



Universidad Politécnica de Puebla

Maestría en Ingeniería en Sistemas y Cómputo Inteligente

**SISTEMA DE VISIÓN ARTIFICIAL PARA EL
RECONOCIMIENTO EMOCIONES FACIALES
APLICADO AL ESTUDIO DE EXPRESIONES
DEL USUARIO**

Alumno:

Emmanuel Baleón Flores

Asesor:

Dr. Jorge de la Calleja Mora

Índice general

1. Planteamiento del problema	6
1.1. Introducción	6
1.2. Objetivos	7
1.3. Justificación y Motivación	7
1.4. Contribuciones esperadas	8
2. Marco teórico	9
2.1. Visión por computadora	9
2.1.1. Procesamiento de imágenes	10
2.1.2. Reconocimiento de objetos	12
2.1.3. Reconocimiento de rostros	13
2.2. Métodos de extracción de características	14
2.2.1. Patrones locales binarios	15
2.2.1.1. LBP Uniforme	16
2.2.2. Filtros de Gabor	17
2.2.3. Análisis de componentes principales	19
2.3. Aprendizaje Automático	20
2.3.1. Máquinas de vectores de soporte	20
2.3.2. Redes neuronales artificiales	23
2.3.3. K-vecinos más cercanos	24
2.4. Métricas de evaluación	26
2.4.1. Matriz de confusión	26
2.5. OpenCV	28
2.6. Expresiones faciales y emociones	28
2.7. Experiencia del usuario y usabilidad	29
2.8. Estado del Arte	30
3. Metodología	35
3.1. Recolección del conjunto de imágenes	35
3.2. Pre-Procesamiento de imágenes	36
3.3. Extracción de características	36
3.4. Clasificación de expresiones usando aprendizaje automático	39

<i>ÍNDICE GENERAL</i>	2
4. Resultados Experimentales	40
4.1. Resultados experimentales de la primera etapa	40
4.1.1. Resultados de la clasificación utilizando LBP básico	41
4.1.2. Resultados de la clasificación utilizando PCA	44
4.1.3. Resultados de la clasificación utilizando Filtros de Gabor	44
4.1.4. Resultados de la clasificación utilizando LBPUr	49
4.2. Resultados experimentales de la segunda etapa	52
4.3. Comparación de resultados	53
5. Conclusiones y trabajo futuro	58
A. Sistema de reconocimiento de expresiones faciales	60
A.1. Funcionamiento General	60

Índice de figuras

2.1. Etapas de la VC	10
2.2. Funcionamiento general del procesamiento de imágenes	11
2.3. Ejemplo de procesamiento de imágenes	12
2.4. Ejemplos de reconocimiento específico y general	13
2.5. Detección de rostros usando OpenCV	14
2.6. Ejemplo del operador LBP básico	15
2.7. Método LBP e histograma de código LBP	16
2.8. Descripción grafica del método LBP aplicado a imágenes con rostros . .	16
2.9. Ejemplos de medición de uniformidad	17
2.10. LBP uniforme	17
2.11. Filtro de Gabor de 2 dimensiones aplicado a una imagen	18
2.12. Banco de filtros clásico de Gabor	19
2.13. Funcionamiento general del método PCA	20
2.14. Ejemplo de SVM en un problema linealmente separable	21
2.15. Ejemplo de la clasificación linealmente separable utilizando SVM	22
2.16. Ejemplo de SVM en un problema no linealmente separable	23
2.17. Ejemplo de redes neuronales	24
2.18. Ubicación de un dato nuevo entre datos conocidos [Aha <i>et al.</i> , 1991] . .	25
2.19. Predicción de la clase de un dato nuevo con respecto al vecino más cercano[Aha <i>et al.</i> , 1991]	25
2.20. Predicción de la clase de un dato nuevo con respecto al vecino más cercano, los datos conocidos contienen ruido [Aha <i>et al.</i> , 1991]	25
2.21. Predicción de la clase de un dato nuevo con respecto a los cinco vecinos más cercanos, los datos conocidos tienen ruido [Aha <i>et al.</i> , 1991]	26
2.22. Matriz de confusión de la clasificación entre manzanas y plátanos . . .	27
3.1. Metodología general	35
3.2. Detección y recorte de rostros del conjunto de imágenes creado	36
3.3. Etapas de los Métodos LBP y LBPUr	37
3.4. Procesamiento de imágenes con PCA	38
3.5. Etapas del método de Gabor	39

