamente explicados podemos observarlos en la figura 4.7, el bloque de detección de rostro usa el algoritmo de viola jones explicado en la sección 2.1.1, el bloque de detector de perfil se explica a detalle en la sección 3.1, la detección automática de las regiones de interés se explica en la sección 3.2 y el bloque decisión se explica en la sección 3.3, solo falta explicar la extracción de características que se realizo en este sistema que se explicara en la siguiente subsección.

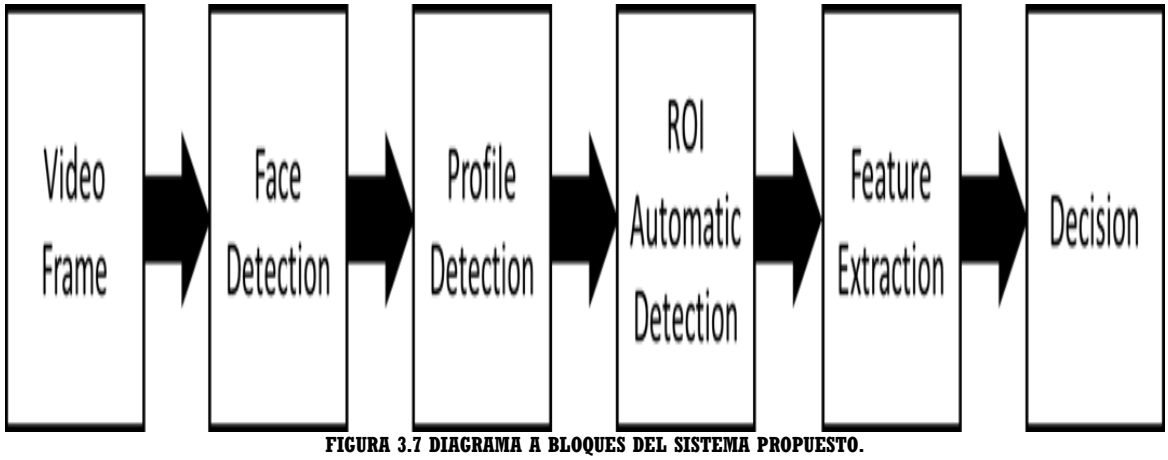


FIGURA 3.7 DIAGRAMA A BLOQUES DEL SISTEMA PROPUESTO.

3.2.1.1 Extracción de Características

En este caso para la realizar la extracción de características de cada una de las regiones de interés, las dividiremos en bloques de 35x 40 pixeles, los cuales serán caracterizados a través del valor modal de cada bloque, con esto nosotros creamos un vector por cada ROI, al final las concatenaremos para aplicar el PCA y la estimación del vector característico descritas en las secciones 3.1.1.2 y 3.1.1.3 respectivamente.

3.2.2 RESULTADOS

El esquema propuesto se probo con la base de datos HOHA que consta de 8 acciones de diferentes actores, con diferentes fondos, es decir con condiciones no controladas, lo cual dificulta el reconocimiento. Las figuras de (3.8-3.11) muestran 4 frames del video 1, donde la acción es besar. Los resultados de dichas figuras son similares, esto se debe a que la expresión es similar en estos frames, pero la figura 3.11 el resultando es diferente, esto no nos debe importar por que el objetivo es reconocer la expresión facial de todo el video completo. El resultado final del video es mostrado en la figura 3.12.

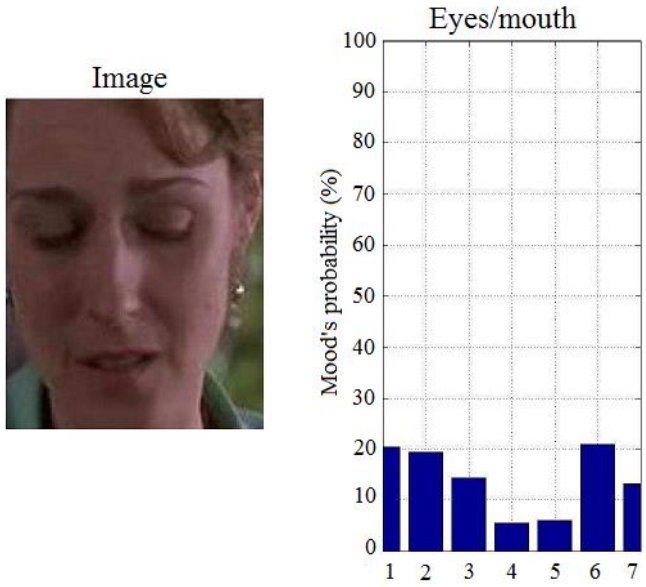


FIGURA 3.8 FRAME 116 DE "AS GOOD AS IT GETS -01766.AVI"1: MIEDO, 2: ENOJO, 3: MOLESTIA, 4: FELIZ, 5: NEUTRAL, 6: TRISTE, 7: SORPRESA.

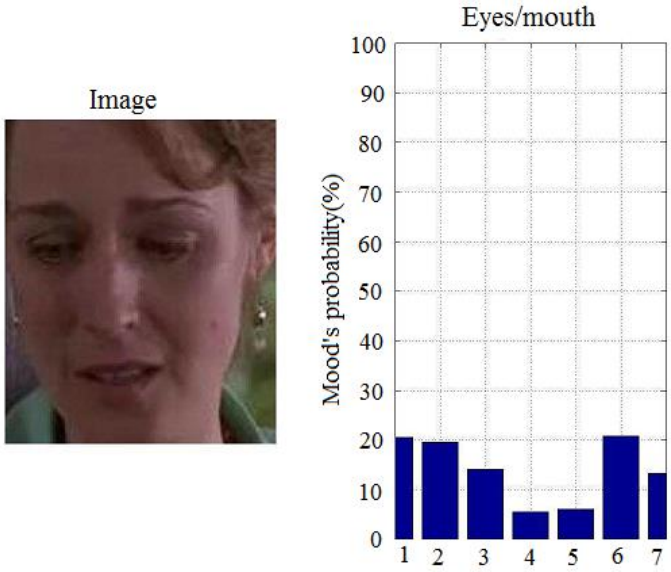


FIGURA 3.9 FRAME 114 DE "AS GOOD AS IT GETS -01766.AVI", 1: MIEDO, 2: ENOJO, 3: MOLESTIA, 4: FELIZ, 5: NEUTRAL, 6: TRISTE, 7: SORPRESA.

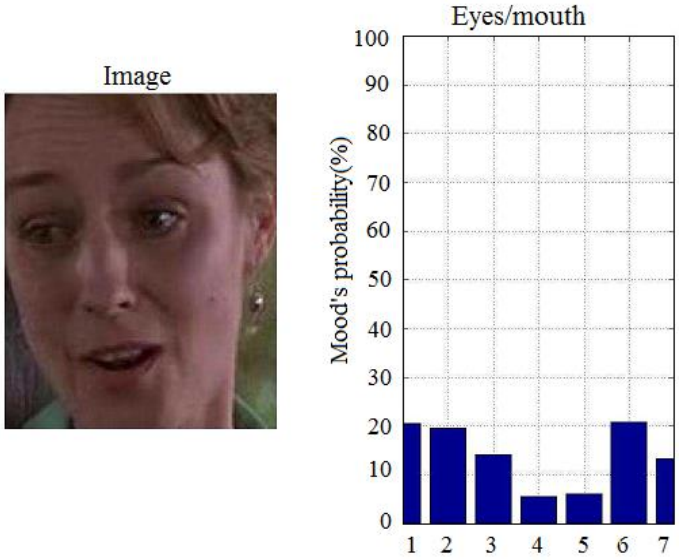


FIGURA 3.10 FRAME 165 DE "AS GOOD AS IT GETS -01766.AVI 1: MIEDO, 2: ENOJO, 3: MOLESTIA, 4: FELIZ, 5: NEUTRAL, 6: TRISTE, 7: SORPRESA.

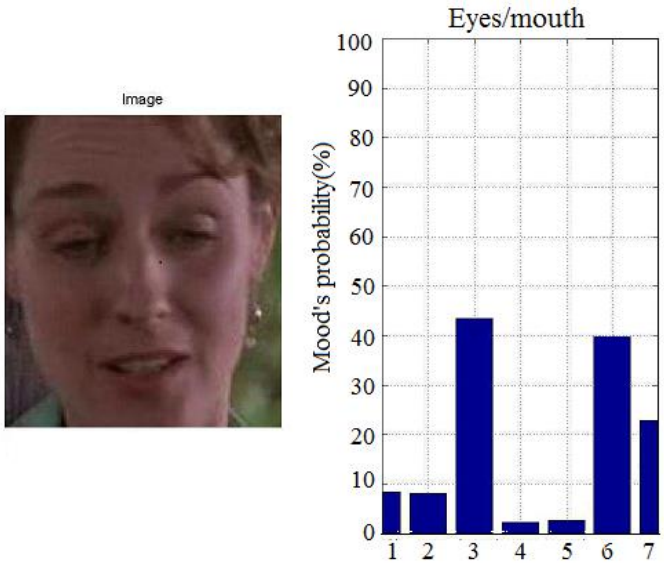


FIGURA 3.11 2 FRAME 174 DE "AS GOOD AS IT GETS -01766.AVI" HERE 1: MIEDO, 2: ENOJO, 3: MOLESTIA, 4: FELIZ, 5: NEUTRAL, 6: TRISTE, 7: SORPRESA

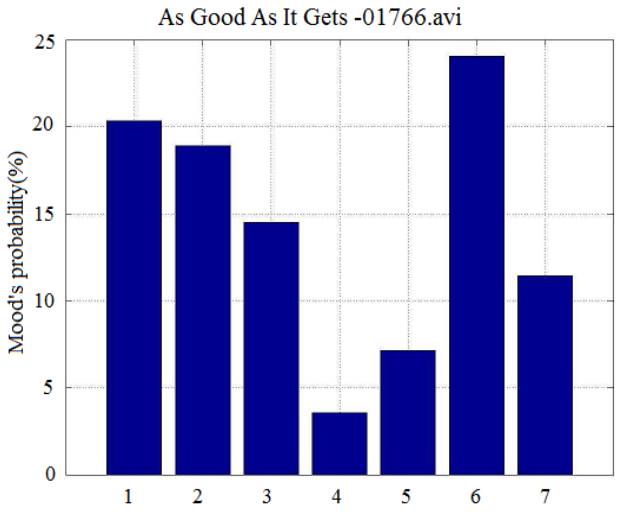


FIGURA 3.12 PROMEDIO DE RECONOCIMIENTO DE EXPRESIONES FACIALES EN LOS FRAMES 60-177 Y 191-194 OF "AS GOOD AS IT GETS -01766.AVI", ACTION: TALKING. HERE 1: MIEDO, 2: ENOJO, 3: MOLESTIA, 4: FELIZ, 5: NEUTRAL, 6: TRISTE, 7: SORPRESA.

Para constatar el resultado obtenido se invita al lector a buscar el video en la base de datos, con el fin de que observe que el resultado final que da el sistema es correcto, ya que en toda la escena la persona se encuentra triste. Otro video que se analizo fue el siguiente "Butterfly Effect, The - 02093.avi ", en este caso el video muestra a una mujer molesta, el resultado final obtenido lo podemos ver en la siguiente figura.

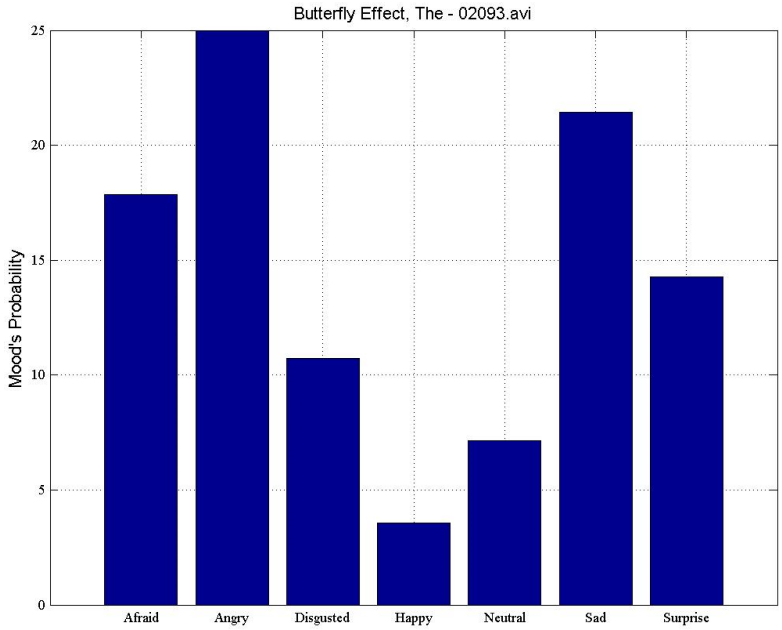


FIGURA 3.13 PROMEDIO DE RECONOCIMIENTO DE EXPRESIONES FACIALES EN LOS FRAMES 60-177 AND 191-194 OF "BUTTERFLY EFFECT, THE – 02093. AVI.".

En este punto el sistema es capaz de reconocer para cada video la expresión facial, el siguiente paso aprovechando que los videos de la base da datos que se uso están etiquetados, es obtener para todos los videos que comparten la misma etiqueta los resultados, estos resultados se muestran en las siguientes figuras, cada figura corresponde a cada una de las acciones analizadas.

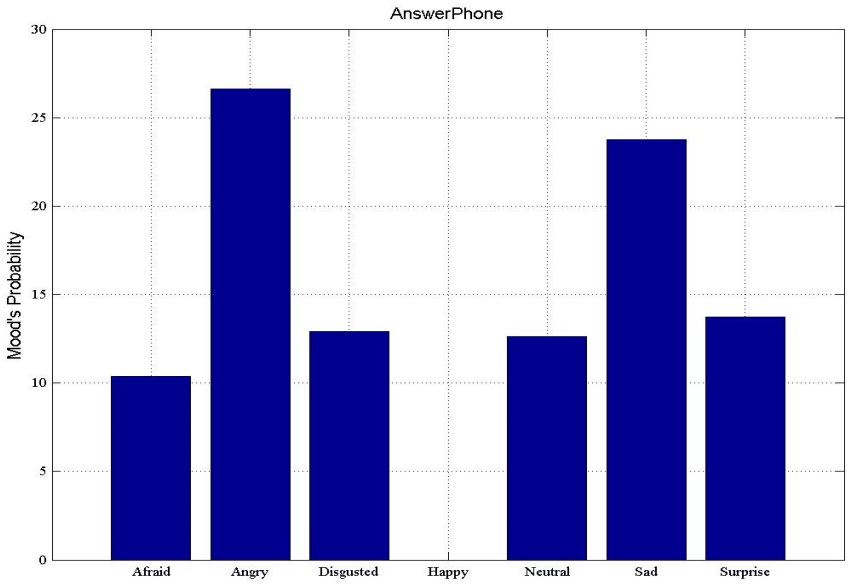


FIGURA 3.14 PROBABILIDAD DEL ESTADO DE ANIMO DE ANSWER PHONE.

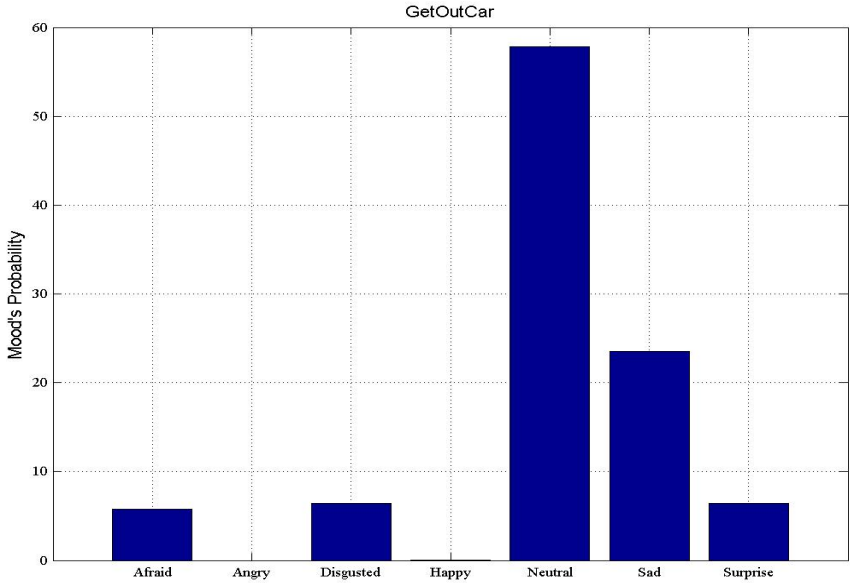


FIGURA 3.15 PROBABILIDAD DEL ESTADO DE ANIMO DE GET OUT CAR.

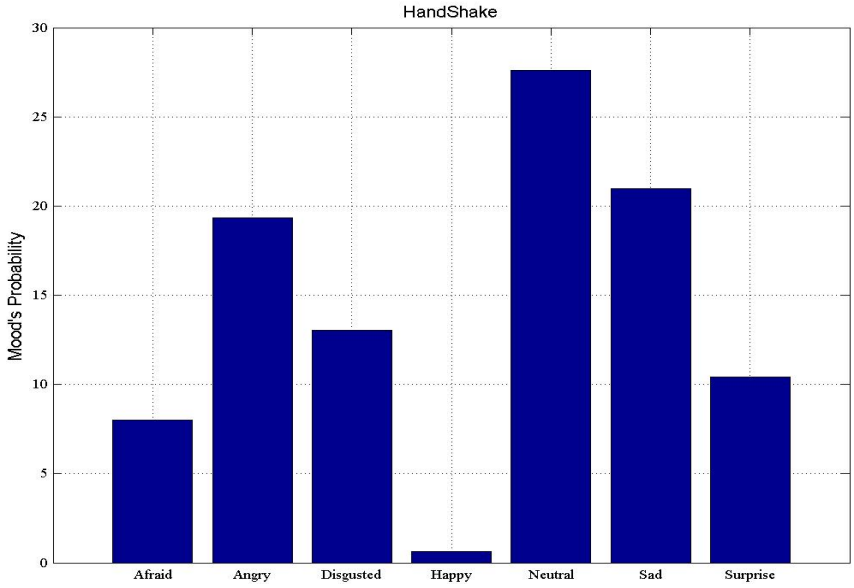


FIGURA 3.16 PROBABILIDAD DEL ESTADO DE ANIMO DE HAND SHAKE.

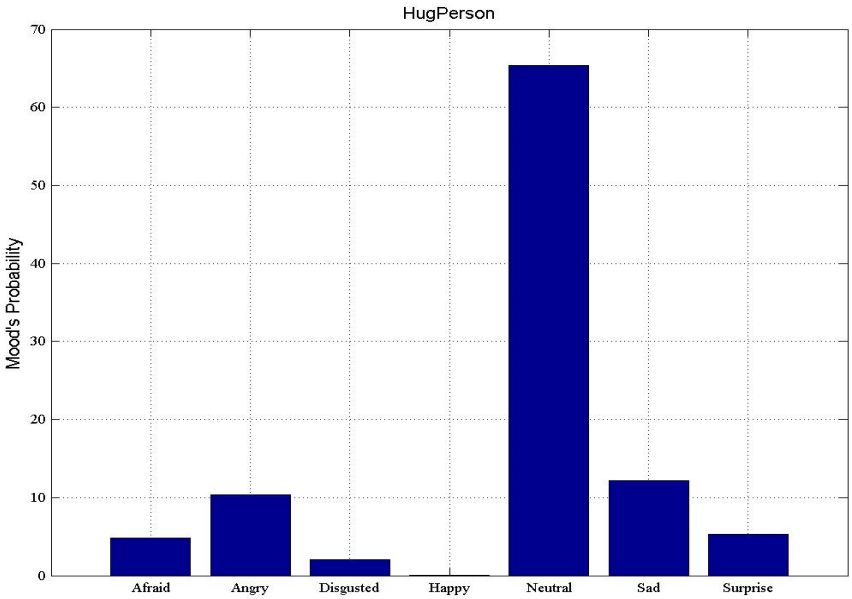


FIGURA 3.17 PROBABILIDAD DEL ESTADO DE ANIMO DE HUGH PERSON.

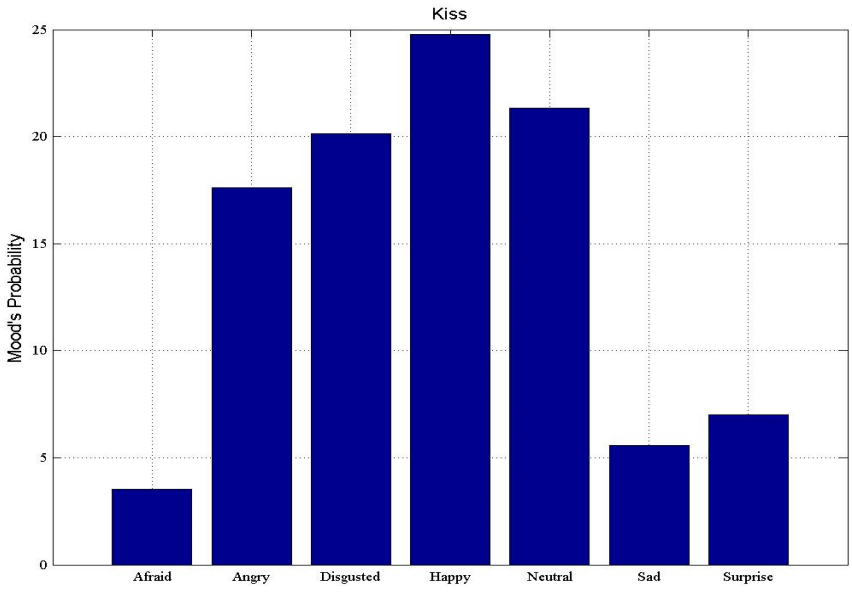


FIGURA 3.18 PROBABILIDAD DEL ESTADO DE ANIMO DE KISS.

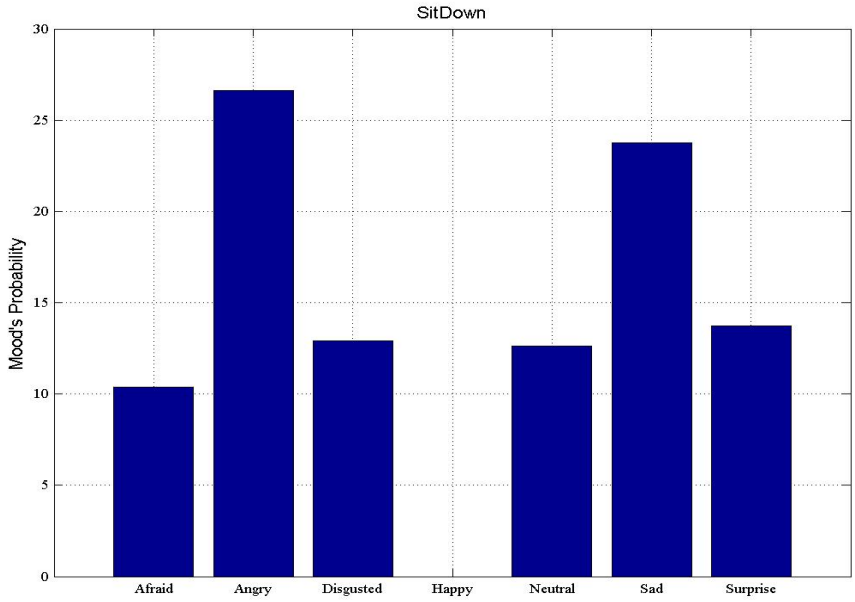


FIGURA 3.19 PROBABILIDAD DEL ESTADO DE ANIMO DE SIT DOWN.

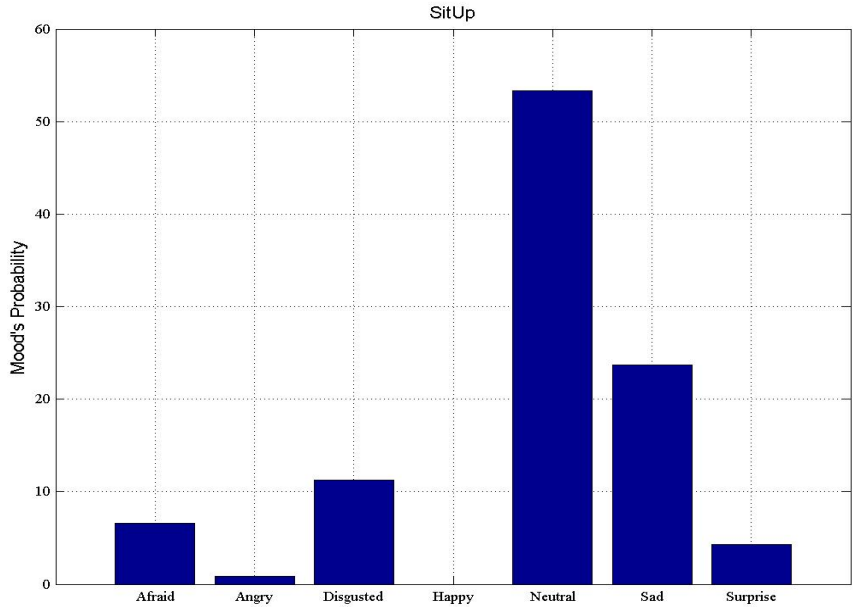


FIGURA 3.20 PROBABILIDAD DEL ESTADO DE ANIMO DE SIT UP

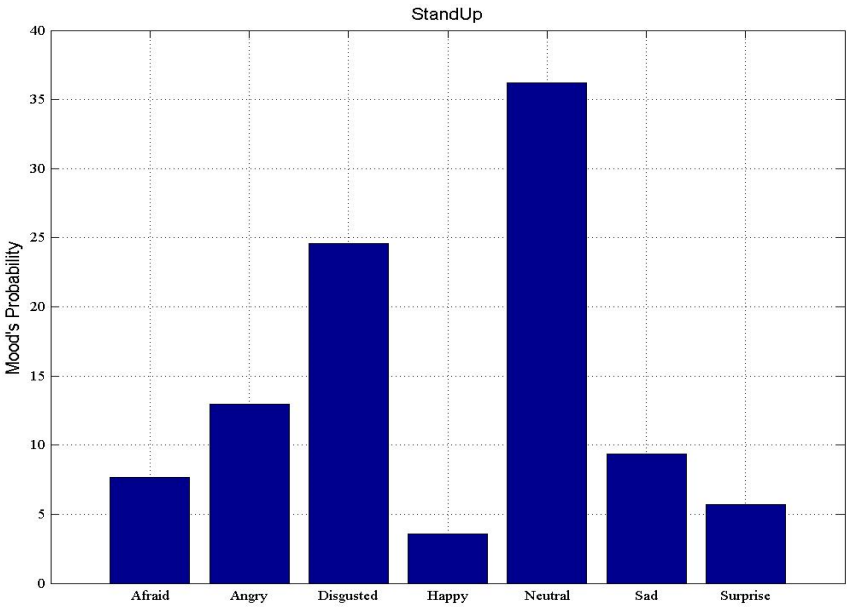


FIGURA 3.21 PROBABILIDAD DEL ESTADO DE ANIMO DE STAND UP.

Los figuras anteriores nos muestran que es posible encontrar alguna relación entre el estado de animo cuando se están realizando cierta acciones especificas, es muy importante señalar que los resultados obtenidos son para la base de datos que se uso.

3.3 CONCLUSIONES

Podemos concluir que el clasificador de bajo costo computacional tiene mejores resultados que una Red Neuronal Artificial,. Como nos muestra la tabla 3.1 donde se supera a el porcentaje de reconocimiento, como en los tiempos de entrenamiento. Al realizar una comparación con la literatura de la tabla 3.5 podemos concluir que el sistema propuesto es superior a [57] y [58], porque se obtuvo un porcentaje mayor de reconocimiento en todos los posibles casos y además se reconoce una expresión facial mas (neutral), en la tabla 3.6 realizamos una comparación con otros sistemas que utilizan todo el rostro para el reconocimiento de expresiones faciales, presentamos resultados similares a los de la literatura de dicha tabla, en esta ocasión ambos reconocemos 7 expresiones faciales. Por lo cual es posible concluir que, nuestro sistema es capaz de reconocer de manera adecuada las expresiones faciales con un porcentaje mayor al 97%, ya sea tomando todo el rostro que en nuestro caso son las 2 regiones de interés concatenadas, o responder de manera adecuada a una oclusión parcial del rostro es decir solo tomando en cuenta una de las regiones de interés propuestas.

