

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación



Análisis de Expresiones Faciales para la Detección de
Estados de Ánimo

Tesis para obtener el título de:

Maestro en Ciencias de la Computación

Presenta:

Jesús García Ramírez

Directores de tesis:

Dr. Ivan Olmos Pineda

Dr. José Arturo Olvera López

Septiembre 2017

RESUMEN

El procesamiento de imágenes digitales junto con la inteligencia artificial son áreas de interés por la comunidad científica, comúnmente sus aplicaciones son en áreas diferentes a la computación (por ejemplo, la medicina o psicología). Las expresiones faciales proveen mucha información sobre los estados de ánimo que demuestra una persona, por la razón anterior, la visión por computadora puede ser utilizada para detectar el estado de ánimo que demuestra una persona en una imagen.

En este trabajo de tesis se presenta una metodología para la detección de estados de ánimo a partir de imágenes digitales, que consta de tres etapas: pre-procesamiento, extracción de características y clasificación.

En la etapa de pre-procesamiento se utilizaron técnicas de detección de bordes y umbralización para segmentar las regiones de la boca y de las cejas, las cuales aportan información relevante sobre los estados de ánimo de las personas, adicionalmente se presenta una comparación con otro método de segmentación de boca en la que se muestra que nuestro enfoque es competitivo al respecto.

La etapa de extracción de características consiste en obtener atributos descriptivos de las regiones segmentadas, por lo regular atributos numéricos. Para este enfoque se utilizó el método de interpolación de diferencias divididas de Newton para obtener coeficientes y puntos que describan las regiones, estas características no se han reportado en la literatura, por lo que este es una forma novedosa de caracterizar estas regiones, las características son evaluadas en la etapa de clasificación.

En la etapa de clasificación se utilizaron dos conjuntos de características una con los coeficientes de interpolación y otro con puntos encontrados a partir de esos coeficientes. Se realizaron pruebas con diferentes clasificadores (Redes Neuronales, Máquinas de Vectores Soporte y Clasificador Simple de Bayes). En general, el conjunto de puntos tuvo mejor desempeño respecto a la clasificación, mientras que las redes neuronales encontraron una mejor precisión.

En general, el enfoque utilizado en todas las etapas tiene un buen desempeño en cuanto a la segmentación de ROIs y clasificación, en conjunto puede ser utilizado para la detección de estados de ánimo a partir de imágenes digitales.

El tiempo de ejecución es la principal desventaja durante el proceso de segmentación, ya que dicho tiempo es proporcional al tamaño de imágenes a procesar.

ABSTRACT

Digital image processing and artificial intelligence are interesting areas for computer research community, commonly its applications are in different areas (such as medicine or psychology). Facial expressions, provide us much information of the emotion that a person shows, for the last reason, computer vision can be used in order to detect the emotion of a person based on digital images.

In this thesis a methodology for detecting emotions based on digital images is presented, the full methodology consist of three stages: pre-processing, feature extraction and classification.

In pre-processing stage edge detection approaches and threshold techniques are used for segmenting mouth and eyebrows regions of interest, these regions provide relevant information about the emotion of a person, additionally a comparison against other method is presented.

Feature extraction stage consist in obtain descriptive information using the segmented regions, regularly numeric values are used. In this approach the divided differences method is used in order to obtain coefficients and points to describe the regions, these sets of features are not reported in previous work, so these features are a novel method to depict the ROIs. This features are evaluated in classification process.

Two features sets are used in classification stage, the first one contains the interpolated coefficients, the second one contains extracted point with the coefficients. Different classifiers are tested with the features sets as input (Neural Networks, Support Vector Machines and Naive Bayes). In general, the points set have better performance with the tested classifiers, on the other hand Neural Networks find the best accuracy.

In general, this approach have a good performance in pre-processing and classification stages, and it can be used to find the emotion of the person in a digital image.

The runtimes are the main disadvantage in the pre-processing stage, and it depends on the image dimension. The higher dimension of the image, the higher runtime is spent for processing the image..

AGRADECIMIENTOS

A Consejo Nacional de Ciencia y Tecnología por apoyar económicamente este proyecto de tesis durante dos años.

A la Benemérita Universidad Autónoma de Puebla, en particular a la Facultad de Ciencias de la Computación que me abrió las puertas para realizar mis estudios de maestría.

A todos los amigos que encontré en la facultad quienes compartimos un sueño común.

A todos los profesores que me impartieron clases durante este proceso, así como al comité tutorial por sus revisiones y consejos durante las evaluaciones.

A mis directores de tesis el Dr. José Arturo Olvera López y el Dr. Ivan Olmos Pineda, por brindarme la confianza para el desarrollo de este proyecto de tesis, gracias por los consejos, enseñanzas y apoyo durante este proyecto.

A mi esposa que me acompaña en este sueño, gracias por estar a mi lado en todo momento, sin tí nunca hubiera alcanzado esta meta, te amo.

A mis padres que me han apoyado siempre, gracias por sus enseñanzas, por estar conmigo en todos los momentos que los he necesitado, por ayudarme a conseguir mi sueño. Los amo y les debo todo lo que soy.

CONTENIDO

1	Introducción	1
1.1	Motivación	2
1.2	Planteamiento del problema	3
1.3	Objetivos	4
1.3.1	Objetivo general	4
1.3.2	Objetivos específicos	4
1.4	Trabajos relacionados	5
1.5	Organización del documento de tesis	8
2	Marco teórico	9
2.1	Procesamiento de imágenes digitales	10
2.1.1	Histograma de una imagen	11
2.1.2	Operadores puntuales	12
2.1.3	Convolución y operadores regionales	12
2.2	Reconocimiento de patrones	13
2.2.1	Clasificación	14
2.2.2	Redes neuronales artificiales	15
2.2.3	Máquinas de vectores de soporte	18
2.2.4	Clasificador simple de Bayes	21
2.3	Herramientas de programación paralela	23
2.3.1	Message Passing Interface (MPI)	23
2.3.2	Hilos de ejecución	23
2.4	Diferencias divididas de Newton	24
3	Metodología para detectar estados de ánimo	25
3.1	Etapas de pre-procesamiento	26
3.1.1	Reducción del espacio de búsqueda	26
3.1.2	Detección de bordes	27
3.1.3	Algoritmo basado en densidad para eliminar ruido	29
3.1.4	Detección de bordes en cejas	30
3.1.5	Detección de bordes en boca	32
3.1.6	Paralelización	34
3.2	Etapas de extracción de características	34
3.2.1	Extracción de características en cejas	34
3.2.2	Extracción de características en boca	36

3.3	Etapa de clasificación	37
4	Resultados	41
4.1	Etapa de pre-procesamiento	42
4.1.1	Reducción de espacio de búsqueda	42
4.1.2	Pre-procesamiento en cejas	42
4.1.3	Pre-procesamiento en boca	44
4.2	Clasificación	46
5	Conclusiones y trabajo futuro	53

LISTA DE FIGURAS

1.1	Enfoque comúnmente usado para la detección de expresiones faciales, a) imágenes de entrada tomadas de [2], b) fase de pre-procesamiento, c) extracción de características, d) modelo creado por el clasificador con la información de c).	4
2.1	Imagen en escala de grises junto a su histograma normalizado.	11
2.2	Imagen a color en formato RGB con sus histogramas normalizados.	12
2.3	Proceso de clasificación.	14
2.4	Representación de un perceptrón.	15
2.5	Espacio de hipótesis en el que se buscan los mejores pesos w_0, w_1, \dots, w_n que conducen a un menor error $E(x)$.	16
2.6	Unidad sigmoide.	17
2.7	Ejemplo de Red Neuronal con arquitectura feedforward.	18
2.8	Conjunto de entrenamiento en 2D linealmente separable.	20
2.9	Margen grande y pequeño para un conjunto de entrenamiento en 2D linealmente separable.	20
2.10	Arquitectura paralela con MPI.	23
3.1	Imagen en RGB con sus canales, a) Imagen en modelo de color RGB, imágenes en escala de grises con los valores del canal B b), canal G c), canal R d).	28
3.2	Modificación a la región a la que se aplicará la convolución, a) Región con $u = 1$, b) Región con $u = 2$.	29
3.3	Procesamiento de aplicado en cada núcleo de procesamiento.	35
3.4	Ejemplo de interpolación con polinomio cúbico para las cejas.	35
3.5	Procesamiento de las características para la región de las cejas.	36
3.6	Ejemplo de interpolación con dos polinomios cuadráticos para la boca.	37
3.7	Procesamiento de las características para la región de la boca.	37
3.8	Procesamiento paralelo de la etapa de clasificación.	39
4.1	Ejemplo de imágenes resultantes de aplicar el algoritmo de Viola & Jones.	43
4.2	Pre-procesamiento de la ceja: a) Imagen de entrada (Canal R), b) Función tangente hiperbólica, c) Operación de cierre, d) Algoritmo Otsu, e) detección del cluster.	43

4.3	Pre-procesamiento de la boca: a) Filtro gradiente, b) Función seno, c) Quitar ruido menor a la media, d) Algoritmo basado en densidad, e) Función XOR.	45
4.4	Ejemplo de ensamble de las regiones de interés pre-procesadas. . . .	46
4.5	Comparativa contra los resultados reportados en [38] (marcados con "+") y nuestros resultados (marcados con "o").	47
4.6	Gráfica de dispersión para Naïve Bayes con CP.	48
4.7	Gráfica de dispersión para Naïve Bayes CC.	48
4.8	Gráfica de dispersión para SVM con kernel polinomial y CP.	49
4.9	Gráfica de dispersión para SVM con kernel RBF y CP.	49
4.10	Gráfica de dispersión para SVM con kernel polinomial y CC.	50
4.11	Gráfica de dispersión para SVM con kernel RBF y CC.	50
4.12	Gráfica de dispersión para NN con CP con los parámetros de la primera fila de la Tabla 4.6.	51
4.13	Gráfica de dispersión para NN con CP con los parámetros de la segunda fila de la Tabla 4.6.	51
4.14	Gráfica de dispersión para NN con CC con los parámetros de la primera fila de la Tabla 4.6.	52
4.15	Gráfica de dispersión para NN con CC con los parámetros de la segunda fila de la Tabla 4.6.	52

LISTA DE TABLAS

3.1	Puntos para encontrar las ROIs en imágenes de rostros.	28
4.1	Resultados de los experimentos de la implementación del algoritmo Viola & Jones.	42
4.2	Resultados de los experimentos del procesamiento de imágenes de cejas.	44
4.3	Media y desviación estándar para las diferencias de los coeficientes del polinomio de la región de las cejas.	44
4.4	Resultados de los experimentos del procesamiento de imágenes de cejas.	45
4.5	Media y desviación estándar para las diferencias de los coeficientes de los polinomios de la región de la boca.	46
4.6	Resultados de los experimentos con CP y redes neuronales.	51
4.7	Resultados de los experimentos CC y redes neuronales.	52

LISTA DE ALGORITMOS

1	Algoritmo de entrenamiento retro programación.	19
2	Detección de bordes.	30
3	Detectar regiones densas en una imagen binaria.	31
4	Detección de la región de las cejas.	33
5	Detección de bordes para la región de la boca.	34

CAPÍTULO 1

INTRODUCCIÓN

CONTENIDO		
1.1	Motivación	2
1.2	Planteamiento del problema	3
1.3	Objetivos	4
1.3.1	Objetivo general	4
1.3.2	Objetivos específicos	4
1.4	Trabajos relacionados	5
1.5	Organización del documento de tesis	8

En este capítulo se presentan las siguientes secciones referentes a este proyecto de tesis: inicialmente se presenta la introducción, en la que se describe la motivación, posteriormente se introduce la descripción del problema a resolver, el objetivo general y los objetivos específicos, así como los trabajos relacionados al reconocimiento de expresiones faciales.

1.1 MOTIVACIÓN

En años recientes, se han desarrollado diferentes aplicaciones sobre el procesamiento de información, en especial, aquellas que respectan al procesamiento de imágenes digitales, tales como el reconocimiento de expresiones faciales (Facial Expression Recognition, FER). Algunas aplicaciones de los sistemas de FER son la detección de desórdenes mentales, determinar si una persona está mintiendo en una entrevista, reconocimiento de emociones, entre otras [7].

Por lo regular un sistema de reconocimiento de expresiones trabaja con imágenes digitales, en el cual se aplica un procesamiento de las mismas y comúnmente se realiza la segmentación de regiones de interés (Regions Of Interest, ROIs), que contienen información relevante para las expresiones faciales que posteriormente es analizada. Las ROIs que se extraen comúnmente para el FER respectan a: las cejas, ojos, nariz, boca y mejillas, éstas son regiones descriptivas del rostro humano y pueden ser utilizadas para determinar la expresión demostrada por una persona.

Un sistema de FER está constituido por las etapas que se enuncian a continuación: se comienza con el pre-procesamiento de imágenes faciales, entre los principales filtros que se aplican en esta fase se encuentran: suavizado, cambio de escala, conversión de imágenes a escala de grises, entre otras. La siguiente etapa es la extracción de características; primero se segmentan las ROIs de la imagen usando técnicas para eliminar el fondo y segmentar el rostro en ella. Finalmente, se encuentran características importantes para el análisis de expresiones faciales. Comúnmente las imágenes utilizadas en el FER se representan de acuerdo al modelo de color RGB (Red, Green, Blue). Sin embargo, se puede realizar una transformación a otro modelo de color, ya que éstos pueden proporcionar información sobre bordes, texturas o luminosidad, diferente a la proporcionada por el modelo RGB. Por esta razón es importante analizar imágenes en modelos como HSV o YUV. En la siguiente etapa, se utiliza la segmentación realizada en la fase anterior para encontrar valores característicos (por lo regular atributos de tipo numérico) de las ROIs. Los valores obtenidos son utilizados como entrada de clasificadores de tipo supervisado para encontrar un modelo que permita realizar una predicción de los estados de ánimo.

En el procesamiento de imágenes digitales cada pixel es tomado en cuenta para realizar una modificación en su valor de intensidad, dependiendo del tamaño de la imagen, su procesamiento puede ser computacionalmente costoso, en especial si se procesa vídeo, ya que se tienen que aplicar dicho procesamiento a un gran número de imágenes por segundo. Por ejemplo, existen dispositivos para la captura de vídeo que almacenan hasta 1500fps (Frames per Second, imágenes por segundo), por lo tanto a mayor número de imágenes captados por el dispositivo, el tiempo de procesamiento incrementa y se requieren más recursos computacionales.

Una alternativa para implementar soluciones a problemas computacionalmente costosos son los enfoques paralelos. Algunos de éstos, como la programación multicores o las implementaciones con la interfaz de paso de mensajes (MPI) son de los más usados, en los últimos años el uso de Unidades de Procesamiento Gráfico (Graphic Processing Units, GPU) ha sido muy notorio, ya que, es una alternativa

de bajo costo para implementaciones paralelas.

En este trabajo de tesis se presenta una metodología para la detección de estados de ánimo asociados a expresiones faciales. En el que se implementan las etapas de pre-procesamiento de imágenes de rostros, encontrando información sobre los bordes de las regiones como cejas y boca; en la siguiente etapa se extraen datos característicos de las ROIs tomando en cuenta la información del pre-procesamiento; finalmente, se presentan las pruebas realizadas con distintos clasificadores para encontrar un modelo con el que se realice una clasificación con una precisión aceptable.

1.2 PLANTEAMIENTO DEL PROBLEMA

La interacción humano-computadora tiene como principal objetivo que ésta se realice de manera natural entre un humano y un sistema computacional híbrido (Hardware y Software), es decir, que la interacción entre ellos sea como la comunicación que existe entre los seres humanos. La comunicación entre los seres humanos puede ser: hablada, escrita o mediante gestos. Sin embargo, existe otra forma para comunicarse, la cual aporta información sobre los estados de ánimo expresados por una persona al interactuar con otra, esta forma de comunicación son las expresiones faciales. De acuerdo a [1], las expresiones faciales aportan el 55% de la información que una persona quiere comunicar, mientras que el 38% es aportado por la entonación de voz y el 7% por las palabras expresadas.

La comunicación entre seres humanos tiene la peculiaridad de que se expresan emociones o estados de ánimo durante una conversación. Por otro lado, las computadoras no tienen la cualidad de expresar ni entender emociones, por esta razón la interacción natural entre un sistema computacional y una persona es difícil, computacionalmente hablando.

La interacción entre el humano y el sistema puede llevarse a cabo mediante un dispositivo de captura de vídeo, después el sistema se encarga de clasificar qué estado de ánimo demuestra el individuo en una imagen determinada. Dependiendo del tamaño de la imagen y los fotogramas que puede capturar el dispositivo (por segundo), el procesamiento es mayor. Por lo que el procesamiento paralelo puede ser útil para reducir el tiempo de ejecución, además de que el balance de trabajo se puede distribuir en los núcleos de una red de computadoras mediante MPI.

El método utilizado por diferentes autores para la detección de expresiones faciales tiene como entrada una imagen o un conjunto de imágenes (vídeo) las cuales son sometidas a un pre-procesamiento en el cual se aplican filtros para la detección de bordes y distintos métodos de umbralización para su posterior binarización. Este pre-procesamiento se realiza para encontrar información relevante del rostro de un individuo o de sus ROIs.

La siguiente fase es la extracción de características, donde se codifica la imagen para guardar valores numéricos o información relevante del rostro. Finalmente, el sistema usa un modelo, previamente construido por un clasificador que tiene como entrada los valores codificados para determinar el estado de ánimo mostrado por el individuo en la imagen. El proceso anteriormente descrito se puede observar en la Figura 1.1.

En este trabajo de tesis se presenta un nuevo método de segmentación de ROIs en rostros humanos, basados en filtros de umbralización y detección de bordes, así mismo se proponen atributos numéricos diferentes a los presentados en la literatura, para ser posteriormente evaluadas con diferentes algoritmos de clasificación supervisada.

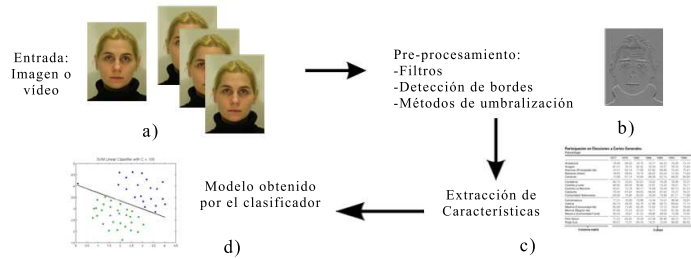


Figura 1.1: Enfoque comúnmente usado para la detección de expresiones faciales, a) imágenes de entrada tomadas de [2], b) fase de pre-procesamiento, c) extracción de características, d) modelo creado por el clasificador con la información de c).