4.1. Comparación de los mejores resultados para la clasificación del conjunto C1	55
4.2. Comparación de los mejores resultados para la clasificación del conjunto C2	55
4.3. Comparación de los mejores resultados para la clasificación del conjunto C3	56
4.4. Comparación de los mejores resultados para la clasificación del conjunto C4	56
4.5. Resultados experimentales de la clasificación del conjunto M1	57
4.6. Resultados experimentales de la clasificación del conjunto Ms1	57
A.1. Ventana principal del sistema de reconocimiento	61
A.2. Ventana de configuración	62
A.3. Ventana de información general del sistema	62
A.4. Ejemplo del reconocimiento de la expresión feliz	64
A.5. Ejemplo del reconocimiento de la expresión normal	65
A.6. Ejemplo del reconocimiento de la expresión repugnancia	66

Índice de tablas

2.1. Trabajos relacionados	34
4.1. Descripción general de los conjuntos de entrenamiento binarios.	41
4.2. Descripción general del conjunto de entrenamiento multiclase.	41
4.3. Configuración de los algoritmos de clasificación para la experimentación con el método LBP básico	42
4.4. Promedios de métricas utilizando LBP básico aplicado a los conjuntos C1, C2, C3 y C4	43
4.5. Configuración de los algoritmos de clasificación para la experimentación con el método PCA	45
4.6. Promedios de métricas utilizando PCA aplicado a los conjuntos C1, C2, C3 y C4	46
4.7. Configuración de los algoritmos de clasificación para la experimentación con el método filtros de Gabor	47
4.8. Promedios de métricas utilizando filtros de Gabor aplicado a los conjuntos C1, C2, C3 y C4	48
4.9. Configuración de los algoritmos de clasificación en Weka para los conjunto C1,C2,C3,C4,M1	50
4.10. Promedios de métricas utilizando LBPUr aplicado a los conjuntos C1, C2, C3 y C4	51
4.11. Descripción general de conjunto Ms1	52
4.12. Configuración de SVM utilizada en OpenCV para la experimentación de LBPUr	53
4.13. Configuración de SVM utilizada en OpenCV para la experimentación de Filtros de Gabor	53
4.14. Resultados experimentales de SVM utilizando LBPUr y Filtros de Gabor	53

Capítulo 1

Planteamiento del problema

1.1. Introducción

La visión por computadora forma parte de la inteligencia artificial y está definida como una disciplina que involucra diferentes ciencias como la física, las matemáticas, la electrónica y la computación [García, 2008]. La implementación de algoritmos y el procesamiento de imágenes hacen que esta sea una ciencia en constante crecimiento.

El objetivo principal de la visión por computadora o visión artificial es imitar la capacidad que tienen los humanos para ver una escena y entenderla [García, 2008]; por ello, es necesario implementar el procesamiento digital de imágenes, clasificación de imágenes, extracción de características y reconocimiento de patrones, sólo por mencionar algunas de las técnicas implementadas por la visión por computadora.

Generalmente el reconocimiento mediante visión artificial es empleado para la identificación de objetos, cuyo fin es encontrar o identificar objetos dentro de imágenes o videos. Dentro del reconocimiento de objetos ha surgido el interés por reconocer rostros, con el objetivo de desarrollar sistemas con múltiples aplicaciones, como en sistemas de video vigilancia, sistemas de entretenimiento y sistemas enfocados al comportamiento humano [Sucar y Gómez, 2011].

Con base a lo antes descrito, para poder identificar expresiones faciales como primer objetivo es necesario identificar el rostro de una persona, para ello los enfoques más usados son: los métodos basados en características geométricas y los métodos basados en apariencias. El enfoque basado en características geométricas representa la forma y lugares de los componentes faciales. Por otra parte, los métodos basados en apariencia se enfocan en filtros de imágenes, que sirven para extraer cambios en la apariencia facial [Shan *et al.*, 2009].

Partiendo de lo anterior, es necesario implementar técnicas de aprendizaje automático para realizar el reconocimiento de expresiones faciales. El aprendizaje automático

es definido por [Mitchell, 1997] como una disciplina que integra a la inteligencia artificial, donde el objetivo es desarrollar programas computacionales que puedan aprender a realizar una tarea automáticamente y con respecto a la experiencia mejoren su desempeño. El aprendizaje es una característica que los seres humanos tenemos desarrollada y el método es simple, tal como adquirir información para posteriormente convertirla en conocimiento, habilidades y descubrir nuevos hechos [Mitchell, 1997], estos procesos son estudiados y modelados en computadoras por el aprendizaje automático (*machine learning*).