Capítulo 4 Conclusiones

Este tesis presenta 3 aportaciones relacionadas con el reconocimiento de expresiones faciales, la primera de ellas: el sistema capaz de detectar el perfil del rostro, esto ayuda a mejorar los porcentajes de reconocimiento ya que al tener rostro viendo de frente o casi de frente a la cámara la información con la que se cuenta para detectar la expresión facial es mayor, dando además la posibilidad de crear sistemas de reconocimiento de expresiones faciales para cuando las personas se encuentran con el rostro girado hacia algún lado. Por otra parte se propone un sistema que detecta y extrae de manera automática las regiones de interés del rostro, sin importar las diferentes condiciones de luminosidad en las imágenes, el genero de la persona o la expresión facial que estén haciendo, siendo estos problemas que se mencionaron en el capitulo 1 a los cuales siempre se les había afectado en este tema, otro problema que se logro resolver es la oclusión de alguna parte del rostro ya que con solo tener una región de interés del rostro completa el sistema propuesto es capaz de decir la expresión facial de la persona. Por ultimo se propone un clasificador basado en clustering y lógica difusa, que al usar algoritmos de clustering su costo computacional es muy bajo comparado con los clasificadores clásicos de la literatura, dando a demás la posibilidad gracias a la lógica difusa de dar porcentajes de pertenencia con las clases, algo que por ejemplo no sucede con las redes neuronales clásicas.

Capítulo 5 Trabajos futuros

En trabajo futuro se pueden considerar tres líneas a desarrollar , la primera de ellas es realizar un pre procesamiento a las imágenes antes de realizar los filtros de Gabor y utilizar algún otro extractor de características de los que se menciona en la literatura, por otra parte se debe resolver alguna oclusión parcial en alguna de las regiones de interés, por si las dos regiones propuestas están ocluidas, ya que esto no permitiría el reconocimiento de la expresión facial, finalmente resolver el problema cuando un persona no está viendo de frente a la cámara, es decir cuando el rostro sufre algún grado de rotación en el sentido vertical, que es cuando se tiene mayor perdida de información, así como realizar los sistemas de reconocimientos de expresiones faciales cuando el rostros se encuentra rotado de manera horizontal..

Capítulo 6 Publicaciones

Publicación en revista internacional indexada

Andres Hernandez-Matamoros, Andrea Bonarini, Enrique Escamilla-Hernandez, Mariko Nakano-Miyatake, Hector Perez-Meana, Facial expression recognition with automatic segmentation of face regions using a fuzzy based classification approach, Knowledge-Based Systems, Volume 110, 2016, Pages 1-14, ISSN 0950-7051, http://dx.doi.org/10.1016/j.knosys.2016.07.011.

A supervised classifier scheme based on clustering algorithms and fuzzy logic, Andres HERNANDEZ-MATAMOROS[a], Mariko NAKANO-MIYATAKE[a], Hector PEREZ-MEANA (En revisión en Pattern Analysis and Applications, Springer)

Knowledge-Based Systems 110 (2016) 1–14

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Facial expression recognition with automatic segmentation of face regions using a fuzzy based classification approach

Andres Hernandez-Matamoros[a], Andrea Bonarini[b], Enrique Escamilla-Hernandez[a], Mariko Nakano-Miyatake[a], Hector Perez-Meana[a,1,*]

[a] Instituto Politecnico Nacional, Av. Santa Ana 1000 Mexico D. F., 04430, Mexico
[b] Politecnico di Milano, Via Ponzio 34/5, 20133 Milano Italy

ARTICLE INFO

Article history:
Received 28 February 2016
Revised 27 May 2016
Accepted 6 July 2016
Available online 7 July 2016

Keywords:
Robust facial expression recognition
2D Gabor functions
Automatic ROI segmentation
PCA
Low complexity classifier
Horizontal projective integral

ABSTRACT

This paper proposes a facial expression recognition algorithm that automatically detects the facial image contained in a color picture and segments it in two regions of interest (ROI)—the forehead/eyes and the mouth—which are then divided into non-overlapping $N \times M$ blocks. Next, the average of the first element of the cross correlation between 54 Gabor functions and each one of the $N \times M$ blocks is estimated to generate a matrix of dimension $L \times NM$, where L is the number of training images. This matrix is then inserted into a principal component analysis (PCA) module for dimensionality reduction. Finally, the resulting matrix is used to generate the feature vectors, which are inserted into the proposed low complexity classifier based on clustering and fuzzy logic techniques. This classifier provides recognition rates close to those provided by other high performance classifiers, but with far less computational complexity. The experimental results show that proposed system achieves a recognition rate of about 97% when the feature vector from only one ROI is used, and that the recognition rate increases to approximately 99% when the feature vectors of both ROIs are used. This result means that the proposed method can achieve an overall recognition rate of approximately 97% even when one of the two ROIs is totally occluded.

1. Introduction

Facial expressions are one of the most powerful ways that human beings notify others of their emotional states. Approximately 55% of the messages related to an individual's feelings are delivered via facial expressions [1]. Facial expressions are generated by contracting facial muscles that temporally deform facial components such as eyebrows, lips, the nose or mouth, etc. These expressions have been grouped into seven basic classes that are universal across human ethnicities: anger, disgust, fear, happiness, sadness, surprise and neutrality [1]. Facial expression recognition has received a great deal of attention during the last decade because it is an important tool when automatic interactions between humans and machines are critical, such as in developing hospital nurse robot assistants [2], automatic animation, and intelligent tutoring systems, among others [1,2].

As a result, several efficient facial emotion recognition (FER) algorithms have been developed. In many cases these use approaches already widely adopted for facial recognition applications such as Gabor functions, discrete wavelet transforms, local binary patterns (LBP), Weber Local Descriptors (WLD), discriminant sparse local spline approaches, and key points based feature extraction methods [2–10]. Ali et al. [2] proposed using empirical mode decomposition, which is a form of nonlinear and non-stationary data analysis for facial emotion classification. The major advantage of this technique, which has previously been used to detect epileptic seizures using EEG signals to diagnose Alzheimer's disease, is that the basis function can be directly derived from the signal itself using local data characteristics, providing a fully data-driven approach. In this study, the face image was first preprocessed to segment it from the background using a rectangle based on a face model [2] scaled to a fixed size, with its intensity normalized. Next, a Radon transform of the resulting image was estimated and its successive projections decomposed using the empirical mode decomposition (EMD) approach. Then, the resulting data were fed into three different dimensionality reduction schemes: Principal Component Analysis (PCA) [11] combined with linear discriminant analysis (LDA), PCA with local Fisher discriminant analysis (LFDA) and kernel LFDA (KLFDA) [2]. Finally, the feature vectors obtained from these schemes were fed into a classifier stage to make the final decision about which of the seven types of facial expression was present in the image under analysis. This system achieved

* Corresponding author.
E-mail address: hmpm@prodigy.net.mx (H. Perez-Meana).
[1] http://www.posgrados.esimecu.ipn.mx

http://dx.doi.org/10.1016/j.knosys.2016.07.011
0950-7051/

a recognition rate of approximately 99% when operating on non-occluded images.

Luo et al. [3] proposed an FER system that first detects the face in an image using the Viola-Jones algorithm [4] and whose pixel values are normalized to reduce the effects of varying illumination. The resulting image is then characterized using the local binary patterns (LBP) algorithm with a sliding window of 3×3 pixels. After scanning the face image using the LBP approach, the resulting matrix is characterized using the histogram of the resulting LBP values [3]. Because each LBP value has eight bits, the resulting histogram becomes a vector of size 256, which is fed into the PCA module for dimensionality reduction. Finally, the resulting vector is fed into a support vector machine (SVM) to carry out the recognition task. This scheme provides a recognition rate of approximately 90%.

Several of the FER systems described above provide fairly good performance when operated with non-occluded facial images taken under controlled conditions. However, building robust FER systems that can perform well on partially occluded images is still a challenging task. Partial face occlusions can be classified into temporary and systematic occlusions. In temporary occlusions, the partial occlusions are produced by head, hair, hand and other movements that partially occlude the face, while systematic occlusions result from people wearing sunglasses, scarves, and so on. [5,6]. In both cases it is important to train the system using non-occluded face images but to test with both occluded and non-occluded images. In many situations, FER systems must be able to operate with occluded face images [5,6]. To this end, Zhang et al. [7] proposed an FER system in which the face image is first extracted using the well-known Viola–Jones face detection algorithm [4] and, then, scaled it to a fixed size of 48×48 pixels. Next, a set of face images with occluded regions were simulated by adding masks to the extracted facial regions. These images were then convolved using a bank of Gabor filters with eight scales and four orientations to estimate the Gabor images. The Gabor images were then fed into the Monte Carlo algorithm to generate a set of randomly sampled Gabor templates that were partially influenced by the occlusions. These templates served as a pool of local features used to replace the occluded template during testing [7]. The salient features least influenced by the occlusion were further determined during the learning process using a support vector machine (SVM). During testing, template matching was used to find the stored local features that most closely resembled the features located within the space around a partially occluded template [7]. This allows replacing the partially occluded templates with nearby non-occluded templates, resulting in a higher recognition rate. Given that four different template sizes and 1000 templates of each size are used for each of the seven emotions mentioned above, the final set contains 28,000 templates [7]. However, using this number of templates for matching may prove to be too large for some applications.

Benitez-Garcia et al. [12] proposed an FER scheme that takes the problem of partial occlusion into account. The system proposed in [12] first receives the face image and segments it into four regions using the approach proposed by Vukadinovic et al. [13], which detects the iris position. Then, using this information, it estimates the eyes, mouth, forehead and nose regions. These are characterized using the sub-block eigenphases method [14]. Next, the SVM is used to make a partial decision for each region. These decisions are then used together to make a final decision using their modal values. This approach is based on segmenting the face image into several regions instead of characterizing the entire facial image [12], which improves the emotion recognition rate for slightly occluded faces. The improvement occurs because the method makes a partial decision for each region, while the global decision is made using the modal values of all the individual decisions. Evaluation results showed that using the eyes, mouth, forehead and the all-face regions yields a recognition rate of 92%; however, even when the face region is partially occluded, the method still provides a recognition rate of 87% [12].

This paper proposes a FER algorithm that extends the scheme proposed in [15], in which the face image is segmented into two ROIs—the forehead/eyes and mouth regions. This segmentation makes it possible to make accurate decisions even if one of the two ROIs is partially or totally occluded. In the proposed system, after the ROI estimation, each region is segmented into a set of $N \times M$ blocks that are correlated with a set of Gabor functions [16,17] to obtain an $L \times NM$ feature matrix, where L is the number of training images. The resulting factures matrix is then applied to a PCA [11] for dimensionality reduction. We also propose a classifier that has low computational cost and is based on a fuzzy logic approach. This classifier provides recognition rates similar to those provided by other high-performance classifiers; however, its low computational cost allows the proposed scheme to be implemented even in smart devices with low computational power and in situations that require an immediate response. The proposed algorithm was evaluated on the KDEF data base [18,19] which consists of 490 color images of 70 people grouped by seven facial expressions. The database is used to carry out two different evaluations. In the first, the FER system was evaluated using only one ROI assuming that the other one was occluded. In the second, both ROIs were used for evaluation. The evaluation results show that when using only one ROI, the proposed system provides a recognition rate of 98%; however, when using both ROIs the recognition rate increases to 99%.

The rest of the paper is organized as follows: Section 2 describes the system framework. Section 3 shows the experimental results, and Section 4 provides conclusions.

2. Proposed system

Fig. 1 shows the system framework for the proposed (FER) system. First, a received image is fed into the face extraction stage, which detects the face image using the Viola-Jones algorithm [4]. Next, the detected face image is passed to the region of interest (ROI) segmentation algorithm, which first estimates the dimensions of the face using either the $I_R(x, y)$ and $I_G(x, y)$ planes in the RGB color space or the $I_Y(x, y)$, $I_U(x, y)$ or $I_V(x, y)$ planes in the YUV color space. Then, using this information, the ROIs are automatically segmented to obtain the mouth region and the forehead/eyes region using the image moments [15,20] and projective integrals [21,22]. These two ROIs are then segmented into $N \times M$ non-overlapping blocks that are then cross correlated with a set of $M_g = 54$ Gabor functions. Next, the first value of the estimated cross-correlation matrixes between each block and the set of M_g Gabor functions are averaged and the resulting vector of size NM is used to estimate the feature vectors of each ROI. These features are then concatenated into three different vectors: mouth, forehead/eyes, and mouth + forehead/eyes. Next, the three vectors are independently processed by the PCA stage for dimensionality reduction. Finally, the resulting vectors are fed into the proposed classifier stage to make the final decision. The following sections provide a more complete description of all the stages of the proposed system.

2.1. Face segmentation

The image received by the FER system is first processed by the Viola–Jones algorithm [4] to segment the face from the background. However, some areas of the detected face image such as the hair, ears, or background may contain noise that is not relevant for facial expression recognition. To eliminate unneeded

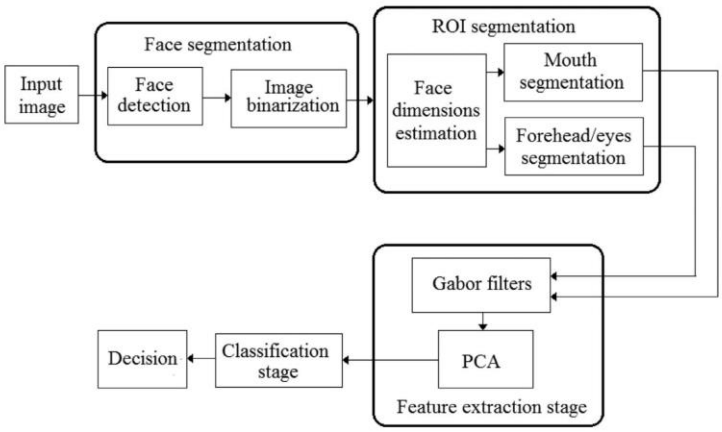


Fig. 1. Block diagram of the proposed algorithm.

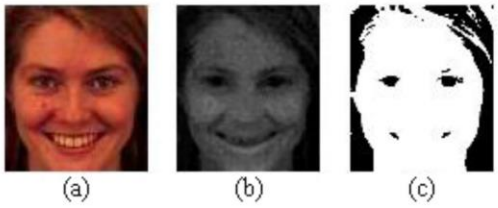


Fig. 2. (a) Original image, (b) Image obtained by the subtracting the green component from the red one, (c) a binarized image using a specified threshold.

content that may reduce the recognition rate of the proposed facial expression recognition system, a more accurate estimation of the facial dimensions is carried out by adjusting the dimensional parameters. To do this using the RGB color space, the face image is split into its three color components—the red, green and blue channels. Next, the green channel is subtracted from the red one [15], $I_{RG}(x, y)$, to highlight skin tones. Then, the resulting image is binarized as shown in Eq. (1) using an experimentally determined threshold, Th, $[0.1\overline{I_{RG}}(x, y) \leq Th \leq 0.3\overline{I_{RG}}(x, y)]$. Fig. 2 illustrates the process, and Eq. (1) is shown below:

$$I_B(x, y) = \begin{cases} 0 & I_{RG}(x, y) < Th \\ 255 & I_{RG}(x, y) \geq Th \end{cases}. \tag{1}$$

To use the YUV color space for face image segmentation, we follow the same procedure described above but replace the plane $I_{RG}(x, y)$, in Eq. (1) with one of the planes $I_Y(x, y)$, $I_U(x, y)$ or $I_V(x, y)$ with the threshold Th. The Th threshold is given by $[0.1\overline{I_U}(x, y) \leq Th \leq 0.3\overline{I_U}(x, y)]$ when using planes U or V or by $[0.1\overline{I_Y}(x, y) \leq Th \leq 0.3\overline{I_Y}(x, y)]$ when using the Y plane.

2.2. Adjusting the face dimension

After the image binarization shown in Fig. 2(c), the moments of the resulting binarized image are estimated as follows [15,20]:

$$M_{pq} = \sum_{x=1}^{N_B} \sum_{y=1}^{M_B} x^p y^q I_B(x, y), \tag{2}$$

where $I_B(x, y)$ is the binary image intensity at position (x, y) and N_B and M_B are the number of columns and rows in the image, respectively, while p and q denote the moment order of the image. Next, using Eq. (2), the centroid can be estimated as follows:

$$x_c = M_{1,0}/M_{0,0}, \tag{3}$$

$$y_c = M_{0,1}/M_{0,0}. \tag{4}$$

Then, using Eqs. (2)–(4), the following variables are defined

$$a = \frac{M_{2,0}}{M_{0,0}} - x_c^2, \tag{5}$$

$$b = 2\left(\frac{M_{1,1}}{M_{0,0}} - x_c y_c\right), \tag{6}$$

$$c = \frac{M_{0,2}}{M_{0,0}} - y_c^2. \tag{7}$$

Next, using Eqs. (5)–(7), the width of the face image can be estimated as follows:

$$W = 2\sqrt{\frac{(a + b) - \sqrt{b^2 + (a - c)^2}}{2}}. \tag{8}$$

Using W, the left (x_l) and right (x_r) edges of the face image can be estimated as

$$x_l = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil \tag{9}$$

and

$$x_r = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil + \lceil W \rceil. \tag{10}$$

Then, using W, the upper edge of the face image can be estimated as follows:

$$y_u = \lceil y_c \rceil - 0.84\left\lceil \frac{W}{2} \right\rceil. \tag{11}$$

Using Eqs. (9)–(11), the face image is segmented as illustrated in Figs. 3 and 4.

2.3. Forehead/eye segmentation

Forehead/eye face segmentation is a critical part of the proposed facial expression recognition system. To segment this area, the segmented face region is divided into three horizontal (A, B

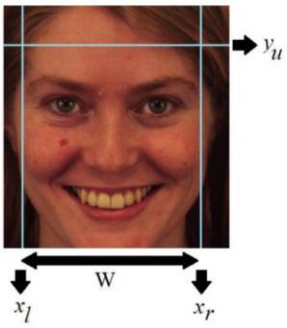


Fig. 3. Segmented face region.

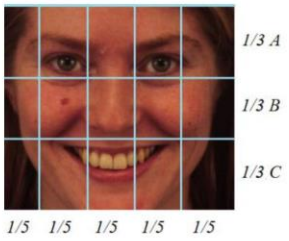


Fig. 4. Symmetrical relationship of the face image.

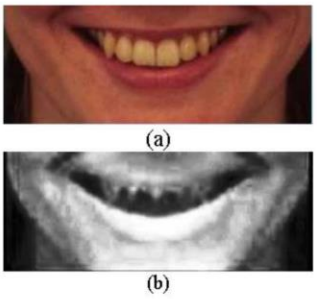


Fig. 5. (a) Original image and (b) equalized version of the image obtained by subtracting the red and green planes.

and C) regions, starting from the top at the point y_u given by Eq. (11) and shown in Figs. 3 and 4, where region A is the forehead/eyes ROI.

2.4. Mouth segmentation

To perform the segmentation of the mouth region, consider the detected face region, which is divided into three horizontal sections of the same size starting at the edge y_u, as describe in Section 2.3. Region C as shown in Fig. 4 is the mouth ROI. However, unlike the forehead/eyes region, in this case, it is necessary to segment only the mouth region. To this end, a histogram equalization of the image $I_{RG}(x, y)$, is performed [23] as shown in Fig. 5.

The next step in the automatic segmentation of the mouth region is to estimate the horizontal projective integral [21,22], which

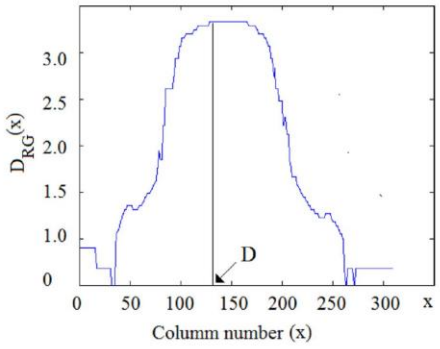


Fig. 6. Horizontal projective integral of the mouth ROI.

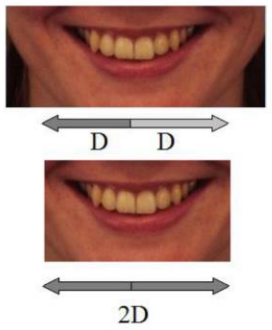


Fig. 7. Detection of the mouth ROI.

is the average of the pixel values of each column. This is a vector containing the average value of the pixels in each column of the image inside the ROI. Fig. 6 shows the horizontal projective integral estimated using the equalized image shown in Fig. 5b. Next, we obtain the position of maximum value of the projective integral,

$$D = x|D_{RG}(x) \rightarrow \max, \tag{12}$$

which is given by

$$|D_{RG}(x)|_{\max} = \left| \frac{1}{N_1} \sum_{y=1}^{N_1} I_{RG}(x, y) \right|_{\max} \tag{13}$$

Then, using the value *"D,"* the left border of the ROI containing the mouth is estimated by subtracting D from x_c, while the right border is obtained by adding it to x_c, keeping the original image height, as shown in Fig. 7. This procedure allows the automatic extraction of the mouth ROI

When using the YUV color space, the forehead/eye and mouth ROI segmentations can be carried out using the same procedure described in Sections 2.2–2.4, replacing the plane $I_{RG}(x, y)$ with one of the planes $I_Y(x, y)$, $I_U(x, y)$ or $I_V(x, y)$.

2.5. Feature extraction

To perform feature extraction, each of the detected ROIs is divided into $N \times M$ blocks, which are characterized by the average of the first term of the cross correlations between each block and 54

Gabor functions. Next, the resulting feature vectors of each of the L training ROIs with their NM elements are arranged in an $L \times NM$ matrix and applied to a PCA stage for dimensionality reduction. The following sections provide a brief description of these stages.

2.5.1. Gabor functions

The Gabor functions are widely used in many image processing applications such as texture analysis and face recognition tasks [16,17] because they are robust to luminescence changes. These functions have frequency responses with specific orientations, frequency-selective properties, and joint optimum resolution in both spatial and frequency domains. The 2D Gabor functions are given by

$$h(x, y, i, k) = g(x'y') \exp(j2\pi F_i x'), \tag{14}$$

where the parameters (x, y) express the location in the spatial domain, $F_i = \pi/2^{(i+1)}$, $i = 1,2,\ldots,N_F$ is the spatial frequency, $\phi_k = k\pi/N_\phi$, $k = 1,2,\ldots,N_\phi$ is the rotation angle, and $g(x', y')$ is the 2D Gaussian function given by

$$g(x', y') = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right), \tag{15}$$

where $\sigma = N/2$, and N is the number of blocks in the x axis and

$$(x', y') = x \cos\phi_k + y \sin\phi_k - x \sin\phi_k + y \cos\phi_k. \tag{16}$$

Thus, using the Gabor functions in Eqs. (15)–(17), the (n, m)th block of the ROI can be characterized as follows:

$$W_{mn} = \frac{1}{N_F N_\phi} \sum_{i=0}^{N_F} \sum_{k=0}^{N\phi} W(m, n, i, k), \tag{17}$$

where

$$W(m, n, i, k) = \left| \sum_{x=0}^{R-1} \sum_{y=0}^{Q-1} f(Rn + x, Qn + y) h(x, y, i, k) \right|. \tag{18}$$

2.6. Principal component analysis (PCA)

Principal component analysis is one of the most widely used methods for dimensionality reduction. It obtains a set of $L+1$ feature vectors, where L is the number of training images [11]. To accomplish this, it first represents the feature vector of the qth training image (given by Eq. (17)) in a vector form of size MN as follows:

$$W^{(q)} = \left[w_0^{(q)}, w_1^{(q)}, w_2^{(q)}, w_3^{(q)}, \ldots, w_r, \ldots, w_{NM-1}^{(q)}\right], \tag{19}$$

where $r = nM + m$, $m = 0, 1,\ldots,M-1$, $n = 0, 1,\ldots,N-1$, and MN is the feature vector size of each ROI. Next, a matrix $\mathbf{G}$ of size $L \times (NM)$ is constructed, concatenating all vectors $\mathbf{W}^{(q)}$ contained in the training set, where $L = TS < NM$, T is the number of classes, and S is the number of training images for each class, as shown below:

$$\mathbf{\Psi} = \begin{bmatrix} \psi_{0,0}, & \psi_{0,1}, & \psi_{0,2} & \ldots\psi_{0,r}\ldots & . & \psi_{o,NM-1} \\ \psi_{1,0}, & \psi_{1,1}, & \psi_{1,2} & \ldots\psi_{1,r}\ldots & . & \psi_{1,NM-1} \\ & \ldots & & \ldots & . & \ldots \\ \psi_{L,0}, & \psi_{L,1}, & \psi_{L,2} & \ldots\psi_{L,r}\ldots & . & \psi_{L,NM-1} \end{bmatrix}, \tag{20}$$

where

$$\psi_{r,l} = w_r^{(l)} - \bar{w}_r, \tag{21}$$

and

$$\bar{w}_r = \frac{1}{L} \sum_{k=0}^{L-1} w_r^{(l)}. \tag{22}$$

Next, the eigenvectors and eigenvalues of the covariance matrix given by $\mathbf{\Psi}^T\mathbf{\Psi}$ associated with the $L+1$ largest eigenvalues are estimated. These are sorted from largest to smallest and used to generate a dominant feature matrix $\mathbf{\Phi}$ of size $L \times NM$, where L corresponds to the total number of training vectors. Finally, the feature vector of the image under analysis is given by

$$\mathbf{\Gamma}_i = \begin{bmatrix} \phi_0^{(0)}, \phi_1^{(0)}, \phi_2^{(0)} & \ldots\phi_r^{(0)}\ldots & . & \phi_{NM-1}^{(0)} \\ \phi_0^{(1)}, \phi_1^{(1)}, \phi_2^{(1)} & \ldots\phi_r^{(1)}\ldots & . & \phi_{NM-1}^{(1)} \\ \ldots & \ldots & . & \ldots \\ \phi_0^{(L)}, \phi_1^{(L)}, \phi_2^{(L)} & \ldots\phi_r^{(L)}\ldots & . & \phi_{NM-1}^{(L)} \end{bmatrix} \begin{bmatrix} w_0^i \\ w_1^i \\ . \\ . \\ w_{NM-1}^i \end{bmatrix}, \tag{23}$$

where $\mathbf{\Gamma}_i$ is the resulting feature vector of size L that is fed into the classification stage, $[\ \phi_0^{(q)}, \phi_1^{(q)}, \phi_2^{(q)} \quad \ldots\phi_r^{(q)} \quad \phi_{NM-1}^{(q)}\ ]$ is the qth eigenvector, and $[w_0^i, w_1^i, \ldots, w_{NM-1}^i]^T$ is the feature vector of the image under analysis belonging to the ith stage, as given by Eq. (19).

2.7. Classification stage

The proposed classifier (Fig. 8) operates in a supervised form to generate several clusters for each class. The number of classes, I, and the number of training patterns, K, for each class are assumed to be known, although the number of clusters that will be generated for each class is unknown beforehand. It is also important to mention that in the proposed classifier the clusters of each class are trained independently of the other classes. Thus if a new class is added, the already trained clusters remain unchanged, and it is necessary to estimate only the clusters for the new class.

To develop the training algorithm, consider the following matrix:

$$\mathbf{\Gamma} = [\mathbf{\Gamma}_0, \mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \ldots, \mathbf{\Gamma}_i, \ldots, \mathbf{\Gamma}_I], \tag{24}$$

where

$$\mathbf{\Gamma}_i = \begin{bmatrix} \Gamma_{i,0,0} & \Gamma_{i,0,1} & \ldots & \Gamma_{i,0,B} \\ \Gamma_{i,1,0} & \Gamma_{i,1,1} & \ldots & \Gamma_{i,1,B} \\ \ldots & \ldots & \ldots & \ldots \\ \Gamma_{i,K,0} & \Gamma_{i,K,1} & \ldots & \Gamma_{i,K,B} \end{bmatrix} \tag{25}$$

denotes the features vectors of the ith class, $\Gamma_{i,n,m}$ corresponds to the mth component of the nth feature vector of the ith class, and $B = NM$. Before training the classifier, the vector containing the number of clusters created for each class $\mathbf{V}(i)$, the parameter denoting the total number of clusters in the system, γ, and the class, i, are set to zero. Because the clusters belonging to each class are independently estimated, system training begins by initializing the clusters belonging to the class under analysis.