1.3 OBJETIVOS

En las siguientes secciones se presenta el objetivo general y los objetivos específicos de este trabajo de tesis.

1.3.1 OBJETIVO GENERAL

Proponer una metodología para la solución del problema de detección de estados de ánimo mediante el análisis de expresiones faciales utilizando técnicas de programación paralela en las etapas de pre-procesamiento y clasificación para el análisis de regiones descriptivas de expresiones faciales a partir de imágenes digitales en personas adultas.

1.3.2 OBJETIVOS ESPECÍFICOS

Los objetivos específicos de este trabajo de tesis se presentan a continuación:

1. Analizar los trabajos existentes sobre el tema de detección de expresiones faciales, en cada una de las etapas del sistema (pre-procesamiento, extracción de características y clasificación) y determinar cuáles ofrecen los mejores resultados.
2. Analizar los métodos fundamentales para el pre-procesamiento de imágenes de rostros, para identificar los que reporten mejores resultados.
3. Analizar la información de imágenes de rostros en modelos de color diferentes al RGB para determinar qué información proporcionan, así como los canales RGB por separado y evaluar su desempeño.

4. Determinar los componentes no holísticos del enfoque propuesto que se pueden paralelizar mediante plataformas basadas en arquitecturas paralelas para reducir los tiempos de ejecución.
5. Utilizar técnicas de extracción de características para obtener los atributos descriptivos de expresiones faciales y construir bases de entrenamiento para la extracción de patrones para detectar estados de ánimo.
6. Realizar pruebas con diferentes clasificadores para determinar cuál tiene mejor precisión respecto a la detección del estado de ánimo mostrado por un individuo en una imagen determinada.
7. Realizar pruebas del sistema completo y determinar los ajustes pertinentes, así las mejoras para que se tenga un mejor desempeño en cuanto la precisión en la etapa de clasificación.

1.4 TRABAJOS RELACIONADOS

En los últimos años el procesamiento de imágenes digitales, junto con la inteligencia artificial se combinaron dando como resultado una nueva área de investigación, la cual se conoce como visión artificial por computadora, este trabajo de tesis respecta a dicha área de la computación.

El análisis de expresiones faciales ha sido un tema que ha tenido gran auge en los últimos años, existen distintas revisiones que ayudan a conocer cómo se ha abordado este tema [3, 4, 5], basados en las revisiones antes mencionadas y en búsquedas de trabajos relacionados al análisis de expresiones faciales, en esta sección se presentan distintos enfoques utilizados sobre este tema.

Desde 1994 se comenzó a realizar investigación sobre el análisis de expresiones faciales, inicialmente Paul Ekman presentó un enfoque totalmente psicológico, teniendo como resultado dos grandes aportes, los cuales siguen vigentes actualmente, dichos aportes son descritos a continuación: se establecieron las seis expresiones universales (alegría, enojo, tristeza, desagrado, sorpresa y miedo), dichas expresiones, por lo regular son clasificadas en los distintos enfoques propuestos, algunos trabajos añaden un estado neutral [6]; el segundo aporte de Paul Ekman es el Sistema de Codificación para Expresiones Faciales (Facial Action Coding System, FACS), este sistema describe distintos movimientos faciales con una ventana de tiempo, los cuales son relacionados a expresiones faciales. Estos movimientos son llamados Unidades de Acción (Action Units, AU), que son ponderados para determinar que tanto contribuyen a la expresión demostrada, en el FACS se especifican: 9 AU para la parte posterior del rostro, 18 para la parte inferior, adicionalmente se consideran 14 movimientos de la cabeza, 9 posiciones de los ojos, 9 descriptores de acción, 9 tipos de comportamiento y 5 acciones adicionales [7].

Como se muestra en la Figura 1.1 la primera etapa de los sistemas de FER es el pre-procesamiento de dichas imágenes. Sin embargo, antes de aplicarlo se puede realizar un cambio de modelo de color, los más usados son: HSV, YUV o $L^*a^*b^*$. Torres et al. [8] presentan un enfoque en el que realizan transformaciones a otros

modelos de color (concretamente HSV y YUV), después de aplicar estas transformaciones, se separan los canales y se obtiene una imagen en escala de grises por cada uno. Las imágenes resultantes son analizadas para determinar cuales dan más información sobre los rostros. En la siguiente etapa se aplicó un análisis de componentes principales (Principal Component Analysis, PCA) y se encontraron las eigenfaces de cada imagen en el conjunto de entrenamiento, las cuales son componentes que describen a una imagen para no analizar toda pixel a pixel. Para la clasificación se utilizó una función de distancia, donde se comparó la eigenface de una nueva imagen y cada uno de los componentes del conjunto de entrenamiento. Como resultado se encontró que los canales del formato de color YUV y los canales de SV del formato HSV tuvieron un mejor desempeño con 88.14% de efectividad.

Xiaoming et al. [1] presentan un trabajo en el que se utilizan diferentes modelos de color entre los cuales se destacan YCbCr y el formato CIE L*a*b*, a las imágenes resultantes se le aplicaron cuatro métodos de detección de bordes (Canny, Laplace, Sobel y Robert), el detector Robert es el que tuvo mejor desempeño. De esta manera se puede concluir que la detección de bordes puede ser una buena opción para el pre-procesamiento de las imágenes.

Bourel et al. [9] muestran un enfoque distinto en el que busca puntos de interés en el rostro que aparece en una imagen, como lo son las ojos, nariz, boca y cejas. Una vez que se encuentran estos puntos de interés en la imagen se buscan distancias entre algunos de ellos y se asigna uno de tres posibles estados (incremental, decremental y estable), los últimos datos son guardados en forma de vector, para después ser utilizados como ejemplos de entrenamiento. Finalmente, en la etapa de clasificación se utilizó el clasificador vecinos mas cercanos con $k=6$, teniendo como variación que algunos de los atributos tienen mayor peso que otros.

También existen enfoques en los que modelos 3D son utilizados para la detección de expresiones faciales. Un ejemplo de éstos lo muestra Huiquin et al. [10] en el que se combina encontrar Patrones Locales Binarios (Local Binary Pattern, LBP) con un pseudo modelo 3D del rostro, en este enfoque se divide el rostro en seis regiones. Se extraen características de cada región en la que se dividió el rostro y se utiliza una función de similitud para determinar la expresión que muestra la persona en una imagen. Mientras que Essa et al. [11] también utilizan modelos 3D, sin embargo, ellos se basan en encontrar los movimiento que una persona realiza en un ventana de tiempo, tomando en cuenta una serie de imágenes y así determinar el estado de ánimo de la persona.

Las principales Regiones de Interés (Regions Of Interest, ROIs) que se encuentran en imágenes de rostros son: ojos, nariz, boca y cejas, con base en estas regiones se extraen distintos datos, que por lo regular son distancias entre estos puntos, o son analizadas en el espacio de frecuencias, posteriormente se extrae información referente a las ROIs, dicha etapa se conoce como extracción de características. Según Sariyand et al. [12] existen tres formas de guardar los datos de las imágenes procesadas: se guarda el rostro completo (visto como una sola entidad o se guardan regiones completas del mismo), un ejemplo de cómo se registra el rostro de manera completa lo muestra Wenfei et al. [13], en este trabajo se pre-procesa una imagen de rostro de manera que se tenga como resultado una imagen solo con el rostro de la persona,

ésta es segmentada en partes iguales, dejando un número de regiones determinada, después a cada region se le aplican filtros de Gabor, a las nuevas regiones se les aplica reducción de dimensionalidad mediante PCA. Finalmente, se aplican discriminantes lineales de Fisher y se guarda un valor por cada región procesada en forma de vector. Para la clasificación se utilizó el clasificador vecinos más cercanos.

Se han desarrollado enfoques 3D como se muestra en [14], en este enfoque se buscan veinticuatro características de rostros en imágenes 3D, para la clasificación se realiza con ensambles con tres clasificadores (Naïve Bayes, LDA y Vecinos más cercanos) y con Adaboost. Otro trabajo que también utiliza imágenes en 3D consiste en registrar un histograma LBP de la imagen completa, quedando solo un vector para cada imagen, para la clasificación se utilizó máquinas de vectores de soporte (Support Vector Machines, SVM) [15].

Mientras que para el registro de manera local, la información esta relacionado con Motion Units (MU), las cuales son movimientos simples de los músculos del rostro, referentes a las ROIs que se mencionaron anteriormente [16].

Muchos de los clasificadores han sido aplicados para el reconocimiento de expresiones faciales, algunos de ellos ya se mencionaron en los párrafos anteriores, además de los clasificadores ya mencionados se han utilizado algunos otros, por ejemplo, N. Sebe et al. realiza una comparación entre distintos clasificadores: naïve Bayes, redes Bayesianas, árboles de decisión y vecinos más cercanos, teniendo el mejor desempeño árboles de decisión [17].

Tian Y. et al. [18] presentan un enfoque que se basan en encontrar puntos en el rostro y con base en éstos se encuentran distancias y ángulos de apertura en los ojos y boca, la distancia entre los ojos y algunos otros, estos sirven para determinar las AU del rostro en la imagen, una vez que se obtienen estas características para clasificar se utiliza un modelo extraído con redes neuronales.

Otros clasificadores como SVM se han implementado para esta etapa, por ejemplo, un enfoque en el cual el rostro es dividido en 14 regiones, de las cuales es extraído un histograma LBP por cada una y éstos son guardados para clasificar las expresiones faciales mediante SVM, obteniendo buenos resultados, este enfoque clasifica seis expresiones faciales y además en otros experimentos añade el estado neutral [19].

Actualmente no se encuentran muchos trabajos que resuelvan el problema del reconocimiento de expresiones faciales con algún método paralelo, George Sabu [20] propone un método basado en GPU para la segmentación del rostro, en el cual se da como entrada al GPU una imagen procesada con un detector de bordes, en la tarjeta gráfica se realizan operaciones con una matriz de convolución aplicando un filtro gradiente, una vez que se termina la etapa de pre-procesamiento, se buscan puntos en el rostro de los cuales se extraerán características referentes a las AU adoptando un enfoque piramidal, posteriormente se encuentran líneas entre estos puntos, así como ángulos entre dichas líneas. Finalmente, se realiza la clasificación con SVM y se redujeron considerablemente los tiempos de ejecución comparado contra un enfoque secuencial.

En este trabajo de tesis se presenta un enfoque diferente para la segmentación de ROIs en imágenes de rostros, éste se basa en realizar modificaciones a las técnicas

referentes a la detección de bordes y métodos de umbralización, además de que se utilizan métodos de interpolación para encontrar las características de las ROIs, dichas características son evaluadas mediante algoritmos de clasificación supervisada, en los que se aplica un enfoque paralelo para el ajuste de parámetros.

1.5 ORGANIZACIÓN DEL DOCUMENTO DE TESIS

La tesis esta organizada en cinco capítulos, el resto de los capítulos se organizan de la siguiente manera:

- En el capítulo 2 se presentan los conceptos básicos sobre los temas que respectan a este trabajo de tesis: procesamiento digital de imágenes, reconocimiento de patrones, clasificación supervisada, programación paralela con MPI e hilos de procesamiento.
- En el capítulo 3 se presenta el desarrollo del proyecto, el cual esta dividido en tres secciones principales, una para cada etapa del proyecto: etapa de pre-procesamiento en la que se segmentan las ROIs; etapa de extracción de características en la que se extraen datos de la imagen para ser guardados y posteriormente usarlos en la última etapa; etapa de clasificación en la que se usan los datos de la etapa anterior y se encuentra un modelo para clasificar los estados de ánimo que muestra una persona en una imagen determinada.
- En el capítulo 4 se muestran los resultados de la teoría presentada en el capítulo 3, en la etapa de pre-procesamiento se presentan las imágenes resultantes de aplicar el pre-procesamiento y un análisis de los resultados de la segmentación, para la etapa de extracción de características se explica cómo se extrajeron puntos de las ROIs y en la etapa de clasificación se presentan los mejores resultados para los que se ajustó el modelo de clasificación.

CAPÍTULO 2

MARCO TEÓRICO

CONTENIDO

2.1	Procesamiento de imágenes digitales	10
2.1.1	Histograma de una imagen	11
2.1.2	Operadores puntuales	12
2.1.3	Convolución y operadores regionales	12
2.2	Reconocimiento de patrones	13
2.2.1	Clasificación	14
2.2.2	Redes neuronales artificiales	15
2.2.3	Máquinas de vectores de soporte	18
2.2.4	Clasificador simple de Bayes	21
2.3	Herramientas de programación paralela	23
2.3.1	Message Passing Interface (MPI)	23
2.3.2	Hilos de ejecución	23
2.4	Diferencias divididas de Newton	24

Es importante conocer los fundamentos teóricos de las disciplinas que respectan a este proyecto, algunas de ellas son: el reconocimiento de patrones, la visión por computadora y el procesamiento de imágenes digitales. En este capítulo se presentan los conceptos teóricos que soporta este proyecto de tesis, las primeras secciones tienen que ver con el procesamiento digital de imágenes, la cual es la principal disciplina que se aborda, en las siguientes secciones se presenta la teoría sobre la clasificación y posteriormente conceptos básicos sobre programación paralela con MPI e hilos de procesamiento en Java.

2.1 PROCESAMIENTO DE IMÁGENES DIGITALES

Las imágenes se pueden definir como la figura de algún objeto captada por el ojo o algún dispositivo de captura. Dichas imágenes suelen ser capturadas por aparatos ópticos o placas fotográficas, las cuales en un inicio eran analógicas. Con el paso del tiempo éstas han sido sustituidas por cámaras digitales, las cuales llevan a cabo un proceso llamado digitalización que se describe a continuación.

La digitalización consta en dos procesos, los cuales son: el muestreo, que consiste en realizar una subdivisión de una imagen analógica en porciones iguales (por lo regular cuadradas), éstas representan sensores que son sensibles a la luz; el proceso de cuantificación consiste en asignar un color a las divisiones de la imagen encontradas en el proceso de muestreo, esta asignación puede ser con un valor (por ejemplo, una imagen en escala de grises) o un arreglo de tres colores (imagen en RGB), asignando un valor en cada escala de amplitud.

Una imagen digital se puede definir como una colección de puntos (píxeles) los cuales representan intensidad de color dentro de una matriz, de la cual su dimensión es conocida como tamaño de la imagen, además es común referirse a dicha dimensión por la longitud de ancho y alto.

Las imágenes en escala de grises están representadas por un único canal, el cual puede ser visualizado por la intensidad de color negro que existe en un píxel, dicho valor se encuentra en un rango de 0 a 255 si la imagen es de 8 bits, en el que 0 representa ausencia de color (negro) y 255 representa una intensidad de color completa (blanco), ya que los valores de intensidad de color están en un rango entre 0 y 255 solo es necesario un byte para guardar el valor de un píxel.

Las imágenes en el modelo de color RGB están representadas no solo por un canal sino por tres, los canales representan la intensidad de color verde, azul y rojo, éstos son suficientes para representar los colores visibles por el ojo humano. Existen diferentes modelos de color, el más utilizado para imágenes a color es el modelo de color RGB (Red, Green, Blue), por lo que en vez de tener un solo valor como en las imágenes en escala de grises, en el modelo de color RGB se tienen tres intensidades para cada píxel, cada combinación de los tres colores representa un color diferente. El número de colores que se puede generar con el modelo de color RGB es mayor a dieciséis millones.

Otro tipo de imágenes que suelen ser manipuladas son las imágenes en blanco y negro, éstas son conocidas como imágenes binarias, dichas imágenes solo contienen dos intensidades de color negro o blanco (0 y 1 respectivamente), son comúnmente utilizadas para encontrar distancias o extraer características de las imágenes.

Como se mencionó al inicio, una imagen se puede definir como una matriz de píxeles, matemáticamente puede ser vista como una función bidimensional $f(x, y)$, donde x y y representan la posición de un píxel dentro de la imagen.

El procesamiento de imágenes digitales se refiere a manipular la intensidad de color de los píxeles, los procesos más comunes para la manipulación de imágenes digitales tienen que ver con la implementación de filtros, los cuales ayudan a encontrar información relevante dependiendo el objetivo de la implementación.

Diariamente los humanos utilizamos imágenes digitales, ya que es muy accesible

contar con un dispositivo de captura. Muchas personas tienen una interacción más profunda con las imágenes digitales, como los fotógrafos profesionales, los médicos, arquitectos, entre otros, por lo cual el procesamiento de las mismas es indispensable para ayudar a estos profesionistas a facilitar su trabajo.

En las siguientes secciones se presentan conceptos sobre los principios de procesamiento de imágenes digitales, como lo son los filtros, operadores punto y regionales, detectores de bordes, así como las técnicas de convolución que se utilizarán a lo largo de este proyecto.

2.1.1 HISTOGRAMA DE UNA IMAGEN

El histograma de una imagen describe la frecuencia de los valores de intensidad respecto a su ocurrencia en una imagen. Un histograma h de una imagen I en escala de grises con valores en un rango $[0, K - 1]$, donde K es el máximo valor de un punto en una imagen de 8 bits $K = 2^8 = 256$, cada entrada del histograma puede definirse como una función $h(i)$ = frecuencia en la intensidad i en I [21].

El histograma de una imagen puede guardar datos de magnitud muy grande dependiendo del número de píxeles en la imagen, por ello es necesario normalizar el histograma, para que los valores de las intensidades estén en un rango de 0 a 1, esto sirve para una mejor manipulación de los datos del histograma. Para encontrar el histograma normalizado basta con dividir el valor de la intensidad entre el total de píxeles.

La principal utilidad del histograma de una imagen es determinar en que segmentos de intensidad de color se encuentra la mayoría de píxeles en la imagen, como se muestra en la Figura 2.1, en donde se puede observar una imagen en escala de grises junto a su histograma normalizado.

En las imágenes en formato de color RGB se encuentran tres histogramas diferentes, cada histograma representa la intensidad de color en cada canal. Una imagen en formato RGB y sus tres histogramas normalizados se puede observar en la Figura 2.2.

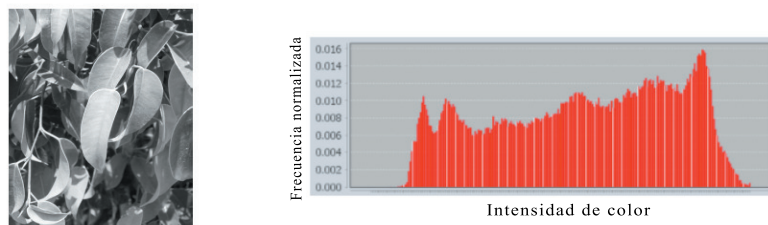


Figura 2.1: Imagen en escala de grises junto a su histograma normalizado.