En este trabajo de investigación se presenta un prototipo de un sistema que identifica tres expresiones faciales básicas (Feliz, normal y repugnancia). El objetivo es implementarlo en usuarios que sean sometidos a pruebas de usabilidad moderada de sistemas o programas informáticos. Para el desarrollo del sistema se estudiaron los métodos de LBP, Filtros de Gabor y Análisis de componentes principales como caracterizadores de las expresiones faciales. Mientras que para la clasificación de características se usaron los algoritmos de K-vecinos más cercanos, Redes neuronales artificiales y máquinas de vectores de soporte. En otro orden de ideas, cuando el sistema finaliza alguna prueba, el prototipo creará un reporte general de todas las expresiones registradas durante la sesión.

1.2. Objetivos

El objetivo general de este trabajo de investigación fue el desarrollar un sistema de reconocimiento de expresiones faciales que apoye a las pruebas de usabilidad moderada, usando visión por computadora y aprendizaje automático. Para lograr el objetivo general se tuvieron los siguientes objetivos particulares.

- 1 Construir una interfaz que permita administrar el prototipo del sistema para el reconocimiento de al menos dos expresiones faciales
- 2 Caracterizar imágenes de expresiones faciales usando los métodos de *LBP* y *Filtros de Gabor* para comparar su desempeño
Caracterizar imágenes de rostros usando los métodos *LBP* y *Filtros Gabor* para evaluar su desempeño en el reconocimiento automático
- 3 Evaluar al menos tres métodos de aprendizaje automático para el reconocimiento de las expresiones faciales

1.3. Justificación y Motivación

El reconocimiento de expresiones faciales dentro de la interacción entre los humanos es de gran importancia para la comunicación. Dentro de la comunicación interpersonal la

identificación de emociones se presenta en un 93 % de acuerdo a [Ratilf y Petterson, 2010] y se ha descubierto que la interpretación de emociones puede alterar significativamente el sentido de una conversación. Las expresiones faciales dan indicios de las emociones que una persona experimenta en el momento de interactuar con otras personas o bien cuando utiliza alguna herramienta de hardware o software. Reconocer las emociones de un humano interactuando con un dispositivo tecnológico o una computadora permite saber cuál fue su experiencia mientras utilizaba el sistema.

Dentro de la Interacción humano computadora, las pruebas realizadas a usuarios sobre la experiencia que tiene al usar un sistema o software permite conocer cuál es la expectativa del usuario con respecto al mismo. Según [Brave y Nass, 2002] las emociones afectan la atención y el rendimiento del usuario cuando realizan valoraciones de un producto, La importancia de identificar las expresiones faciales durante una prueba de usabilidad es poder conocer los motivos que detonaron determinada emoción. El diseño de un producto puede provocar una emoción o emociones de forma explícita.

Por otra parte, el desarrollo de un sistema que pueda reconocer las expresiones faciales de un usuario mientras realiza una evaluación de software, permitiría obtener reportes de forma rápida y formaría parte de las herramientas utilizadas por los observadores de cada prueba.

1.4. Contribuciones esperadas

Para este trabajo las contribuciones esperadas al término del mismo son las siguientes:

1. Un prototipo de un sistema para el monitoreo de expresiones faciales
2. El histograma de las expresiones que fueron reconocidas durante una sesión de prueba
3. Una comparativa de los métodos de caracterización de imágenes de expresiones faciales
4. La evaluación de al menos dos métodos de aprendizaje automático para el reconocimiento de expresiones
5. Una base de datos de imágenes de expresiones creada con estudiantes de la universidad politécnica de Puebla

Capítulo 2

Marco teórico

La organización de este capítulo se expresa de la siguiente manera, entre las secciones 2.1 y 2.4, se describen temas que permiten el desarrollo e investigación de este proyecto como: visión artificial, métodos de extracción de características, aprendizaje automático y métricas de evaluación. En la sección 2.5 se presenta a OpenCV como una herramienta de programación que incluye métodos visión artificial y aprendizaje automático. En la sección 2.6 se exponen las emociones faciales universales definidas por [Ekman y Oster, 1981]. Dentro de la sección 2.7, se describe el área de experiencia de usuario, área donde se pretende impacte el proyecto, por último, se muestran los trabajos relacionados dentro de la sección 2.8.

2.1. Visión por computadora

La visión por computadora (VC) *Computer Vision del termino en inglés*, es una área derivada de la inteligencia artificial y es considerada por [Learned-Miller, 2011] la ciencia que permite dotar a una computadora o máquina con la capacidad de ver. También, la [BMVA, 2014] define a la VC como una extracción y análisis de información importante dentro de una o varias imágenes de forma automática, utilizando algoritmos y técnicas que permitan una percepción visual automática.