Step 1. Initialization process

Consider $\mathbf{Y} = \mathbf{\Gamma}_0$ and $y_{n,m} = \Gamma_{n,m}$, that is,

$$\mathbf{Y} = \begin{bmatrix} y_{0,0} & y_{0,1} & \ldots. & y_{0,B} \\ y_{1,0} & y_{1,1} & \ldots. & y_{1,B} \\ \ldots. & \ldots. & \ldots. & \ldots. \\ y_{L,0} & y_{L,1} & \ldots. & y_{L,B} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_0^T \\ \mathbf{Y}_0^T \\ . \\ \mathbf{Y}_L^T \end{bmatrix}. \tag{26}$$

Next, assume that the index of the first cluster is $p = 0$; then, build a vector $\mathbf{N}_c$ containing the number of elements in each cluster initialized with zeroes, that is,

$$N_c = [0,\ 0,\ 0,\ 0, \ldots.0], \tag{27}$$

Next, build a vector $\mathbf{S}_i$ containing the variance of each cluster—also initialized with zeros,

$$\mathbf{S}_i = [0,\ 0,\ 0,\ 0, \ldots.0], \tag{28}$$

and assign the first row of matrix $\mathbf{Y}$ to the initial value of the first centroid:

$$\mathbf{C}(0, j) = \mathbf{Y}(0, j), \quad j = 0,\ 1,\ 2,\ 3\ldots, B. \tag{29}$$

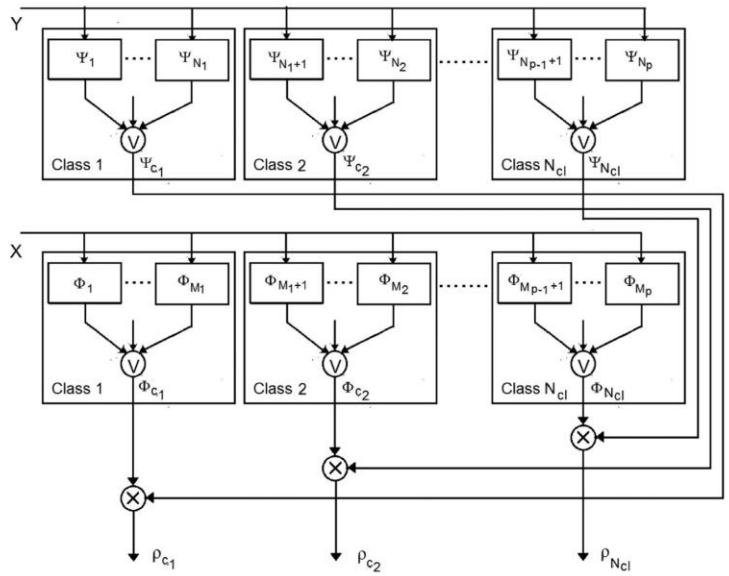


Fig. 8. Proposed classifier based on a fuzzy logic approach when classifying facial expressions using both the mouth and forehead/eyes regions.

Next, set the matrix $\mathbf{Y}^{(1)}$ by eliminating the first row from $\mathbf{Y}$, which was used to estimate the centroid of the first cluster as shown in Eq. (29). That is,

$$\mathbf{Y}^{(1)} = \begin{bmatrix} y^{(1)}_{0,0} & y^{(1)}_{0,1} & .... & y^{(1)}_{0,B} \\ y^{(1)}_{1,0} & y^{(1)}_{1,1} & .... & y^{(1)}_{1,B} \\ .... & .... & .... & .... \\ y^{(1)}_{L_N,0} & y^{(1)}_{L_N,1} & .... & y^{(1)}_{L_N,B} \end{bmatrix}, \tag{30}$$

where $\mathbf{Y}^{(1)}$ is a matrix of $L_N \times B$ and $L_N = L-1$. Next, determine the rows of $\mathbf{Y}^{(1)}$ that have the maximum and minimum distance with respect to C(0,j), that is

$$d_M = \max(d_k), \quad k = 0,\ 1,\ 2, \ldots, L_N, \tag{31}$$

and

$$d_m = \min(d_k), \quad k = 0,\ 1,\ 2, \ldots, L_N, \tag{32}$$

where

$$d_k = \sum_{j=0}^{B} \left(\mathbf{Y}^{(1)}(k,j) - \mathbf{C}(0,j)\right)^2, \quad k = 0,\ 1,\ 2, \ldots, L_N. \tag{33}$$

Then, using the mth row of $Y^{(1)}$ (which corresponds to the minimum distance), modify $N_c(0)$, the centroid C(0,j), and the variance $S_i(0,j)$ as follows:

$$Nc(0) = Nc(0) + 1, \tag{34}$$

$$\mathbf{C}(0,j) = \frac{Nc(0)-1}{Nc(0)}\mathbf{C}(0,j) + \frac{1}{Nc(0)}\mathbf{Y}^{(1)}(m,j), \tag{35}$$

$$\mathbf{S}(0,j) = \frac{Nc(0)-1}{Nc(0)}\mathbf{S}(0,j) + \frac{1}{Nc(0)}\left(\mathbf{Y}^{(1)}(m,j) - \mathbf{C}(0,j)\right)^2. \tag{36}$$

Next, using the Mth row, the second cluster of the first class is estimated as follows:

$$\mathbf{C}(1,j) = \mathbf{Y}^{(1)}(M,j), \quad j = 0,\ 1,\ 2,\ 3..., B. \tag{37}$$

In the second iteration, first compute the matrix $\mathbf{Y}^{(2)}$, deleting the mth and Mth rows from matrix $\mathbf{Y}^{(1)}$. Thus,

$$\mathbf{Y}^{(2)} = \begin{bmatrix} y^{(2)}_{0,0} & y^{(2)}_{0,1} & .... & y^{(2)}_{0,B} \\ y^{(2)}_{1,0} & y^{(2)}_{1,1} & .... & y^{(2)}_{1,B} \\ .... & .... & .... & .... \\ y^{(2)}_{L_N,0} & y^{(2)}_{L_N,1} & .... & y^{(2)}_{L_N,B} \end{bmatrix}, \tag{38}$$

where $\mathbf{Y}^{(2)}$ is a matrix of $L_N \times B$ and $L_N = L-3$. Then, estimate the rows of $\mathbf{Y}^{(2)}$ that have the maximum and minimum distance with respect to C(p,j), $p = 0,\ 1$ using

$$d_{M,N} = \max(d_{k,r}), \quad k = 0,\ 1,\ 2, \ldots, L_N; \quad r = 0,\ 1, \tag{39}$$

and

$$d_{m,n} = \min(d_{k,r}), \quad k = 0,\ 1,\ 2, \ldots, L_N; \quad r = 0,\ 1, \tag{40}$$

where m denotes the row index of $\mathbf{Y}^{(2)}$ and n is the index of the centroid with the smallest distance among them, while M is the row index of the $\mathbf{Y}^{(2)}$, and N is the index of the cluster with the maximum distance among them, and

$$d_{k,r} = \sum_{j=0}^{B} \left(\mathbf{Y}^{(2)}(k,j) - \mathbf{C}(r,j)\right)^2, \quad k = 0,\ 1,\ 2, \ldots, L_N; \quad r = 1,\ 2. \tag{41}$$

Next, using the mth row, which corresponds to the minimum distance, modify N_c(n), the centroid $\mathbf{C}$(n,j), and the variance $\mathbf{S}_i$(n,j), where n and m are given as follows:

$$Nc(n) = Nc(n) + 1, \tag{42}$$

$$\mathbf{C}(n,j) = \frac{Nc(n)-1}{Nc(n)}\mathbf{C}(n,j) + \frac{1}{Nc(n)}\mathbf{Y}^{(2)}(m,j), \tag{43}$$

$$\mathbf{S}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{S}(n,j)+\frac{1}{Nc(n)}\left(\mathbf{Y}^{(2)}(m,j)-\mathbf{C}(n,j)\right)^2 \quad (44)$$

Next, increasing the number of clusters by one (that is, $p=p+1$) and using the row of $\mathbf{Y}^{(2)}$ whose distance with respect to all clusters is the largest, obtain the value of the pth cluster:

$$\mathbf{C}(p,j)=\mathbf{Y}^{(2)}(M,j),\quad j=0,\ 1,\ 2,\ 3...,B. \quad (45)$$

In the third iteration, compute the matrix $\mathbf{Y}^{(3)}$, deleting the mth and Mth rows from $\mathbf{Y}^{(2)}$. Thus,

$$\mathbf{Y}^{(3)}=\begin{bmatrix} y^{(3)}_{0,0} & y^{(3)}_{0,1} & .... & y^{(3)}_{0,B} \\ y^{(3)}_{1,0} & y^{(3)}_{1,1} & .... & y^{(3)}_{1,B} \\ .... & .... & .... & .... \\ y^{(3)}_{L_N,0} & y^{(3)}_{L_N,1} & .... & y^{(3)}_{L_N,B} \end{bmatrix}, \quad (46)$$

where $\mathbf{Y}^{(3)}$ is a matrix of $L_N\times B$ and $L_N=L-5$. Next, estimate the rows of $\mathbf{Y}^{(3)}$ whose distances with respect to C(p,j), $p=0,\ 1,\ 2$, are the maximum and minimum, respectively, as follows:

$$d_{M,N}=\max(d_{k,r}),\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=0,\ 1,\ 2 \quad (47)$$

and

$$d_{m,n}=\min(d_{k,r}),\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=0,\ 1,\ 2, \quad (48)$$

where m denotes the row index of $\mathbf{Y}^{(3)}$, and n denotes the index of the centroid with the smallest distance among them, while M is the index of $\mathbf{Y}^{(3)}$, and N is the index of the cluster with the largest distance among them, and

$$d_{k,r}=\sum_{j=0}^{B}\left(\mathbf{Y}^{(3)}(k,j)-\mathbf{C}(r,j)\right)^2,\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=0,\ 1,\ 2. \quad (49)$$

Next, using the mth row, which corresponds to the minimum distance, modify N_c(n), the centroid C(n,j), and the variance S(n,j), where n and m are determined as follows:

$$Nc(n)=Nc(n)+1, \quad (50)$$

$$\mathbf{C}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{C}(n,j)+\frac{1}{Nc(n)}\mathbf{Y}^{(3)}(m,j), \quad (51)$$

And

$$\mathbf{S}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{S}(n,j)+\frac{1}{Nc(n)}\left(\mathbf{Y}^{(3)}(m,j)-\mathbf{C}(n,j)\right)^2. \quad (52)$$

Next, increment the number of clusters by one (that is, $p=p+1$) and, using the row of $\mathbf{Y}^{(3)}$ whose distance with respect to all clusters in the ith class is the largest, obtain the value of the pth cluster as follows:

$$\mathbf{C}(p,j)=\mathbf{Y}^{(3)}(M,j),\quad j=0,\ 1,\ 2,\ 3...,B. \quad (53)$$

Step 2.

In general, compute the matrix $\mathbf{Y}^{(t+1)}$, deleting from $\mathbf{Y}^{(t)}$ the mth and Mth rows obtained in the previous stage and set $L_N=L_N-2$. Thus

$$\mathbf{Y}^{(t+1)}=\begin{bmatrix} y^{(t+1)}_{0,0} & y^{(t+1)}_{0,1} & .... & y^{(t+1)}_{0,B} \\ y^{(t+1)}_{1,0} & y^{(t+1)}_{1,1} & .... & y^{(t+1)}_{1,B} \\ .... & .... & .... & .... \\ y^{(t+1)}_{L_N,0} & y^{(t+1)}_{L_N,1} & .... & y^{(t+1)}_{L_N,B} \end{bmatrix}, \quad (54)$$

Next, estimate the rows of $\mathbf{Y}^{(t+1)}$ that have the maximum and minimum distance with respect to C(p,j), $p=0,\ 1,\ \ldots t+1$ as follows:

$$d_{M,N}=\max(d_{k,r}),\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=0,\ 1,\ldots,2t \quad (55)$$

and

$$d_{m,n}=\min(d_{k,r}),\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=0,\ 1,\ ,\ldots,2t, \quad (56)$$

where m denotes the row index of $\mathbf{Y}^{(t+1)}$, and n is the centroid index with the smallest distance, while M is the index of $\mathbf{Y}^{(t+1)}$, and N is the index of the cluster with the maximum distance among all clusters belonging to the ith class, and

$$d_{k,r}=\sum_{j=0}^{B}\left(\mathbf{Y}^{(t+1)}(k,j)-\mathbf{C}(r,j)\right)^2,\quad k=0,\ 1,\ 2,\ldots,L_N;\quad r=1,\ 2,\ldots,2t. \quad (57)$$

Next, using the mth row, which corresponds to the minimum distance, modify N_c(n), the centroid C(n,j), and the variance S(n,j), where n and m are as follows:

$$Nc(n)=Nc(n)+1, \quad (58)$$

$$\mathbf{C}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{C}(n,j)+\frac{1}{Nc(n)}\mathbf{Y}^{(t+1)}(m,j), \quad (59)$$

And

$$\mathbf{S}(n,j)=\frac{Nc(n)-1}{Nc(n)}\mathbf{S}(n,j)+\frac{1}{Nc(n)}\left(\mathbf{Y}^{(t+1)}(m,j)-\mathbf{C}(n,j)\right)^2 \quad (60)$$

Next, using the row in $\mathbf{Y}^{(t+1)}$ whose distance with respect to all clusters is the largest, obtain the value of the cluster $p=p+1$ as follows:

$$\mathbf{C}(p,j)=\mathbf{Y}^{(t+1)}(M,j),\quad j=0,\ 1,\ 2,\ 3...,B. \quad (61)$$

If $L_N>2$, then go to Step 2.

After the training vectors belonging to the ith class have been analyzed, the centroids and variances C_i and S_i of each of the resulting p clusters are arranged as follows:

$$\mathbf{C}_i=\left[C^T_{N_{i-1}},C^T_{N_{i-1}+1},C^T_{N_{i-1}+2},\ldots,C^T_{N_i}\right]^T \quad (62)$$

and

$$\mathbf{S}_i=\left[S^T_{N_{i-1}},S^T_{N_{i-1}+1},S^T_{N_{i-1}+2},\ldots,S^T_{N_i}\right]^T, \quad (63)$$

where $N_i=N_{i-1}+p$ denotes the number of clusters of the ith stage. Next, set $W(\gamma)=p$, $\lambda=\lambda+p$ and $\gamma=\gamma+1$. If $\gamma<I$, where I is the total number of classes, go to Step 1; otherwise, the centers and variances of all clusters belonging to any of the I classes are arranged as follows:

$$\mathbf{C}=\left[C^T_0,\ldots,C^T_{N_1},C^T_{N_1+1},\ldots,C_{N_2},C^T_{N_2+1},\ldots,C^T_{N_{\lambda-1}},\ldots,C^T_{N_\lambda}\right]^T \quad (64)$$

and

$$\mathbf{S}=\left[S^T_0,\ldots,S^T_{N_1},S^T_{N_1+1},\ldots,S_{N_2},S^T_{N_2+1},\ldots,S^T_{N_{\lambda-1}},\ldots,S^T_{N_\lambda}\right]^T, \quad (65)$$

where $N_{i+1}-N_i$ denotes the number of clusters of the $(i+1)$th class.

2.7.1. Evaluation stage

During training, the system estimates the N_i-N_{i-1} clusters belonging to the ith class of the kth ROI, where $i=1,\ldots,\lambda_k$ and $k=1,2$. These will be used in the evaluation stage. Here, because any class is classified using N_i-N_{i-1} clusters, the system must first evaluate whether the input pattern belongs to any such clusters. Then, it uses that information to determine if the input pattern belongs to the jth class. To this end, the feature vectors of the kth ROI are first estimated as follows:

$$\mathbf{\Gamma} = \begin{bmatrix} \varphi_0^{(0)}, \varphi_1^{(0)}, \varphi_2^{(0)} & ..\varphi_r^{(0)}.. & . & \varphi_{NM-1}^{(0)} \\ \varphi_0^{(1)}, \varphi_1^{(1)}, \varphi_2^{(1)} & ..\varphi_r^{(1)}.. & . & \varphi_{NM-1}^{(1)} \\ ... & .. & . & .. \\ \varphi_0^{(L)}, \varphi_1^{(L)}, \varphi_2^{(L)} & ..\varphi_r^{(L)}.. & . & \varphi_{NM-1}^{(L)} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ . \\ . \\ w_{NM-1} \end{bmatrix}, \quad (66)$$

where w_i is given by Eqs. (18)–(20). Then, the following expression is evaluated:

$$IF\ \mathbf{\Gamma} \in \mathbf{C}_{i,1}\ \text{OR}\ \mathbf{\Gamma} \in \mathbf{C}_{i,2}\ \text{OR}\ \mathbf{\Gamma} \in \mathbf{C}_{i,3}\ \text{OR}.....\mathbf{\Gamma} \in \mathbf{C}_{i,(N_i - N_{i-1})}\ \text{THEN}\ \mathbf{\Gamma} \in \rho_{c_i}, \quad (67)$$

where ρ_{c_i} denotes the c_i class. To evaluate Eq. (67), taking into account that each cluster is characterized by its center and variance, we can evaluate the membership degree of the input pattern to the ρ_{c_i} class using the vector Γ given by (66) as follows:

$$\psi_{c_i} = \prod_{j=0}^{L} \left(\exp\left(\frac{(\Gamma(j) - C(i,j))^2}{2S^2(i,j)} \right) \right) \quad 1 \le i \le (N_i - N_{i-1}). \quad (68)$$

Then, the membership degree of the input pattern to the ρ_{c_i} is given by

$$\psi_{c_k} = \max\{\psi_{c_i}\}, \quad 1 \le i \le (N_i - N_{i-1}); \quad k = 1, 2, \ldots, \rho_{N_{cl}}. \quad (69)$$

In the particular case in which we have two ROIs (forehead/eyes and mouth), with six classes for each ROI, we have two sets of memberships such that the final decision is made as follows:

$$\rho_{c_k} = \varphi_{c_k} \psi_{c_k}, \quad 0 \le k \le \rho_{N_{cl}}, \quad (70)$$

where

$$\psi_{c_k} = \max\{\psi_i\}, \quad 0 \le k \le \rho_{N_{cl}}, \quad (71)$$

$$\varphi_{c_k} = \max\{\phi_i\}, \quad 0 \le k \le \rho_{N_{cl}}, \quad (72)$$

and

$$\phi_{c_i} = \prod_{j=0}^{L} \left(\exp\left(\frac{(Z(j) - F(i,j))^2}{2V^2(i,j)} \right) \right) \quad 1 \le i \le (N_i - N_{i-1}), \quad (73)$$

where Z(j) denotes the j-th component of the features vector of the ROI under analysis.

The estimated centroids **G** and **F** and the variance **S** and **V** of the clusters of each ROI are given by

$$\mathbf{G} = \left[\mathbf{C}_1^T, \ldots, \mathbf{C}_{N_1+1}^T, \mathbf{C}_{N_1+2}^T, \ldots, \mathbf{C}_{N_2}^T, \mathbf{C}_{N_2+1}^T, \ldots, \mathbf{C}_{N_3}^T, \ldots, \mathbf{C}_{N_{\lambda-1}}^T, \ldots, \mathbf{C}_{N_\lambda}^T \right]^T, \quad (74)$$

$$\mathbf{F} = \left[\mathbf{F}_1^T, \ldots, \mathbf{F}_{N_1+1}^T, \mathbf{F}_{N_1+2}^T, \ldots, \mathbf{F}_{N_2}^T, \mathbf{F}_{N_2+1}^T, \ldots, \mathbf{F}_{N_3}^T, \ldots, \mathbf{F}_{N_{\lambda-1}}^T, \ldots, \mathbf{F}_{N\lambda}^T \right]^T, \quad (75)$$

$$\mathbf{S} = \left[\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_1+1}^T, \mathbf{S}_{N_1+2}^T, \ldots, \mathbf{S}_{N_2}^T, \mathbf{S}_{N_2+1}^T, \ldots, \mathbf{S}_{N_3}^T, \ldots, \mathbf{S}_{N_{\lambda-1}}^T, \ldots, \mathbf{S}_{N_\gamma}^T \right]^T, \quad (76)$$

and

$$\mathbf{V} = \left[\mathbf{V}_1^T, \ldots, \mathbf{S}_{N_1+1}^T, \mathbf{V}_{N_1+2}^T, \ldots, \mathbf{V}_{N_2}^T, \mathbf{V}_{N_2+1}^T, \ldots, \mathbf{V}_{N_3}^T, \ldots, \mathbf{V}_{N_{\lambda-1}}^T, \ldots, \mathbf{V}_{N_\lambda}^T \right]^T. \quad (77)$$

Eq. (70) provides the membership degree of the facial expression under analysis to the C_k class. Thus, because each class corresponds to a different emotional state, the proposed system is able not only to determine the emotional state of the person under analysis but also its degree. For example, if the person under analysis is sad, the proposed system is able to determine not only that the person is sad but also the degree of sadness. Thus, Eq. (70) can be associated with the membership degree of each one of the emotional states under analysis.

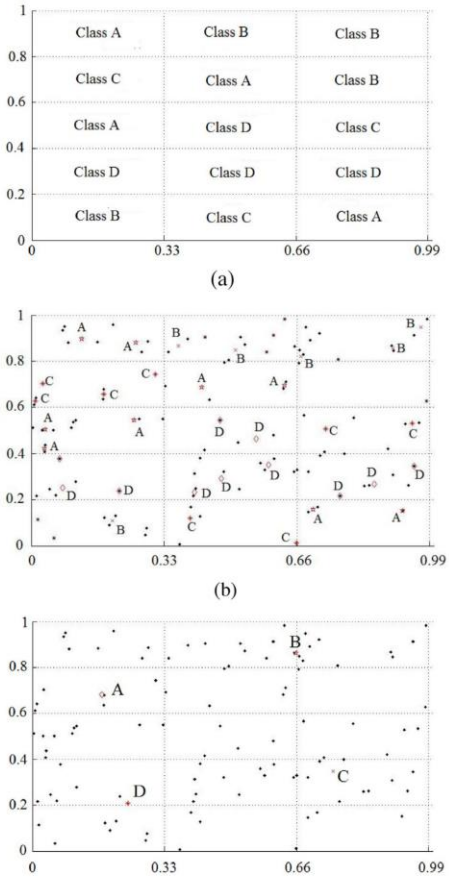


Fig. 9. Performances of the proposed scheme when classifying 4 disjoint classes. (a) classes, (b) clusters generated by the proposed scheme, (c) clusters generated by the *K*-means algorithm.

3. Evaluation results

To evaluate the cluster estimation capability of the proposed classifier and compare it with the conventional *K*-means algorithm, both schemes were required to classify 100 numbers [24]. These numbers were randomly distributed between 0 and 1 and belonged to 4 different classes distributed as shown in Fig. 9a. After training both the proposed classifier and the *K*-means algorithm, the schemes were evaluated using a different set of numbers randomly distributed between 0 and 1. Fig. 9b and c show the testing data along with the generated centroids. From these figures, it is evident that the proposed scheme can correctly classify the testing data even when the classes are disjointed, but that a conventional scheme cannot.

Next, we evaluate the emotion recognition capability of the proposed algorithm. To this end the KDEF data base [18,19], which consists of 490 color images of 70 people grouped by seven

facial expressions, was used to carry out various evaluations. First, the ability of the proposed algorithm to properly segment the face images when they were stored using either the RGB or YUV color spaces was evaluated. This is important because an accurate ROI segmentation strongly depends on accurate face segmentation and binarization—the facial dimensions depend on proper estimation of the moments.

Fig. 10 shows the binarization results using different threshold values for input images coded using the RGB color space. Figs. 11–14 show the segmented face images using several threshold values for input images represented using both the RGB and YUV color spaces. In Fig. 11 the image is segmented using the $I_{RG}(x,y)$ plane for input images saved in the RGB color space, as described in Section 2.1. In contrast, Figs. 12–15 show the face segmentation performance of the proposed algorithm using the planes $I_Y(x,y)$, $I_V(x,y)$ and $I_U(x,y)$, respectively, for face images coded using the YUV color space. The evaluation results show that proposed scheme (described in Section 2) provides fairly good segmentation performance regardless of whether the input image is encoded using the RGB or the YUV color space.

After the face image segmentation, estimating the forehead/eyes and mouth ROIs is an important task carried out using the moments estimated from the binarized images. Fig. 16 shows the estimated ROIs under different illumination conditions. The evaluation results show that proposed scheme (described in Section 2) provides a fairly good ROI estimation performance under different illumination conditions.

To carry out the recognition performance evaluation of the proposed algorithm, after the face region detection, the mouth ROIs were reduced to 180×90 pixels, while the forehead/eyes ROIs were resized to 300×150 pixels. Starting with the goal of finding the optimal window sizes for the Gabor filters, different window sizes were analyzed in both face regions, using 50 images for training and 20 for testing. Fig. 17 shows that the mouth region achieved a recognition rate of approximately 97.85% when the Gabor window sizes were between 10×10 and 30×30, while for the forehead/eyes region, the recognition rate was 99.28%. In all cases the Gabor functions were used with $N_F = 6$ *and* $N_\phi = 9$. Thus, a suitable window size is 30×30 pixels because the computational cost is lower. The same experiment was performed using the ANN as a classifier, with results for the mouth region of 97.14%. The training was performed using 50 images and Gabor filter windows of 30×30, while the forehead/eyes region obtained a recognition rate of 91.42% using 50 training images and a window size of 50×50. Both recognition percentages are lower than those obtained using the proposed classifier.

Fig. 18 shows the evaluation results obtained with different numbers of training patterns. It is important to mention that for the 3 cases under analysis (the mouth region using 30×30 windows, the forehead/eyes region using 50×50 windows, and the mouth region using 30×30 windows + the forehead/eyes region using 50×50 windows) the recognition rates are higher than 97% when both regions—mouth and forehead/eyes—are used. The evaluation results show that the highest recognition rate is obtained when the mouth and forehead regions are jointly used, achieving a recognition rate of approximately 99%.

Fig. 19 shows a recognition performance comparison between the proposed facial expression recognition scheme and other recently proposed schemes. Among these other schemes are the FER proposed by Ali et al. [2] which provides a fairly good recognition performance using the entire face without occlusions; the FER presented by Benitez et al [14], which performs fairly well even when there is some occlusion of the face image; and the FER provided by Zhang et al. [7], which is robust to face occlusions because it replaces partially occluded templates with non-occluded templates stored in a database. This system provides a fairly good performance, although its complexity is high because it must match the 28,000 stored templates [7]. The performance was evaluated using the mouth and the forehead/eyes regions. Other efficient methods compared to the proposed scheme are described in [5,25].

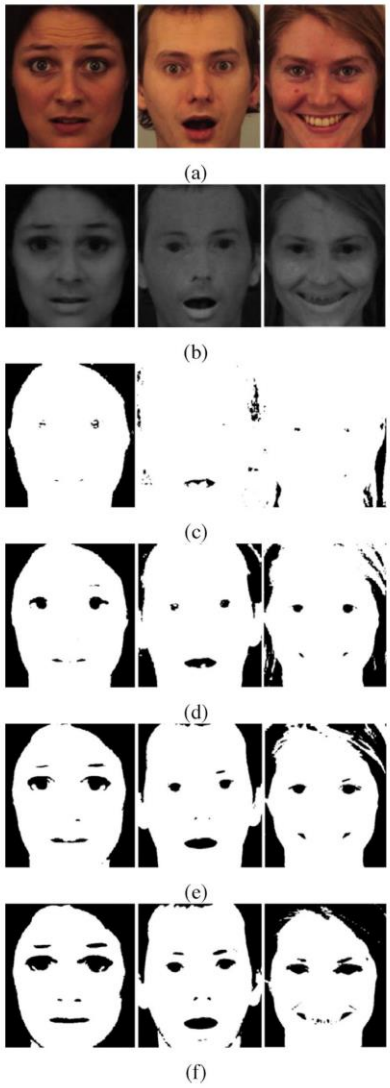


Fig. 10. Binarized image results using several threshold values (a) detected images, (b) images obtained after subtracting the green component from the red component, (c) binarized image using the threshold $Th = 0.1\bar{x}$, (d) binarized image using the threshold $Th = 0.2\bar{x}$, (e) binarized image using the threshold $Th = 0.3\bar{x}$, (f) binarized image using the threshold $Th = 0.4\bar{x}$.