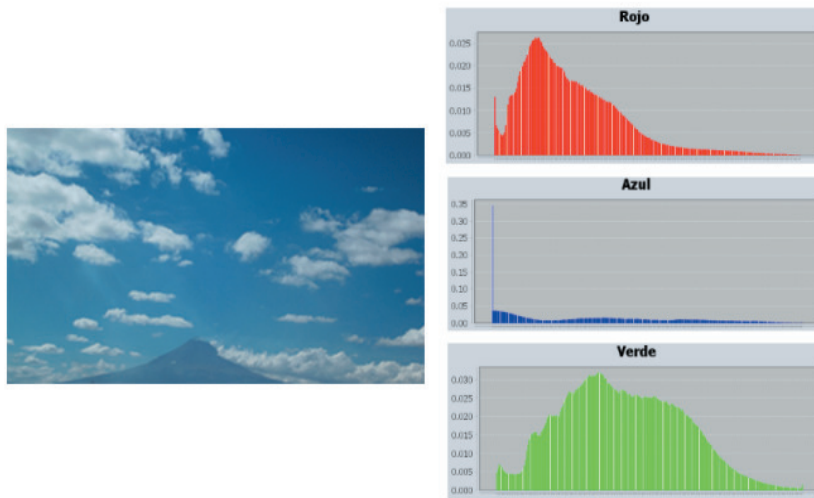


Figura 2.2: Imagen a color en formato RGB con sus histogramas normalizados.

2.1.2 OPERADORES PUNTUALES

Los filtros de imágenes realizan una modificación a la intensidad de color de los píxeles. Los operadores de tipo puntuales realizan una modificación directa al píxel, tomando en cuenta únicamente la intensidad de éste. Un filtro se define como una función $f(x) = I'(x, y)$, siendo $I'(x, y)$ la nueva intensidad del píxel de una imagen I en la posición (x, y) .

Los operadores que se utilizan comúnmente tienen que ver con la modificación en el brillo y el contraste de una imagen; hacer una corrección de color dejando la imagen lista para ser procesada posteriormente mediante técnicas regionales; transformaciones a otros modelos de color que ofrecen información diferente al modelo RGB; aplicar una umbralización para encontrar bordes o información relevante en una imagen.

Una operación muy utilizada en este trabajo de tesis es la umbralización, esta operación consiste en realizar un cambio en la intensidad de color en los píxeles dependiendo de su intensidad inicial con una función $I'(x, y)$, la cual se muestra en la ecuación 2.1,

$$I'(x, y) = \begin{cases} a_0 & \text{si } I(x, y) > \gamma \\ a_1 & \text{si } I(x, y) \leq \gamma \end{cases} \quad (2.1)$$

Donde γ es el umbral para modificar la intensidad de color de un píxel, en la posición (x, y) y a_0, a_1 los nuevos valores de intensidad de la imagen [21].

2.1.3 CONVOLUCIÓN Y OPERADORES REGIONALES

Los filtros regionales toman en cuenta los píxeles vecinos para realizar un cambio en la intensidad del píxel central, la operación se denota con un polinomio y se conoce como convolución, ésta es realizada de manera discreta, la operación se realiza con una matriz de convolución H , comúnmente tiene una dimensión 3×3 , el operador

\otimes (que representa la convolución) realiza una multiplicación de sus elementos en la misma posición y al final se suman todos los resultados, la forma de aplicar la operación se muestra en la ecuación 2.2. Dicha operación solo se realiza a las regiones en las que se pueda aplicar, ya que en los bordes de la imagen no posible debido a que los pixeles no tienen ocho vecinos.

$$f'(i, j) = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \otimes \begin{bmatrix} x_{(i-1,j-1)} & x_{(i-1,j)} & x_{(i,j+1)} \\ x_{(i,j-1)} & x_{(i,j)} & x_{(i,j+1)} \\ x_{(i+1,j-1)} & x_{(i+1,j)} & x_{(i+1,j+1)} \end{bmatrix} = \dots \quad (2.2)$$

$$\dots = x_{11} * x_{(i-1,j-1)} + x_{12} * x_{(i-1,j)} + x_{13} * x_{(i,j+1)} + x_{21} * x_{(i,j-1)} + x_{22} * x_{(i,j)} + x_{23} * x_{(i,j+1)} + x_{31} * x_{(i+1,j-1)} + x_{32} * x_{(i+1,j)} + x_{33} * x_{(i+1,j+1)}$$

Se debe tomar en cuenta que los valores de intensidad se encuentran en un rango de 0 a 255 por lo que si el valor es mayor a 255 la nueva intensidad del pixel tomará el valor 255, por otro lado, si el pixel es menor a 0 el nuevo valor del pixel será 0.

La principal diferencia entre los operadores puntuales y los operadores regionales es que se toman en cuenta los pixeles vecinos para realizar un cambio en el pixel central respecto a su intensidad de color [21]. Para aplicarla se utiliza la operación de convolución antes descrita y se utiliza una matriz de convolución, por lo regular la matriz de convolución es de dimensión 3x3. Sin embargo, se puede extender el tamaño de dicha matriz para problemas específicos y abarcar una mayor región.

Dependiendo de la operación que se quiera realizar se cambian los valores de la matriz de convolución, algunos de los filtros más utilizados son los de suavizado que sirven para quitar ruido dentro de una imagen (por ejemplo, filtro mediana o el promedio de una región), filtros de detección de bordes que encuentran las transiciones de intensidad de color en la imagen (por ejemplo, el filtro Sobel).

2.2 RECONOCIMIENTO DE PATRONES

Dentro de la computación existe una rama llamada Inteligencia Artificial (IA), la cual ha ido ampliando sus objetivos con el paso del tiempo, inicialmente Bellman se refiere a la IA como la automatización de actividades que se asocian con el pensamiento humano, es decir, que un sistema pueda tomar decisiones, resolver problemas o aprender por si mismo [22]. Kurzweil, se refiere a que se desarrollen sistemas que realicen operaciones que requieran inteligencia, haciendo el trabajo más fácil para los humanos [23], Winston define a la IA como la capacidad de percibir, razonar y actuar refiriéndose a que un sistema computacional puede pensar racionalmente [24], mientras que Poole presenta un enfoque en el que agentes inteligentes son programados [25].

La IA tiene aplicaciones en muchas disciplinas como los son: el procesamiento del lenguaje natural, robótica, visión por computadora, entre otras. Sin embargo, todas aplican el aprendizaje computacional, que según Jiaewi Jian et al. [26] se encarga de investigar cómo los programas computacionales pueden aprender a partir de un

conjunto de datos. El aprendizaje automático se encarga de que a partir de datos los programas computacionales encuentren patrones y toman decisiones inteligentes.

Dentro del área de reconocimiento de patrones existe una parte que se dedica a la clasificación, la cual se basa en algoritmos (clasificadores) que encuentran un modelo para clasificar o predecir comportamientos a partir de un conjunto de datos (conjunto de entrenamiento), cuando el modelo es encontrado se valida con otro conjunto de datos (conjunto de validación), el cual es utilizado para evaluar la clasificación respecto al porcentaje de precisión. Finalmente, nuevos ejemplos son sometidos al modelo encontrado a qué grupo pertenece.

Cuando se tiene el conjunto de entrenamiento donde los datos están categorizados formando grupos que comparten características, el tipo de aprendizaje se llama supervisado, cuando los datos no están categorizados el aprendizaje es llamado no supervisado. En las siguientes secciones se describen los conceptos básicos de clasificación y sobre los clasificadores utilizados en este proyecto de tesis.

2.2.1 CLASIFICACIÓN

La clasificación es un proceso que consiste en dos pasos: el aprendizaje y la clasificación. Como entrada del proceso de clasificación se tiene un conjunto de datos, los cuales pueden estar categorizados en grupos (clases) o no, dicho conjunto de datos es utilizado completamente o parte de él para la etapa de aprendizaje (conjunto de entrenamiento), en esta etapa se extrae un modelo con base en el conjunto de entrenamiento utilizando un algoritmo llamado clasificador, finalmente se utiliza el modelo extraído por el clasificador para determinar a qué clase pertenecen nuevos ejemplos, este proceso se puede observar en la Figura 2.3 [26].

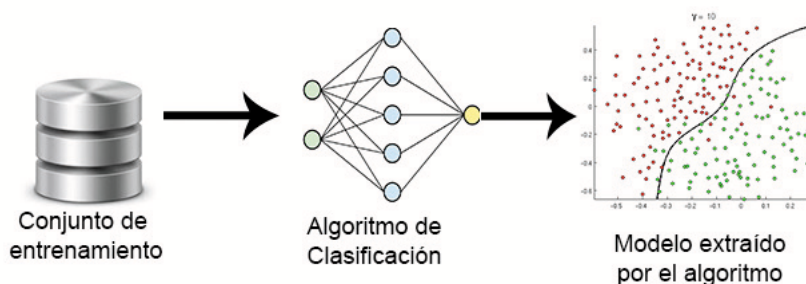


Figura 2.3: Proceso de clasificación.

Como se mencionó anteriormente los datos del conjunto de entrenamiento pueden estar categorizados por grupos de acuerdo a sus características, este tipo de aprendizaje se denomina supervisado, mientras que si los datos no lo están el aprendizaje es de tipo no supervisado, este último agrupa los datos mediante sus similitudes. Si los datos que se tienen son de tipo numérico se utiliza la distancia euclidiana para determinar la similitud entre ellos, si los datos son nominales se utilizan otro tipo de funciones [28].

2.2.2 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales son un algoritmo de clasificación supervisada, que se ha convertido en uno de los más utilizados de estos tiempos. Este algoritmo está inspirado en las interconexiones de las neuronas del cerebro humano. El procesamiento que llevan a cabo para realizar la clasificación se presenta en los siguientes párrafos. La información fue extraída de [27].

Las unidades fundamentales en una red neuronal son llamadas perceptrones, un perceptrón tiene como entrada un vector de valores reales, calcula una combinación de estas entradas utilizando un polinomio además de sus coeficientes que son llamados pesos, como salida se tiene una función escalón mostrada en la fórmula 2.3.

$$o(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{si } w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n < 0 \\ -1 & \text{en otro caso} \end{cases} \quad (2.3)$$

Donde w_i son los pesos que determinan la contribución de la entrada x_i en la salida del perceptrón. Note que el w_0 es un valor de sesgo el cual determina si algunos valores son superiores al umbral (en el caso anterior el umbral es cero). La representación del perceptrón descrito anteriormente se muestra en la figura 2.4.

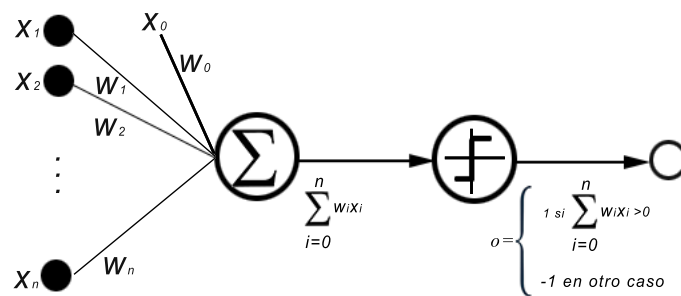


Figura 2.4: Representación de un perceptrón.

Para que un perceptrón asigne el valor de los pesos necesita una regla de entrenamiento. Inicialmente los pesos se eligen de manera aleatoria, después se utiliza la regla de entrenamiento para encontrar los valores de éstos que conducen a una mejor respuesta de la red. En el siguiente paso, iterativamente se modifican los valores de acuerdo a la regla de aprendizaje. El proceso se repite hasta que el perceptrón clasifica los ejemplos correctamente o se establece un número de iteraciones. Cada peso w_i es modificado de acuerdo a la regla de la fórmula 2.4 la cual es denominada regla de aprendizaje, ya que, ajusta los pesos de la unidad.

$$w_i = w_i + \Delta w_i \quad (2.4)$$

donde:

$$\Delta w_i = \eta(t - o)x_i \quad (2.5)$$

En la fórmula 2.5, t es la salida del ejemplo de la clase actual, o es la salida del perceptrón y η es la tasa de aprendizaje. La tasa de aprendizaje modera el grado con el que se cambia el peso en cada paso. Usualmente se asignan valores pequeños (por ejemplo 0.1). El proceso que se describió anteriormente se utiliza cuando los datos son separables de manera lineal.

Cuando se necesitan separar datos que no son linealmente separables se necesita realizar el proceso de entrenamiento con el gradiente descendente y utilizar una función de activación distinta a la escalón, usualmente de tipo sigmoide. Primero se necesita establecer la salida del perceptrón o , en este caso se utiliza una unidad lineal y se define como un producto punto entre el vector de pesos, con el vector de entrada como se muestra en la fórmula 2.6.

$$o(\vec{o}) = \vec{w} \cdot \vec{x} \quad (2.6)$$

También hay que establecer una función de evaluación de la función anterior, para ello se utiliza el error cuadrático medio como se muestra en la fórmula 2.7.

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad (2.7)$$

Para entender porque se utiliza el gradiente descendente para entrenar la red neuronal cuando no se tienen datos linealmente separables a continuación se describen algunas funciones. En la figura 2.5 se muestra el espacio de hipótesis, donde en el eje $E(w)$ representa el error de la función de salida, mientras que los ejes w_0 y w_1 representan el espacio completo de hipótesis. Al utilizar el gradiente descendente para entrenar una red neuronal se busca reducir el error en el espacio de hipótesis, es decir se realiza la derivada en un determinado punto de la función, por lo que busca minimizar error (eje $E(w)$).

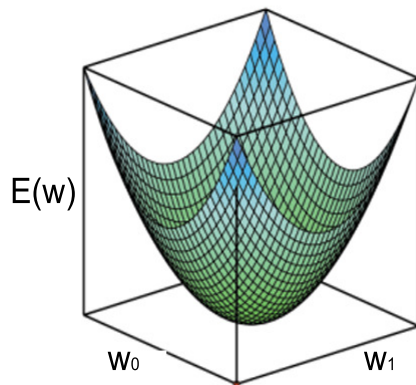


Figura 2.5: Espacio de hipótesis en el que se buscan los mejores pesos w_0, w_1, \dots, w_n que conducen a un menor error $E(x)$.

Para calcular la dirección en la que se debe guiar la búsqueda, se encuentra un vector con la derivada parcial de cada uno de los componentes del vector \vec{w} , este vector es llamado gradiente de E respecto a \vec{w} , y éste se muestra en la fórmula 2.8.

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0} + \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right] \quad (2.8)$$

Tomando en cuenta la información anterior se deben realizar modificaciones a la regla de entrenamiento, estas modificaciones se muestran en las fórmulas 2.9 y 2.10. Debido a que el Gradiente realiza un paso hacia arriba, se necesita realizar un paso descendente, por lo que en la fórmula 2.10 se utiliza un signo negativo.

$$\vec{w} = \vec{w} + \Delta \vec{w} \quad (2.9)$$

donde:

$$\Delta \vec{w} = -\eta \nabla E(\vec{w}) \quad (2.10)$$

Además del cambio en el entrenamiento con el gradiente descendente, es necesario cambiar la función escalón para la salida del perceptrón. En el caso de que los datos no sean linealmente separables se debe utilizar una función sigmoide. En la figura 2.6 se muestra la unidad sigmoide junto con su función de salida.

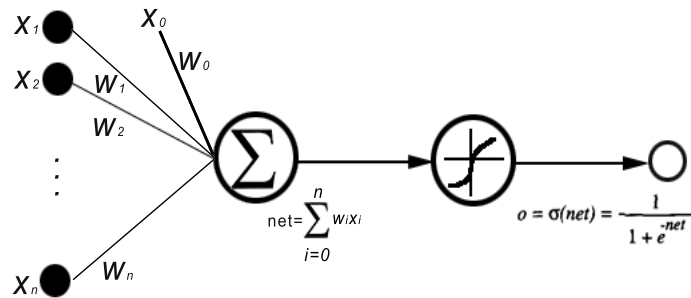


Figura 2.6: Unidad sigmoide.

Una vez que se comprendieron las dos funciones para la separabilidad de los datos, se comienza a proponer una arquitectura en donde existe una capa de neuronas de entrada, capas ocultas (se ha demostrado que con una capa oculta es suficiente) y una capa de salida como se muestra en la figura 2.7, la cual tiene cuatro neuronas en la capa de entrada, tres neuronas en la capa oculta y dos neuronas de salida. Este tipo de arquitectura es llamado feedforward y se caracteriza por que todas las neuronas de una capa están conectadas con todas las neuronas de la siguiente capa y de la capa de entrada.

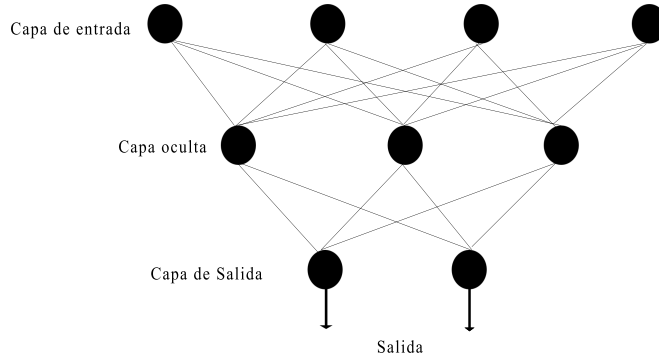


Figura 2.7: Ejemplo de Red Neuronal con arquitectura feedforward.

Para entrenar este tipo de arquitectura suele utilizarse el algoritmo de retro propagación (Backpropagation). El cual se muestra en el Algoritmo 1. Este algoritmo consiste en inicializar los valores de los pesos con valores aleatorios y con base en las entradas, se calcula el error y realiza una retroalimentación de los pesos hacia atrás.

Existen dos formas de realizar la condición de paro mostrada en la línea 3 del Algoritmo 1, una consiste en partir el conjunto de entrenamiento y realizar una prueba cada vez que se realiza una actualización en los pesos, se establece un umbral en cuanto a la precisión y cuando éste se alcanza la etapa de entrenamiento termina. El otro criterio de paro consiste en un número de épocas, una época consiste en aplicar una vez el algoritmo de retro propagación.

Finalmente, se introduce un último parámetro llamado momentum, éste sirve para guiar la búsqueda en el espacio de hipótesis ya que toma en cuenta la magnitud del error en un paso anterior para llevar a cabo el ajuste actual. Para aplicar el momentum en el Algoritmo 1 se cambia la función de entrenamiento por la expresión mostrada en la fórmula 2.11, donde $\alpha \in [0, 1]$.

$$\Delta w_{ij}(n) = \eta \delta_j x_{ji} + \alpha \Delta w_{ij}(n-1) \quad (2.11)$$

2.2.3 MÁQUINAS DE VECTORES DE SOPORTE

El clasificador máquinas de vectores de soporte (Support Vector Machine, SVM) es un método de clasificación supervisado, que se basa en realizar transformaciones de datos a un espacio donde los datos están más separados respecto a los datos originales utilizando funciones llamadas kernel. De esta manera el problema de separabilidad de los datos se basa en encontrar hiperplanos para la separación de datos en el espacio al que se llegó mediante la función kernel.