The evaluation results show that proposed scheme performs better than some other previously proposed methods such as [5,7,12] and close to that reported in [2]. It is worth mentioning

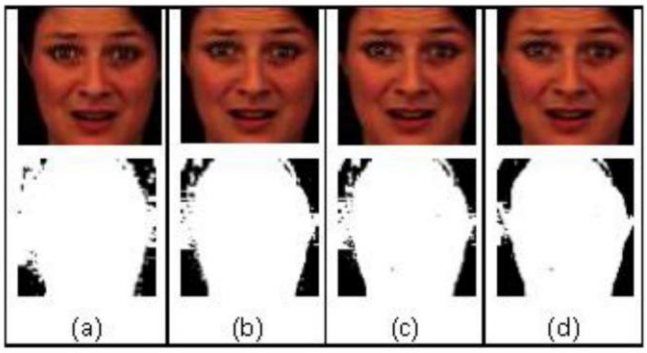


Fig. 11. Segmented and binarized images performed in the $I_{RG}(x, y)$ plane using threshold values equal to: (a) $Th = 5$, (b) $Th = 6$, (c) $Th = 7$, (d) $Th = 8$.

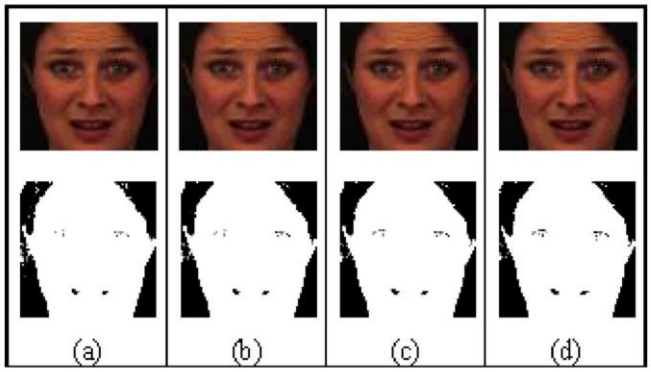


Fig. 12. Segmented and binarized images performed in the $I_Y(x, y)$ plane using threshold values equal to: (a) $Th = 4$, (b) $Th = 5$, (c) $Th = 6$, (d) $Th = 7$.

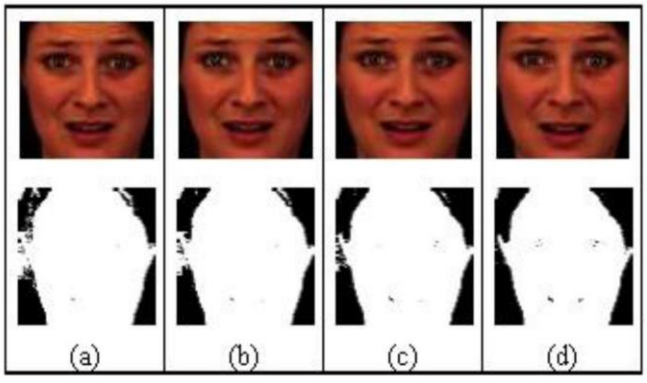


Fig. 13. Segmented and binarized images performed in the $I_V(x, y)$ plane using threshold values equal to: (a) $Th = 4$, (b) $Th = 5$, (c) $Th = 6$, (d) $Th = 7$.

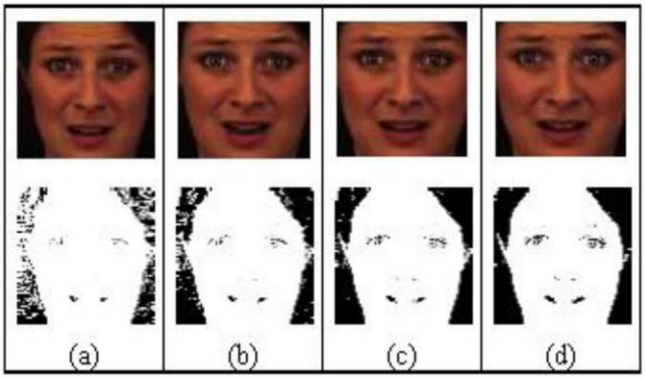


Fig. 14. Segmented and binarized images performed in the $I_U(x, y)$ plane using threshold values equal to: (a) $Th=4$, (b) $Th=5$, (c) $Th=6$, (d) $Th=7$.

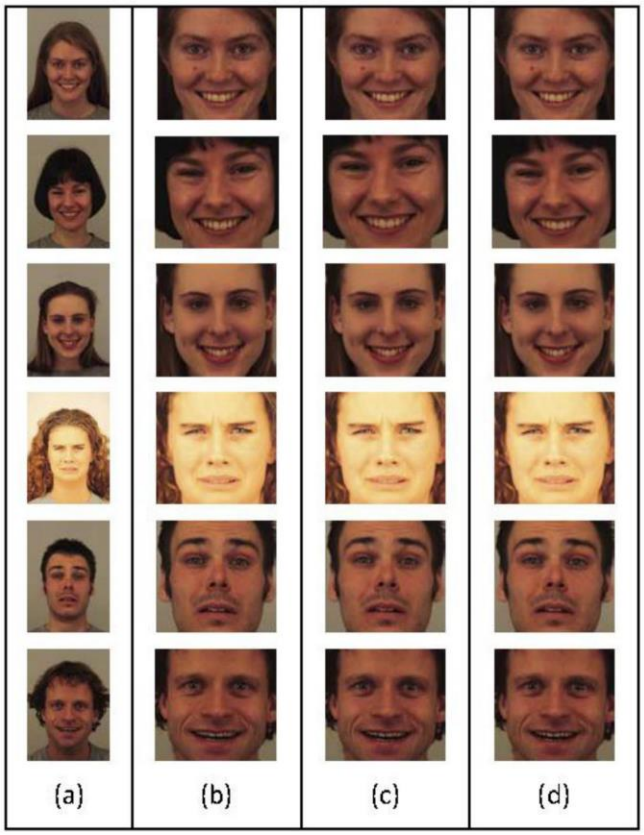


Fig. 15. Segmented images using the proposed scheme when the input face image is encoded using the YUV color space using the following planes and threshold values: (b) $I_Y(x, y)$ and $Th=10$, (c) $I_U(x, y)$ and $Th=4$, (d) $I_V(x, y)$ and $Th=4$. The original images are shown in (a).

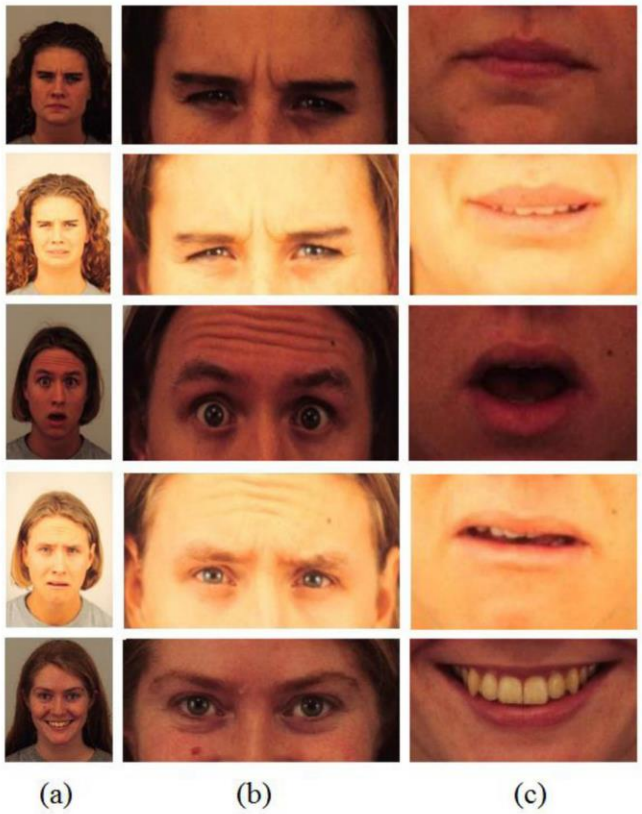
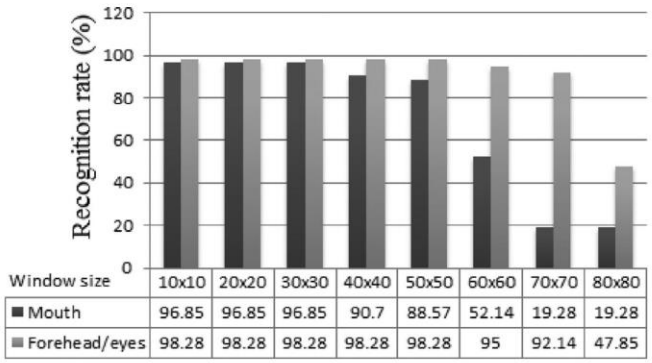
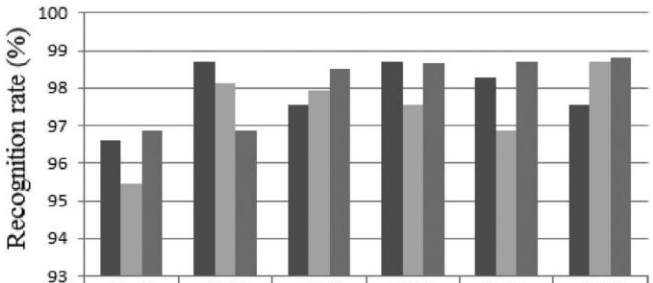


Fig. 16. Face region segmentation stage obtained using the proposed scheme, (a) original images, (b) forehead/eyes segmentation, (c) mouth segmentation.



Window size	10x10	20x20	30x30	40x40	50x50	60x60	70x70	80x80
Mouth	96.85	96.85	96.85	90.7	88.57	52.14	19.28	19.28
Forehead/eyes	98.28	98.28	98.28	98.28	98.28	95	92.14	47.85

Fig. 17. Average recognition rates with different window sizes.



	10 TP	20 TP	30 TP	40 TP	50 TP	60 TP
M	96.61	98.71	97.57	98.71	98.28	97.57
FH/E	95.47	98.14	97.92	97.57	96.85	98.71
M+FH/E	96.85	96.85	98.52	98.64	98.71	98.81

Fig. 18. Average recognition rates with different training patterns.

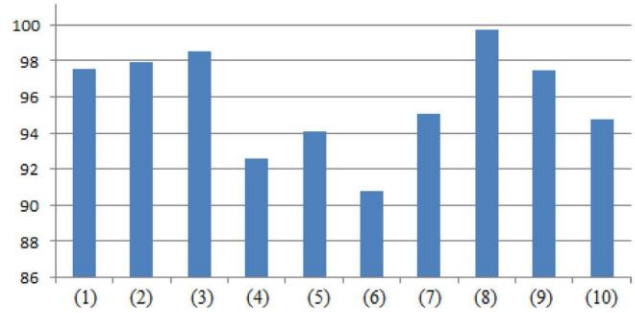


Fig. 19. Recognition performance of (1) proposed scheme using the mouth region, (2) proposed scheme using the forehead/eyes, (3) Proposed scheme using both regions. The performance of the algorithms proposed by (4) Benitez et al. [12], (5) Zhang et al. using the mouth region [7], (6) Zhang et al. using the forehead/eyes region [7], (7) Zhang et al. using both regions [7], (8) Ali et al. [2], (9) Wang and Zhang [25] and (10) Buciu and Pitas [5] are also shown for comparison.

Table 1
Evaluation performance of proposed scheme with and without occlusions.

Method	Feature extraction	Classifier	Occlusion		
			No	Mouth	Eyes
Zhang et al. [7]	Gabor template	SVM	95.5%	94.1%	90.8%
Buciu et al. [5]	Gabor filter	MCC	94.5%	92.3%	87.2%
Proposed	Gabor function	Proposed	98.8%	96.2%	97.2%

that the scheme proposed in [2] was developed and evaluated using only images without occluded regions.

Table 1 shows the recognition performance of the proposed scheme when the mouth region is occluded, when the forehead/eyes region is occluded, and without occlusion. The recognition performance of FER scheme proposed by Zhang et al. [7] and Buciu et al. [5] are also shown for comparison. Evaluation results show that proposed scheme provides better performance that above mentioned schemes when all of them operate under same kind of occlusions.

4. Conclusions

This paper proposes a facial expression recognition algorithm that performs an automatic segmentation of the facial regions of interest. To achieve this goal, it first segments the face image using the Viola-Jones algorithm. Then, using the face symmetry, it extracts the ROIs. To do this, it first represents the color image using the RGB color space, subtracting the green plane from the red plane. Next, using the image moments and horizontal projective integral, it automatically extracts two ROIs: the forehead/eyes and the mouth regions. It is important to mention that this procedure achieves proper ROI extraction even under varying illumination conditions. Illumination was one of the main problems encountered by Zhang et al. [7] when using these regions for facial expression recognition. Next, each extracted ROI is divided into NM blocks. These are characterized using 54 2D Gaussian functions together with the PCA. Finally, the extracted features are fed into a proposed low-complexity classifier that performs better than a multilayer perceptron ANN in terms of both recognition rate and training time. In comparison with other FER systems recently described in the literature, we can conclude that the proposed system performs better than those proposed in [5,8,7,25-27] both because it achieves a higher recognition percentage in all possible cases and because it recognizes facial expressions more accurately. The FER system proposed by Zhang et al. [2] provides slightly better performance than the proposed scheme, but only when the face under analysis is not occluded. The proposed FER provides better performance when only the mouth or the forehead/eyes ROI is available. Thus, it is possible to conclude that proposed system is able to

provide facial recognition rates higher than 97% for either entire faces without occlusions, or for faces that are partially occluded—that is, when only one of the ROIs is considered. It is important to point out that the recognition performance using the YUV color space is quite similar to that obtained when the RGB color space is used.

Acknowledgements

We would like to thank the National Science and Technology Council of Mexico (CONACYT) and to the Instituto Politecnico Nacional for their financial support during the course of this research.

References

[1] Y. Tian, T. Kanade, J.F. Cohn, Facial expressions analysis, in: Stan Z. Li, Anil K. Jain (Eds.), Handbook of Face Recognition, Springer-Verlag, 2004.
[2] H. Ali, M. Hariharan, S. Yaacob, A. Hamid-Adom, Facial recognition using empirical mode decomposition, Expert Syst. Appl. 42 (3) (2015) 1261–1277.
[3] Y. Luo, C. Wu, Y. Zhang, Facial expression recognition based on fusion features of PCA and LPB with SVM, Optik 124 (2013) 2767–2770.
[4] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, Comput. Vision Pattern Recognit. (2001) 511–518.
[5] K. Buciu, I. Pitas, An analysis of facial expression recognition under partial face image occlusion, Image Vision Comput. 26 (7) (2008) 1052–1067.
[6] Y. Miyakoshi, S. Kato, Facial emotion detection considering partial occlusion of face using Bayesian network, in: Proceedings of IEEE Symposium on Computers & Informatics (ISCI), 2011, pp. 96–101.
[7] L. Zhang, D. Tjondronegoro, V. Chadran, Gabor based templates for facial expression recognition in images with facial occlusion, Neurocomputing 145 (5) (2014) 451–464.
[8] Z. Zhang, L. Wang, Q. Zhi, S. Chen, Y. Chen, Pose invariant face recognition using facial landmarks and Weber local descriptor, Knowl. Based Syst. 84 (2015) 78–88.
[9] Y. Lei, H. Han, X. Hao, Discriminant sparse local spline embedding with application to face recognition, Knowl. Based Syst. 89 (2015) 47–55.
[10] Y. Alvarez-Betancurt, M. Garcia-Silvente, A key points-based feature extraction method for iris recognition under variable image quality conditions, Knowl. Based Syst. 92 (2016) 169–182.
[11] I. Jolliffe, Principal Component Analysis, Second ed., Springer, 2002.
[12] G. Benitez-García, G. Sanchez-Perez, H. Perez-Meana, K. Takahashi, M. Kaneko, Facial expressin recognition based on facial región segmentation and modal value approach, IEICE Trans. Inf. Syst. E97 (4) (2014) 928–935.
[13] D. Vukadinovick, M. Pantic, Fully automatic facial point detector using Gabor feature based boosted classifiers, in: Proceedings of the IEEE International Conference on System, Man and Cybernetics, 2005, pp. 1692–1697.
[14] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, M. Nakano-Miyatake, H. Perez-Meana, A sub-block based eigenphases algorithm with optimum sub-block size, Knowl. Based Syst. 37 (1) (2013) 415–426.
[15] A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, A facial expression recognition with automatic segmentation of face regions, in: H. Fujita, G. Guizzi (Eds.), Intelligent Software Methodology, Tools and Techniques, 532, Communications in Computer and Information Science, 2015, pp. 529–530.
[16] W. Li, K. Mao, H. Zhang, T. Chai, Selection of Gabor filters for improved texture feature extraction, in: Proceedings of the IEEE International Conference on Image Processing, 2010, p. 361,364.
[17] M. Pakdel, F. Tajeripour, Texture classification using optimal Gabor filters, in: Proceedings of the International Conference on Computer and Knowledge Engineering, 2011, pp. 208–213.
[18] M. Calvo, Facial expression of emotion (KDFE): identification under different display duration conditions, Behav. Res. Methods 40 (1) (2008) 109–115.
[19] D. Lundqvist, A. Flykt, A. Öhman, The Karolinska directed emotional faces – KDEF (CD-ROM) from department of clinical neuroscience, Psychology Section, Karolinska Institutet, Stockholm, Sweeden, 1998.
[20] S. Hameed Al-azawi, J.H. Al-Ameri, Face features recognition system considering central moments, Int. J. Comput. Eng. Res. 3 (1) (2013) 52–57.
[21] S. Adwan, H. Arof, Modified integral projection method for eye detection using dynamic time warping, Int. J. Innov. Comp. Inf. Control 8 (1) (2012) 187–200.
[22] G. Garcia-Mateos, R. Ruiz-García, O Lopez-de Teruel, Face detection using integral projection models, Lect. Notes Comput. Sci. 2396 (2012) 644–653.
[23] M. Pizer, Adaptive histogram equalization and its variations, computer vision, Graph. Image Process. 39 (1987) 355–368.
[24] A. Hernandez-Matamoros, E. Escamilla-Hernandez, K. Perez-Daniel, K.M. Nakano-Miyatake, H. Perez-Meana, A supervised classifier scheme based on clustering algorithms, in: Proceedings of IEEE Central America and Panama Convention (CONCAPAN XXXIV), 2014, pp. 1–5.
[25] Z. Wang, Q Rao, Facial expression based on orthogonal local Fisher discriminant analysis, in: Proceedings of International Conference on Signal Processing (ICSP), 2010, pp. 1358–1361.
[26] X. Zhao, S. Zhang, Facial expression recognition based on local binary patterns and kernel discriminant isomaps, Sensors (2011) 9573–9588.
[27] W. Freeman, K. Tanaka, J. Ohta, K. Kyuma, Computer vision for computer games, in: International Conference on Automatic Face and Gesture Recognition, 2008, pp. 100–105.

CONGRESOS

A. A. Hernandez-Matamoros, E. Escamilla-Hernandez, K. Perez-Daniel, M. Nakano-Miyatake and H. Perez-Meana, "A supervised classifier scheme based on clustering algorithms," *2014 IEEE Central America and Panama Convention (CONCAPAN XXXIV)*, Panama City, 2014, pp. 1-5.

B. A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, A Facial Expression Recognition with Automatic Segmentation of Face Regions, *SoMeT ,* **CCIS 532** (2015), 529–540

C. Andres Hernandez-Matamoros, Takayuki Nagai, Muhammad Attamimi, Mariko Nakano, Hector Perez-Meana, "Facial Expression Recogntion in Unconstrained Environment" SoMeT 2017, Frontiers in Artificial Intelligence and Applications, Volume 297: New Trends in Intelligent Software Methodologies, Tools and Techniques, 525 – 538

D. Andres Hernandez-Matamoros, Takayuki Nagai, Muhammad Attamimi, Hector Perez-Meana, "Facial Expression Recogntion in the wild" Mini-Conference for Exchange Students on Informatics & Engineering and Information Systems No. 34. Poster presentation.

E. Andres Hernandez-Matamoros, Takayuki Nagai, Muhammad Attamimi, Hector Perez-Meana, "Facial Expression Recogntion in the wild" Mini-Conference for Exchange Students on Informatics & Engineering and Information Systems No. 35. Oral presentation.

A Supervised Classifier Scheme Based on Clustering Algorithms

A. Hernandez-Matamoros, E. Escamilla-Hernandez, K. Perez-Daniel, M. Nakano-Miyatake, and H. Perez-Meana, *Senior Member, IEEE*

***Abstract*— This paper proposes a new classifier scheme based on classical clustering algorithms, such as the *Batchelor & Wilkins* y *K-means* algorithms which are trained in a similar form that the artificial neural network (ANN) or support vector machines (SVM). Proposed scheme has the advantage that if a new class is added, it is not necessary to train he classifier completely, but only add a new class. Experimental results show that the proposed scheme provides classification rates quite similar to those provided by the SVM with much less computational complexity.**

***Index Terms*—Supervised training, self-organizing maps, pattern recognition, support vector machines.**

I. Introducción

En la actualidad el reconocimiento de patrones es un problema que se ha intentado resolver con distintos tipos de algoritmos, desafortunadamente en la mayoría de los casos al agregar una nueva clase se requiere reentrenar todo el sistema [1]-[4], en muchos casos tiene un alto costo computacional. En esta área las Redes Neuronales Artificiales (RNA) han sido aplicadas en múltiples propósitos, su diseño se ha caracterizado por un mecanismo de prueba y error, el cual puede originar un desempeño bajo. Por otro parte, los algoritmos de retro propagación y otros basados en el gradiente descendiente, presentan una desventaja: no pueden resolver problemas no continuos, ni problemas multimodales. De lo anterior en [5] aplican algoritmos evolutivos para diseñar de manera automática una RNA capaz de resolver esos problemas.

Por esta razón en este articulo se propone un nuevo clasificador basado en algoritmos de agrupamientos capaz de resolver los problemas de los clasificadores clásicos manteniendo un alto porcentaje de reconocimiento. La estructura del presente artículo es la siguiente: en la sección 2 se dedica a métodos de agrupamiento y menciona las distancias comúnmente utilizadas; para así en la sección 3 presentar el clasificador propuesto, para en la sección 4 mostrar los resultados obtenidos con 2 sistemas, por último se encuentran las conclusiones.

Manuscrito recibido Junio 25, 2014. Versión actualizada Octubre 18, 2014. Recomendado para publicación por miembros evaluadores del Programa Técnico del CONCAPAN 2014. Este trabajo fue financiado por el Consejo Nacional de Ciencia y Tecnología, CONACYT y por el Instituto Politécnico Nacional, IPN.

A. Hernández-Matamoros está con el Instituto Politécnico Nacional, México D. F. 00430 México. Email andresmatamoros1986@hotmail.com.

E. Escamilla-Hernández está con el Instituto Politécnico Nacional, México D. F. 00430 México. Email eescamillah@hotmail.com.

K. Pérez-Daniel está con el Instituto Politécnico Nacional, México D. F. 00430 México. Email krperezd@hotmail.com.

M. Nakano-Miyatake está con el Instituto Politécnico Nacional, México D. F. 00430 México. Email mnakano@ipn.mx.

H. Pérez-Meana está con el Instituto Politécnico Nacional, México D. F. 00430 México, teléfono +52-55-5656-2058, fax +52+55+5656+2058. Email hmperezm@ipn.mx,

II. Métodos de Agrupamiento

En las dos primeras partes de esta sección se explicaran los algoritmos de *K-means* y el algoritmo de *Batchelor & Wilkins* respectivamente, que en ambos casos evalúan las distancias que existen entre los centroides estimados para cada clase y patrón bajo análisis para llevar a cabo la clasificación de patrones. Finalmente en la tercera partes se menciona la distancia Euclidiana, a cual es utilizada para desarrollar el clasificador propuesto.

A. K-Means

El método de K-means [6] es uno de los más usados en aplicaciones científicas e industriales. El nombre viene porque representa cada uno de los grupos por la media de sus puntos, es decir por su centroide, usando como criterio de optimización una función de error cuadrático. En la tabla I muestra el pseudocódigo de este método.

Para este algoritmo es necesario definir, de antemano, el número de grupos o conjuntos que se desea crear.

B. Clasificador de Batchelor&Wilkins

A diferencia de K-means, este es un método de agrupamiento en el cuál el número de clases es desconocido de antemano, como mayor desventaja que su comportamiento esta sesgado por el orden de la presentación de los patrones. En la Tabla II se muestra el pseudocódigo de este.

TABLE I
K-MEANS CLUSTERING ALGORITHM

1.-Definir el numero de conjuntos.
2.-Asignar aleatoriamente los k conjuntos iníciales.
3.-Calcular los centroides para cada uno de los conjuntos .
4.-Asignar a cada punto el conjunto cuyo centroide se encuentre más cerca.
5.-Regresar al paso 2 hasta que ya no sufran cambios los centroides.
6.-Fin.

978-1-4799-7584-6/14/$31.00 ©2014 IEEE

TABLA II
BATCHELOR AND WILKINS CLASSIFICATION AÑGORITHM [7]

Parámetros	
f	Fracción de la distancia media entre agrupaciones
Algoritmo	
Primer agrupamiento: Patrón escogido al azar	
Segundo agrupamiento: Patrón más alejado del primer agrupamiento	
	Mientras se creen nuevos agrupamientos Obtener el patrón más alejado de los agrupamientos existentes (máximo a las distancias mínimas de los patrones a los agrupamientos)
	Si la distancia del patrón escogido al conjunto de agrupamientos es mayor que una fracción de la distancia media entre los agrupamientos, crear un agrupamiento con el patrón seleccionado
Asignar cada patrón a su agrupamiento más cercano	

Los resultados obtenidos por este algoritmo dependen en gran medida de los parámetros con los cuales se ejecute el algoritmo.

C. *Distancia Euclidiana*

Dentro de la literatura existen gran variedad de distancias para medir la diferencia entre 2 puntos en un espacio de N dimensiones, las más comunes son Manhattan, Promedio, Euclidiana Cuadrada, con base en [8] se tomara la distancia Euclidiana para realizar las pruebas.

$$\sqrt{\sum_{k=1}^{n}(A_k - B_k)^2} \tag{1}$$

III. Clasificador Propuesto

El clasificador propuesto está basado en el aprendizaje supervisado [9]-[11], en el cual es necesario conocer el número de clases a las cuales pertenecen los patrones a clasificar, así como el número de elementos por cada clase usados para el entrenamiento, siendo fundamental una adecuada selección del conjunto de patrones usados para el entrenamiento a fin de que el sistema sea capaz obtener los modelos que sean capaces de representarlos. A este conjunto de patrones de entrenamiento lo nombraremos como **X**, donde las filas pertenecen a los patrones y las columnas al número de elementos que tiene cada patrón. Es importante señalar que todos los patrones deben tener la misma longitud para un correcto funcionamiento del clasificador.

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,n} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ X_{m,1} & X_{m,2} & & X_{m,n} \end{bmatrix} \tag{2}$$

Dentro de **X** existe el número de clases deseadas a clasificar (S_1, S_2, S_3,..... S_i), esto es

$$(S_1, S_2, S_3, \ldots, S_i) \in \mathbf{X}, \tag{3}$$

Donde i<m. Además, supondremos que cada una de cuenta con un número indeterminado de patrones, esto es

$$(P_{i,1}, P_{i,2}, P_{i,3}, \ldots P_{i,j}) \in S_i, \tag{4}$$

De tal modo que los patrones por entrenar están dentro de la matriz **X.**

A. *Entrenamiento*

Para este proceso cada uno de los subconjuntos de S entrará por separado al clasificador propuesto que se muestran en el algoritmo de la Tabla III.

Es importante mencionar que en cada centroide es el promedio de los valores asignados a este. Con el algoritmo de la Tabla III se obtendrá una matriz que llamaremos **C** por cada uno de los $\mathbf{S}_i$ subconjuntos o clases, obteniendo lo siguiente.

$$(S_1 \in C_1, S_2 \in C_2, \ldots, S_i \in C_i) \tag{5}$$

En este punto es donde se obtiene más de un modelo característico para cada una de las clases, terminado el proceso de entrenamiento cuando se tengan las *"i"* matrices consideradas de antemano. Esto es

$$(C_1, C_2, \ldots C_i) \tag{6}$$

Lo siguiente para finalizar el entrenamiento es concatenar todas las matrices C_k, $k=1,2,..,i$, en una matriz E que se define en la ec. (7).

$$E = \left[C_1 | C_2 | C_3 | .. | C_i\right] \tag{7}$$

Para realizar un ejemplo grafico, de cómo es que agrupa el método propuesto comparado con uno de los algoritmo más usados de su tipo, el algoritmo de k-means, se utilizaron 100 patrones de valores aleatorios entre 0 y 1 de dos dimensiones, teniendo estos la siguiente clasificación propuesta, es importante mencionar que el número de elementos entre las clases variaba; teniendo la clase A 25 elementos, la clase B cuenta con 29 elementos, mientras que la clase C tiene 16 elementos y por último la D tiene 30 elementos.

TABLA III
CLASIFICADOR PROPUESTO/FACE DE ENTRENAMIENTO

Inicio
Con el primer patrón de [X] se forma un centroide.
Mientras existan patrones por agrupar
    Obtener la distancia de todos los datos con todos los centroides.
    Para la mayor distancia
        Crear un nuevo centroide.
    Termina
    Para la menor distancia
        Asigna el patrón al centroide correspondiente
        Recalcula el centroide
    Termina
Termina
Fin

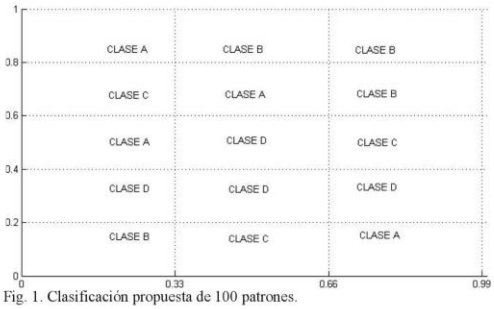


Fig. 1. Clasificación propuesta de 100 patrones.

La Fig. 2 muestra los centroides creados al aplicar el clasificador propuesto, es importante notar que éste es capaz de crear los centroides necesarios para modelar cada una de las subclases de la misma clase, sin importar que se encuentren alejadas entre estas mismas, lo cual no es posible realizar con el clasificador K-means. Esto se debe a que, debido a que cada una de las clases está formada por superficies disjuntas, no es posible agrupar sus elementos por medio de un solo centroide, como se muestra en la Fig. 3.

B. Pruebas

Para la fase de pruebas se tiene un patrón P necesariamente con el mismo número de elementos que los renglones de **X**, la cual se evalúa usando el algoritmo mostrado en la Tabla IV.

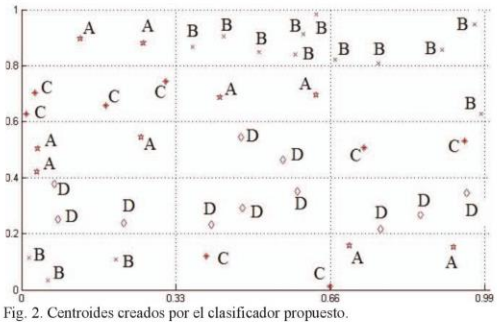


Fig. 2. Centroides creados por el clasificador propuesto.

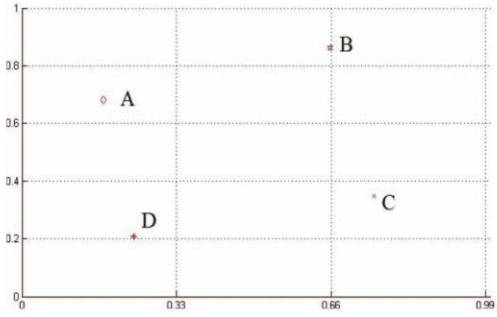


Fig. 3. Centroides creados por el algoritmo de k-means.

TABLA IV
CLASIFICADOR PROPUESTO/ ETAPA DE PRUEBA

Inicio
Obtener la distancia de patrón "P" con los elementos de **E**
Para la menor distancia
Obtener el índice del elemento para buscar a que clase corresponde dicho índice
Termina
Fin

IV. Resultados

Para evaluar el funcionamiento del sistema propuesto, inicialmente se requirió que este clasificara un conjunto de datos bidimensionales, pertenecientes a cuatro clases distribuidas como se muestra en la Fig. 1. Los resultados obtenidos muestran que el clasificador propuesto es capaz de clasificarlos correctamente, como se muestra en la Fig. 4, mientras que en la Fig. 5 se muestra los resultados obtenidos con el K-Means para comparación.

Con el fin de evaluar el funcionamiento del clasificador propuesto en aplicaciones reales, se reprodujeron experimentos en un laboratorio y con un mismo ordenador para base de datos previamente estudiadas con otros clasificadores que se encuentran en la literatura. La tabla V muestra los resultados obtenidos cuando el clasificador propuesto es requerido a identificar sonidos ambientales [13], coeficientes cepstrales en frecuencia mel (MFCC) y matching pursuit (MP) usando características individuales.

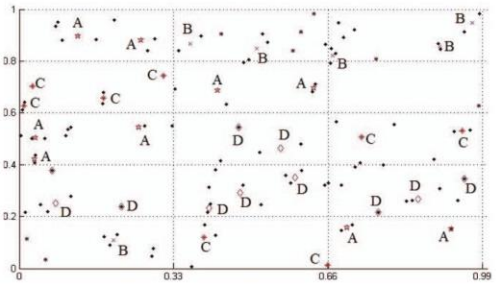


Fig. 4. Resultados obtenidos con el clasificador propuesto

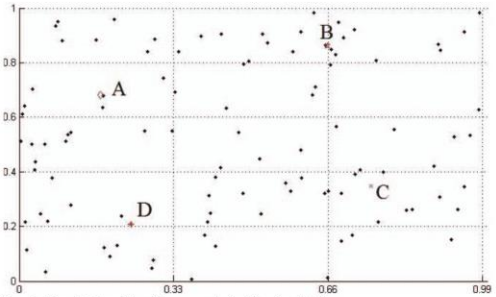


Fig. 5. Resultados obtenidos con el clasificador k-Means.

Los resultados obtenidos usando maquinas de soporte vectorial (SVM), redes neuronales artificiales (ANN), algoritmo de K-means y regresión logística (RL) se muestran para su comparación. Por su parte la Tabla VI muestra los resultados obtenidos para los nueve sonidos de la base de datos caracterizados como se mencionó anteriormente. La base de datos consiste de 9 clases de sonidos diferentes, teniéndose 300 sonidos de cada una de las clases. Todos los clasificadores se entrenaron con el mismo número de sonidos, mostrando los resultados obtenidos para su comparación. Los resultados experimentales muestran que el porcentaje de reconocimiento obtenido es muy similar al proporcionado por una SVM, poco menos del 1%, aunque con un costo computacional mucho menor costo.

Otra aplicación que requiere el empleo de clasificadores eficientes y en lo posible de bajo costo computacional, es el reconocimiento de rostros, el cual ha sido ampliamente usado en controles de acceso, sistemas de seguridad biométrica, etc. Con el fin de llevar a cabo la evaluación del clasificador en esta aplicación se empleó una base de datos de rostros con visión estéreo con diferentes niveles de fusión [14] para su reconocimiento, siendo posible crear un sistema de 2D si solo toma la imagen derecha o la imagen izquierda, se reemplazo el SVM por el sistema propuesto. Aquí para caracterizar cada una de las imágenes bajo análisis se emplean dos diferentes técnicas de extracción; las funciones de Gabor y el método de componentes principales (PCA). Así una vez obtenidos los vectores característicos, estos se procesan usando ya sea la SVM o el clasificador propuesto para tomar la decisión final. Los resultados experimentales proporcionados en la tabla VIII muestran que el clasificador propuesto proporciona un porcentaje de reconocimiento muy similar al proporcionado por la SVM, con una menor complejidad computacional, lo cual se ve reflejado en el tiempo de entrenamiento como muestra la tabla VII. En este caso la base de datos consiste de 50 personas diferentes, teniéndose 15 imágenes de cada persona. Tanto la SVM como el clasificador propuesto se entrenaron con el mismo número de personas.

TABLA V
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO Y DE OTROS CLASIFICADORES.

Clasificador	SVM	ANN	LR	Propuesto	K-means
Reconocimiento (%)	98.4	95	74.3	97.5	3.92

TABLA VI
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO, DEL K-MEANS Y DEL SVM CUANDO SE EMPLEAN PARA RECONOCER SONIDOS AMBIENTALES

	SVM	Propuesto	K-means
Gato	96.6%	96.5%	0%
Perro	99.8%	98.1%	3.6%
Risa	96.1%	95:2%	16.3%
Sirena	99.5%	99.9%	0%
Rotura de cristal	97.3%	94.6%	0.3%
Llanto	98.3%	99.2%	6.3%
Explosión	99.8%	99.9%	0%
Gritos	99.1%	99.1%	0%
Voz	97.2%	96.1%	8.6%

TABLA VII.
COMPARACIÓN DEL TIEMPO DE ENTRENAMIENTO DEL SISTEMA PROPUESTO Y DEL SVM CON DIFERENTES NIVELES DE FUSIION

Tiempo de entrenamiento	SVM	Propuesto
Fusión a nivel sensor	13.09s	8.37s
Fusión a nivel concatenación	13.12s	11.5s
Fusión a nivel decisión	5.97s	4.18s

TABLA VIII
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO Y DEL SVM EMPLEADOS PARA RECONOCER ROSTROS CON DIFERENTES NIVELES DE FUSIIÓN.

Niveles de Fusión	Reconocimiento (%) Usando funciones de Gabor		Reconocimiento (%) Usando PCA	
	SVM	Propuesto	SVM	Propuesto
Fusión a nivel sensor	95	99.3	96.2	98.3
Fusión a nivel concatenación	94.6	99.3	91.7	98.3
Fusión a nivel decisión	93.7	99.3	91.5	98

V. Conclusión

Podemos concluir que el método propuesto a diferencia de otro método de agrupamiento (k-means) muy usado en muchas aplicaciones, tiene una diferencia porcentual de hasta el 99.9%. Por otro lado el método propuesto es capaz de competir con clasificadores clásicos, ya que como se demostró en las tablas VI y VIII, tiene porcentajes de reconocimiento similares a estos teniendo la ventaja que el tiempo de entrenamiento es menor, lo cual se muestra en la tabla VII; así como, si se desea agregar una nueva clase al sistema solo es necesario entrenar esta y agregarla a la matriz E, también se demostró que es capaz de clasificar de manera correcta, distintos tipos de patrones cada uno con diferente numero de clases. Entre los patrones usados se tienen tanto sonidos como imágenes de rostros, incluyendo rostros tomados con una cámara de visión estéreo. lo cual hace capaz al sistema de poder reconocer rostros en 2D Y3D, en 3D es capaz de variar en muy poco el porcentaje de reconocimiento obtenido a diferentes niveles de fusión algo que no sucede con los métodos clásicos, es importante hacer notar que estos últimos, patrones no tiene la misma longitud.

Referencias

[1] S. Sergey, "Programming the algorithm learning neural network back propagation", Perspective Technologies and Methods in MEMS Design, MEMSTECH, pp. 77 – 78, 2009..

[2] Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery 2, pp. 121-167, 1998.

Los resultados obtenidos usando maquinas de soporte vectorial (SVM), redes neuronales artificiales (ANN), algoritmo de K-means y regresión logística (RL) se muestran para su comparación. Por su parte la Tabla VI muestra los resultados obtenidos para los nueve sonidos de la base de datos caracterizados como se mencionó anteriormente. La base de datos consiste de 9 clases de sonidos diferentes, teniéndose 300 sonidos de cada una de las clases. Todos los clasificadores se entrenaron con el mismo número de sonidos, mostrando los resultados obtenidos para su comparación. Los resultados experimentales muestran que el porcentaje de reconocimiento obtenido es muy similar al proporcionado por una SVM, poco menos del 1%, aunque con un costo computacional mucho menor costo.

Otra aplicación que requiere el empleo de clasificadores eficientes y en lo posible de bajo costo computacional, es el reconocimiento de rostros, el cual ha sido ampliamente usado en controles de acceso, sistemas de seguridad biométrica, etc. Con el fin de llevar a cabo la evaluación del clasificador en esta aplicación se empleó una base de datos de rostros con visión estéreo con diferentes niveles de fusión [14] para su reconocimiento, siendo posible crear un sistema de 2D si solo toma la imagen derecha o la imagen izquierda, se reemplazo el SVM por el sistema propuesto. Aquí para caracterizar cada una de las imágenes bajo análisis se emplean dos diferentes técnicas de extracción; las funciones de Gabor y el método de componentes principales (PCA). Así una vez obtenidos los vectores característicos, estos se procesan usando ya sea la SVM o el clasificador propuesto para tomar la decisión final. Los resultados experimentales proporcionados en la tabla VIII muestran que el clasificador propuesto proporciona un porcentaje de reconocimiento muy similar al proporcionado por la SVM, con una menor complejidad computacional, lo cual se ve reflejado en el tiempo de entrenamiento como muestra la tabla VII. En este caso la base de datos consiste de 50 personas diferentes, teniéndose 15 imágenes de cada persona. Tanto la SVM como el clasificador propuesto se entrenaron con el mismo número de personas.

TABLA V
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO Y DE OTROS CLASIFICADORES.

Clasificador	SVM	ANN	LR	Propuesto	K-means
Reconocimiento (%)	98.4	95	74.3	97.5	3.92

TABLA VI
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO, DEL K-MEANS Y DEL SVM CUANDO SE EMPLEAN PARA RECONOCER SONIDOS AMBIENTALES

	SVM	Propuesto	K-means
Gato	96.6%	96.5%	0%
Perro	99.8%	98.1%	3.6%
Risa	96.1%	95:2%	16.3%
Sirena	99.5%	99.9%	0%
Rotura de cristal	97.3%	94.6%	0.3%
Llanto	98.3%	99.2%	6.3%
Explosión	99.8%	99.9%	0%
Gritos	99.1%	99.1%	0%
Voz	97.2%	96.1%	8.6%

TABLA VII.
COMPARACIÓN DEL TIEMPO DE ENTRENAMIENTO DEL SISTEMA PROPUESTO Y DEL SVM CON DIFERENTES NIVELES DE FUSIION

Tiempo de entrenamiento	SVM	Propuesto
Fusión a nivel sensor	13.09s	8.37s
Fusión a nivel concatenación	13.12s	11.5s
Fusión a nivel decisión	5.97s	4.18s

TABLA VIII
COMPARACIÓN DEL FUNCIONAMIENTO DEL SISTEMA PROPUESTO Y DEL SVM EMPLEADOS PARA RECONOCER ROSTROS CON DIFERENTES NIVELES DE FUSIIÓN.

Niveles de Fusión	Reconocimiento (%) Usando funciones de Gabor		Reconocimiento (%) Usando PCA	
	SVM	Propuesto	SVM	Propuesto
Fusión a nivel sensor	95	99.3	96.2	98.3
Fusión a nivel concatenación	94.6	99.3	91.7	98.3
Fusión a nivel decisión	93.7	99.3	91.5	98

V. Conclusión

Podemos concluir que el método propuesto a diferencia de otro método de agrupamiento (k-means) muy usado en muchas aplicaciones, tiene una diferencia porcentual de hasta el 99.9%. Por otro lado el método propuesto es capaz de competir con clasificadores clásicos, ya que como se demostró en las tablas VI y VIII, tiene porcentajes de reconocimiento similares a estos teniendo la ventaja que el tiempo de entrenamiento es menor, lo cual se muestra en la tabla VII; así como, si se desea agregar una nueva clase al sistema solo es necesario entrenar esta y agregarla a la matriz E, también se demostró que es capaz de clasificar de manera correcta, distintos tipos de patrones cada uno con diferente numero de clases. Entre los patrones usados se tienen tanto sonidos como imágenes de rostros, incluyendo rostros tomados con una cámara de visión estéreo. lo cual hace capaz al sistema de poder reconocer rostros en 2D Y3D, en 3D es capaz de variar en muy poco el porcentaje de reconocimiento obtenido a diferentes niveles de fusión algo que no sucede con los métodos clásicos, es importante hacer notar que estos últimos, patrones no tiene la misma longitud.

Referencias

[1] S. Sergey, "Programming the algorithm learning neural network back propagation", Perspective Technologies and Methods in MEMS Design, MEMSTECH, pp. 77 – 78, 2009..

[2] Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery 2, pp. 121-167, 1998.

[3] P. Ponce Cruz, "Inteligencia Artificial con aplicaciones a la ingeniería", 1er edición, Alfaomega Grupo Editor, Mexico, Julio 2010.
[4] C. M. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006.
[5] GARRO, Beatriz A.; SOSSA, Humberto; VAZQUEZ, Roberto A.. Diseño Automático de Redes Neuronales Artificiales mediante el uso del Algoritmo de Evolución Diferencial (ED). Polibits, México, n. 46, dic. 2012.
[6] Kanungo, Tapas; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y., "An efficient k-means clustering algorithm: analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.24, no.7, pp.881,892, Jul 2002.
[7] Batchelor, B.G., "Method for location of clusters of patterns to initialise a learning machine," Electronics Letters , vol.5, no.20, pp.481,483, October 2 1969.
[8] Hernández Matamoros A.G., Pérez Daniel K.R, "Aprendizaje de un Objeto mediante imágenes obtenidas de internet usando aprendizaje no supervisado" Research in Computing Science, Tendencias Tecnológicas en Computación, vol. 66, pp 39-48, Octubre 2013.
[9] Russell, S. and Norvig, P. Inteligencia Artificial. Un enfoque moderno. Prentice Hall.1996.
[10] J.R. Hilera y V.J. Martínez , Redes Neuronales Artificiales "Fundamentos, modelos y aplicaciones". Alfaomega ,2000.
[11] Gallardo Campos Margarita. Aplicación de técnicas de Clustering para la mejora del aprendizaje. Universidad Carlos III de Madrid . 2009.
[12] M. Cowling, R. Sitte, "Comparison of techniques for environmental sound recognition", Pattern Recognition Letters, Vol. 24, No. 15, pp. 2895-2907, noviembre 2003.
[13] C. A. Ruiz-Martinez, M. T. Akhtar, and E. Escamilla-Hernandez, "A study on environmental sound recognition using mel frequency cepstral coefficients and support vector machines," in Proc. The 27th Signal Processing (SIP) Symposium, November 28-30, 2012, Japan.
[14] E. García-Rios, E. Escamilla-Hernandez, G. Aguilar-Torres, O. Jacobo-Sanchez, M. Nakano-Miyatake, H. Perez-Meana, "Multi-biometric Face Recognition System Using Levels of Fusion", Int. Journal of Computers, vol. 7, no. 3, pp. 99-108, 2013.

Andrés Hernández-Matamoros recibió el título de Ingeniero en Electrónica de la Universidad Autónoma Metropolitana de la Ciudad de México en 2011, el grado de Maestría en Microelectrónica de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional en diciembre de 2013. De enero a junio de 2013 realizó una estancia de investigación en la Universidad de Milán Italia. Actualmente es un estudiante de doctorado en la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional. Sus intereses en investigación están en los campos de reconocimiento de patrones y procesamiento de imágenes

Enrique Escamilla Hernández recibió el título de Ingeniero en Electrónica de la Universidad Autónoma Metropolitana del al Ciudad de México en 1997. Posteriormente en 2002 y 2006 recibió los grados de Maestría en Microelectrónica y Doctorado en Comunicaciones y Electrónica respectivamente, de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional. De 1997 a 1998 estuvo con el Departamento de Capacitación de Teléfonos de México. De septiembre de 1999 a julio de 2007 fue profesor asociado en el Departamento de Ingeniería Eléctrica de la Universidad Autónoma Metropolitana Unidad Iztapalapa. En 2006 se unió al Centro de Investigación en Tecnologías Informáticas y Sistemas de la Universidad Autónoma de Hidalgo, México, donde estuvo hasta 2008 cuando se unió a la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacan, donde actualmente es profesor titular. De 2002 a 2003 realizó una estancia de investigación en la Universidad de Electro-Comunicaciones de Tokio, Japón. Su interés en investigación está en los campos de reconocimiento de patrones, procesamiento de imágenes y diseño electrónico.

Karina Pérez-Daniel recibió el grado de Ingeniero en Electrónica de la Universidad Autónoma de Hidalgo en 2007. En diciembre de 2010 recibió el grado de Maestría en Microelectrónica de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional, donde actualmente es una estudiante de doctorado. De octubre de 2009 a agosto de 2009 realizó una estancia de investigación en la Universidad de Electro-Comunicaciones de Tokio. De marzo a diciembre de 2013 realizó uan estancia de investigación en al Universidad e Burdeos, Francia y de marzo a julio de 2014 realizó un intership en Microsoft en Seattle, Estados Unidos. Su interés en investigación está en los campos de procesamiento de imágenes, indexado de información digital.

Mariko Nakano-Miyatake recibió el grado de Maestra en Ciencias en Ingeniería Eléctrica de The University of Electro-Communications, Tokio Japón en 1985, y su Doctorado en Ciencias en Ingeniería Eléctrica de la Universidad Autónoma Metropolitana (UAM), Ciudad de México en 1998. Desde julio 1992 a febrero 1997 fue profesora del Departamento de Ingeniería Eléctrica de la UAM México. En febrero 1997, se integró a la Sección de Postgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional. Sus áreas de investigación son sistemas adaptables, redes neuronales y marca de agua. Ella es miembro del IEEE, RISP y del Sistema Nacional de Investigadores de México.

Héctor Pérez-Meana recibió el grado de M.S. de The University of Electro-Communications, Tokio Japón, el grado de Doctor en Ingeniería Eléctrica de The TokyoInstitute of Technology, Tokio, Japón, en 1989. En 1981 se integró como profesor titular en el Departamento de Ingeniería Eléctrica de la Universidad Autónoma Metropolitana. Desde marzo 1989 a Septiembre 1991, fue investigador visitante de Fujitsu Laboratories Ltd, Kawasaki, Japón. En febrero 1997, se integró como profesor de la Sección de Posgrado e Investigación de Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional (IPN). En 1991 recibió el Premio al Mejor Artículo de IEICE y en 1999 y 2000 el Premio de Investigación del IPN. En 1998 fue presidente del comité organizador de the ISITA'98. Sus principales áreas de investigación son filtros adaptables, procesamiento de imágenes, reconocimiento de patrones. El Dr. Pérez-Meana es Senior member del IEEE, IEICE, Sistema Nacional de Investigadores de México y de la Academia Mexicana de Ciencias.

A Facial Expression Recognition with Automatic Segmentation of Face Regions

Andres Hernandez-Matamoros[1], Andrea Bonarini[2], Enrique Escamilla-Hernandez[1], Mariko Nakano-Miyatake[1], and Hector Perez-Meana[1(✉)]

[1] Instituto Politecnico Nacional, Av. Santa Ana 1000, Mexico D.F. 04430, Mexico
{mnakano, hmperezm}@ipn.mx
[2] Politecnico Di Milano, Via Ponzio 34/5, 20133 Milan, Italy
http://www.posgrados.esimecu.ipn.mx

Abstract. This paper proposes a facial expression recognition algorithm, which automatically detects and segments the face regions of interest (ROI) such as the forehead, eyes and mouth, etc. Proposed scheme initially detects the image face and segments it in two regions: forehead/eyes and mouth. Next each of these regions is segmented into N × M blocks which are characterized using 54 Gabor functions that are correlated with each one of the N × M blocks. Next the principal component analysis (PCA) is used for dimensionality reduction. Finally, the resulting feature vectors are inserted in a proposed classifier based on clustering techniques which provides recognition results closed to those provided by the support vector machine (SVM) with much less computational complexity. The experimental results show that proposed system provides a recognition rate of about 98 % when only one ROI is used. This recognition rate increases to about 99 % when the feature vectors of all ROIs are concatenated. This fact allows achieving recognition rates higher than 97 %, even when one of the two ROI are totally occluded.

Keywords: Facial expression recognition · Gabor functions · PCA · Classifier methods · Face detection · Facial ROI segmentation

1 Introduction

The use of smart devices in the solution of several problems has increased recently given as a result the development of very efficient systems that can be used in many practical applications. Among them the facial expression recognition (FER) systems have been used to recognize the mood. This is because several problems can be avoided if it is possible to accurate detect the mood of a person, i.e., if a given person has a nervous breakdown, if he is tired, angry or happiness, etc. For this reason, during the last several years the interest for developing these kinds of systems has increased [1–4]. A very important part of such systems is the detection of face regions because an accurate detection of such regions may improve the performance of FER. Currently, in the literature exists some algorithms able to detect faces in an image and even smiles

H. Fujita and G. Guizzi (Eds.): SoMeT 2015, CCIS 532, pp. 529–540, 2015.
DOI: 10.1007/978-3-319-22689-7_41

must of them based on Viola-Jones [5] algorithm. Unfortunately these schemes are not enough accurate to detect the facial expression and thereby to achieve accurate mood detection. This happens because when someone does a mood a expression, it could be strong or not, some movement of the face muscles is done involuntarily. This movement is, in general, different in each facial expression doing it possible to determine the regions of interest of the face in each case. Several problems are present in facial expression recognition; some of them are related with the face orientation related to the camera, because if the person isn't looking straightforward to the camera partial occlusion of the face may occur [6], or the presence of shadows due to poor illumination conditions. Then it is necessary that the FER system be able to achieve a recognition rate higher than 95 % even if only one of the available regions of interest remains without occlusion. For this reason, we propose a FER algorithm that is able to segment the face ROI under different illumination conditions which does it possible to take accurate decisions even if one of the two ROI of the face has a partial or even total occlusion. In proposed system, after the ROI estimation, each region is segmented in a set of N × M blocks which are correlated with a set of Gabor functions. The resulting factures matrix is then applied to a PCA for dimensionality reduction. We also propose a classifier with low computational cost which provides recognition rates similar to those provided by other high performance classifiers such as the SVM and ANN. This fact allows implementing the proposed scheme even in smart devices with low computation power that require an immediate response. The proposed algorithm was evaluated with kdef data base [7] which consists of 490 images which are divided into seven facial expressions of 70 people which are used to carry out different evaluations. In the first one the FER system was evaluated using only one ROI, assuming that the other one was occluded, while in the second one both ROI were concatenated. Evaluation results show that using one ROI the proposed system provides a recognition rate of 98 %, while using both ROI the recognition rate increase to 99 %. The rest of the paper is organized as follows: Sect. 2 describes the system framework, the experimental results are shown in Sects. 3 and 4 provides the conclusion of this work.

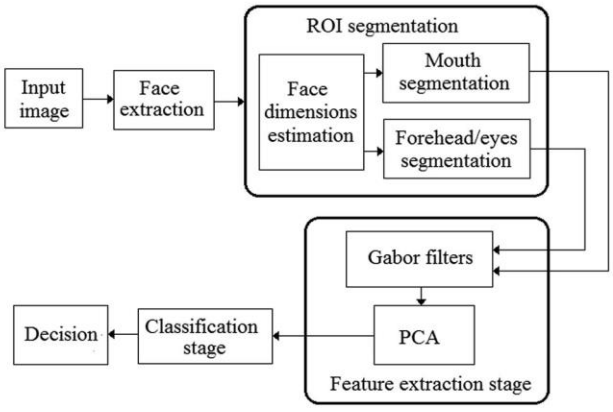


Fig. 1. Proposed facial expression recognition system

2 Proposed System

The System framework of proposed facial expression recognition system (FER) is shown in Fig. 1. Here, firstly the received face image is fed into the face extraction stage, which extracts the face image using the Viola-Jones algorithm [5]. Next the detected face region is inserted into the region of interest (ROI) detection stage, which firstly estimates the face dimensions. Then using this information the ROIs are automatically segmented to get the mouth and Forehead/Eyes using the image moments and projective integrals. These ROIs are then segmented in N × M no overlapping blocks which are then cross-correlated, as mentioned above, with a set of 54 Gabor functions. Next the first value of each estimated cross correlation mentioned above is used for the feature extraction of each ROI, which are then concatenated in 3 different vectors: mouth, forehead/eyes, mouth + front/eyes. Next the feature vectors are independently process by a PCA stage for dimensionality reduction. Finally, the resulting vectors are fed in to the proposed a classifier stage to take the final decision. Next sections provide a description of all stages of proposed system.

2.1 ROI Segmentation

The image received by the FER is firstly processed by the Viola Jones algorithm to segment the face from the background. However, the detected face image may contain noise, such as the hair or ears, which does not contain relevant information for the facial expressions; or the background where the photograph was taken. In order to eliminate this problem that may decrease the recognition rate of the proposed face expression recognition system, a more accurate estimation for face dimension is carried out as shown in Fig. 3.