Existen dos casos en los que se puede realizar la clasificación con máquinas de vectores de soporte: donde los datos son linealmente separables y donde los datos no lo son. En esta sección se presenta como trabaja el algoritmo para los dos casos.

Algoritmo 1 Algoritmo de entrenamiento retro programación.

Entrada: Ejemplos de entrenamiento T , η , n_{in} , n_{out} , n_{hidden}

Salida: Modelo para clasificar nuevo ejemplo

- 1: Crear una red neuronal con arquitectura feedforward con n_{in} neuronas de entrada, n_{hidden} neuronas en la capa oculta y n_{out} neuronas de salida
 - 2: Inicializar los pesos con números aleatorios pequeños.
 - 3: Realizar el siguiente proceso hasta la condición de paro:
 - 4: **para** Cada valor en T de la forma $\langle \vec{x}, \vec{t} \rangle$ **hacer**
 - 5: Propagar la entrada hacia la capa oculta y la capa de salida:
 - 6: Encontrar las salida o_u de cada neurona dada la instancia \vec{x}
 - 7: Propagar hacia atrás el error en la red neuronal:
 - 8: Por cada neurona en la capa de salida k , encontrar el error:
 $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$
 - 9: Por cada neurona en la capa oculta h encontrar el error:
 $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh} \delta_k$
 - 10: Actualizar los pesos de la red:
 $w_{ji} \leftarrow w_{ji} + \Delta w_{ij}$
 donde: $\Delta w_{ij} = \eta \delta_j x_{ij}$
 - 11: **fin para**
-

Para el caso en donde los datos son linealmente separables se dará un ejemplo en donde existen dos clases (+1 y -1) en un conjunto de datos D con un conjunto de tuplas X asociadas a una clase y_i para el entrenamiento $(X_1, y_1), (X_2, y_2), \dots, (X_{|D|}, y_{|D|})$. Para este caso y_i puede tomar cualquiera de los dos valores de clases ($y_i \in \{+1, -1\}$). En la Figura 2.8 se muestra un conjunto de datos linealmente separables en 2D (puntos negros representan +1, puntos blancos representan -1), donde cualquier línea punteada puede ser un plano de separación de los datos, se debe notar que existe un número infinito de planos de separación, ahora hay que encontrar el mejor plano de separación para que los datos puedan ser clasificados de manera correcta.

En la Figura 2.9 se pueden observar dos planos de separación de datos, la imagen izquierda muestra un margen de separación pequeño y la imagen de la derecha se muestra un plano de separación grande. El clasificador busca el máximo margen de separación.

Un margen de separación se puede definir como se muestra en la fórmula 2.12, donde W es un vector de pesos, $W = \{w_1, w_2, \dots, w_n\}$; n es el número de atributos y b es un número escalar que se refiere al sesgo. Tomando en cuenta tuplas en 2-D ($X = x_1, x_2$), donde x_1 y x_2 son los valores de los atributos A_1 y A_2 , si se toma en cuenta b como un peso adicional la fórmula 2.12 se puede reescribir como se muestra en la fórmula 2.13. Entonces cada punto que se encuentre sobre el plano de separación satisface la ecuación de la fórmula 2.14, mientras que los puntos debajo del plano debe satisfacer la ecuación 2.15.

$$W \cdot X + b = 0 \tag{2.12}$$

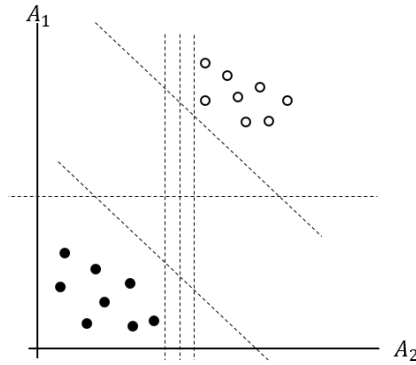


Figura 2.8: Conjunto de entrenamiento en 2D linealmente separable.

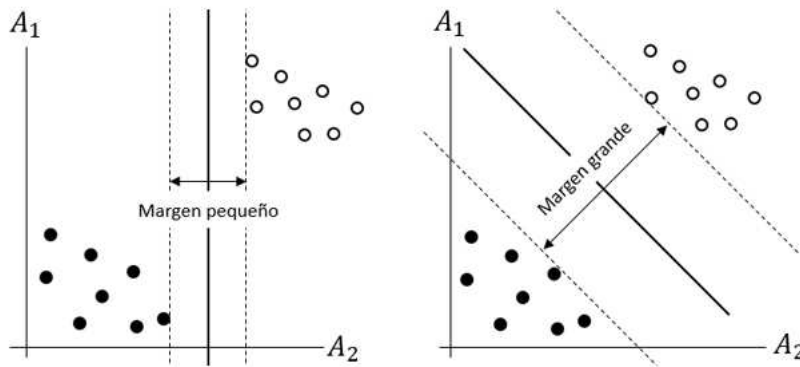


Figura 2.9: Margen grande y pequeño para un conjunto de entrenamiento en 2D linealmente separable.

$$w_0 + w_1x_1 + w_2x_2 = 0 \quad (2.13)$$

$$w_0 + w_1x_1 + w_2x_2 > 0 \quad (2.14)$$

$$w_0 + w_1x_1 + w_2x_2 < 0 \quad (2.15)$$

Los pesos pueden ser ajustados ya que cada uno de los mostrados en la fórmula 2.16, las tuplas de entrenamiento que caen sobre los hiperplanos H_1 y H_2 son llamadas vectores de soporte. También se puede encontrar el tamaño del margen máximo. La distancia del hiperplano de separación a cualquier punto de H_1 es $\frac{1}{\|W\|}$, donde $\|W\|$ es la norma Euclidiana de W ($\sqrt{W \cdot W}$). Entonces el máximo plano de separación es $\frac{2}{\|W\|}$.

$$\begin{aligned} H_1 : w_0 + w_1x_1 + w_2x_2 &\geq 1 \text{ para } y_i = +1 \\ H_2 : w_0 + w_1x_1 + w_2x_2 &\leq -1 \text{ para } y_i = -1 \end{aligned} \quad (2.16)$$

Para el caso en el que los datos no son linealmente separables, se puede extender el procesamiento del algoritmo con dos pasos, los cuales se describen a continuación: primero se realiza una transformación para que los datos estén mejor separados con una transformación no lineal, después se busca un plano de separabilidad dentro del nuevo espacio.

Para realizar la clasificación con datos en los que existen más de dos dimensiones se debe realizar un proceso diferente. Considere el siguiente ejemplo, se tiene un vector de datos en 3-D $X = (x_1, x_2, x_3)$ es mapeado en espacio 6-D (Z). Sin embargo esto es computacionalmente costoso, de esta manera se utilizan las funciones kernel que realizan una transformación, la cual hace este proceso menos costoso. Los kernels más utilizados se muestran en la fórmula 2.17.

$$\begin{aligned} \text{Kernel de tipo Gaussiano } k &= e^{\frac{-\|X_i - X_j\|^2}{2\sigma^2}} \\ \text{Kernel polinomial de grado } k &= K(X_i, X_j) = (X_i \cdot X_j + 1)^h \\ \text{Kernel de tipo Sigmoide } k &= \tanh(\kappa X_i \cdot X_j - \delta) \end{aligned} \quad (2.17)$$

2.2.4 CLASIFICADOR SIMPLE DE BAYES

El clasificador simple de Bayes (Naïve Bayes) es un clasificador estadístico, que se basa en el teorema de Bayes para determinar la pertenencia de una tupla a una clase en particular. En este clasificador se asume que los valores de los atributos son independientes de otros.

Como se mencionó anteriormente, este clasificador se basa en el teorema de Bayes y este se puede definir de la siguiente manera. Sea X una tupla de datos, H es la hipótesis de que X pertenece a una clase C . Para problemas de clasificación se tiene que encontrar $P(H|X)$ la probabilidad que la tupla X pertenezca a la clase C , dada la descripción de los datos de X .

$P(H|X)$ es la probabilidad a posteriori de H condicionada por X , $P(H)$ es la probabilidad priori de H , finalmente, $P(X)$ es la probabilidad apriori de X , el teorema de Bayes se muestra en la fórmula 2.18 y en los siguientes párrafos se explica como trabaja el clasificador simple de Bayes.

$$P(H|X) = \frac{P(H|X)P(H)}{P(X)} \quad (2.18)$$

El clasificador simple de Bayes funciona de la siguiente manera:

1. Sea D un conjunto de entrenamiento que contiene tuplas respectivamente etiquetadas con la clase. Cada tupla esta representada con vector de atributos n -dimensional $X = (x_1, x_2, \dots, x_n)$, cada tupla contiene n atributos A_1, A_2, \dots, A_n .
2. Dentro de D existen m clases C_1, C_2, \dots, C_m . Dada una tupla X , el clasificador predecirá que X pertenece a una clase con la probabilidad a posteriori más grande condicionada por X . El clasificador predice que X pertenece a una clase C_i si y solo si $P(C_i|X) > P(C_j|X)$ para $1 \leq j \leq m, j \neq i$.

Lo que se quiere entonces es maximizar $P(C_i|X)$, la cual es llamada máxima hipótesis aposteriori, basado en la fórmula 2.18, se encuentra la ecuación para encontrar la probabilidad aposteriori de la clase y se muestra en la fórmula 2.19.

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (2.19)$$

3. Ya que $P(X)$ es constante para todas las clases, solo se tiene que maximizar $P(X|C_i)P(C_i)$. Si no se conoce la probabilidad apriori, es común asumir que las probabilidades de las clases son iguales $P(C_1) = P(C_2) = \dots = P(C_m)$, entonces se maximiza $P(X|C_i)$. La probabilidad apriori de las clases se puede encontrar de la siguiente manera $P(C_{i,D}) = |C_{i,D}|/|D|$, donde $C_{i,D}$ es el número de tuplas de la clase C_i en D .
4. Ya que computar de $P(X|C_i)$ puede llegar a ser costoso en conjuntos grandes de datos se realiza la suposición de que los atributos son condicionalmente independientes, entonces $P(X|C_i)$ se puede calcular como se muestra en la fórmula 2.20, donde x es cada atributo.

$$P(C_i|X) = \prod_{k=1}^n P(x_k|C_i) \quad (2.20)$$

Para encontrar $P(X|C_i)$ se consideran los siguientes dos puntos:

- Si los datos son de tipo nominal, $P(x_k|C_i)$ es el número de tuplas de la clase C_i en D teniendo el valor x_k para A_k , dividido entre $|C_{i,D}|$.
- Si los datos son de tipo numérico, se asume que los atributos siguen una distribución Gaussiana con una media μ y una desviación estándar σ definida en la fórmula 2.21, por lo tanto, $P(x_k|C_i)$ se encuentra como se muestra en la fórmula 2.22.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \mu)^2}{2\sigma^2}} \quad (2.21)$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad (2.22)$$

5. Para predecir la clase de X , se evalúa $P(X|C_j)P(C_j)$ para cada clase C_i , se predice la clase C_i para los valores de la tupla X si y solo si se cumple la expresión de la fórmula 2.23.

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ para } 1 \leq j \leq m, j \neq i \quad (2.23)$$

2.3 HERRAMIENTAS DE PROGRAMACIÓN PARALELA

Debido a que el procesamiento de imágenes y la etapa de clasificación, son procesos computacionalmente costosos y muy tardados, una alternativa es la programación paralela, en este proyecto se utiliza la librería MPI, ya que esta implementada para el lenguaje de programación Python, ésta es utilizada para la etapa de pre-procesamiento. En la etapa de clasificación se utilizan hilos de procesamiento, éstos son utilizados en Java junto con el código de Weka. El objetivo de las implementaciones paralelas es reducir el tiempo de ejecución en cada etapa.

2.3.1 MESSAGE PASSING INTERFACE (MPI)

La programación paralela tiene un papel importante cuando la cantidad de datos a procesar es demasiado grande para el procesamiento que se tiene en una computadora. Algunos ejemplos son el procesamiento de imágenes y algoritmos de clasificación supervisada y no supervisada (DBscan o Redes Neuronales), en esos casos la programación paralela puede ayudar a reducir tiempos de ejecución.

MPI es una interfaz para codificar programas paralelos mediante el intercambio de datos entre nodos de procesamiento, esta interfaz tiene implementaciones en distintos lenguajes de programación como C/C++, Java o Python [29].

Las implementaciones con MPI pueden ser realizadas dentro de una red de computadoras o en una computadora con un procesador que conste de varios núcleos, la arquitectura que se sigue comúnmente (ver figura 2.10) consta de un nodo principal, el cual se encarga de enviar a cada nodo la información para ser procesada, en algunos casos se puede realizar el balanceo de carga en éste. Los nodos a los que llega la información se encargan de procesar la información enviada y finalmente el resultado es enviado al nodo principal para mostrarlo.

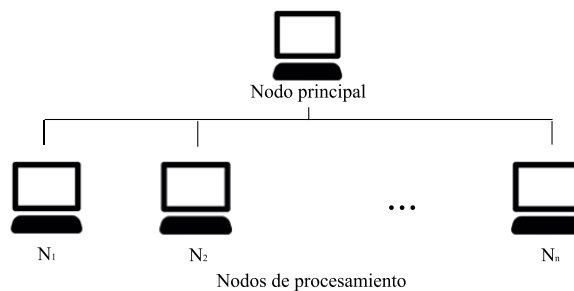


Figura 2.10: Arquitectura paralela con MPI.

2.3.2 HILOS DE EJECUCIÓN

Una de las implementaciones más utilizadas en el lenguaje de programación Java para programas paralelos son los hilos de ejecución, cada hilo de ejecución es un proceso que es enviado a un núcleo de la computadora.

Además de ser utilizados para programación paralela, se pueden aplicar principios concurrentes, es decir, los recursos compartidos por los hilos de ejecución sean utilizados por un solo hilo y no sea modificado por otro.

En este proyecto de tesis los hilos de ejecución son utilizados para la etapa de clasificación, en concreto para el ajuste de parámetros en redes neuronales. El código del programa Weka es utilizado para realizar la implementación de los algoritmos de clasificación supervisada. También se utiliza la concurrencia la implementación para utilizar el conjunto de parámetros que se evalúa y realizar la escritura en el archivo donde se guarden los resultados.

2.4 DIFERENCIAS DIVIDIDAS DE NEWTON

Los métodos de interpolación son utilizados para aproximar funciones utilizando puntos de control que sean parte de dicha función. El método de diferencias divididas de Newton es iterativo y sirve para realizar interpolación de polinomios de grado n , la principal ventaja es que, comparado con otros métodos, éste es más rápido, este método se describe en los siguientes párrafos.

De la misma manera que otros métodos de interpolación se necesitan $n+1$ puntos para realizar la interpolación de gran n , para realizar la interpolación se deben tomar en cuenta las siguientes fórmulas:

$$P(x_0) = a_0 \quad (2.24)$$

además,

$$P(x_1) = a_0 + a_1(x - x_0) \quad (2.25)$$

donde,

$$a_1 = \frac{f(x_1) - P(x_1)}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (2.26)$$

Las fórmulas anteriores sirven para encontrar los coeficientes de un polinomio de grado 1, es decir, una línea, de esta forma se puede generalizar para un grado n , para encontrar los puntos se utiliza la fórmula 2.27, donde $P(x_n)$ el valor y de punto en x , a_n los coeficientes encontrados con el método y x_n los valores de x de los puntos utilizados para encontrar el polinomio, mientras que para encontrar los coeficientes se utiliza la fórmula 2.28 [30].

$$\begin{aligned} P(x_n) &= a_0 + a_1(x - x_0) + \dots + a_n(x - x_0) \cdot \dots \cdot (x - x_{(n-1)}) \\ &= a_0 + \sum_{j=1}^n a_j \prod_{k=0}^{j-1} (x - x_k) \end{aligned} \quad (2.27)$$

$$\begin{aligned} a_0 &= f(x_0) \\ a_j &= \frac{f(x_j) - P(j-1)(x_j)}{\prod_{k=0}^{j-1} (x_j - x_k)}, \quad 1 \leq j \leq n \end{aligned} \quad (2.28)$$

CAPÍTULO 3

METODOLOGÍA PARA DETECTAR ESTADOS DE ÁNIMO

CONTENIDO

3.1	Etapa de pre-procesamiento	26
3.1.1	Reducción del espacio de búsqueda	26
3.1.2	Detección de bordes	27
3.1.3	Algoritmo basado en densidad para eliminar ruido	29
3.1.4	Detección de bordes en cejas	30
3.1.5	Detección de bordes en boca	32
3.1.6	Paralelización	34
3.2	Etapa de extracción de características	34
3.2.1	Extracción de características en cejas	34
3.2.2	Extracción de características en boca	36
3.3	Etapa de clasificación	37

En este capítulo se muestran las etapas del enfoque que se desarrolló en este proyecto de tesis, en la primera parte del capítulo se da una introducción a la teoría utilizada en la etapa de pre-procesamiento en la cual se aplicaron técnicas de detección de bordes junto con la paralelización de estos métodos. La segunda parte muestra la etapa de extracción de características, donde se extraen atributos numéricos del rostro de una persona y finalmente se describe el proceso seguido para el entrenamiento de algoritmos de clasificación supervisada.

3.1 ETAPA DE PRE-PROCESAMIENTO

En esta etapa se aplicaron distintas técnicas de procesamiento de imágenes digitales, sobre todo aquellas que tienen que ver con detección de bordes, esta parte del proyecto de tesis consiste en segmentar las ROIs del rostro para dejar la imagen lista para la extracción de características. Para realizar la detección de bordes en imágenes de rostros humanos se hicieron variaciones en técnicas clásicas, las cuales se mostrarán en las siguientes secciones. Las imágenes utilizadas fueron obtenidas de las bases de datos MMI [2], JAFFE [31] y VidTIMIT [32].

La detección de bordes en imágenes de rostros es una tarea importante, ya que los bordes en las imágenes proporcionan información sobre los estados de ánimo y las reacciones que tienen las personas, dichos bordes son normalmente utilizados para extraer características sobre las regiones de interés. En esta etapa se implementaron dos enfoques para la segmentación de ROIs, ya que al aplicar las técnicas clásicas, éstas no tuvieron un buen desempeño, debido a que no se obtuvieron los bordes bien definidos en las imágenes resultantes.