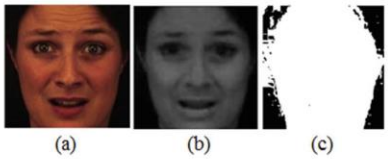


Fig. 2. (a) Original image, (b) Image obtained by the subtraction, (c) binarized image.

Adjustment of face dimension. To adjust the face dimension parameters, firstly the color face image is divided into its three color components: Red, Green and Blue channels. Next the red and green channels are subtracted from them to highlight the skin as shown in Fig. 2b. Finally, the resulting image is binarized using the equation

$$I(x,y) = \begin{cases} 0 & I(x,y) < 1 \\ 255 & I(x,y) \geq 1 \end{cases} \quad (1)$$

After the image binarization, shown in Fig. 2c, the moments of the resulting image are estimated as follows [8]

$$M_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^p y^q I(x, y) \tag{2}$$

where $I(x,y)$ is the image intensity at position (x,y), N is the number of columns and M is the number of rows in the image; while p and q define the moment of the image. Next using the Eq. (2) the centriod can be estimated as follows:

$$x_c = {}^{M_{1,0}}\!/_{M_{0,0}} \tag{3}$$

$$M = {}^{M_{0,1}}\!/_{M_{0,0}} \tag{4}$$

Next, using (3) and (4) the following variables are defined

$$a = \frac{M_{2,0}}{M_{0,0}} - x_c^2 \tag{5}$$

$$b = 2\left(\frac{M_{1,1}}{M_{0,0}} - x_c y_c\right) \tag{6}$$

$$c = \frac{M_{0,2}}{M_{0,0}} - y_c^2 \tag{7}$$

Next using (5)–(7) the face image width can be estimated as follows:

$$W = 2\sqrt{\frac{(a+b) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{8}$$

Using W, the left, x_l, and right, x_r, edges of the face image can be estimated as

$$x_l = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil \tag{9}$$

$$x_r = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil + \lceil W \rceil \tag{10}$$

Next, using W the upper edge of the face image can be estimates as follows

$$y_u = \lceil y_c \rceil - 0.84 \left\lceil \frac{W}{2} \right\rceil \tag{11}$$

From (9)–(11) the face image can be segmented as follows

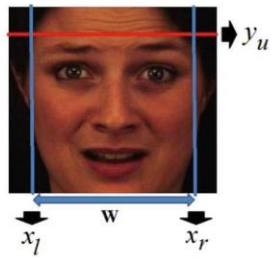


Fig. 3. Segmented face region

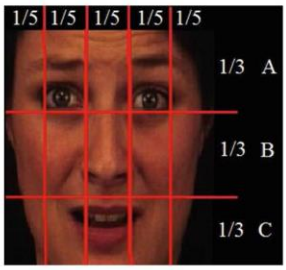


Fig. 4. Symmetrical relationship of the face

Forehead/eye segmentation. A very important part of proposed face expression recognition system is the forehead/face segmentation. To this end, the segmented face region is divided into three regions from the top (A, B and C) as shown in Fig. 4, where the region A is the ROI in the case of the Forehead/eyes segmentation task.

Mouth segmentation. To perform the segmentation of the mouth region, consider the segmented face region which is divided into three regions, of same high, and take the C region of Fig. 4 as our ROI, but unlike the Forehead/eyes region, in this case it is necessary to segment only the mouth region. To this end, the Red and Green image's channels are subtracted among them, then a histogram equalization was performed [9] of image obtained above, obtaining an image as shown in Fig. 5.

Fig. 5. Equalized version of the image obtained from the subtraction of red and green planes.

The next step for the automatic segmentation of the mouth region is the estimation of the horizontal projective integral [10] which is the average of the pixel values of each column. This is a vector containing the average value of the pixels in each column of the image inside the ROI. Figure 6 shows the horizontal projective integral estimated using the equalized image shown in Fig. 5.

Next we obtain the maximum value of the projective error, which will be denoted as "D". Then using the value "D" the left border of the ROI containing the mouth is estimating by subtracting D from x_c, i.e. while the right border is obtained adding it to x_c, keeping the original image height, as shown in Fig. 7, with this procedure the region of interest is extracted automatically.

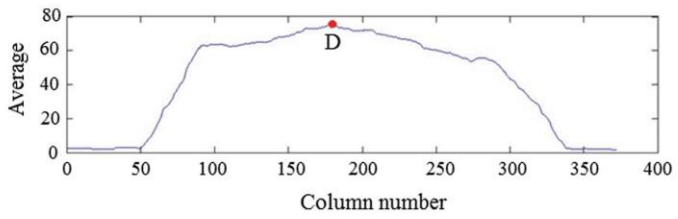


Fig. 6. Horizontal projective integral of the mouth ROI

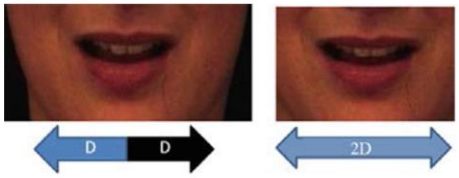


Fig. 7. Detection of mouth ROI.

2.2 Feature Extraction

To perform the feature extraction, each one of the detected ROIs is divided in N × M blocks which are characterized by the average of the first term of the cross correlations between such block and 54 Gabor functions. Next the resulting features vector of each training ROI, with N × M elements, are arranged in a matrix form and applied to a PCA stage for dimensionality reduction. Next sections provide a brief description of these stages.

Gabor functions. The Gabor functions are widely used in many image processing applications such as texture analysis and face recognition tasks [11], because they are robust to luminescence changes. These functions have frequency responses with specific orientations, frequency-selective properties and joint optimum resolution in both spatial and frequency domains. The 2D Gabor functions are given by

$$h(x, y, i, k) = g(x'y') \exp(j2\pi F_i x') \tag{12}$$

where the parameters (x, y) expressed its location in the spatial domain, $F_i = \pi/2^{(i+1)}$, $i = 1,2,..,N_F$ is the spatial frequency, $\phi_k = k\pi/N_\phi$, $k = 1,2,...,N_\phi$ is the rotation angle and $g(x',y')$ is the 2D Gaussian function given by

$$g(x', y') = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \tag{13}$$

where $\sigma = N/2$ and N is the number of blocks in the x axis.

$$(x', y') = x \cos\phi_k + y \sin\phi_k - x \sin\phi_k + y \cos\phi_k. \tag{14}$$

Thus using the Gabor functions given by (12)–(14), the (n,m)-th block of the ROI can be characterized

$$W_{mn} = \frac{1}{N_F N\phi} \sum_{i=0}^{N_F} \sum_{k=0}^{N\phi} W(m, n, i, k) \tag{15}$$

Where

$$W(m, n, i, k) = \left| \sum_{x=0}^{R-1} \sum_{y=0}^{Q-1} f(Rn + x, Qn + y) h(x, y, i, k) \right| \tag{16}$$

Principal component analysis. The principal component analysis is one of the most widely used dimensionality reduction methods [12]. To this end, firstly, the feature vectors given by (15) are represented as a one-dimensional vector $\mathbf{W} = [w_0, w_1, .., w_r, w_{NM}]^T$, where $r = nM + m$, $m = 0,1,...,M-1$, $n = 0,1,...,N-1$, where $M \times N$ is the ROI size. Next a matrix $\mathbf{G}$ of size $(NM) \times Q$ is constructed concatenating the all vectors $\mathbf{W}$ contained in the training set, where (NM) is the dimensionality of $\mathbf{W}$, and $Q = TS < NM$, where T is the number of classes and S the number of training images for each class. Next, the eigenvectors and eigenvalues of the covariance matrix given by $\mathbf{G}^T\mathbf{G}$ are estimated and used to generate a dominant feature matrix $\mathbf{\Phi}$ of size $L \times NM$, where $L = Q$ corresponds to the number of the most representative eigenvectors contained in the original image [12]. Finally, the feature vector of each image is given by

$$\mathbf{Y} = \mathbf{\Phi W} \tag{17}$$

where $\boldsymbol{Y}$ is the resulting feature vector of size $L \times 1$, Φ is the dominant matrix, and $\mathbf{W}$ is the vector containing the characteristics of the ROI under analysis given by (15).

2.3 Classification Stage

A low computational complexity classification method is proposed which uses a supervised training approach, like the ANN or SVM approaches, with the characteristic that if a new class must be added, it is not necessary to train the system with all patterns again but only with the patterns belonging to the new class [14].

Training. To develop the proposed classifier consider the set of training patterns

$$\mathbf{Y} = \begin{bmatrix} y_{1,1} & y_{1,2} & \ldots. & y_{1,B} \\ y_{2,1} & y_{2,2} & \ldots. & y_{2,B} \\ \ldots. & \ldots. & \ldots. & \ldots. \\ y_{L,1} & y_{L,2} & \ldots. & y_{L,B} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_0^T \\ \mathbf{Y}_1^T \\ . \\ \mathbf{Y}_L^T \end{bmatrix}, \tag{18}$$

and the centroid matrix that is initialized as follows

$$\mathbf{C} = \begin{bmatrix} y_{1,1} & y_{1,2} & \ldots. & y_{1,B} \\ c_{2,1} & c_{2,2} & \ldots. & c_{2,B} \\ \ldots. & \ldots. & \ldots. & \ldots. \\ c_{P,1} & c_{P,2} & \ldots. & c_{P,B} \end{bmatrix} = \begin{bmatrix} y_{1,1} & y_{1,2} & \ldots. & y_{1,B} \\ 0 & 0 & \ldots. & 0 \\ 0 & 0 & \ldots. & 0 \\ 0 & 0 & \ldots. & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1^T \\ \mathbf{C}_2^T \\ . \\ \mathbf{C}_P^T \end{bmatrix}, \tag{19}$$

where L is equal to the total number training patterns, and the initial centroids number is et equal to one. Next, the remaining $P-1$ centroids are estimated as follows:

i. While L > 0, compute

$$\delta_{q,r} = \left|\mathbf{Y}_q - \mathbf{C}_{r_k}\right| \quad q = 1, 2, \ldots, L; \quad , r_k = N_{k-1} + j_k; \quad 1 \le j_k \le (N_k - N_{k-1}) \tag{20}$$

$$\delta_{\min} = \min\left(\delta_{q,r_k}\right) = \delta_{m,r_n} \tag{21}$$

$$\delta_{\max} = \max\left(\delta_{q,r_k}\right) = \delta_{M,r_N} \tag{22}$$

$$\mathbf{C}_{r_N+1} = \mathbf{Y}_M, \tag{23}$$

$$\text{Next for } \mathbf{Y}_\mathrm{m}, \text{ update } \mathbf{C}_{r_n} = (\mathbf{C}_{r_n} + \mathbf{Y}_\mathrm{m})/2 \tag{24}$$

set $L = L - 1$, $P = P + 1$ and $N_j = N_j + 1$.

Finally the resulting clusters are arranged into N_p classes with M_k clusters per class as follows

$$\mathbf{G} = \left[\mathbf{C}_1^T, \ldots, \mathbf{C}_{N_1+1}^T, \mathbf{C}_{N_1+2}^T, \ldots, \mathbf{C}_{N_2}^T, \mathbf{C}_{N_2+1}^T, \ldots, \mathbf{C}_{N_3}^T, \ldots, \mathbf{C}_{N_{p-1}}^T, \ldots, \mathbf{C}_{N_p}^T\right]^T \quad (25)$$

Testing. For a given input pattern **Z**, compute

$$\delta_{\min} = \left|\mathbf{Z} - \mathbf{C}_{N_{k-1}+j_k}\right|_{\min} \quad (26)$$

Then, if $N_{k-1} < j_k \leq N_k$, where $1 < N_k \leq N_p$, the pattern under analysis belongs to the N_k-*th* class.

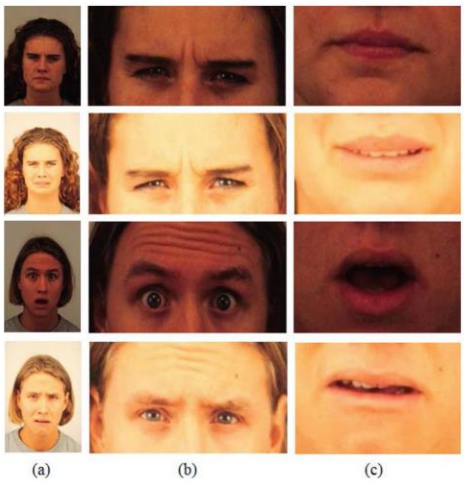


Fig. 8. Performance of face region segmentation stage, (a) original images, (b) eyes/forehead, segmentation, (c) and (f) mouth segmentation.

3 Evaluation Results

An important task of proposed algorithm is the face regions segmentation in order to allow an accurate feature extraction. Figure 8 shows the face region segmentation performance of proposed scheme with different illumination conditions. Evaluation results show that the illumination conditions do not affect the automatic extraction of facial regions.

To carry out the recognition performance of proposed algorithm, after the face region detection, the mouth is reduced to 180 pixels wide and 90 high, while the Front/eyes regions were resized to 300 pixels wide and 150 high. Starting with the goal of find the optimal windows sizes of Gabor filters, different window sizes were analyzed in both face regions, using 50 images for training and 20 testing. Figure 9 shows that for the mouth region a recognition rate of about 97.85 % was obtained when Gabor

windows whose sizes where from 10 × 10 to 30 × 30; while for Front/Eyes regions the recognition's rates is 99.28 %. Thus a suitable window size is 30 × 30 pixels because the computational cost is lower. The same experiment was performed using the ANN as classifier, given as results for the mouth 97.14 % recognition, training with 50 images and using windows for Gabor filters 30 × 30, while for the Forehead/Eyes region was obtained a recognition rate of 91.42 % with 50 training images and windows size of 50 × 50. Both the recognition's percentage is lower than that obtained using the proposed classifier.

Mouth	97.85	97.85	97.85	90.7	88.57	52.14	19.28	19.28
Forehead/Eyes	99.28	99.28	99.28	99.28	99.28	95	92.14	47.85

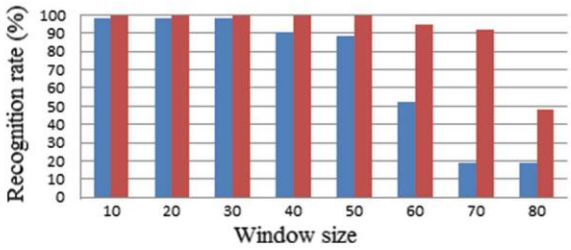
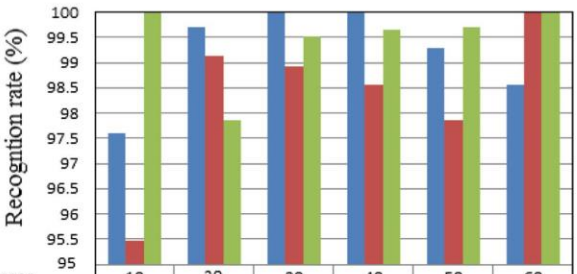


Fig. 9. Average Recognition with different size of windows



Training patterns	10	20	30	40	50	60
Mouth	97.61	99.71	98.57	99.71	99.28	98.57
Forehead/eyes	95.47	99.14	98.92	98.57	97.85	99.71
Mouth+Forehead/eyes	97.85	97.85	99.52	99.64	99.71	99.52

Fig. 10. Average Recognition with different training's patterns

Figure 10 shows the evaluation results obtained with a different number of training patterns. It is important to mention that for the 3 cases under analysis (mouth region using windows 30 × 30, Forehead/Eyes region using windows 50 × 50, mouth region using windows of 30 × 30 + Region Forehead/Eyes using windows 50 × 50) the recognition rates are higher than 97 %, when both regions of mouth and front/eyes are used. Evaluation results show that the highest recognition rate is obtained when both,

the mouth and forehead regions are jointly used providing a recognition rate of about 99 %.

Figure 11 shows the comparison of recognition performance of proposed facial expression recognition scheme and other recently proposed schemes, [6, 13] reports global recognition rates using the whole face, while [2] reports the recognition performance using the mouth region and the forehead/eyes regions. Evaluation results show that proposed scheme performs better than other previously proposed methods [2, 6] and quite close to that reported in [13], with less computational complexity.

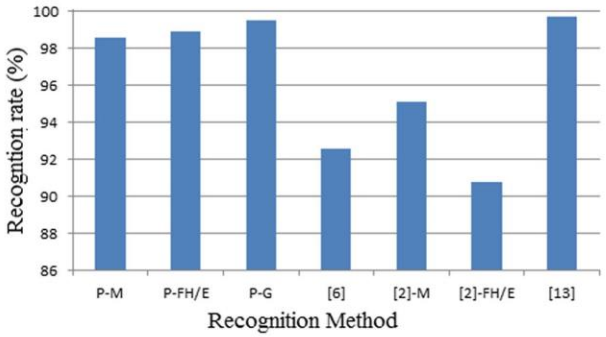


Fig. 11. Recognition performance of proposed scheme using the mouth region P-M, the forehead/eyes P-FH/E and both of them P-G. The performance of other previously proposed schemes are [2, 6] and [13] are also shown for comparison.

4 Conclusions

This paper presents an algorithm for recognizing facial expressions, performing an automatic segmentation of facial regions of interest to achieve this, first what is proposed is a segmentation of the face image obtained through the algorithm Viola Jones, then based on the symmetry of the face and using the integral projective automatically remove the 2 regions of interest, the first region is the Forehead/Eye and the second the Mouth, here is important that adequate extraction of regions even with different luminescence is achieved, this one of the main problems that present for facial expression recognition, moreover a classifier is proposed with low computational cost which performs better than an ANN, in both the percentage of recognition, as in training time. When making a comparison with the literature we can conclude that the proposed system performs better than [6] and [2], because a higher percentage of recognition in all possible cases was obtained and also a facial expression is recognized more accurately. Proposed system provides similar performance that [13] which use the whole image. Thus, it is possible to conclude that our system is able to recognize adequately the facial expressions with a percentage higher than 97 %, either taking the whole face, which in our case consists of regions of interest concatenated, or with partial occlusion, that is only considering one of the regions of interest proposals.

Acknowledgements. We thank the National Science and Technology Council of Mexico (CONACYT) and to the Instituto Politecnico Nacional for the financial support during the realization of this research.

References

1. Tian, Y., Kanade, T., Cohn, J.F.: Facial expressions analysis. In: Li, S.Z., Jain, A.K. (eds.) Handbook of Face Recognition. Springer, London (2004)
2. Zhang, L., Tjondronegoro, D., Chandran, V.: Random Gabor based templates for facial expression recognition in images with facial occlusion. Neurocomputing **145**, 451–464 (2014)
3. Buciu, K., Pitas, I.: An analysis of facial expression recognition under partial face image occlusion. Image Vis. Comput. **26**(7), 1052–1067 (2008)
4. Miyakoshi, Y., Kato, S.: Facial emotion detection considering partial occlusion of face using Bayesian network. In: 2011 IEEE Symposium on Computers & Informatics (ISCI), pp. 96–101 (2011)
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, pp. 511–518 (2001)
6. Benitez-Garcia, G., Sanchez-Perez, G., Perez-Meana, H., Takahashi, K., Kaneko, M.: Facial expression recognition based on facial region segmentation and modal value approach. IEICE Trans. on Inf. Syst. **E97-D(4)**, 928–935 (2014)
7. Lundqvist, D., Flykt, A., Öhman, A.: The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet (1998). ISBN 91-630-7164-9
8. Freeman, W., Tanaka, K., Ohta, J., Kyuma, K.: Computer vision for computer games. In: International Conference on Automatic Face and Gesture Recognition, pp. 100–105 (2008)
9. Pizer, M.: Adaptive histogram equalization and its variations. Comput. Vis. Graph. Image Process. **39**, 355–368 (1987)
10. Li, W., Mao, K., Zhang, H., Chai, T.: Selection of Gabor filters for improved texture feature extraction. In: IEEE International Conference on Image Processing, pp. 361–364 (2010)
11. Pakdel, M., Tajeripour, F.: Texture classification using optimal Gabor filters. In: International Conference on Computer and Knowledge Engineering, pp. 208–213 (2011)
12. Jolliffe, I.: Principal Component Analysis, 2nd edn. Springer, New York (2002)
13. Hasimah, A., Muthusamy, H., Yaacob, S., Adom, A.: Expert Syst. Appl. **42**, 1261–1277 (2015)
14. Hernandez-Matamoros, A., Escamilla-Hernandez, E., Perez-Daniel, K., Nakano-Miyatake, K.M., Perez-Meana, H.: A supervised classifier scheme based on clustering algorithms. In: IEEE Central America and Panama Convention (CONCAPAN XXXIV), pp. 1–5 (2014)

Facial Expression Recogntion in Unconstrained Environment

Andres HERNANDEZ-MATAMOROS[a,b], Takayuki NAGAI[b], Muhammad Attamimi[b]
Mariko NAKANO[a,], Hector PEREZ-MEANA[a,1]
[a]*Mechanical and Electrical Engineering School Culhuacan Campus,*
Instituto Politecnico Nacional, Mexico City, Mexico
[b]*Mechanical Engineering and Intelligent Systems,*
The University of Electro-Communications, Tokyo Japan

Abstract. The facial expression recognition has been a topic of active researches given a result the proposal of several efficient algorithms; however in most cases they remain limited to controlled conditions situations. In this study, we tackle the challenge of recognizing emotions through the facial expression into activities in-the-wild adding the accuracy rate for each expression. To this end we an algorithm that allows accurate face expression recognition in an uncontrolled environments. Proposed scheme firstly detects and segments automatically the face regions of interest (ROI) together with the detection of different profile of face (left, frontal and right). Then it uses only the frames in which the face profile is frontal to carry out the emotion recognition. We use a classifier based on clustering, it has the advantage that if a new class (emotion) is added, it is not necessary to train this completely. Proposed scheme was evaluated using short video clips of several pictures together with description sentences describing the main activity in the video. The evaluation results show that the proposed scheme is able to recognize the principal emotions in unconstrained video sequences.

Keywords. Face expression recognition, projective integral, modal analysis, profile estimation, face dimension estimation.

1. Introduction

The development of signal processing and pattern recognition schemes, suitable for implementation in smart devices has been a topic of active research during the las several decades, given as a result the development of several efficient algorithms that can be used in many practical applications. Among them the facial expression recognition (FER) systems have been used to recognize the mood. This is because several problems can be avoided if can accurately if a given person has a nervous breakdown, is tired, angry or happiness, etc.

A very important part of face expression recognition systems (FER) is the detection of face regions because an accurate detection of such regions may improve the FER performance. Currently several face detection algorithms have been proposed that are able to detect faces images in pictures and even smiles, most of them based on

[1] Corresponding Author, SEPI Mechanical and Electrical Engineering School, Instituto Politecnico Nacional, Av. Santa Ana 1000, Coyoacan, Mexico City, Mexico; E-mail: hmperezm@ipn.mx.

the Viola-Jones algorithm [6]. Unfortunately these schemes do not provide enough accurate face detection to be directly used in a FER system. This is because when a person has a given mood, some movement of the face muscles may be present, as shown [14]. This movement is, in general, different in each facial expression doing it possible to use them to determine the regions of interest of the face to estimate the different facial expression. Besides ROI estimation, several other problems remain in facial expression recognition. Some of them are related with the face orientation related to the camera, because if the person is not looking straightforward to the camera, partial occlusion of the face may occur; or the presence of shadows due to poor illumination conditions.

To reduce the problems described above, we propose a FER algorithm that is able to detect the face orientation in the frame under analysis, such that if the face is perpendicular to the camera, the ROI is estimated, after the ROI estimation each region is segmented into a set of N×M blocks to get the feature vector using the modal value. The resulting features matrix is then applied to a PCA [12] and LDA [13] for dimensionality reduction. Next using the classifier proposed in [7], with low computational cost which provides recognition rates similar to those provided by other high performance classifiers such as the SVM and ANN, is used. The proposed algorithm was trained using the KDEF data base [15] which consists of 490 images which are divided into 7 facial expressions (Afraid, Angry, Disgusted, Happy, Sad, Surprise and Neutral) of 70 people. Finally the proposed scheme tested using the HOHA database [16] which consists of 150 videos of 32 movies, which are divided in 8 actions (Answer Phone, Get Out Car, Hand shake, Hug Person, Kiss, Sit Down, Sit Up and Stand Up). The evaluation results show that the proposed system provide recognition rates of about 90% when it is required to perform facial expression recognition in video sequences.

The rest of the paper is organized as follows: Section 2 describes the proposed scheme, the experimental results are shown in Section 3 and Section 4 provides the conclusion of this work.

2. Proposed System

The proposed facial expression recognition (FER) system is shown in Fig. 1. Here, firstly the received frame is fed into the face extraction stage, which extracts the face image using the Viola- Jones algorithm [6]. Next the extracted face image is fed into the profile detection module to determine if the profile is a half left, straight or half right profile. These are then inserted into a classifier stage to determine the profile type. Next, if the classifier determines that the profile of the image under analysis corresponds to a straight one, the face image is inserted into the region of interest (ROI) automatic detection stage, which firstly estimates the face dimensions. Then using this information the ROIs are automatically segmented to get the mouth and Forehead/Eyes using the image moments [10] and projective integrals [10]. These ROIs are then segmented in 35×40 no overlapping blocks. The modal value is then calculated for each block to get as a result a vector with 1400 dimensions for each ROI. Next the feature vectors are independently process by a PCA and LDA for dimensionality reduction, after that the vector is concatenated. Finally, this vector is inserted into the classifier stage to take the final decision to each frame. Next sections provide a description of all stages of proposed system.

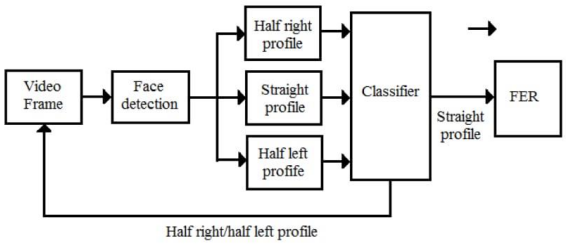


Figure 1. Proposed face expression recognition system

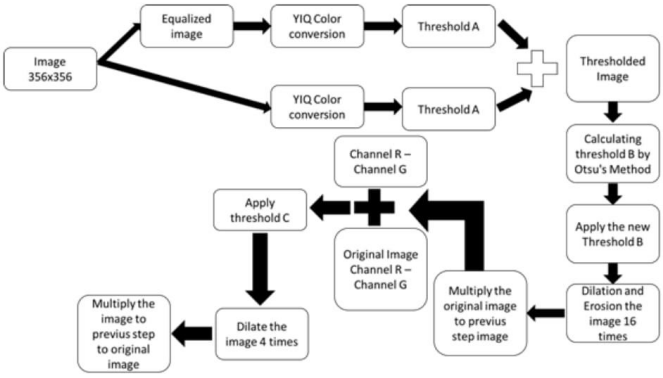


Figure 2. Block diagram of proposed profile detection system.

2.1. Profile detection

Here, firstly we use the Viola-Jones Algorithm, this extracts the face on image, so we apply this algorithm to detect the face in a frame on a video, we implemented this algorithm using minimum window size at a third of its width and height respectively. The dimension of the face image detected by the Viola-Jones algorithm is re-dimensioned to a size of 350×350. Next we equalize the image and apply the YIQ color space conversion given by (1)-(3) on both, the original and the equalized images. The profile detection system is shown in the Fig. 2.

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

$$I = 0.5960 * R - 0.2740 * G - 0.3220 * B \quad (2)$$

$$Q = 0.2110 * R * 0.5230 * G + 0.3210 * B \quad (3)$$

In order to detect the face parts in an image, the threshold A is applied to the pixels of both images, where the threshold value is given by the following equation:

$$(60 < Y < 200) AND (20 < I < 50) \tag{4}$$

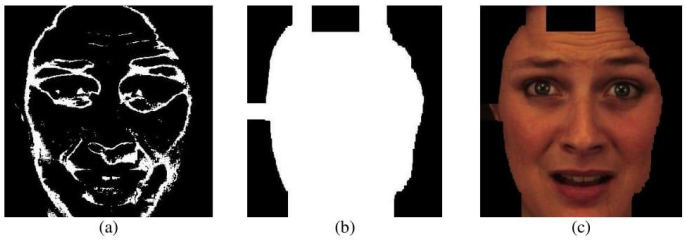
Next both thresholded images are added (Fig. 3, right image), so we have only one image. Subsequently the Otsu's method is applied to get a new threshold and applying this threshold, with the goal to fill the center of the face, dilation and erosion is applied 16 times to obtained the dilate image.