Cada etapa para el pre-procesamiento de las imágenes de rostros se presenta a continuación:

1. Se implementó el algoritmo de Viola & Jones para localizar el rostro en una imagen, por lo que se realiza una reducción de espacio de búsqueda al encontrar solo la región del rostro.
2. Ya que la relación entre las partes del rostro tiene una proporción geométrica, es decir, las ROIs siempre estarán localizadas en determinados puntos de la imagen se implementó una máscara para localizarlas.
3. Debido a que algunos canales del modelo de color RGB no aportan mucha información respecto a los bordes, se utilizaron las intensidades de los canales R y G.
4. Se aplicó el detector de bordes Gradiente en dos direcciones (vertical y horizontal) y se encontró la magnitud entre ellos.
5. Se realiza una binarización de la imagen basada en la media respecto a la frecuencia de intensidades.
6. Se implementó un algoritmo basado en el algoritmo de clusterización DBScan para encontrar regiones densas en imágenes binarias para encontrar las ROIs.

3.1.1 REDUCCIÓN DEL ESPACIO DE BÚSQUEDA

El procesamiento de imágenes digitales es costoso computacionalmente, ya que para modificar el valor de intensidad de un pixel por lo menos éste debe ser tomado en cuenta, en otros casos como los filtros regionales los vecinos de un pixel son utilizados para modificar su intensidad, por lo que se deben analizar solo las ROIs de las imágenes.

En esta tesis las imágenes analizadas son imágenes de rostros, este tipo de imágenes no se necesitan analizar en su totalidad, por lo que es necesario encontrar la región del rostro. Viola & Jones proponen un enfoque para localizar la región del rostro en imágenes digitales, dicho proceso es descrito en los siguientes párrafos [33]. Inicialmente, de ser necesario, se deben redimensionar las imágenes para que éstas sean del mismo tamaño.

El algoritmo que presentan Viola & Jones utiliza un resumen de regiones de una imagen, creando una nueva llamada imagen integral, una vez que se encuentra esta imagen se eligen algunas características basadas en las intensidades de los ojos y boca, estas regiones son mas oscuras respecto a las regiones que las rodean. Finalmente, se aplica un proceso de clasificación el cual tiene varias etapas, en cada una se descarta si existe un rostro en la región o no.

Para procesar la imagen integral se realiza la sumatoria de las intensidades de los pixeles en una determinada región de la imagen y este valor es el que se utilizará posteriormente como característica. Este tipo de características resumen información de la región, y pueden encontrarse rostros en forma horizontal, vertical y diagonal.

Para la elección de las características se aplica el algoritmo Adaboost para elegir las más representativas, haciendo más robusto el algoritmo, para encontrar la región del rostro se encontraron únicamente dos características: la intensidad de los ojos es mayor en la zona de los ojos con respecto a la zona media de la nariz y la zona horizontal de las mejillas.

En la siguiente etapa se construye un modelo respecto a las características encontradas anteriormente. En la etapa de clasificación se realiza un árbol de decisión, en el cual se va descartando si la región encontrada corresponde a un rostro o no, esta etapa se realiza para reducir el número de falsos positivos. En este enfoque únicamente se encuentran rostros de manera horizontal, existe una extensión de este trabajo en el cual es posible encontrar rostros en distintos ángulos [34].

Con el proceso descrito anteriormente se encuentra la región del rostro en la imagen. Sin embargo, aún se tienen imágenes de dimensiones grandes, además de que existen regiones que no vale la pena procesar. Por la razón anterior, se implementó una máscara para encontrar las ROIs respecto a la dimensión de la imagen obtenida del proceso anterior. Los valores para encontrar las ROIs se presentan en la Tabla 3.1, en dicha tabla se muestran los puntos inicial y final de la región rectangular de cada ROI, estos valores respectan a las características geométricas de las personas en las imágenes de las bases de datos utilizadas. Sin embargo, si realizan pruebas con otros individuos, estos valores pueden ser diferentes y deben ser ajustados.

Existe un traslape en las regiones de los ojos y de las cejas, las cuales pueden ser retroalimentadas unas con otras. Una vez que se tienen localizadas las ROIs se comienza a aplicar algoritmos de segmentación de ROIs, los cuales se presentan en las siguientes secciones.

3.1.2 DETECCIÓN DE BORDES

Para caracterizar las ROIs es necesario encontrar los bordes de las ROI. La tarea de detección bordes en imágenes de rostros no es fácil, debido a que las transiciones

Tabla 3.1: Puntos para encontrar las ROIs en imágenes de rostros.

	Punto inicial de la región	Punto Final de la Región
Boca	$(y + alto * .6, x + ancho * .2)$	$(y + alto, x + ancho * .8)$
O. Derecho	$(y + alto * .28, x + ancho * .1)$	$(y + alto * .5, x + ancho * .5)$
O. Izquierdo	$(y + alto * .28, x + ancho * .5)$	$(y + alto * .5, x + ancho * .9)$
C. Derecha	$(y + alto * .2 + ancho * .1)$	$(y + alto * .4, x + ancho * .5)$
C. Izquierda	$(y + alto * .2 + ancho * .5)$	$(y + alto * .4, x + ancho * .9)$

entre el fondo y el rostro son muy tenues.

Las imágenes en el modelo de color RGB están constituidas por tres canales, donde cada uno aporta diferente información, para este caso se observó que las intensidades de los canales R y G aportan mayor información que el canal B o la conversión de escala de grises tomando en cuenta el promedio de los tres canales, en la Figura 3.1 se muestra la comparación de las intensidades de color de cada canal de una imagen en RGB. En esta etapa cada canal es procesado por separado y al final se fusionan los resultados.



Figura 3.1: Imagen en RGB con sus canales, a) Imagen en modelo de color RGB, imágenes en escala de grises con los valores del canal B b), canal G c), canal R d).

Para la detección de bordes es necesario aplicar la convolución como se muestra en la sección 2.2.3. Sin embargo, hay que realizar una modificación en los píxeles que serán a tomar en cuenta para aplicar la convolución, debido a que si se aplica de la manera tradicional no se tiene un buen resultado. Primero, la dimensión de la región a la que se va a aplicar el proceso de convolución esta dada por la dimensión $(2u + 2) \times (2u + 1)$, donde u determina la dimensión de la región.

En la Figura 3.2 se muestra la modificación realizada al proceso de convolución en la imagen, en la Figura 3.2a se muestra la región con un valor $u = 1$, con este valor se realiza la convolución de la manera tradicional. En la figura 3.2b se aplica la modificación con un valor $u = 2$, en esta región se hace una extensión de la región utilizada para aplicar la convolución, donde se utilizan los mismos valores, puede notarse que debido a los coeficientes con valor 0, la región a considerar es mayor respecto a $u = 1$.

Al aplicar un detector de bordes de manera tradicional es difícil que los bordes en algunas imágenes se encuentren de buena manera, al aplicar la modificación anterior

-1	-2	-1
0	0	0
1	2	1

a)

-1	0	-2	0	-1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	0	2	0	1

b)

Figura 3.2: Modificación a la región a la que se aplicará la convolución, a) Región con $u = 1$, b) Región con $u = 2$.

se cubre una mayor área, por lo que es más fácil encontrar bordes. El detector que se aplicó a las imágenes es el detector Gradiente en dos direcciones, las matrices de convolución se presentan en la fórmula 3.1. Finalmente, se encuentra la magnitud entre el gradiente en dos direcciones y éste será el nuevo valor del pixel.

$$GradienteV. = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, GradienteH. = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.1)$$

El siguiente paso es aplicar un filtro de aclarado para hacer más notorios los bordes encontrados, en este caso se utilizó el filtro seno el cual es mostrado en la fórmula 3.2, después se realiza un proceso para quitar ruido de la imagen y binarizar los bordes. Para realizar este procesamiento es necesario encontrar el histograma de la imagen donde se detectaron los bordes, donde se tomará en cuenta como ruido son las intensidades menores a la media μ y el nuevo valor será 255.

Finalmente, se realiza una binarización con $\gamma = 127$ como se muestra en la fórmula 2.1. El proceso descrito anteriormente se presenta en el Algoritmo 2. Después aplicar este proceso se tiene una imagen con ruido y manchones, en la siguiente etapa se quitarán aquellos que sean considerados como ruido y se tendrá una imagen únicamente con los bordes de la ROI.

$$z' = 255 * Seno\left(\frac{\pi * z}{2 * 255}\right) \quad (3.2)$$

3.1.3 ALGORITMO BASADO EN DENSIDAD PARA ELIMINAR RUIDO

Las imágenes binarias, están formadas únicamente por pixeles negros y blancos, donde alguno de los dos colores representa el fondo y el otro color información relevante (para este caso el fondo es de color blanco y la información de las ROIs es de color negro). El proceso presentado anteriormente proporciona como resultado imágenes binarias con ruido e información que no es relevante. Por lo que se propone un algoritmo basado en densidad para quitar el ruido e información no relevante en imágenes binarias.

El algoritmo propuesto utiliza como base el algoritmo de agrupamiento DBScan [35], este algoritmo encuentra regiones densas tomando en cuenta dos parámetros

Algoritmo 2 Detección de bordes.

Entrada: Imagen en RGB I

Salida: Imagen binaria con los bordes de la boca I_s

- 1: Separar los canales de la imagen $I=I_R, I_G, I_B$
- 2: **para** Cada valor en I_R y en I_B **hacer**
- 3: G_V =Aplicar la matriz de Gradiente Vertical (Fórmula 3.1)
- 4: G_H =Aplicar la matriz de Gradiente Horizontal (Fórmula 3.1)
- 5: Valor= $\sqrt{(G_V)^2 + (G_H)^2}$
- 6: El nuevo valor del pixel será: $255 * \text{Sen}(\frac{3.1416*Valor}{2*255})$
- 7: **fin para**
- 8: Encontrar el histograma de cada una de las nuevas imágenes H_R, H_G
- 9: Encontrar la media $\mu: \frac{\text{ancho*alto}}{255}$
- 10: **para** Cada valor en I_R y en I_B **hacer**
- 11: **si** La frecuencia en el valor de intensidad del Histograma es menor a μ **entonces**
- 12: El nuevo valor del pixel en I_s será 255
- 13: **fin si**
- 14: **fin para**
- 15: Binarización con $\gamma = 127$
- 16: Retornar I_s

los cuales son el mínimo de puntos para que una región sea considerada como densa (min) y el radio de la región (e). En este caso una imagen binaria puede ser vista como un plano bidimensional donde los puntos que representan las ROIs son puntos en dicho plano. Por lo que las regiones densas serán las que tengan puntos negros cercanos entre si. Los valores utilizados en este enfoque son $min = 1$ y $e = 1$.

En los siguientes párrafos se describe el algoritmo propuesto. Se tiene como entrada una imagen binaria, de la cual se extraen las coordenadas de los puntos negros en dicha imagen. El procesamiento de los pixeles comienza tomando un punto arbitrario y éste es añadido a un grupo que será llamado cluster y se revisa si sus vecinos están dentro del conjunto de puntos. Cada que se añade un pixel a un cluster, éste debe ser marcado como visitado.

Una vez que se añadieron los vecinos al grupo se toman los pixeles que se han añadido al grupo y se repite el proceso de revisión de los vecinos, cuando se terminaron de analizar todos los pixeles del grupo se elige un nuevo punto que ha sido visitado y se repite el proceso anterior. El proceso termina cuando todos los puntos negros de la imagen han sido analizados.

Como resultado se tiene una lista con todos cluster encontrados en la imagen, dependiendo de las necesidades se puede solo quitar ruido o quedarse con los clusters más densos. Este enfoque se muestra en el Algoritmo 3.

3.1.4 DETECCIÓN DE BORDES EN CEJAS

Debido a que existen huecos en las imágenes binarias que respectan a la región de las cejas, además de que esta región no es muy densa se necesita adoptar un enfoque

Algoritmo 3 Detectar regiones densas en una imagen binaria.

Entrada: Imagen binaria I

Salida: Lista con los grupos densos en I ($Clusters$)

- 1: Se crea una lista $Puntos$ con las coordenadas de los pixeles negros en I
 - 2: **para** Cada x en $Puntos$ **hacer**
 - 3: **si** x no está visitado **entonces**
 - 4: Crear un cluster y añadir a $Clusters$
 - 5: Añadir las coordenadas al cluster actual junto con los vecinos que estén en $Puntos$
 - 6: Se marcan como visitados los pixeles que se añadieron al cluster
 - 7: **para** Cada valor en el cluster actual **hacer**
 - 8: Añadir las coordenadas al cluster actual de los vecinos si no han sido visitados y marcar como visitados
 - 9: **fin para**
 - 10: **fin si**
 - 11: **fin para**
 - 12: Retornar $Clusters$
-

diferente para esta región. Adicionalmente hay casos donde las transiciones son muy suaves respecto a la región del parpado. En los siguientes párrafos se presenta el enfoque que se desarrolló para encontrar la región de las cejas.

El primer paso es realizar un análisis de las intensidades de los canales de las imágenes, ya que la entrada del algoritmo son imágenes en formato de color RGB. Como se puede observar en la Figura 3.1 el canal que aporta más información con respecto a las regiones oscuras, como las cejas, es el canal R, por lo que éste será usado para segmentar esta región.

El siguiente paso será aplicar un filtro que haga mas notorias las regiones oscuras, el filtro elegido fue la tangente hiperbólica con un valor de $\alpha = 0.03$. La función sigmoide es una operación que hace mas oscuras las regiones oscuras y más claras las regiones claras y la ecuación para ser aplicada a cada pixel se muestra en la fórmula 3.3.

$$z' = \frac{255}{2} \left[1 + \tanh \left[\alpha \left(z - \frac{255}{2} \right) \right] \right] \quad (3.3)$$

El siguiente paso será realizar una operación de aclarado. Sin embargo, las regiones oscuras deben quedarse intactas (en este caso se utiliza la función seno para aclarar), esto servirá como corrección de sombras, la función se muestra en la fórmula 3.4. Donde $f'(x, y)$ es una función de aclarado y th es el umbral que separa las regiones claras de las oscuras.

$$f(x) = \begin{cases} f(x, y) & \text{si } f(x, y) < th \\ f'(x, y) & \text{si } f(x, y) > th \end{cases} \quad (3.4)$$

Después se debe aplicar una operación de morfología de cierre para realizar

un suavizado y eliminar huecos pequeños en dicha región, este filtro se basa en la dilatación y la erosión [36], las cuales utilizan un elemento estructura constituido por ceros y unos, en el cual se realiza una operación parecida a la convolución, para este caso se utiliza el elemento estructura mostrado en la fórmula 3.5.

$$\textit{ElementoEstructura} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.5)$$

La operación con el elemento estructura se aplica de la siguiente manera: se realiza la traslación por toda la imagen de manera similar a la convolución, sin embargo, en morfología solo se toman en cuenta los valores que estén marcados con 1 en el elemento estructura.

Para el caso de la dilatación el nuevo valor del pixel pivote (por lo regular, el del centro) será el valor máximo de los que estén marcados con 1 en el elemento estructura, para el caso de la erosión el nuevo valor del pixel pivote será el mínimo.

La operación de cierre consiste en una operación de dilatación a una imagen en escala de grises, seguida de una erosión, dando como resultado un suavizado en la imagen además de eliminar huecos pequeños, lo que ayuda a que la ceja se convierta en una región más densa respecto a la imagen original.

Como resultado del paso anterior se tiene una imagen corregida con las regiones de cejas definidas de mejor manera que en la imagen original. El siguiente paso es aplicar un algoritmo de umbralización automático, en este caso se utilizó el algoritmo Otsu, éste es un algoritmo iterativo donde se busca la máxima separabilidad de dos regiones en el histograma de una imagen [37].

Hasta este punto ya se tiene la imagen binaria con la información y ruido que tiene que ver con el cabello y los ojos. Finalmente se aplica el algoritmo basado en densidad y se realiza un análisis tomando en cuenta las funciones de aspecto de radio, tamaño de los clusters con respecto a la imagen original y densidad para determinar los clusters que son cejas de los que no lo son. El proceso descrito anteriormente se muestra en el Algoritmo 4.

3.1.5 DETECCIÓN DE BORDES EN BOCA

La boca es una parte del rostro que aporta mucha información sobre la expresión que demuestra una persona, el objetivo de procesar esta región es encontrar el borde de los labios para extraer atributos característicos sobre esta región, sin embargo, esta tarea es difícil, ya que las transiciones entre los labios y la piel son muy tenues.

Para esta región únicamente se utilizan las intensidades de los canales R y G debido a que aportan más información para la detección de bordes esta región, las cuales se procesan por separado y al final se fusionan sus resultados.

Inicialmente, se realiza el proceso mostrado en el Algoritmo 2 con la imagen de la región en el modelo de color RGB. El siguiente paso es aplicar el algoritmo basado en densidad presentado en el Algoritmo 3 y se hace un análisis de los dos clusters que se quedaron en cada una de las imágenes, primero se quitan los que estén más cercanos a las esquinas, ya que no representan información acerca de la boca, debido

Algoritmo 4 Detección de la región de las cejas.

Entrada: Imagen en RGB I

Salida: Imagen binaria con los clusters de la ceja I_s

- 1: Separar los canales de la imagen $I=I_R, I_G, I_B$
 - 2: Aplicar el filtro Tangente Hiperbólica a la imagen.
 - 3: **para** Cada valor en I_R **hacer**
 - 4: Aplicar la umbralización de la Fórmula 3.4
 - 5: **fin para**
 - 6: Aplicar la operación morfológica de cierre con el elemento estructura de la Fórmula 3.5 I_M
 - 7: Encontrar el Histograma H de I_M y aplicarle el algoritmo de umbralización Otsu, teniendo como resultado un valor de umbral th
 - 8: **para** Cada valor x en I_R **hacer**
 - 9: **si** x es menor a th **entonces**
 - 10: El nuevo valor del pixel será 255
 - 11: **si no**
 - 12: El nuevo valor del pixel será 0
 - 13: **fin si**
 - 14: **fin para**
 - 15: Regresar la imagen I con los clusters que quedaron en el conjunto.
-

que la boca se localiza en el centro de la imagen, y se utilizan los dos clusters más densos.

El proceso anterior se aplica a las imágenes en los dos canales (R y G) una vez que se tienen las dos imágenes binarias creadas se realiza una fusión de las dos imágenes con una función XOR (se pintan de negro los pixeles negros de las dos imágenes en una sola). Una vez que se tienen esas imágenes procesadas se tiene que determinar cuales regiones pertenecen a la boca.

Después del procesamiento anterior aún puede existir ruido, el cual es eliminado con la proporción que existe entre el área del cluster y la imagen, y es calculado con la fórmula 3.6. Donde x_{max} y x_{min} son los valor máximo y mínimo en x del cluster, mientras que y_{max} y y_{min} son los valores máximo y mínimo del cluster, ancho y alto se refieren a la imagen de la región.

$$proporcion = \frac{(x_{max} - x_{min}) * (y_{max} - y_{min})}{ancho * alto} \quad (3.6)$$

Se encontró que la proporción con respecto a la imagen segmentada de la boca no supera el valor 0.35, por lo que si la proporción con respecto a la imagen es menor a 0.35 se toma como ruido y no es tomado en cuenta para la imagen final, de lo contrario el cluster aparecerá en la imagen. El proceso mostrado en esta sección se muestra en el Algoritmo 5.

Algoritmo 5 Detección de bordes para la región de la boca.

Entrada: Imagen en RGB I

Salida: Imagen binaria con los bordes de la boca I_s

- 1: Realizar el procesamiento del Algoritmo 2
 - 2: Realizar el procesamiento del Algoritmo 3 y quedarse con los clusters más densos
 - 3: Quitar los clusters que estén cercanos a las esquinas
 - 4: Se fusionan las dos imágenes con una función XOR.
 - 5: **para** Cada Cluster **hacer**
 - 6: Encontrar la proporción con respecto a la imagen (Formula 3.6)
 - 7: Si la proporción es menor a 0.35 quitar el cluster del conjunto
 - 8: **fin para**
 - 9: Regresar la imagen I con los clusters que quedaron en el conjunto.
-

3.1.6 PARALELIZACIÓN

El procesamiento de imágenes es una tarea que es muy tardada debido a que las intensidades de los píxeles tienen que ser procesados, en algunos casos se tienen que tomar en cuenta los valores de los píxeles vecinos para realizar el cambio de intensidad. Por esta razón es importante aplicar enfoques paralelos, para que los tiempos de ejecución sean menores.

Para la etapa de pre-procesamiento se realizó un procesamiento paralelo utilizando la biblioteca MPI con el lenguaje de programación Python, siguiendo la arquitectura mostrada en la Figura 2.10, en estos experimentos se utilizaron dos computadoras con cuatro núcleos cada una y las imágenes se procesaron de la siguiente manera: a cada núcleo se le envía una imagen la cual será procesada; en cada uno de los núcleos se aplica la máscara para encontrar cada región del rostro; después se realiza el procesamiento para las ROIs presentadas en este capítulo como se muestra en la Figura 3.3; finalmente, se tiene como salida una imagen binaria por cada ROI segmentada.

3.2 ETAPA DE EXTRACCIÓN DE CARACTERÍSTICAS

La etapa de extracción de características consiste en encontrar atributos descriptivos de las ROIs, por lo regular de tipo numérico. En esta etapa se utilizaron las imágenes resultantes de la etapa de pre-procesamiento y se utilizaron polinomios encontrados con el método diferencias divididas de Newton, además se definieron dos conjuntos de características; uno con los coeficientes y otro con diez puntos, ambos fueron extraídos con los polinomios.

3.2.1 EXTRACCIÓN DE CARACTERÍSTICAS EN CEJAS

Como se mencionó en la sección anterior la segmentación de cejas es diferente a la utilizada en las demás ROIs de la imagen. Como resultado de la segmentación se tiene una imagen binaria para cada una de las cejas, a continuación se describe el proceso de extracción de características para estas regiones.

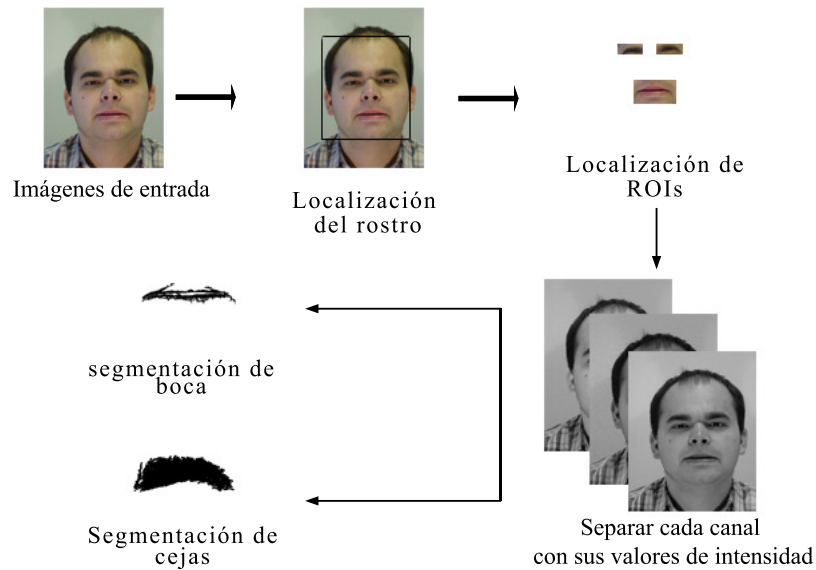


Figura 3.3: Procesamiento de aplicado en cada núcleo de procesamiento.

En este proceso se tiene como entrada una imagen binaria con la región de cada ceja, para las dos cejas se sigue el mismo proceso, se realiza un recorrido por toda la imagen tomando en cuenta un incremento Δ en cada paso sobre el eje x . En cada incremento se analizan los puntos negros en el eje y sobre dicho punto, después se guardan las coordenadas de los puntos negros y se realiza un promedio de ellos en el eje y , con la operación anterior se encuentra el punto medio, al final se tienen una serie de puntos que serán utilizados para realizar interpolación.

Para caracterizar las cejas se utiliza un polinomio cúbico debido a que éste se ajusta mejor a la forma de la ceja, para encontrar este tipo de polinomios son necesarios cuatro puntos de los obtenidos en el paso anterior. La selección de los puntos es de la siguiente manera: se seleccionaron los dos puntos mas extremos y dos puntos intermedios encontrados en proporción a la región de la ceja. Los puntos son trasladados al origen con un vector de traslación, que tendrá los valores negativos del primer punto. Un ejemplo de la interpolación con un polinomio cúbico se muestra en la Figura 3.4.

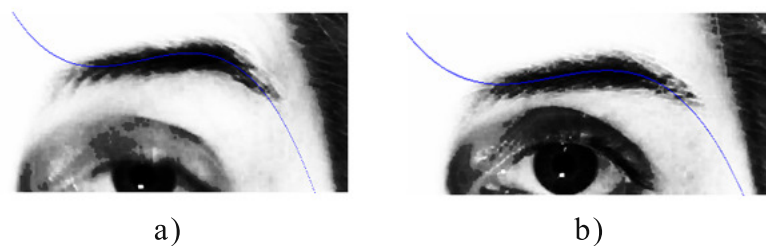


Figura 3.4: Ejemplo de interpolación con polinomio cúbico para las cejas.

Después de trasladar los puntos, éstos son utilizados para realizar la interpolación cúbica mediante método diferencias divididas de Newton, este enfoque fue utilizado debido a que es rápido y no tiene el problema donde no exista una matriz inversa para encontrar los valores del polinomio. Dicho método de consiste en tener $n + 1$ puntos donde n es el grado del polinomio que se quiere obtener y se encuentran con la expresión mostrada en la fórmula 3.7. Finalmente, se encuentra el polinomio y con base en él se distribuyen diez puntos entre los extremos de las cejas respecto al eje x . En la Figura 3.5 se muestra el proceso para extraer las características para la región de las cejas.

$$\begin{aligned}
 P(x_0) &= y_0 = a_0 \implies a_0 = y_0 \\
 P(x_1) &= y_1 = a_0 + a_1(x_1 - x_0) \implies a_1 = \frac{y_1 - y_0}{x_1 - x_0} \\
 P(x_2) &= y_2 = a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \implies a_2 = \frac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}
 \end{aligned}
 \tag{3.7}$$

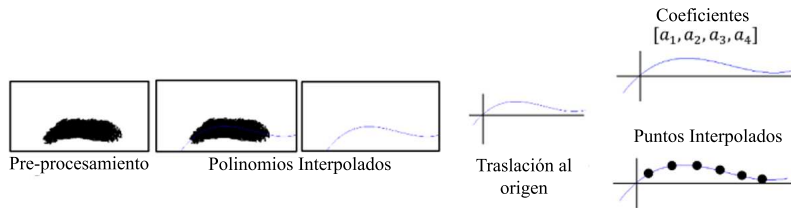


Figura 3.5: Procesamiento de las características para la región de las cejas.

3.2.2 EXTRACCIÓN DE CARACTERÍSTICAS EN BOCA

Para la extracción de características se aplica un enfoque parecido al de las cejas, como entrada se tiene la imagen binaria con los bordes de la boca que se encontró en la etapa del procesamiento. Para caracterizar la región de la boca se utilizan dos polinomios cuadráticos como se muestra en la figura 3.6. Se realiza un procesamiento parecido al utilizado en la región de las cejas, la principal diferencia es que se encuentran dos polinomios (uno para la parte inferior y otro para la parte superior), de la misma manera que en la región de las cejas se utilizan incrementos con el valor Δ , sin embargo, para la región de la boca se encuentran los puntos máximos y mínimos en cada incremento.

Para encontrar los dos polinomios cuadráticos, se necesitan únicamente tres puntos, por lo que se utilizan los dos puntos extremos y el punto medio para realizar la interpolación. De igual manera que en el procesamiento de las cejas se utilizó el método de las diferencias divididas de Newton para realizar la interpolación.

Una vez que se encontraron los polinomios se buscan diez puntos para cada polinomio, después son trasladados al origen tomando en cuenta el primer punto

de manera similar al procesamiento de la boca. Los diez puntos serán utilizados posteriormente para la etapa de clasificación. En la Figura 3.6 se muestra un ejemplo de la interpolación de la boca con dos polinomios cuadráticos y en la Figura 3.7 se muestra el procesamiento descrito en esta sección.



Figura 3.6: Ejemplo de interpolación con dos polinomios cuadráticos para la boca.

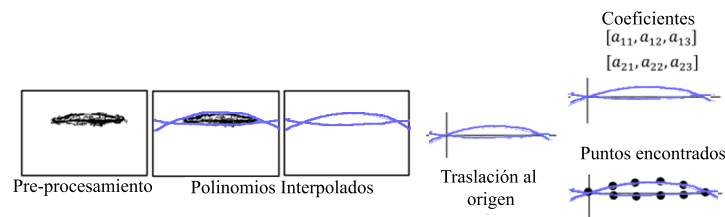


Figura 3.7: Procesamiento de las características para la región de la boca.

3.3 ETAPA DE CLASIFICACIÓN

En la etapa de extracción de características se encontraron dos conjuntos de características, uno con los coeficientes y otro con diez puntos por cada polinomio encontrado. Estas características son evaluados en la etapa de clasificación con algoritmos de aprendizaje supervisado como redes neuronales o máquinas de vectores de soporte, de esta manera si se obtiene un porcentaje de clasificación alto las características son buenas para ser aplicadas otro conjunto de datos.

Para esta etapa se realizaron experimentos con distintos algoritmos de aprendizaje supervisado como: máquinas de vectores de soporte, redes neuronales y el clasificador simple de Bayes, los cuales están diseñados para datos de tipo numérico. En los siguientes párrafos se describe que parámetros de cada algoritmo fueron ajustados para encontrar un modelo que tenga un alto porcentaje de clasificación.

El conjunto de datos fue normalizado entre un rango entre cero y uno mediante el método $z - score$. Posteriormente se aplicaron los algoritmos de clasificación mencionados en el párrafo anterior con una validación cruzada con $k = 5$ y finalmente se encontraron las matrices de confusión para cada prueba realizada, así como el porcentaje de clasificación correcta.

El clasificador simple de Bayes está basado en probabilidades y se aplicó de la manera clásica, de esta manera no hay parámetros que ajustar, por lo que únicamente se realizó solo un experimento corriendo el algoritmo con los datos normalizados.

Para el clasificador máquinas de vectores de soporte se cambio el tipo de kernel utilizado. Se experimentó con un kernel polinomial y uno binomial, de esta manera se puede comparar cual es el mejor dependiendo de la matriz de convolución y la precisión encontrada.

Otro clasificador utilizado es redes neuronales para el cual se utilizó una arquitectura feedforward con una capa oculta y se realizó un proceso de ajuste de parámetros, en el cual se estableció un número de neuronas en la capa oculta utilizando un rango $\pm \frac{\text{atributos} + \text{clases}}{2}$, sobre la un número pivote encontrado con la ecuación: $\frac{\text{atributos} + \text{clases}}{2}$. Además, se establecieron un rango para el momentum (0.001-0.06) y otro para la tasa de aprendizaje (0.01-0.4), mientras que para las épocas utilizadas 500, 1000 y 1500.

Los resultados para los clasificadores utilizados se muestran en el sección 4.3, donde se analizan los resultados respecto a la precisión y las matrices de confusión para los estados de ánimo clasificados.

Dentro de la clasificación con redes neuronales se tiene que realizar un proceso en el que se realiza un ajuste de parámetros, en esta etapa se experimenta con distintos conjuntos de parámetros para determinar cuales tienen un mejor precisión obtenida.

Para esta etapa se utilizó la implementación de Weka, el cual está desarrollado en el lenguaje de programación Java, por lo que se implementaron hilos de ejecución para realizar el procesamiento paralelo, utilizando principios de exclusión mutua, en el que se utiliza una estructura de cola para que los distintos parámetros se vayan evaluando.

En la Figura 3.8 se muestra el procesamiento utilizado, inicialmente se establecen k hilos de ejecución, además de un conjunto de parámetros a ser evaluados, los resultados se concentran en un archivo excel, en el que se muestra el conjunto de parámetros y la precisión encontrada con la red neuronal.

Si se requiere el modelo encontrado por la red neuronal se puede correr el experimento con los parámetros mostrados en el archivo excel o se puede guardar el modelo. Finalmente, se puede mostrar la matriz de confusión y diferentes métricas de evaluación.

El objetivo de realizar el procesamiento paralelo es reducir el tiempo de ejecución al evaluar los parámetros, el tiempo de ejecución depende del total de conjuntos de parámetros utilizados en cada ejecución, además de los recursos con los que cuenta la computadora utilizada para realizar los experimentos.

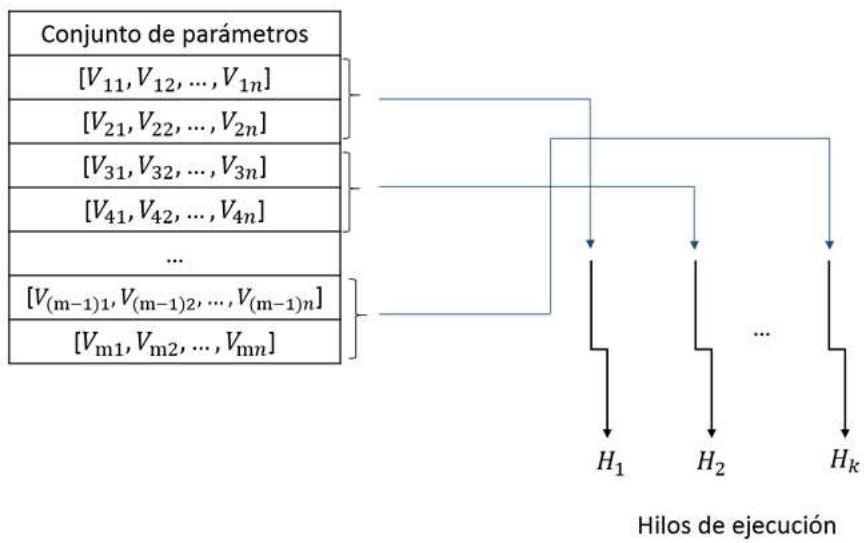


Figura 3.8: Procesamiento paralelo de la etapa de clasificación.

CAPÍTULO 4

RESULTADOS

CONTENIDO

4.1	Etapa de pre-procesamiento	42
4.1.1	Reducción de espacio de búsqueda	42
4.1.2	Pre-procesamiento en cejas	42
4.1.3	Pre-procesamiento en boca	44
4.2	Clasificación	46

En este capítulo se presentan los resultados para cada etapa mostrada en el Capítulo 3 (segmentación de ROIs, extracción de características y clasificación) adicionalmente se realizaron comparaciones con otro enfoque para la segmentación de boca.

4.1 ETAPA DE PRE-PROCESAMIENTO

En esta etapa se realizó un proceso para encontrar las ROIs en imágenes de rostros, en el que se utilizó el algoritmo de Viola & Jones para encontrar el rostro en una imagen. Una vez que se encontró la región del rostro se creó una máscara para localizar las ROIs. En las siguientes etapas se aplicaron algoritmos de umbralización y detección de bordes para segmentar solamente la parte de los bordes de la ROI sin información ruidosa y sin el fondo.

4.1.1 REDUCCIÓN DE ESPACIO DE BÚSQUEDA

Como se mencionó anteriormente se necesita reducir el espacio de búsqueda dentro de la imagen, ya que no es necesario procesar todos los píxeles de la imagen. El primer paso para reducir el espacio es utilizar el algoritmo de Viola & Jones, este algoritmo realiza un procesamiento para encontrar el rostro de una persona en una imagen digital. Para determinar la efectividad de la implementación utilizada, la cual está contenida en la biblioteca OpenCV con el lenguaje de programación Python, se realizaron experimentos con la base de datos MMI, dicha base de datos contiene imágenes y vídeo (para los experimentos se utilizaron las imágenes que son 494). Los resultados de los experimentos se muestran en la Tabla 4.1 como una matriz de confusión.

Tabla 4.1: Resultados de los experimentos de la implementación del algoritmo Viola & Jones.

	Verdadero	Falso
Verdadero	468	14
Falso	11	NA

Como se puede observar en la matriz de confusión el porcentaje de rostros encontrados es de 94.93% de precisión, mientras que los falsos positivos y falsos negativos tienen un porcentaje muy parecido con 2.83% y 2.22% respectivamente, como no se utilizaron imágenes en las que no aparecieran rostros el último dato de la matriz de confusión mostrada en la Tabla 4.1 no aplica. En la Figura 4.1 se muestran cuatro imágenes en las que se aplicó el algoritmo Viola & Jones, dejando en un recuadro la región del rostro encontrada por el algoritmo.

4.1.2 PRE-PROCESAMIENTO EN CEJAS

Para la región de las cejas se utiliza un enfoque basado en métodos de umbralización y este es presentado en el capítulo 3. En esta sección se muestran los resultados de aplicar el enfoque propuesto para la segmentación de esta región.

En el proceso mostrado en la Figura 4.2 corresponde al pre-procesamiento de la región de las cejas. La Figura 4.2a corresponde a la imagen de entrada, para este proceso la imagen corresponde a una imagen en escala de grises con las intensidades



Figura 4.1: Ejemplo de imágenes resultantes de aplicar el algoritmo de Viola & Jones.

del canal R, utilizando una imagen en el modelo de color RGB, en 4.2b se muestra la imagen resultante de aplicar la función puntual tangente hiperbólica, la cual realiza un aclarado a las regiones claras y un oscurecimiento a las zonas oscuras. La imagen 4.2c muestra el resultado de aplicar la operación morfológica de cierre, la cual realiza un suavizado y elimina huecos pequeños, la imagen 4.2d muestra la aplicación del algoritmo Otsu a la imagen 4.2c. Finalmente, en 4.2e se muestra la detección del cluster correspondiente a la región de las cejas aplicando el algoritmo 3.