(a) (b)

Figure 3. (a) Original image, (b) Resulting image after thresholding operation

Next, the original image is multiplied by the dilate image shown in Fig. 4(b), giving a result the image shown Fig. 4 (c). Subsequently the resulting image, shown in Fig. 4(c), is decomposed in its RGB components and then the channel R is subtracted to channel G, given as a result the image $I_T(x,y)$, shown Fig. 5 (a). Next the original image $I_o(x,y)$ is also decomposed in its three color planes RGB, subtracting the plane G from the R one to obtain the image shown in Fig. 5(b). Finally the resulting image is added to previously obtained image, shown in Fig. 5(a) giving a result the image shown in Fig. 5(c).

In the next step we use a threshold C of value 75, because if a threshold lower than 75 is used, the image would be less wide but in this situation the eyes and mouth are lost. The Fig. 6 shows the difference when different threshold values C is used.

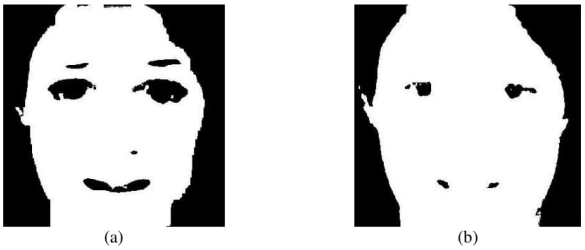


(a) (b) (c)

Figure 4 (a) Resulting image using the Otsu's threshold scheme, (b) dilated image, (c) product of original and dilated images, $I_T(x,y)$.

(a) (b) (c)

Figure 5 (a) Image obtained subtracting the channel G form channel R of the image $I_T(x,y)$, (b) Image obtained subtracting the channel G form channel R of original image, (c) image resulting adding (a) and (b).



(a) (b)

Figure 6 Image obtained using the Otsu's method with a threshold value C equal (a) 75 and (b) 30

After the threshold C is applied, the images are dilated 4 times and fill the spaces insides the face image, such that the final result is shown in the figure 7. With the final mask we can calculate the projective integral to the image on the columns as shown the figure 8, using the projective integral is possible detect the kind of profile.

(a) (b)

Figure 7 (a) Final mask, (b) final image

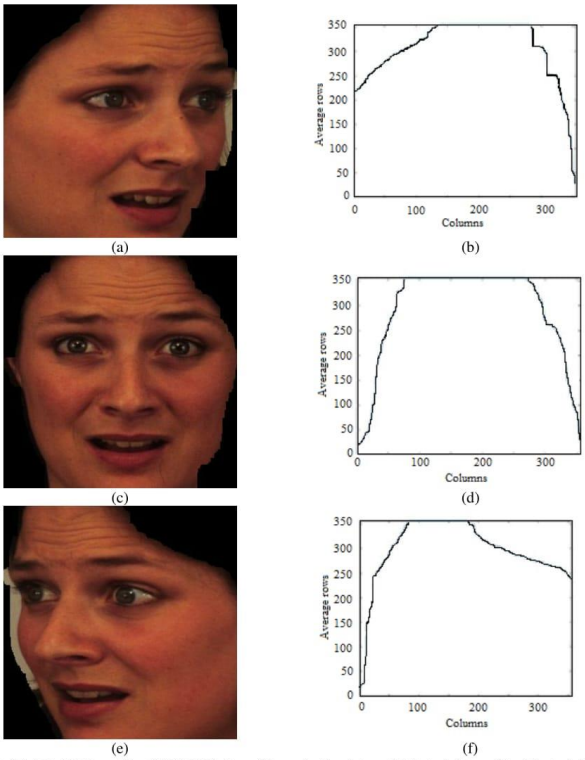


Figure 8 (a) Half left profile, (b)) Half left profile projective integral (c) straight profile, (d) straight profile projective integral, (e) half right profile, (f) half right profile projective integral.

To train the system profile detection we create a matrix using the feature vectors obtained after the PCA algorithm is applied for dimensionality reduction. If use the classifier based on clustering proposed in [7, 18] to identify the profiles with 260 patterns and 230 patterns for testing we obtain the results shown in the Table 1.

Table 1. Performance of detection profile algorithm

Left	Center	Right	Average
92.85	98.57	90	93.8

2.2. Straight Profile Facial Expression Recognition

The detected face image may contain noise, such as the hair or ears, which does not contain relevant information for the facial expressions; or the background where the photograph was taken. In order to eliminate this problem that may decrease the recognition rate of the proposed face expression recognition system (Figure 9), a more accurate estimation for face dimension is carried out. To adjust the face dimension parameters, firstly the color face image is divided into its three color components: Red, Green and Blue channels [17, 18]. Next the red and green channels are subtracted among them to highlight the skin. Finally the resulting image is binarized using the equation

$$I(x,y)=\begin{cases}0 & I(x,y)<1\\ 255 & I(x,y)\geq 1\end{cases} \tag{5}$$

After the binarized image, the moments of the resulting image are estimated [18]

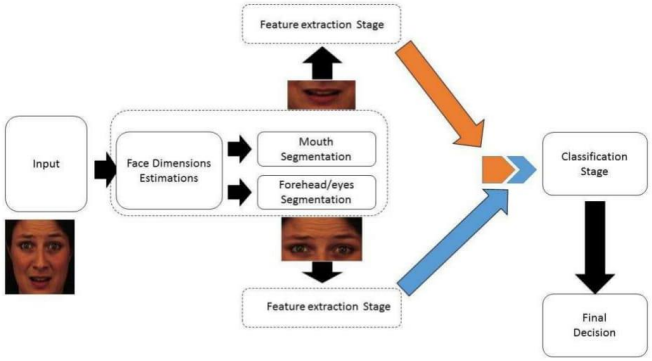


Figure 9 Straight profile proposed FER system.

$$M_{pq}=\sum_{x=1}^{N}\sum_{y=1}^{M}x^{p}y^{q}I(x,y) \tag{6}$$

Where $I(x,y)$ is the image intensity at position (x,y), N is the number of columns and M is the number of rows in the image; while p and q define the moment of the image. Next using the eq. (2) the center of gravity can be estimated as follows:

$$x_c=M_{1,0}/M_{0,0} \tag{7}$$

$$M=M_{0,1}/M_{0,0} \tag{8}$$

Next, using (7) and (8) the following variables are defined [7, 17, 18]

$$a = \frac{M_{2,0}}{M_{0,0}} - x_c^2 \tag{9}$$

$$b = 2\left(\frac{M_{1,1}}{M_{0,0}} - x_c y_c\right) \tag{10}$$

$$c = \frac{M_{0,2}}{M_{0,0}} - y_c^2 \tag{11}$$

Thus, the face image width can be estimated as follows:

$$W = 2\sqrt{\frac{(a+b) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{12}$$

Using W, the left, x_l, right, x_r and upper edges of the region of interest (ROI) of the face image can be estimated as follows [18]:

$$x_l = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil \tag{13}$$

$$x_r = \lceil x_c \rceil - \left\lceil \frac{W}{2} \right\rceil + \lceil W \rceil \tag{14}$$

$$y_u = \lceil y_c \rceil - 0.84\left\lceil \frac{W}{2} \right\rceil \tag{15}$$

Thus the estimated ROI is shown in Fig. 10.

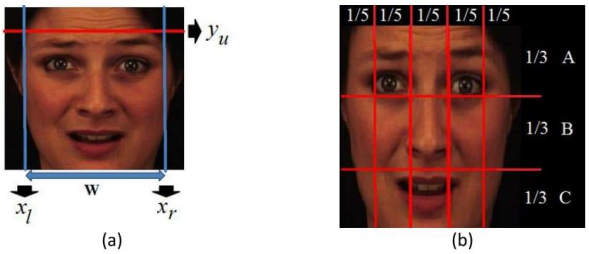


Figure 10. Segmented face region, (b) Symmetrical relationship of the face

2.2.1. Forehead/ eye segmentation

A very important part of proposed face expression recognition system is the forehead/face segmentation. To this end, the segmented face region is divided into three regions from the top (A, B and C) as shown in Fig. 10, where the region A is the ROI in the case of the Forehead/eyes segmentation task.

2.2.2. Mouth segmentation

The mouth region is segmented dividing the face region into three regions of same high and take the C region of Fig. 10(b). Next, because it is necessary to segment only the mouth region, the Red and Green image's channels are subtracted among them and the resulting image histogram is equalized obtaining the image shown in Fig11.

Figure 11. Equalized image obtained from the subtraction of red and green planes.

The next step for the automatic segmentation of the mouth region is the estimation of the horizontal projective integral [17] which is the average of the pixel values of each column inside the ROI. Figure 12 shows the horizontal projective integral estimated using the equalized image shown in Fig. 11.

Next we obtain the maximum value of the projective error, which will be denoted as "D". Then using the value "D" the left border of the ROI containing the mouth is estimating by subtracting D from x_c, i. e. while the right border is obtained adding to x_c, keeping the original image height, as shown in Fig. 7, with this procedure the region of interest is extracted automatically.

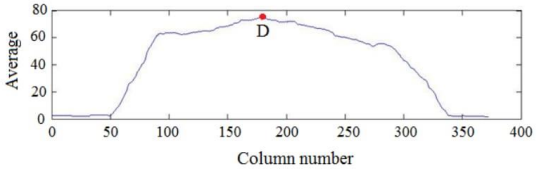


Figure 12. Horizontal projective integral of the mouth ROI

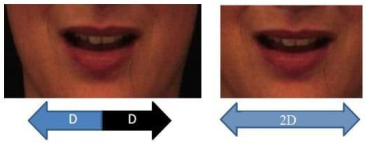


Figure 13. Detection of mouth ROI

3. Proposed System

To perform the feature extraction, each one of the detected ROIs is divided in 35x40 blocks which are characterized by the modal value of the pixels for each block. Next the resulting features vector of each training ROI, with 35x40 elements, are arranged in a matrix form and applied PCA and LDA stage for dimensionality reduction. Next sections provide a brief description of these stages.

Principal component analysis (PCA) [12] is a standard tool in modern data analysis in diverse fields from computer science because it is a simple, non-parametric method for extracting relevant information from con- fusing data sets. That uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables, for this paper the number of principal components is N-1 original variables.

Linear discriminant analysis (LDA) [18] is a method used in pattern recognition to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification. LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities

The classifier [7, 18] has a low computational complexity, which uses a supervised training approach, like the ANN or SVM approaches, with the characteristic that if a new class must be added, it is not necessary to train the system with all patterns again but only with the patterns belonging to the new class. The Classification stage give us the decision frame by frame, also it is possible take a decision video by video and action by action. To get this in a video, the modal value is taken for all frames on this video and the last part to get the percentage for each action, the modal value is taken for all frames on this video.

4. Evaluation Results

The proposed scheme tested using the HOHA database [16] which consists of 150 videos of 32 movies, which are divided in 8 actions (Answer Phone, Get Out Car, Hand shake, Hug Person, Kiss, Sit Down, Sit Up and Stand Up). Figures 14-17 show the results to 4 frames in the video As Good as It Gets - 01766.avi where the action is talking inside the car. The results to the Figs. 14-16 are similar, that is because the facial expression is similar. However although in the Fig. 17 the result is different, the goal is recognize the facial expression in the full video frame, whose result is shown in the Fig. 18. Finally the Figures 17-24 show the results for each action after to analyze all videos.

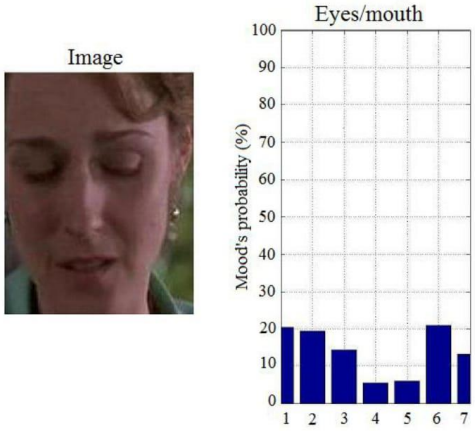


Figure 14. Detected women and FER in frame 116 of "As Good As It Gets -01766.avi" Action: talking: Here 1: afraid, 2: angry, 3: disgusted, 4: happy, 5: neutral, 6: sad, 7: surprise.

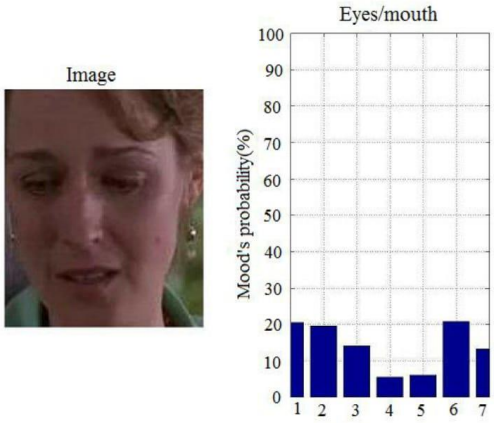


Figure 15. Detected women and FER in in frame 114 of "As Good As It Gets -01766.avi", Action: talking: Here 1: afraid, 2: angry, 3: disgusted, 4: happy, 5: neutral, 6: sad, 7: surprise.

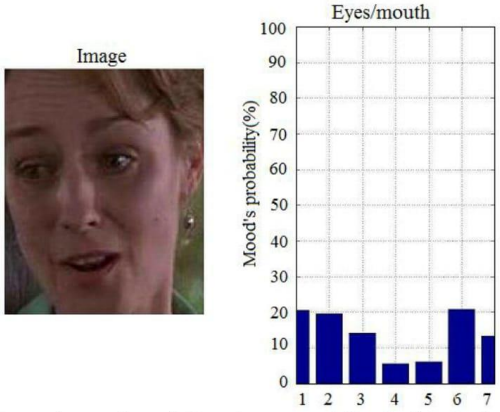


Figure 16. Detected women face and FER in frame 165 of "As Good As It Gets -01766.avi", Action: talking: Here 1: afraid, 2: angry, 3: disgusted, 4: happy, 5: neutral, 6: sad, 7: surprise.

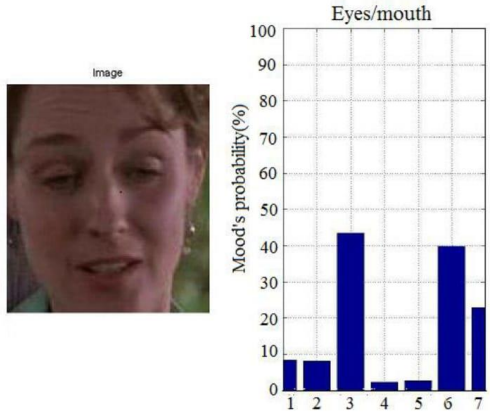


Figure 17. Detected women face and FER in frame 174 of "As Good As It Gets -01766.avi", Action: talking. Here 1: afraid, 2: angry, 3: disgusted, 4: happy, 5: neutral, 6: sad, 7: surprise.

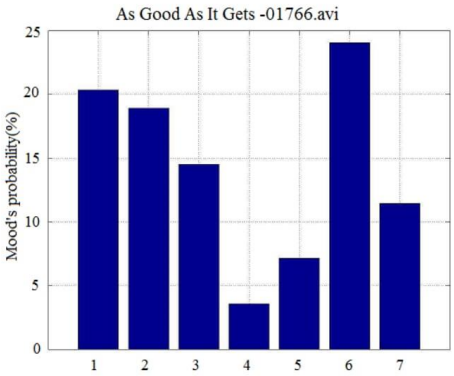


Figure 18. Average face expression recognition (FER) of detected women face in frames 60-177 and 191-194 of "As Good As It Gets -01766.avi", Action: talking. Here 1: afraid, 2: angry, 3: disgusted, 4: happy, 5: neutral, 6: sad, 7: surprise.

5. Conclusions

The proposed scheme tested using the HOHA database [16] which consists of 150 videos of 32 movies, which are divided in 8 actions (Answer Phone, Get Out Car, Hand shake, Hug Person, Kiss, Sit Down, Sit Up and Stand Up). Figures 14-17 show the results to 4 frames in the video As Good as It Gets - 01766.avi where the action is talking inside the car. The results to the Figs. 14-16 are similar, that is because the facial expression is similar. However although in the Fig. 17 the result is different, the goal is recognize the facial expression in the full video frame, whose result is shown in the Fig. 18. Finally the Figures 17-24 show the results for each action after to analyze all videos.

Acknowledgements

The authors would like to thank to the National Science and Technology Council of Mexico, to JASSO of Japan, the National Polytechnic Institute and the University of Electro-Communications of Japan for the financial support during the realization of this research.

References

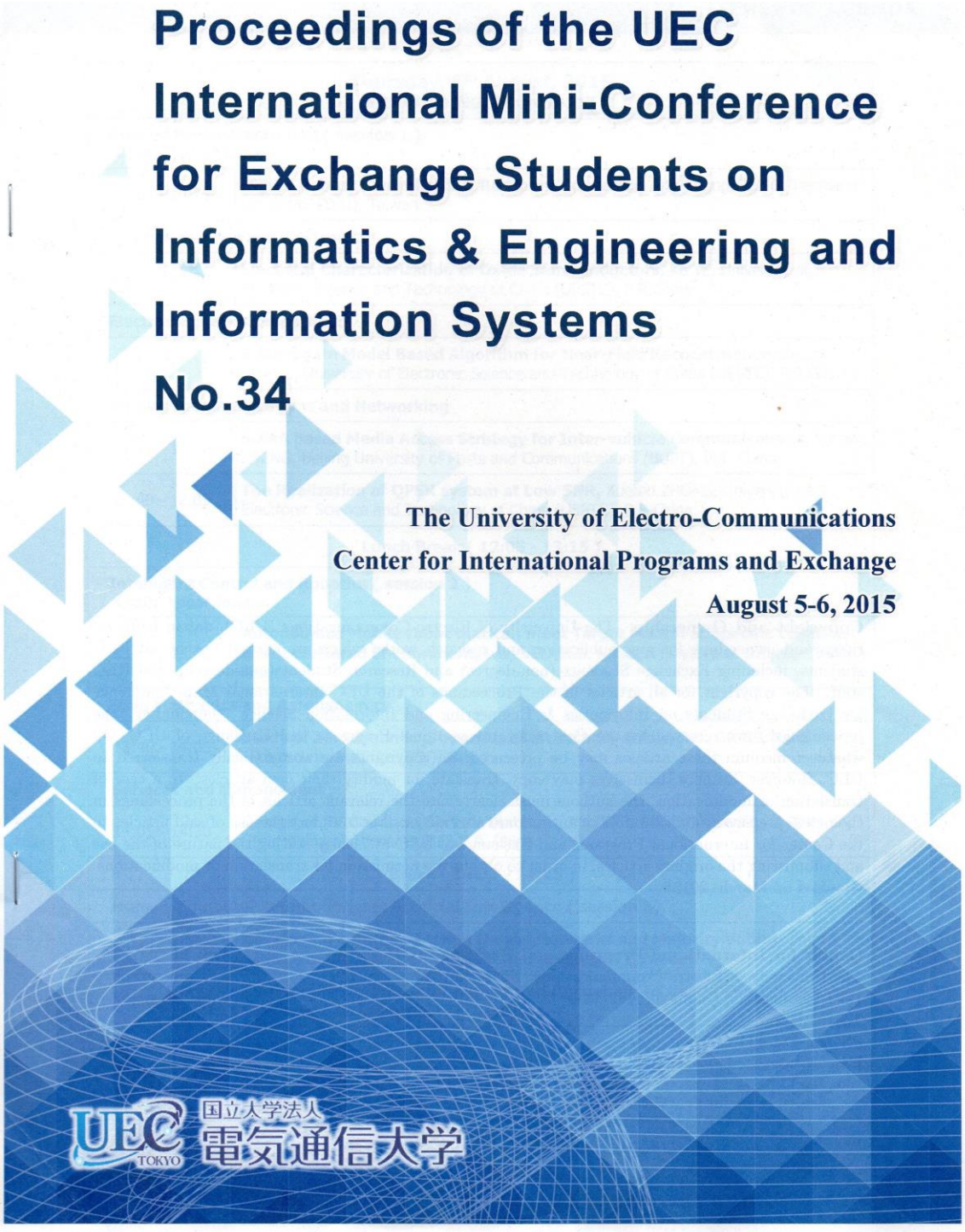
[1] Y. Tian, T. Kanade and J.F. Cohn, Facial Expressions *Analysis Handbook of Face Recognition,* eds. Stan Z.Li and Anil K. Jain, Springer-Verlag, Berlin, 2004.

[2] B. Fasel and J. Luettin, Automatic facial expression analysis: A survey, *Pattern Recognition* (2003) 259-275.

[3] Kotsia, I. Buciu, and I. Pitas, An analysis of facial expression recognition under partial face image occlusion, *Image Visual Computing* **26** (2008), 1052-1067.
[4] Y. Miyakoshi and S. Kato, Facial emotion detection considering partial occlusion of face using Bayesian Network, *IEEE Symposium on Computers and* Informatics (2011), 96-101.
[5] Gary Bradski, Adrian Kaehler. Learningb. *O'Reilly Media*, Inc.(2008).
[6] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, Proceedings of the IEEE Computer Society Conference (2001), I- 511,I-518.
[7] A. Hernandez-Matamoros, E. Escamilla-Hernandez, K. Perez-Daniel M. Nakano-Miyatake, H. Perez-Meana, A supervised classifier scheme based on clustering algorithms, *IEEE Central America and Panama Convention (CONCAPAN XXXIV)*, (2014) , 12-14..
[8] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, Computer Vision for Computer Games, *Int. Conf. on Automatic Face and Gesture Recognition* (1996),
[9] M. S. Pizer, Adaptive Histogram Equalization and its Variations, *Computer Vision, Graphics and Image processing*, **39** (1987) , 355-368.
[10] Gary R. Bradski, Computer Vision Face Tracking For Use in a Perceptual User Interface, CAMSHIFT, Int. Journal of Technology **Q2** (1998) , 1-15.
[11] Md. Al-Amin Bhuiyan Face, Detection and Facial Feature Localization for Human-machine Interface. *National Institute of Informatics,* (2003).
[12] I.T. Jolliffe, *Principal Component Analysis,* 2nd ed., Springer, Berlin, (2002).
[13] Tae-Kyun Kim; Kittler, J., Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, (2005), pp.318-327.
[14] Contreras Flores, ARTNATOMY,www.artnatomia.net,Victoria, SPAIN (2005).
[15] The Karolinska, Directed Emotional Faces - KDEF, *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, ISBN 91-630-7164-9.
[16] I. Laptev, M. Marsza, C. Schmid, B. Rozenfeld, Learning Realistic Human Actions from Movies, *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.
[17] A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, A Facial Expression Recognition with Automatic Segmentation of Face Regions, *SoMeT* , **CCIS 532** (2015), 529–540.
[18] A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, Face Emotion Recognition with Automatic Segmentation of Face Using a Fuzzy Based Classification Approach, *Knowledge Based Systems*, **110**, 1-14.

Proceedings of the UEC International Mini-Conference for Exchange Students on Informatics & Engineering and Information Systems No.34

The University of Electro-Communications
Center for International Programs and Exchange
August 5-6, 2015



Facial expression recognition in the wild

Andres Gerardo HERNANDEZ MATAMOROS *
UEC Student No. 1595002
National Polytechnic Institute (IPN)
Mexico City, Mexico

Takayuki NAGAI
Department of Mechanical Engineering
and Intelligent Systems
The University of Electro-Communications
Tokyo, Japan

August 5^{th}-6^{th}, 2015

Keywords: Facial Expression Recognition (FER), Youtube, Clustering, Viola-Jones Algorithm.

Abstract

Recently, the study of facial expression has grown up but it remains limited to narrow small vocabularies of emotion into videos.In this study, we tackle the challenge of recognizing emotions from the video which includes activities "in-the-wild". We propose a solution that takes a short video clip along with brief sentences, that describe the main activity in the video.It is possible to set the relation ship between the actions and emotions since we have a paired videos and sentences.To recognize the facial expression we modify our previous work to automatically detect and segment the region of interest (ROI) of forehead/eyes and mouth.We then combine a classifier based on clustering.Our proposed classifier is able to train online if a new class (emotion) is added.For future work, we plan to evaluate our method on a Hollywood database and show that it is able to improve the acuuracy rate of the expressions and actions.

*the author is supported by JASSO Scholarship.

Facial expression recognition in the wild

Andres Gerardo HERNANDEZ-MATAMOROS
Student ID:1595002
SEPI,ESIME Culhuacan
National Polytechnic Institute
Mexico City, Mexico

Takayuki NAGAI
Department of Mechanical Engineering
and Intelligent Systems
The University of Electro-Communications
Tokyo, Japan

Introduction

Despite a recent push towards emotion recognition, it remains limited to narrow small vocabularies of emotion into video. In this study, we tackle the challenge of recognizing the emotions from the video which includes activities "in-the-wild". The proposed method improves the accuracy rates of each expression.

Objectives

-Recognition of frontal face in the video.
-Recognition of face parts(forehead/eyes and mouth).
-Determine the accuracy rate for each expression.

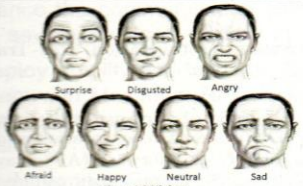


Figure 1.ROI's face.

Methods and Materials

-Viola-Jones Algorithm for face detection
-Principal Component Analysis (PCA) to compress the length of vectors.
-Classifier based on clustering.

Preliminary Results

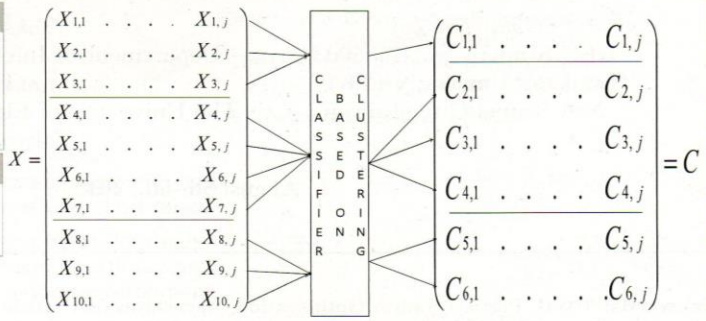
Here, we used a database (1) to learn the model because it contains the frontal face. To characterize the face, we take the proposal of (2) and ussing the classifier based on clustering. It is possible to improve the accuracy rates of each expression.

Expected Results

-Selecting frontal face from the image automatically.
-Set the relationship between the actions and the facial expressions in a video.

Conclusion and Discussion

-We have proposed an algorithm for recognizing facial expressions, performing an automatic extraction of facial regions of interest in a video.
-The proposed algorithm is able to provide 2 regions of interest; the first one is forehead /eyes and the second one is mouth.



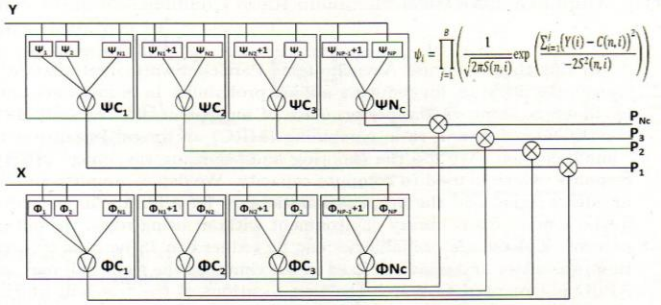


Table 1. Preliminary Results , accuracy rates for each expression.

Expression									
			Joint probability			Joint probability			Joint probability
Afraid	***28.96%***	***100%***	***100%***	7.35%	0.0%	0.0%	15.68%	0%	0%
Angry	14.64%	0.0%	0.0%	3.61%	0.0%	0.0%	9.78%	0.0%	0.0%
Disgusted	21.42%	0.0%	0.0%	1.71%	0.0%	0.0%	18.29%	0.0%	0.0%
Happy	8.37%	0.0%	0.0%	0.79%	0.0%	0.0%	***45.39%***	***100.0%***	***100.0%***
Neutral	4.86%	0.0%	0.0%	4.2%	0.0%	0.0%	2.62%	0.0%	0.0%
Surprise	1.18%	0.0%	0.0%	***81.78%***	***99.9%***	***100%***	3.01%	0.0%	0.0%
Sad	20.57%	0.0%	0.0%	0.57%	0.01%	0.0%	5.23%	0.0%	0.0%

Contact

Andres G. Hernandez Matamoros
National Polytechnic Institute (Mexico)
ahernandezm1131@hotmail.com

References

1. The Karolinska Directed Emotional Faces Lundqvist, D., Flykt, A., & Öhman, A. (1998). The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.
2. A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, H.Perez-Meana .Expression Recognition with Automatic Segmentation of Face Regions. SOMET 2015.
3. ARTNATOMY/ARTNATOMIA,www.artnatomia.net,Victoria Contreras Flores,ESPAÑA,2005.
4. Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* , vol.1, no., pp.I-511,I-518 vol.1, 2001.