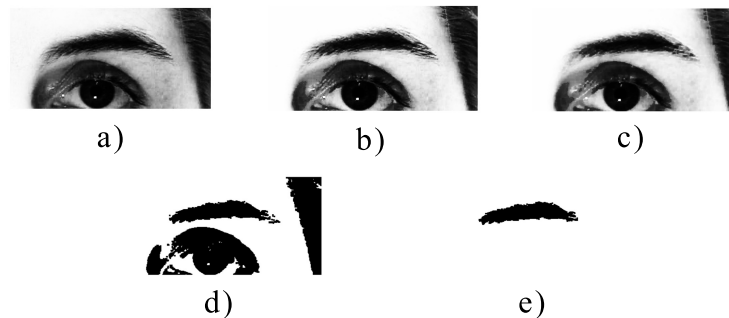


Figura 4.2: Pre-procesamiento de la ceja: a) Imagen de entrada (Canal R), b) Función tangente hiperbólica, c) Operación de cierre, d) Algoritmo Otsu, e) detección del cluster.

Los resultados después de aplicar el procesamiento anterior se muestra en la tabla 4.2 (esta selección se realizó de manera visual). Se tiene un total de 958 imágenes de cejas, de las cuales este enfoque funcionó para 804, para 38 imágenes de cejas se segmentó la ceja junto con la región del ojo. Finalmente en 116 imágenes no se segmentó la región debido a que no se encontró la suficiente información para extraer las características correspondientes.

Se segmentó manualmente un conjunto de imágenes de la base de datos que consiste de cinco imágenes por individuo, después se encontraron los polinomios con el método diferencias divididas de Newton como se muestra en la etapa de extracción de características para el conjunto de imágenes segmentadas manualmente y para

Tabla 4.2: Resultados de los experimentos del procesamiento de imágenes de cejas.

Segmentación	Porcentaje
Completa	83.94%
Segmentado con ruido	3.96%
Sin segmentar	12.10%

Tabla 4.3: Media y desviación estándar para las diferencias de los coeficientes del polinomio de la región de las cejas.

	Coefficiente 1	Coefficiente 2	Coefficiente 3
Media	0.190	0.0016	0.000019
Desviación estándar	0.177	0.0018	0.000035

las imágenes segmentadas con el enfoque propuesto.

Una vez que se encontraron los coeficientes de los polinomios, se realizó la diferencia entre los dos conjuntos y finalmente se encontró la media y desviación estándar de las diferencias, se debe tomar en cuenta que el primer punto fue trasladado al origen, por ese motivo el primer coeficiente siempre será cero y no será tomado en cuenta para la comparación, los resultados se muestran en la Tabla 4.3, en esta tabla se puede observar que los valores son muy pequeños, esto quiere decir que los valores de los coeficiente son muy parecidos entre si, por lo tanto la segmentación se realiza de buena manera.

4.1.3 PRE-PROCESAMIENTO EN BOCA

En el procesamiento en la región de la boca se aplicaron técnicas de detección de bordes, el proceso seguido en la región se muestra en la sección 3.1.5, donde se presenta el algoritmo propuesto para detección de esta región, en la Figura 4.3 se muestra el resultado de aplicar cada una de las etapas a una imagen de la base de datos MMI, se debe tomar en cuenta que para este proceso el enfoque propuesto se debe aplicar a dos imágenes en escala de grises con las intensidades de los canales R y G, al final se hace una operación XOR entre ellas para tener una imagen final con los detalles encontrados en cada canal.

En todos los incisos de la Figura 4.3 se muestra del lado derecho la imagen del canal R y del lado izquierdo del canal G. En 4.3a se puede observar el resultado de aplicar el filtro gradiente en dos direcciones, en 4.3b se muestra el resultado de aplicar la función puntual Seno a las imágenes en 4.3a, en 4.3c se observa la imagen resultante después del procesamiento de la línea 10 mostrada en el algoritmo 5 (si la frecuencia del pixel es menor a la media se considera ruido), en 4.3d se muestra el resultado final del algoritmo basado en densidad. Finalmente, en 4.3e se observa la función XOR de las imágenes mostradas en 4.3d. Como resultado adicional en

la Figura 4.4 se muestra el ensamble de las regiones de interés de cuatro rostros de ejemplo.

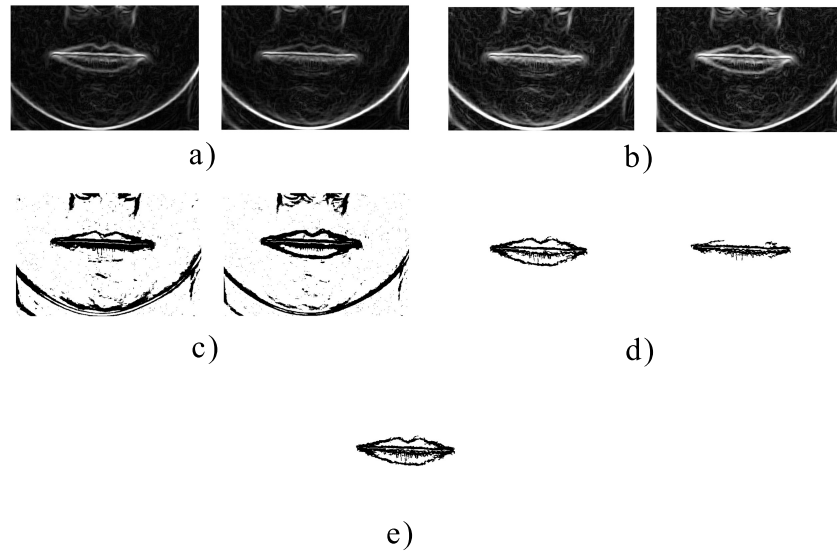


Figura 4.3: Pre-procesamiento de la boca: a) Filtro gradiente, b) Función seno, c) Quitar ruido menor a la media, d) Algoritmo basado en densidad, e) Función XOR.

El algoritmo fue aplicado a las 479 imágenes de la base de datos MMI teniendo como resultado que en la mayoría se segmentó bien la región de la boca. En la tabla 4.4 se muestra el porcentaje de imágenes bien segmentadas y de imágenes sin segmentar (esta comparación se hizo de manera visual), de un total de 479 imágenes se segmentó bien la región de la boca en 402, mientras que en 77 se introdujo ruido o no se segmentó la región de la boca.

Tabla 4.4: Resultados de los experimentos del procesamiento de imágenes de cejas.

Segmentación	Porcentaje
Completa	83.92%
Sin segmentar	16.08%

De la misma manera que se hizo para la región de las cejas, se segmentaron manualmente algunas imágenes de la base de datos MMI y se realizó una comparación entre los coeficientes obtenidos por la segmentación manual y la segmentación del enfoque propuesto, estos resultados se muestran en la Tabla 4.5 en donde se puede observar que los valores para la media y desviación estándar son pequeños por lo que las regiones encontradas son muy parecidas, por lo que la segmentación se realizó de buena manera.



Figura 4.4: Ejemplo de ensamble de las regiones de interés pre-procesadas.

Finalmente, se realizó una comparación contra el enfoque propuesto en [38] en el que se utilizaron imágenes de las bases de datos VidTIMIT y JAFFE, la comparación se muestra en la Figura 4.5, el resultado de nuestro enfoque esta marcado con "o", mientras que el enfoque propuesto en [38] esta marcado con "+", la primera fila corresponde a imágenes de la base de datos Jaffe y la segunda fila corresponda a imágenes de la base de datos VidTIMIT. En la imagen se puede observar que el enfoque propuesto es competitivo a pesar de que las imágenes hayan sido tomadas en escala de grises y el tamaño sea pequeño. Es importante señalar que se realizó un ajuste en las imágenes antes de aplicar el enfoque propuesto, dicho ajuste consiste en aplicar filtros de suavizado. Para las imágenes a color esto se realizó a los canales utilizados, mientras que para las imágenes en escalas de grises se aplicó a toda la imagen.

Tabla 4.5: Media y desviación estándar para las diferencias de los coeficientes de los polinomios de la región de la boca.

	Coefficiente 1	Coefficiente 2
Media región superior	0.101	0.0007
Media región inferior	0.160	0.00104
Desviación estándar región superior	0.105	0.00071
Desviación estándar región inferior	0.1518	0.001005

4.2 CLASIFICACIÓN

En la etapa de extracción de características, fueron extraídos dos conjuntos de datos, los dos se encontraron con polinomios de interpolación, ahora en la etapa de clasificación se tiene que evaluar si estas características sirven para caracterizar los estados de ánimo mostrado por las personas en una imagen. Por lo que se realizaron experimentos con tres clasificadores diseñados para datos de tipos numérico, los cuales son: clasificador simple de Bayes, máquinas de vectores de soporte y redes neuronales.

Las características se concentraron en dos archivos .arff, en uno se utilizaron los valores de los coeficientes de los polinomios para cada región, mientras que en otro se pusieron los diez puntos para cada región, para los dos se utilizó el siguiente

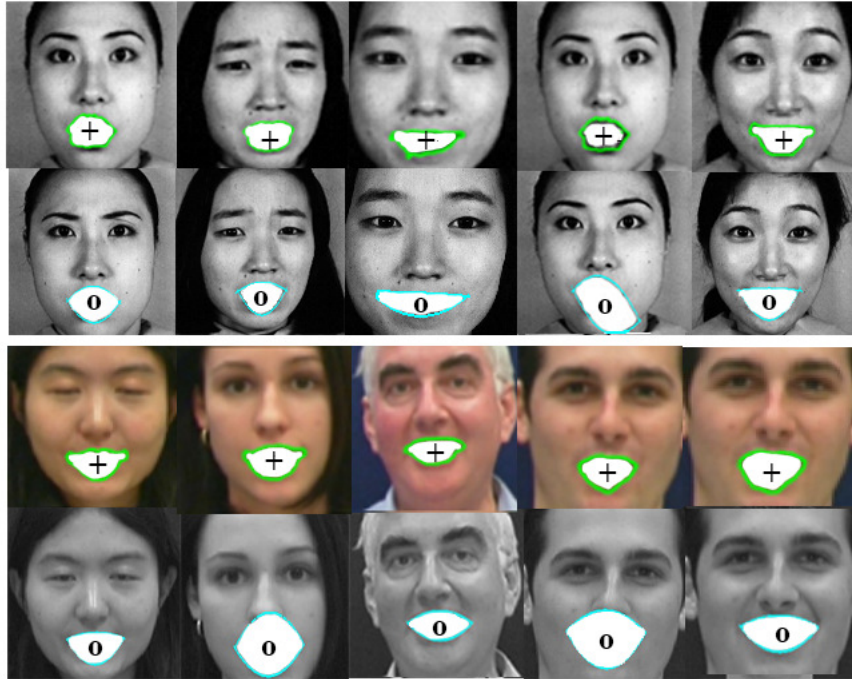


Figura 4.5: Comparativa contra los resultados reportados en [38] (marcados con "+") y nuestros resultados (marcados con "o").

orden: primero los puntos de la ceja izquierda, después los puntos de la ceja derecha y finalmente los puntos de la boca.

Estos valores fueron utilizados como entrada a los tres clasificadores que se enunciaron anteriormente, el clasificador simple de Bayes se aplicó de manera tradicional, en las máquinas de vectores de soporte se cambio el tipo de kernel utilizado entre un binomial y un polinomial, para las redes neuronales se variaron los parámetros del clasificador(número de épocas, momentum, tasa de aprendizaje, número de neuronas en la capa oculta).

No todas las imágenes las personas demuestran una expresión donde algún estado de ánimo sea notorio. Por el motivo anterior se utilizó solo una parte de la base de datos para implementar el enfoque propuesto. En los experimentos se utilizaron 79 imágenes y se clasificaran tres emociones: felicidad, enojo y sorpresa. Para el sentimiento de felicidad se tienen 42 instancias, para enojo 17 y para sorpresa 20. Para diferencias los conjuntos de características se nombrarán como CC (Características de Coeficientes) y CP (Características de puntos).

Para todos los experimentos se realizó una validación cruzada con 5 iteraciones variando los parámetros como se mencionó anteriormente. Se utilizó el código del software Weka, en una computadora con dos procesadores intel core i7 con una frecuencia de reloj de 2.53 GHz y 32 GB de memoria RAM. A continuación se presentan los experimentos realizados con los tres algoritmos de clasificación.

Primero se presentan los resultados de aplicar el clasificador simple de Bayes, en las Figuras 4.6 y 4.7 se muestra las gráficas de dispersión y las matrices de

confusión para los dos conjuntos de características (CP y CC respectivamente). El mejor resultado se obtuvo con CP con un 84.81 % de precisión, el radio del círculo en las gráficas de dispersión corresponden al número de instancias clasificadas que se muestran en las matrices de confusión.

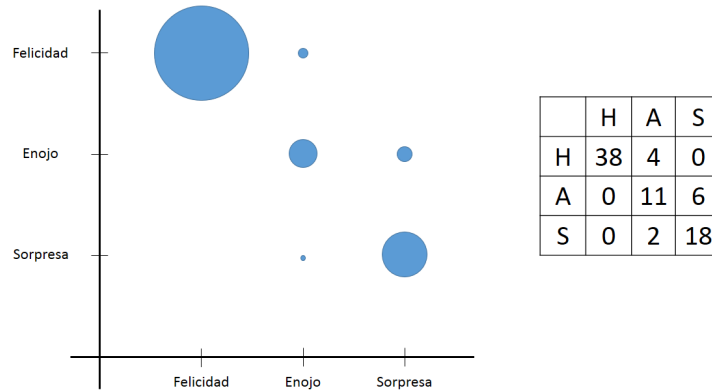


Figura 4.6: Gráfica de dispersión para Naïve Bayes con CP.

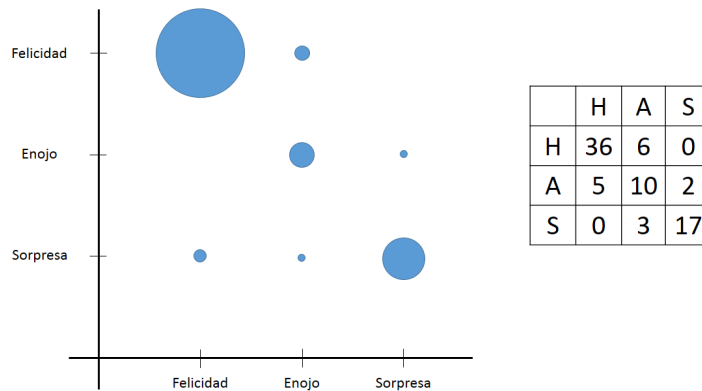


Figura 4.7: Gráfica de dispersión para Naïve Bayes CC.

El segundo clasificador con el que se experimentó fue SVM, con este clasificador se cambió el tipo de kernel utilizado, se utilizó el kernel de tipo RBF y otro de tipo polinomial, las matrices de confusión y gráficas de dispersión para estos experimentos se muestran en las Figuras 4.8 a la 4.11. El experimento con mejor precisión fue utilizando el kernel polinomial.

Finalmente se experimentó con redes neuronales, para este clasificador se realizó un ajuste de los parámetros de éste (momentum, tasa de aprendizaje, neuronas en la capa oculta y número de épocas), se establecieron rangos para el momentum (0

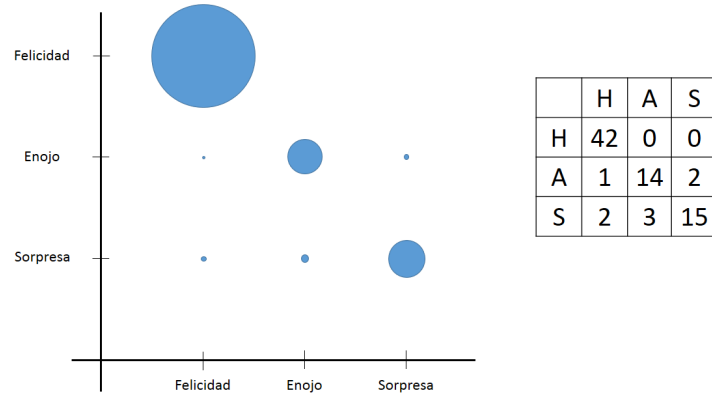


Figura 4.8: Gráfica de dispersión para SVM con kernel polinomial y CP.

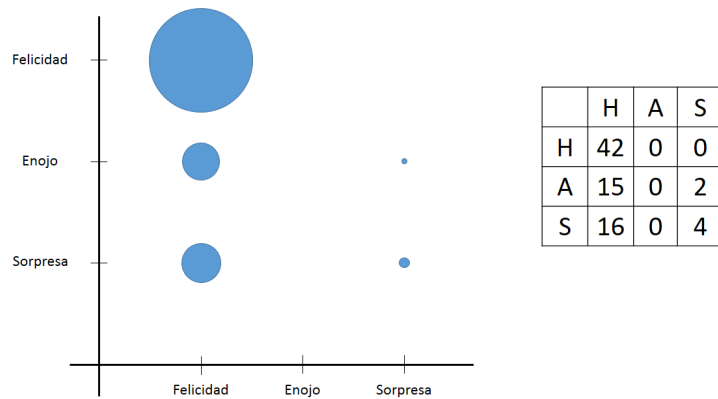


Figura 4.9: Gráfica de dispersión para SVM con kernel RBF y CP.

a 0.4), tasa de aprendizaje (0 a 0.6), neuronas en la capa oculta ($\pm \frac{\text{atributos} + \text{clases}}{2}$ respecto a $\frac{\text{atributos} + \text{clases}}{2}$) y se experimentaron con 500, 1000 y 1500 épocas. Los mejores resultados respecto a la precisión obtenida se muestran en la Tabla 4.6 con CP. Los mejores parámetros se muestran en las dos primeras filas. Sin embargo, la matriz de confusión es diferente para los dos casos, dichas matrices junto con su respectiva gráfica de dispersión se muestran en las Figuras 4.12 y 4.13.

La única diferencia entre las dos matrices de confusión es que en la matriz mostrada en la Figura 4.12 y la Figura 4.13 solo hubo confusión entre sorpresa y enojo, mientras que en la matriz mostrada en b) tiene una confusión con sorpresa y felicidad.

También se realizaron experimentos con CC, de la misma manera que se realizaron los experimentos con CP se establecieron los mismos parámetros, los mejores parámetros se muestran en la Tabla 4.7, de la misma manera se muestran las ma-

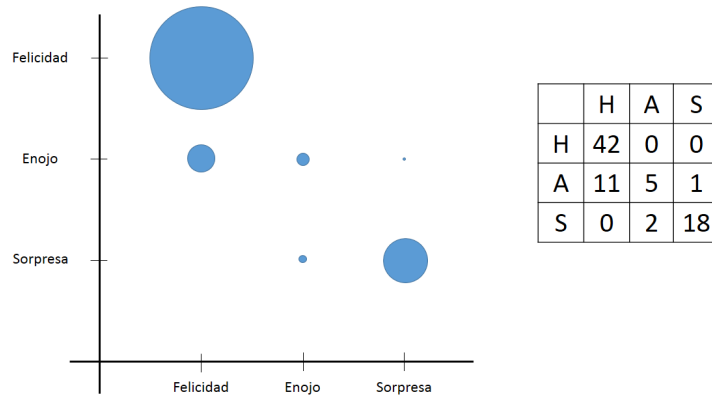


Figura 4.10: Gráfica de dispersión para SVM con kernel polinomial y CC.

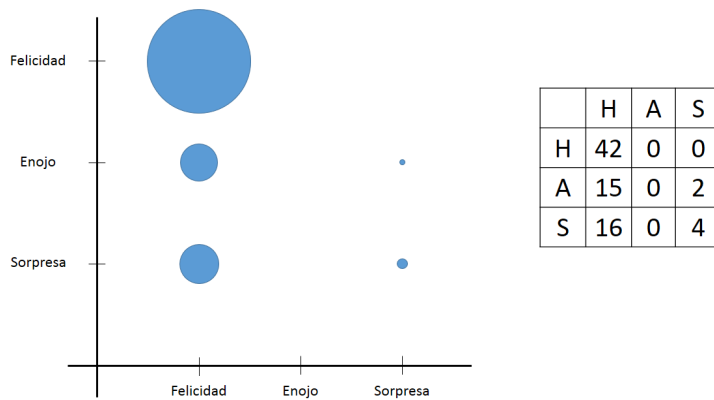


Figura 4.11: Gráfica de dispersión para SVM con kernel RBF y CC.

trices de confusión junto a su gráfica de dispersión en las Figuras 4.14 y 4.15.

Los mejores resultados para la etapa de clasificación son los experimentos realizados con redes neuronales y teniendo como entrada CP con un 92.4 % y los parámetros mostrados en la fila 1 y 2 de la Tabla 4.6.

Tabla 4.6: Resultados de los experimentos con CP y redes neuronales.

Epocas	Neuronas C.O.	Momentum	Learning rate	Precisión
500	39	0.3	0.6	92.4 %
500	44	0.4	0.6	92.4 %
500	45	0.16	0.58	92.3 %
1000	45	0.11	0.6	91.9 %
1500	41	0.1	0.27	91.8 %

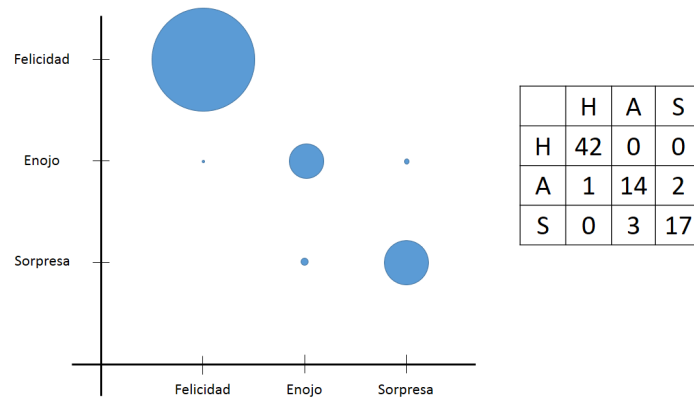


Figura 4.12: Gráfica de dispersión para NN con CP con los parámetros de la primera fila de la Tabla 4.6.

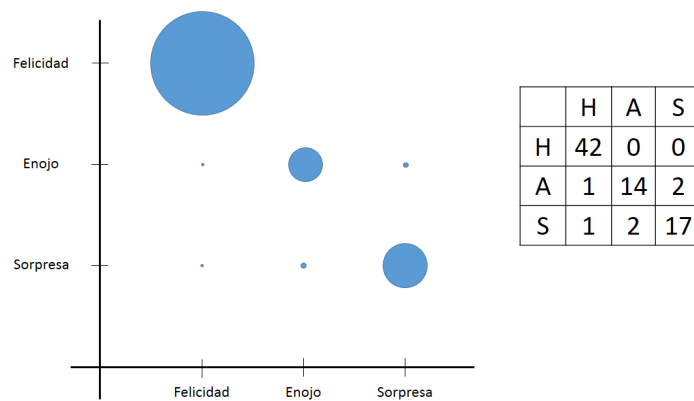


Figura 4.13: Gráfica de dispersión para NN con CP con los parámetros de la segunda fila de la Tabla 4.6.

Tabla 4.7: Resultados de los experimentos CC y redes neuronales.

Epocas	Neuronas C.O.	Momentum	Learning rate	Precisión
1000	36	0.14	0.04	84.81 %
500	41	0.01	0.11	83.54 %
500	36	0.13	0.04	82.53 %
1000	45	0.39	0.08	79.74 %
1500	35	0.18	0.15	78.48 %

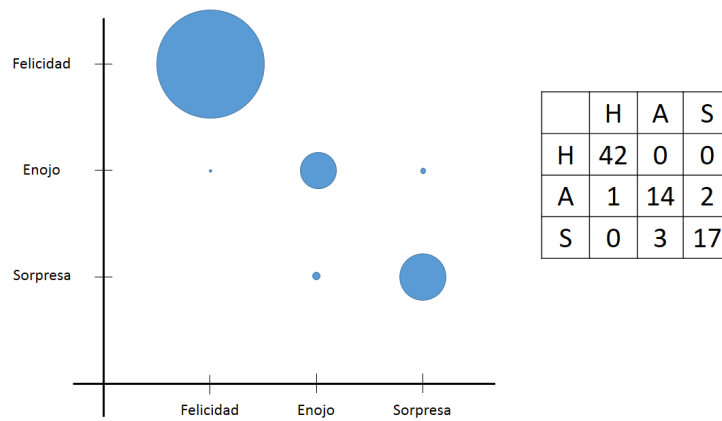


Figura 4.14: Gráfica de dispersión para NN con CC con los parámetros de la primera fila de la Tabla 4.6.

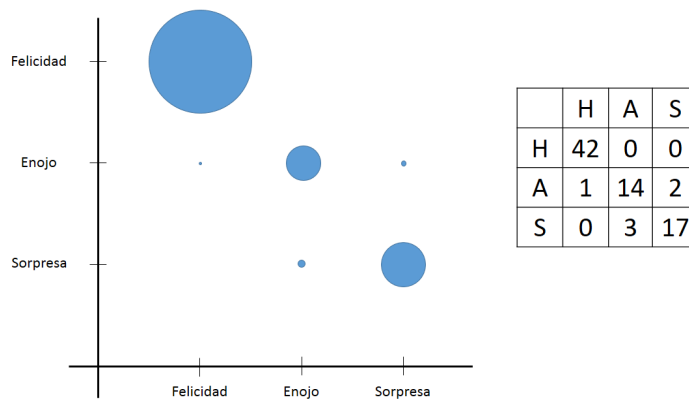


Figura 4.15: Gráfica de dispersión para NN con CC con los parámetros de la segunda fila de la Tabla 4.6.

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

En este trabajo de tesis se presenta una metodología para la detección de estados de ánimo a partir de imágenes digitales, donde se desarrollaron las etapas: pre-procesamiento, extracción de características y clasificación.

En la etapa de pre-procesamiento se presentan los algoritmos propuestos para la segmentación de las ROIs en imágenes de rostros (en este caso cejas y boca), adicionalmente se presenta una comparación con un conjunto de control en el que se segmentaron manualmente las ROIs en el que se compararon los coeficientes de interpolación respecto a las imágenes segmentadas con nuestro enfoque para cada región, en los resultados obtenidos se observó que la diferencia entre los coeficientes encontrados con el conjunto de control respecto a los encontrados con nuestro enfoque fueron muy pequeños por lo que las regiones segmentadas son muy parecidas respecto al resultado obtenido, además se presenta la comparación con otro trabajo para la segmentación de la boca y se concluyó que nuestro enfoque es competitivo respecto a otros procesos reportados en la literatura.

Para la etapa de extracción de características se utilizó el método diferencias sucesivas de Newton para encontrar polinomios característicos de las ROIs, para la región de las cejas se utiliza un polinomio cúbico, mientras que para la región de la boca se utiliza un polinomio cuadrático para la parte inferior y otro para la parte superior. Una vez encontrados los polinomios se extrajeron diez puntos de cada uno, tomando en cuenta los puntos extremos de las ROIs. Este tipo de características no se han reportado en otro trabajo, la extracción de estos atributos es un forma novedosa de caracterizar las ROIs.

En la etapa de clasificación se realizaron experimentos con los clasificados naïve bayes, máquinas de vectores de soporte y redes neuronales, variando los parámetros de cada uno de los clasificadores para encontrar la mejor precisión. La clasificación se realizó con tres de los seis estados de ánimo (Felicidad, Enojo y Sorpresa) y se utilizaron dos conjuntos de características uno con los coeficientes de los polinomios y otro con diez puntos por cada polinomio.

Para las redes neuronales se realizó un proceso de ajuste de parámetros en el que éstos fueron variados para determinar cual era la mejor precisión que se podría

obtener con los conjuntos de características extraídos, dichos parámetros se evaluaron de manera paralela para reducir tiempos de ejecución.

Se realizó un proceso de paralización en las etapas de pre-procesamiento y clasificación. Para el pre-procesamiento se utilizó la librería MPI y se envía una imagen RGB a cada proceso, el cual se encarga de segmentar las imágenes, dejando como resultado una imagen binaria por cada ROI. Para la etapa de clasificación se realizó una implementación con hilos de ejecución para evaluar los parámetros de las redes neuronales.

La principal desventaja del enfoque es el tiempo de ejecución, ya que el tiempo de pre-procesamiento depende del tamaño de las imágenes de entrada. El algoritmo basado en DBScan es muy costoso computacionalmente, debido a que se tienen que analizar un gran número de píxeles. Tomando en cuenta esto, el enfoque propuesto se necesita mejorar para ser aplicado en tiempo real.

Como trabajo futuro se pretende extraer características de otras ROIs (ojos y nariz), junto con características globales (distancias entre ROIs), además de la captura de una base de datos que tenga imágenes de los seis estados de ánimo, así como tomar vídeo tratando de generar algunos de estados de ánimo de manera espontánea. Finalmente, se necesita mejorar el pre-procesamiento debido a que el tiempo de ejecución es alto; se pueden aplicar algoritmos paralelos basados en GPU o disminuir el número de píxeles que se deben analizar aplicando filtros morfológicos.

PUBLICACIONES DERIVADAS DE ESTE TRABAJO DE TESIS

- Jesús García Ramírez, J. Arturo Olvera López, Ivan Olmos Pineda, Georgina Flores Becerra, Adolfo Aguilar Rico: Thresholding Approach based on GPU for Facial Expression Recognition, *Research in Computer Science, Advances in Pattern Recognition*, Vol. 112 pp. 41-51 (2016)
- Jesús García Ramírez, Ivan Olmos Pineda, J. Arturo Olvera López, Manuel Martín Ortíz: Edge Detection for Facial Expression Recognition, *CEUR Workshop*, Vol. 1659 pp. 66-73 (2016)
- Jesús García-Ramírez, J. Arturo Olvera-López, Ivan Olmos-Pineda, Manuel Martín-Ortíz: ROIs Segmentation in Facial Images based on Morphology and Density Concepts. In: *Proceedings of Mexican Conference on Pattern Recognition 2017*, pp. 169-178, LNCS 10267, Springer (2017)
- Jesús García Ramírez, Ivan Olmos Pineda, J. Arturo Olvera-López, Manuel Martín Ortíz: Facial Expression Recognition Using Interpolation Features, *Research in Computing Science, Advances in Pattern Recognition*, Vol. 133 pp. 19-28 (2017)

REFERENCIAS

- [1] Xiaoming C., Wushan C.: *Facial expression recognition based on edge detection*. International Journal of Computer Science & Engineering Survey 6(2), 1-9 (2015)
- [2] Pantic M., Valstar M., Rademaker R., Maat L.: *Web-based database for facial expression Analysis*. In: Proceedings of International Conference on Multimedia and Expo, pp. 5-10 (2005)
- [3] Corneau C., Olliu M., Cohn J., Escalera S.: *Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-related Applicants*, IEEE Transaction on Pattern Analysis and Machine Intelligence 99, 2-20 (2015)
- [4] Aswathy R.: *A Literature review on Facial Expression Recognition Techniques*, IOSR Journal of Computer Engineering 11(1), 61-64 (2013)
- [5] Kumari J., Rajesh J., Pooja K.: *Facial Expression Recognition: A survey*, In: Proceedings of 2nd. International Symposium on Computer Vision and the Internet (VisionNet 2015), pp. 486-491 (2015)
- [6] Ekman P.: *Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique*. Psychological Bulletin 115(2), pp. 268-287 (1994)
- [7] Cohn J., Ambadar Z., Ekman P.: *Observer-based measurement of facial expression with the Facial Action Coding System*. In: J. A. Coan & J. J. B. Allen (Eds.), The handbook of emotion elicitation and assessment. New York, pp. 203-221 (2007)
- [8] Torres L., Reutter J., Lorente L.: *The importance of the color information in face recognition*. In: Proceedings of the International Conference on Image Processing, (ICIP 99), pp. 627-631 (1999)
- [9] Bourel F., Chibelushi C., Low A.: *Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localized Facial Dynamics*. In: Proceedings of the Fifth IEEE International Conference Automatic Face and Gesture Recognition, pp. 106-111 (2002)
- [10] Huiquin Z., Osamu Y.: *A Fusion Approach for Facial Expression Using Local Binary Pattern and a Pseudo 3D Face Model*. In: Proceedings of International

- Conference on Software Engineering, Artificial Intelligence, Networking Parallel/Distributed Computing, pp. 121-126 (2013)
- [11] Essa I., Pentland A.: *Coding Analysis, Interpretation, and recognition of facial expression*, IEEE Transaction on Pattern Analysis and Machine Intelligence 19(7), 757-763 (1997)
- [12] Sariyanidi E., Gunes H., Cavallaro A.: *Automatic Analysis of Facial Affect: A survey of Registration, Representation, and Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence 37(6), 1113-1133 (2015)
- [13] Wenfei G., Cheng X., Venkatesh V., Dong H., Han L.: *Facial expression recognition using radial encoding of local Gabor Features and classifier synthesis*, Pattern Recognition, 45(1), 80-91 (2012)
- [14] Tang H., Huang T.: *3D facial expression recognition based on automatically selected features*. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR 2008), pp. 1-8 (2008)
- [15] Sandbach G., Zafeiriou S., Pantic M.: *Local normal binary patterns for 3D facial action unit detection*. In: proceedings of the International Conference of Image Processing (ICIP 2012), pp. 1813-1816 (2012)
- [16] Littlewort G., Whitehill J., Wu T., Fasel I., Frank M., Movellan J., Bartlett M.: *The computer expression recognition toolbox (CERT)*. In: Proceedings of Face and Gesture Recognition (FG 2011), pp. 298-305 (2011)
- [17] Cohen I., Sebe N., Gozman F, Cirelo M., Huang T.: *Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data*. In: Proceedings of Computer Vision and Pattern Recognition (CVPR 2003), pp. 595-601 (2003)
- [18] Tian Y., Kanade T., Cohn J.: *Recognizing Action Units for Facial Expression Analysis*, Transactions on Pattern Analysis and Machine Intelligence 232(2), 97-115 (2001)
- [19] Ghimire D., Jeon S, Lee J., Park S.: *Facial Expression Recognition based on Local Region Specific Features and Support Vector Machines*, Multimedia Tools and Applications, 75(1), pp. 1-19 (2016)
- [20] Sabu G.: *Geometrical Model of Facial Regions and CUDA based Parallel Face Segmentation for Emotion Recognition*, International Journal of Applied Engineering Research 11(9), 6740-6752 (2016)
- [21] Burger W., Burge M.: *Principles of Image Processing (Fundamental Techniques)*, First Edition (2009)
- [22] Bellman R. E.: *An Introduction of Artificial Intelligence: Can Computers Think?*, San Francisco : Boyd & Fraser, (1978)

-
- [23] Kurzweil R.: *The Age of Intelligent Machines*, First Edition, Viking (1999)
- [24] Winston P.: *Artificial Intelligence*, Third Edition, Addison Wesley (1992)
- [25] Poole d., Mackworth A. Goebel R.: *Computational Intelligence: a Logical Approach*, First Edition, Oxford University Press (1998)
- [26] Jiawei H., Micheline K. Jian P.: *Data Mining, Concepts and Techniques*, Third Edition, Elsevier Inc.(2012)
- [27] Mitchell T.: *Machine Learning*, Third Edition, McGrawHill (2013)
- [28] García J. R., Martínez J. F.: *Extension to c-means algorithm for the use of similarity functions*. In: Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery, pp. 354-359 (1999)
- [29] Nielsen F.: *Introduction to HPC with MPI for Data Science*, First Edition, Springer International Publishing (2016)
- [30] Doron L.: *Introduction to numerical analysis (Technical Report)*, Department of Mathematic and Center for Scientific Computation and Mathematical Modeling (2010)
- [31] Lyons M., Akemastu S., Kamachi M., Gyoba J.: *Coding Facial Expressions with Gabor Wavelets*, In proceedings of: 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205 (1998)
- [32] Sanderson C., Lovell B.: *Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference.*, Lecture Notes in Computer Science (LNCS), Vol. 5558, pp. 199-208 (2009)
- [33] Viola P., Jones P.: *Rapid Object Detection using a Boosted Cascade of Simple Features*, In: Proceedings of Computer Vision and Pattern Recognition, pp. 511-518 (2001)
- [34] Jones M., Viola P.: *Fast Multi-view Face Detection*, In: Proceedings of Computer Vision and Pattern Recognition, pp. 1-11 (2003)
- [35] Ester M., Kriegel H., Sander J., Xu X.: *A density-based algorithm for discovering clusters in large spatial databases with noise* in Kdd 96 (34), 226-231 (1996)
- [36] Gonzalez R., Woods R.: *Digital Image Processing*, Third Edition, Pearson Education (2008)
- [37] Otsu N.: *A threshold selection method from gray-level histogram*, IEEE Transaction on systems, man, and cibernetics 9(1), 62-66 (1979)
- [38] Le H., Savvides M.: *A novel Shape Constrained Feature-based Active Contour model for lips/mouth segmentation in the wild*, Pattern Recognition, 54(1) 23-33 (2016)