Proceedings of the UEC International Mini-Conference for Exchange Students on Informatics & Engineering and Information Systems No.35

The University of Electro-Communications
Center for International Programs and Exchange
March 3-4, 2016

国立大学法人
電気通信大学

Facial expression recognition in the wild

HERNANDEZ MATAMOROS
Andres Gerardo *
UEC Student No. 1595002
PEREZ MEANA Hector
National Polytechnic Institute
Mexico City,Mexico

NAGAI Takayuki
Department of Mechanical Engineering
and Intelligent Systems
The University of Electro-Communications
Tokyo, Japan

Abstract

In the last years the study of facial expression has grown up but it remains limited to narrow small vocabularies of emotion into videos. In this study, we tackle the challenge of recognizing emotions through the facial expression into activities "in-the-wild" adding the accuracy rate for each expression. We propose a solution that takes a short video clip along with description sentences, it describes the main activity in the video. The action, as we have a brief sentence that describe the action of the video, it is possible that we will set a relation between the actions and emotions into the video. To recognize the facial expression we modify our previous work,it detects and segments automatically the regions of interest (ROI) adding the detection of different profile of face(left, straigth and rigth).We then combine a classifier based on clustering, it has the advantage that if a new class (emotion) is added, it is not necessary to train this completely.The obtained results is so interesting, we can recognize the emotion principal in a video using 7 facial expressions, for other hand, we can assign the accuracy rate of facial expression for daily actions for example: answer phone, get our car, sit up .

Keywords: Projectives Integrals,Modal Value, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)

1 Introduction

The use of smart devices in the solution of several problems has increased recently given as a result the development of very efficient systems that can be used in many practical applications. Among them the facial expression recognition (FER) systems have been used to recognize the mood. This is because several problems can be avoided if it is possible to accurate detect the mood of a person, i.e., if a given person has a nervous breakdown, if he is tired, angry or happiness, etc. For this reason, during the last several years the interest for developing these kinds of systems has increased. A very important part of such systems is the detection of face regions because an accurate detection of such regions may improve the performance of FER. Currently, in the literature exists some algorithms able to detect faces in an image and even smiles must of them based on Viola-Jones algorithm. Unfortunately these schemes are not enough accurate to detect the facial expression and thereby to achieve accurate mood detection. This happens because when someone does a mood a expression, it could be strong or not, some movement of the face muscles is done involuntarily. This movement is, in general, different in each facial expression doing it possible to determine the regions of interest of the face in each case. Several problems are present in facial expression recognition; some of them are related with the face orientation related to the camera, because if the person isn't looking straightforward to the camera partial occlusion of the face may occur, or the presence of shadows due to poor illumination conditions. For this reason, we propose a FER algorithm that is able to detect when the face is straigthforward to the camera, after segment the face ROI under different illumination conditions, after the ROI estimation, each region is segmented in a set of NxM blocks to get the characteristic vector using the modal value. The resulting features matrix is then applied to a PCA and LDA for dimensionality reduction. We use a classifier with low computational cost which provides recognition rates similar to those provided by other high performance classifiers such as the SVM and ANN. The proposed algorithm was trained with KDEF data base which consists of 490 images which are divided into seven facial expressions of 70 people and it was tested with HOHA database which consists of 32 videos which are divided in 8 actions. Evaluation results show that using the proposed system, we

*the author is supported by JASSO Scholarship.

able to recognize the facial expression in a video. The rest of the paper is organized as follows: Section 2 describes the system framework, the experimental results are shown in Section 3 and Section 4 provides the conclusion of this work.

2 Proposed System

The System framework of proposed facial expression recognition (FER) system is shown in 1. Here, firstly the received face image is fed into the face extraction stage, which extracts the face image using the Viola-Jones algorithm, in the next stage we determine the profile (half left, straight, half rigth). Next the straight profile is inserted into the region of interest (ROI) detection stage, which firstly estimates the face dimensions. Then using this information the ROIs are automatically segmented to get the mouth and Forehead/ Eyes using the image moments and projective integrals. These ROIs are then segmented in 35x40 no overlapping blocks,The modal value is calculated for each block, this gets as result a vector with 1400 dimensions for each ROI .Next the feature vectors are independently process by a PCA and LDA for dimensionality reduction. Finally, the resulting vectors are fed in to the classifier stage to take the final decision. Next sections provide a description of all stages of proposed system.

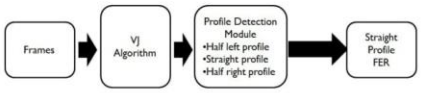


Figure 1: Proposed System

2.1 Profile Detection

Firstly,we use the Viola-Jones Algorithm to detect the face in a frame on a video, we implemented this algorithm using minimum window size at a third of its width and height respectively.The image received by the Viola-Jones algorithm is redimensioned at size 350x350, the next step we equalize the image [1], after we apply the YIQ color conversion using the equations 1-3 on the original image and the equalized image.We can see the system in 2

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

$$I = 0.5960 * R - 0.2740 * G - 0.3220 * B \quad (2)$$

$$Q = 0.2110 * R - 0.5230 * G + 0.3210 * B \quad (3)$$

In order to detect the face parts in an image, the pixels of both images are thresholded [1], the threshold value is given by the following equation.

$$(60 < Y < 200) AND (20 < I < 50) \quad (4)$$

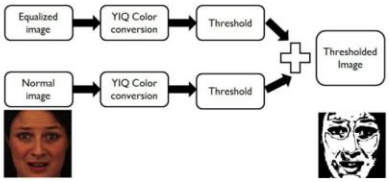


Figure 2: Profile Detection Diagram

The next step is add both umbralized images,so we have only one image, this image is divided widthwise in 7 parts and for each part we determine the percentage of information, with this we can determine the profile of the face namely if the percentage of information is higher in the center than at the sides the profile is straight, for other hand if the percentage of information is higher in the rigth side than at the left sides and center sides the profile is half left profile (that is because we see the left side of the face), by last if the percentage of information is higher in the left side than at the rigth sides and center sides the profile is half rigth profile.In this paper only uses the straigth profile because it have more information than the other profiles.

2.2 ROI Segmentation

The detected face image may contain noise, such as the hair or ears, which does not contain relevant information for the facial expressions; or the background where the photograph was taken. In order to eliminate this problem that may decrease the recognition rate of the proposed face expression recognition system, a more accurate estimation for face dimension is carried out.

2.3 Adjusment of face dimension

To adjust the face dimension parameters, firstly the color face image is divided into its three color components: Red, Green and Blue channels. Next the red and green channels are subtracted from them to highlight the skin as shown in .Finally, the resulting image is binarized using 6-6

$$I(x, y) = 0; I(x, y) < 1 \quad (5)$$

$$I(x, y) = 255; I(x, y) \geq 1 \quad (6)$$

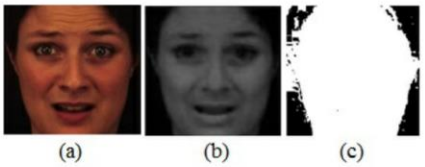


Figure 3: (a) Original image, (b) Image obtained by the subtraction, (c) binarized image

After the image binarization, shown in /refda1(c), the moments of the resulting image are estimated as follows [1]

$$M_{pq} = \sum_{x=1}^{N}\sum_{y=1}^{M} x^p y^q I(x,y) \tag{7}$$

where I(x,y) is the image intensity at position (x,y), N is the number of columns and M is the number of rows in the image; while p and q define the moment of the image. Next using 7 the centriod can be estimated as follows:

$$x_c = \frac{M_{1,0}}{M_{0,0}} \tag{8}$$

$$y_c = \frac{M_{0,1}}{M_{0,0}} \tag{9}$$

Next, using 8 and 9 the following variables are defined

$$a = \frac{M_{2,0}}{M_{0,0}} - x_c^2 \tag{10}$$

$$b = 2(\frac{M_{0,1}}{M_{0,0}} - x_c y_c) \tag{11}$$

$$c = \frac{M_{0,2}}{M_{0,0}} - y_c^2 \tag{12}$$

Next using 10-12 the face image width can be estimated as follows:

$$W = \sqrt{\frac{(a+b) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{13}$$

Using 13, the left, Xl, and right, Xr, edges of the face image can be estimated as

$$Xl = x_c - \frac{W}{2} \tag{14}$$

$$Xr = Xl + W \tag{15}$$

Next, using W the upper edge of the face image can be estimates as follows

$$Yu = y_c - 0.84\frac{W}{2} \tag{16}$$

From 14-16 the face image can be segmented as follows

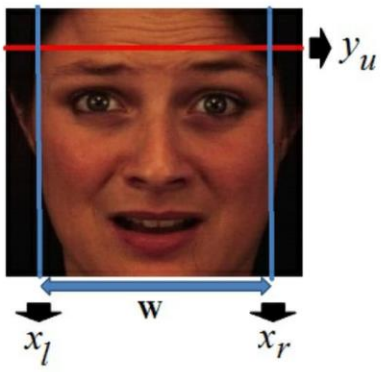


Figure 4: Segmented face region

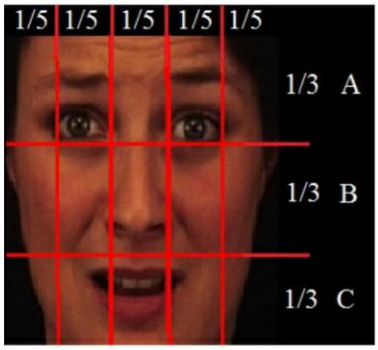


Figure 5: Symmetrical relationship of the face

2.4 Forehead/ eye segmentation

A very important part of proposed face expression recognition system is the fore-head/face segmentation. To this end, the segmented face region is divided into three regions from the top (A, B and C) as shown in 5, where the region A is the ROI in the case of the Forehead/eyes segmentation task.

2.5 Mouth Segmentation

To perform the segmentation of the mouth region, consider the segmented face re-gion which is divided into three regions, of same high, and take the C region of 5 as our ROI, but unlike the Forehead/eyes region, in this case it is necessary to segment only the mouth region. To this end, the Red and Green image's channels are sub-tracted among them, then a histogram equalization was performed of image ob-tained above,

obtaining an image as shown in 6.

Figure 6: Equalized version of the image obtained from the subtraction of red and green planes

The next step for the automatic segmentation of the mouth region is the estima-tion of the horizontal projective integral which is the average of the pixel values of each column. This is a vector containing the average value of the pixels in each column of the image inside the ROI. 7 shows the horizontal projective integral estimated using the equalized image shown in 6. Next we obtain the maximum value of the projective error, which will be denoted as "D". Then using the value "D" the left border of the ROI containing the mouth is estimating by subtracting D from xc, i. e. while the right border is obtained adding it to xc, keeping the original image height, as shown in refda6, with this procedure the region of interest is extracted automatically.

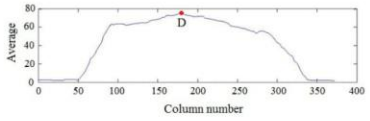


Figure 7: Horizontal projective integral of the mouth ROI

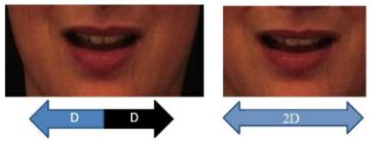


Figure 8: Detection of mouth ROI

3 Feature extraction

To perform the feature extraction, each one of the detected ROIs is divided in 35x40 blocks which are characterized by the modal value of the pixels for each block. Next the resulting features vector of each training ROI, with 35x40 elements, are arranged in a matrix form and applied to a PCA and LDa stage for dimensionality reduction. Next sections provide a brief description of these stages.

3.1 Principal component analysis

Principal component analysis (PCA) is a standard tool in modern data analysis in diverse elds from computer science because it is a simple, non-parametric method for extracting relevant information from confusing data sets. That uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables, for this paper the number of principal components is N-1 original variables.

3.2 Linear Discriminant analysis

Linear discriminant analysis (LDA) is a method used in pattern recognition to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities.

3.3 Classification stage

A low computational complexity classification method is used, which uses a supervised training approach, like the ANN or SVM approaches, with the characteristic that if a new class must be added, it is not necessary to train the system with all patterns again but only with the patterns belonging to the new class.

4 Results

The HOHA database consists in 150 videos, these was analyzed between Figure /refre1- /refre4 , we can see the result to 4 frames in a video As Good As It Gets - 01766.avi with action is Kiss, the results to the figure /refre1- /refre3 are similar, that is because the facial expression is similar but to figure /refre4 the result is so different, The goal is recognize the facial expression ina video, to get this we apply the modal value to all frames in a video to get the result that figure /refre5

shows to this video, In /refre6 we have to the video Butterfly Effect, The - 02093.avi where the action is Stand Up.

5 Conclusions

This paper presents an algorithm for recognizing facial expressions, performing an automatic segmentation of facial regions of interest to achieve this, first what is pro-posed is a segmentation of the face image obtained through the algorithm Viola Jones, then based on the symmetry of the face and using the integral projective automatically remove the 2 regions of interest, the first region is the Forehead / Eye and the second the Mouth, here is important that adequate extraction of regions even with different luminescence is achieved, this one of the main problems that present for facial expression recognition, moreover a classifier is proposed with low computa-tional cost which performs better than an ANN, in both the percentage of recognition, as in training time. When making a comparison with the literature we can conclude that the proposed system performs better than [6] and [2], because a higher percentage of recognition in all possible cases was obtained and also a facial expression is recognized more accurately. Proposed system provides similar performance that [13] which use the whole image. Thus, it is possible to conclude that our system is able to recognize adequately the facial expressions with a percentage higher than 97, either taking the whole face, which in our case consists of regions of interest concatenated, or with partial occlusion, that is only considering one of the regions of interest proposals.

6 Acknowledgments

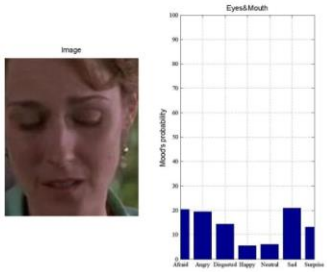
We thank the JASSO for the financial support during the realiza-tion of this research.

References

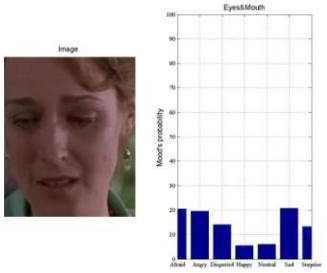
[1] 1. Y. Tian, T. Kanade, and J.F. Cohn, "Facial Expressions Analysis, Handbook of Face Recognition' Springer-Verlag, Dec, 2004.

[2] 2. L. Zhang, D. Tjondronegoro, V. Chandran: Random Gabor based templates for facial expression recognition in images with facial occlusion, Neurocomputing 145 451464,(2014).

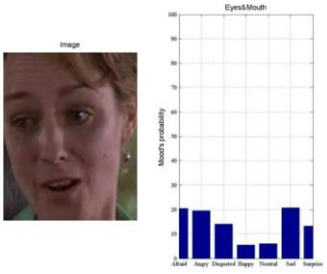
[3] K. Buciu, and I. Pitas: An analysis of facial expression recognition under partial face image occlusion, Image Visi. Comput, 26 (7),1052-1067 (2008).



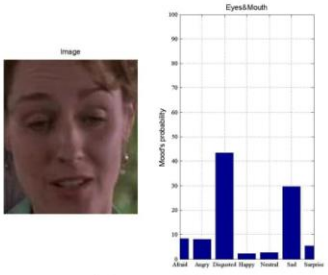
(a) Detected face 3



(b) Detected face 9

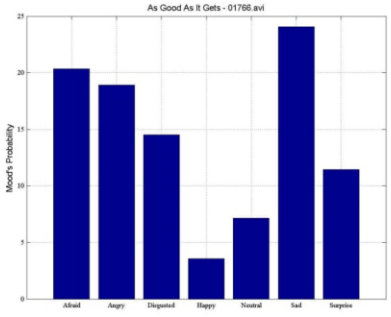


(c) Detected face 21

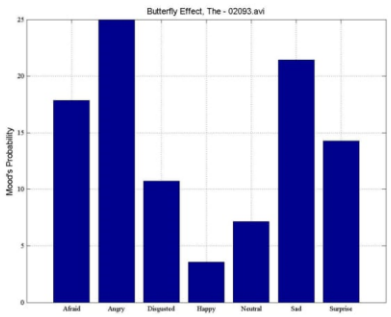


(d) Detected face 26

Figure 9: FER, As Good As It Gets - 01766.avi, Action: Kiss



(a) FER, As Good As It Gets - 01766.avi, Action: Kiss

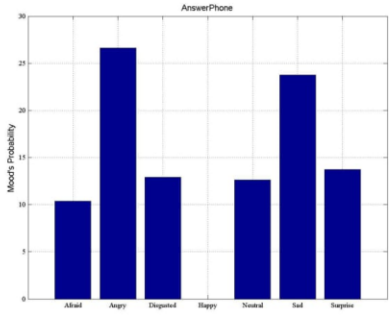


(b) FER, Butterfly Effect, The - 02093.avi, Action: Stand Up

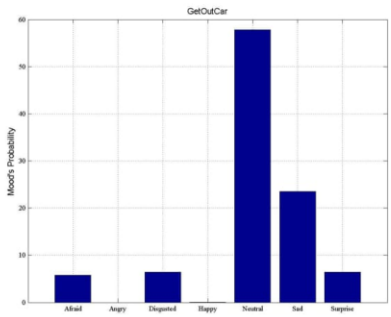
Figure 10: FER, As Good As It Gets - 01766.avi, Action: Kiss

[4] Y. Miyakoshi and S. Kato: Facial emotion detection considering partial occlusion of face using Bayesian Network", 2011 IEEE Symposium on Computers and Informatics (ISCI), 96-101 (2011).

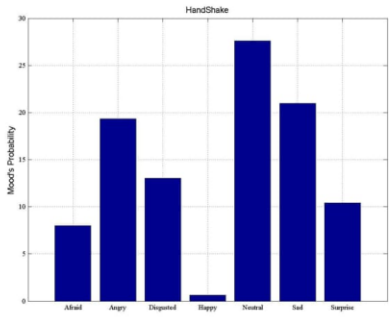
[5] P. Viola and M. Jones: Rapid object detection using a boosted cascade of sim-ple features, Computer Vision and Pattern Recognition, 511-518 (2001).



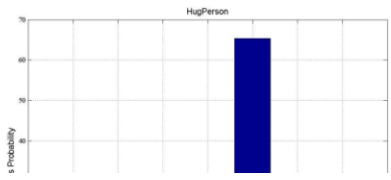
(a) Answer Phone



(b) Get Out Car



(c) Hand Shake



Capítulo 7 Bibliografía

[1] N. M. Oliver, B. Rosario and A. P. Pentland, "A Bayesian computer vision system for modelan human interactions," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 831-843, Aug 2000.

[2] CM Bishop - 2006, Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn, Springer, New York, 2007.

[3] B. Fasel and J. Luettin, Automatic facial expression analysis: A survey, Pattern Recognition (2003) 259-275.

[4] Kotsia, I. Buciu, and I. Pitas, An analysis of facial expression recognition under partial face image occlusion, Image Visual Computing 26 (2008), 1052-1067.

[5] Y. Miyakoshi and S. Kato, Facial emotion detection considering partial occlusion of face using Bayesian Network, IEEE Symposium on Computers and Informatics (2011), 96-101.

[6] Tae-Kyun Kim; Kittler, J., Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image, IEEE Transactions on Pattern Analysis and Machine Intelligence 27, (2005), pp.318-327.

[7] Alonso V.E., Enríquez-Caldera R., Sucar L.E. (2017) A Two-Directional Two-Dimensional PCA Correlation Filter in the Phase only Spectrum for Face Recognition in Video. In: Nasrollahi K. et al. (eds) Video Analytics. Face and Facial Expression Recognition and Audience Measurement. FFER 2016, VAAM 2016. Lecture Notes in Computer Science, vol 10165. Springer, Cham.

[8] Reddy B., Kim YH., Yun S., Jang J., Hong S. (2017) End to End Deep Learning for Single Step Real-Time Facial Expression Recognition. In: Nasrollahi K. et al. (eds) Video Analytics. Face and Facial Expression Recognition and Audience Measurement. FFER 2016, VAAM 2016. Lecture Notes in Computer Science, vol 10165. Springer, Cham.

[9] Open Source Computer Vision Library (OpenCV), http://sourceforge.net/ projects/opencvlibrary/.

[10] Open Source Computer Vision Library Wiki, http://opencvlibrary .sourceforge.net/.

[11] OpenCV discussion group on Yahoo, http://groups.yahoo .com/group/OpenCV.

[12] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, Proceedings of the IEEE Computer Society Conference (2001), I- 511,I-518.

[13] Contreras Flores, ARTNATOMY,www.artnatomia.net,Victoria, SPAIN (2005).

[14] H.A. Rowley. Neural Network-Based Face Detection. PhD thesis, Carnegie Mellon University, 1999.

[15] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, 2nd Quarter 1998.

[16] M. Turk and A. Pentland. Face recognition using eigenfaces. In Proceeding of IEEE Computer Vision and Pattern Recognition, pages 586–590, Maui, Hawaii, 1991.

[17] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In IEEE Intl. Conf. on Computer Vision and Pattern Recognition, CVPR 2001, pages 12–14, Kauai, Hawaii, 2001.

[18] Klette, R. (2014). Concise computer vision: An introduction into theory and algorithms. Springer, 375- 413.

[19] Grauman, K., & Leibe, B. (2011). Visual object recognition. San Rafael, Calif.: Morgan & Claypool.

[20] Mallat SG. (1989). A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (7), 674–693.

[21] Freund, Y., & Schapire, R. (1999). A short Introduction to Boosting.Journal of Japanese Society for Artificial Intelligence, 14(5), 771-780.

[22] Viola, P., & Jones, M. (2001). Robust real-time face detection. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV.

[23] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(1):103–108, 1990.

[24] R. Hietmeyer. Biometric identification promises fast and secure processing of airline passengers. The International Civil Aviation Organization Journal, 55(9):10–11, 2000.

[25] International Biometric Group. Homepage. URL: http://www.biometricgroup.com/.

[26] P. Isasi Viñuela and I. M. Galván León, Redes de neuronas artificiales: un enfoque práctico. Monographs and Textbooks on Probability and Mathematical Statistics, Madrid: Prentice Hall, 2004.

[27] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol. 2, pp. 121–167, 1998.

[28] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification (2nd Edition). Wiley-Interscience, November 2000

[29] S. S. Member and R. J. Fellow, "Similarity measures" 1999.

[30] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Third Edition. Academic Press,2006.

[31] Darío Maravall Gómez – Allende "Reconocimiento de Formas y Visión Artificial" Assison-Wesley Iberoamericana. 1994.

[32] O. Chapelle, B. Sch¨olkopf, and A. Zien, eds.,Semi-Supervised Learning.Cambridge, MA: MIT Press, 2006.

[33] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in COLT' 98: Proceedings of the eleventh annual conference on Computational learningtheory, (New York, NY, USA), pp. 92–100, ACM Press, 1998.

[34] T. M. Mitchell, "The role of unlabeled data in supervised learning," in In Proceedings of the Sixth International Colloquium on Cognitive Science, 1999.

[35] J. Haitian Clustering Algorithms. ISBN 0-471-35645-X, 1975.

[36] M. Haitian, J. y Wong. Algorithm as139: A K-means clustering Algorithm. Ampliad Statistics, Vol. 28, pp100-108,1979.

[37] A. N. Langville and C. D. Meyer, Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, July 2006.

[38] Altrock, C von, Krause, B. Fuzzy logic and neurofuzzy technologies in embedded automotive applications. Fuzzy Logic´93, pp.A113-9, San Francisco California, 1993.

[39] Mandami, E. H., Odtengaard, J. J., Lembessis, E. Use of fuzzy logic for implementing rule-based control of industrial processes. Advances in Fuzzy Sets, Possibility Theory an Applications Plenum Press, 1983.

[40] Eichfeld, H., Künemund, T. A fuzzy controller chip for complex real-time application. 5th IFSA World Congress, pp. 1390-3, 1993

[41] Martín del Brío, B., Sanz Molina, A. Redes neuronales y sistemas borrosos. 3a edición. Ra-Ma.

[42] Pérez Pueyo, Rosanna Descripción General de las Técnicas de Lógica Difusa, cap.2

[43] The Karolinska, Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.

[44] I. Laptev, M. Marsza, C. Schmid, B. Rozenfeld, Learning Realistic Human Actions from Movies, IEEE Conference on Computer Vision & Pattern Recognition, 2008.

[45] Color Space Transformations Philippe Colantoni and Al 2004.

[46]Face Detection and Facial Feature Localization for Human-machine Interface. Md. Al-Amin BHUIYAN National Institute of Informatics . 2003

[46] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[47] van den Boomgard, R, and R. van Balen, "Methods for Fast Morphological Image Transforms Using Bitmapped Images," Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing, Vol. 54, Number 3, pp. 254-258, May 1992.

[48] G. Garcıa Mateos and S. Fructuoso Munoz. A perceptual interface using integral projections. In 7th Intl. Conf. on Pattern Recognition and Image Analysis (PRIA), San Petersburgo, Rusia, 2004.

[49] Md. Al-Amin Bhuiyan Face, Detection and Facial Feature Localization for Human-machine Interface. National Institute of Informatics, (2003).

[50] Abdi. H., & Williams, L.J. (2010). "Principal component analysis" (PDF). Wiley Interdisciplinary Reviews: Computational Statistics. 2 (4): 433–459.

[51] A. Hernandez-Matamoros, E. Escamilla-Hernandez, K. Perez-Daniel M. Nakano-Miyatake, H. Perez-Meana, A supervised classifier scheme based on clustering algorithms, IEEE Central America and Panama Convention (CONCAPAN XXXIV), (2014) , 12-14.

[52] Andres Hernandez-Matamoros, Andrea Bonarini, Enrique Escamilla-Hernandez, Mariko Nakano-Miyatake, Hector Perez-Meana, Facial expression recognition with automatic segmentation of face regions using a fuzzy based classification approach, Knowledge-Based Systems, Volume 110, 2016, Pages 1-14, ISSN 0950-7051, http://dx.doi.org/10.1016/j.knosys.2016.07.011.

[53] M. Pizer, Adaptive histogram equalization and its variations, computer vision, Graph. Image Process. 39 (1987) 355–368.

[54]Fernandez, L. A., Diaz, D., & Depaoli, R. (2005), Optimizacion de la ecualizacion del histograma en el procesamiento de imagenes digitales. In VII Workshop de Investigadores en Ciencias de la Computacion.

[55]HistogramEqualization,http://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf.

[56] Fogel, I.; Sagi, D. (1989). "Gabor filters as texture discriminator". Biological Cybernetics. 61(2). doi:10.1007/BF00204594. ISSN 0340-1200.

[57] Benitez-Garcia, G;Sanchez-Perez, G; Perez-Meana, H;Takahashi K; Kaneko M, "Facial expression Recognition Based on Facial Region Segmentation and Modal Value Approach" IEICE TRANS. INF. &SYST., VOL.E97-D, NO.4 April 2014.

[58] Ligang Zhang, DianTjondronegoro, VinodChandran , " Random Gabor based templates for facial expression recognition in images with facial occlusion " Elsevier Neurocomputing 145 451–464,(2014).Hasimah Ali, Muthusamy Hariharan, Sazali Yaacob, Abdul Hamid Adom, " Expert Systems with Applications", Elsevier Expert Systems with Applications 42, 1261–1277, (2015). Russell, S. and Norvig, P.

[59] H. Ali, M. Hariharan, S. Yaacob, A. Hamid-Adom, Facial recognition using em- pirical mode decomposition, Expert Syst. Appl. 42 (3) (2015) 1261–1277.

[60] Z. Wang, Q Rao, Facial expression based on orthogonal local Fisher discrimi- nant analysis, in: Proceedings of International Conference on Signal Processing (ICSP), 2010, pp. 1358–1361.

[61] K. Buciu, I. Pitas, An analysis of facial expression recognition under partial face image occlusion, Image Vision Comput. 26 (7) (2008) 1052–1067.