La VC está conformada por diferentes elementos, entre ellos se encuentra la memoria, la recuperación de información, el razonamiento, la estimación, el reconocimiento y la coordinación con otros elementos. Según [Learned-Miller, 2011], se considera que un sistema tiene un determinado grado de visión si tiene más de un componente de los descritos anteriormente. Por este motivo, la VC no puede considerarse como un elemento independiente, sino como un conjunto de elementos que forma un sistema que tiene como objetivo imitar las capacidades de ver de los seres vivos.

Para [Sucar y Gómez, 2011], una de las metas principales de la VC es extraer las características de una imagen para su descripción e interpretación. Por ejemplo:

- Determinar la posición y el tipo de objeto en una imagen
- Analizar un objeto para determinar su calidad
- Descomponer una imagen en partes
- Construir una representación tridimensional de un objeto

En la actualidad, la VC es una ciencia que se encuentra en sus primeras fases de desarrollo y tiene objetivos ambiciosos según [Vélez *et al.*, 2003]. Algunas de sus aplicaciones se encuentran en diversas tecnologías como sistemas de seguridad para el reconocimiento de personas, sistemas de detección de objetos implementados en robots de servicios, análisis de calidad de productos en sistemas industriales, entre otras.

Para [Vélez *et al.*, 2003], los sistemas de VC están compuestos por cuatro etapas:

- Captura de imágenes digitales por medio de algún sensor
- Procesamiento de imágenes mediante filtros o métodos geométricos
- Segmentación de la imagen para entender los elementos que componen la imagen
- Reconocimiento o clasificación de los segmentos previamente obtenidos

Estas etapas no siempre se desarrollan de forma secuencial, debido a que existe la posibilidad de realizar una retroalimentación como se muestra en la figura 2.1. Si se considera que alguna de las etapas presenta errores, una opción para corregirlos sería retroceder a etapas anteriores aun si es necesario retroceder a la etapa inicial (captura).

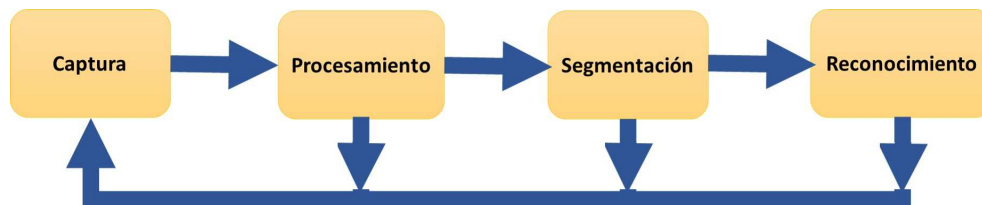


Figura 2.1. Etapas de la VC

2.1.1. Procesamiento de imágenes

El procesamiento de imágenes se considera como una sub-área de la VC ligada a diferentes ciencias como la física, las matemáticas y la computación. Generalmente el procesamiento de imágenes es considerada como sinónimo de la VC, sin embargo, los objetivos del procesamiento digital de imágenes son diferentes.[Sucar y Gómez, 2011] menciona que mejorar la calidad de las imágenes para posteriormente ser analizadas

o interpretadas es el objetivo principal del procesamiento de imágenes, Algunas aplicaciones del procesamiento de imágenes se enfocan en remover defectos por enfoque o movimiento, también en la mejora de propiedades como: color, textura y contraste.

En la Figura 2.2 se puede observar un esquema general del procesamiento de imágenes en su funcionamiento, donde la imagen de entrada es asignada por un usuario o sistema, esta imagen pasa a ser procesada para obtener una imagen de salida con una mejora en su calidad



Figura 2.2. Funcionamiento general del procesamiento de imágenes

La mayoría de proyectos relacionados con la VC como la fotografía y el reconocimiento de objetos por mencionar algunos, son auxiliados por el procesamiento de imágenes con el fin de obtener resultados aceptables [Szeliski, 2010]. La utilidad del procesamiento de imágenes abarca diversas áreas, por ejemplo, procesamiento de imágenes obtenidas para diagnóstico médico, procesamiento de imágenes para el reconocimiento de galaxias, la restauración de imágenes tomadas por celulares o cámaras digitales, entre otras.

En la Figura 2.3 se muestra un ejemplo del procesamiento de imágenes, donde se realiza la mejora de una imagen que tiene una apariencia oscura y se presenta como resultado una imagen con una calidad mejorada en su iluminación.

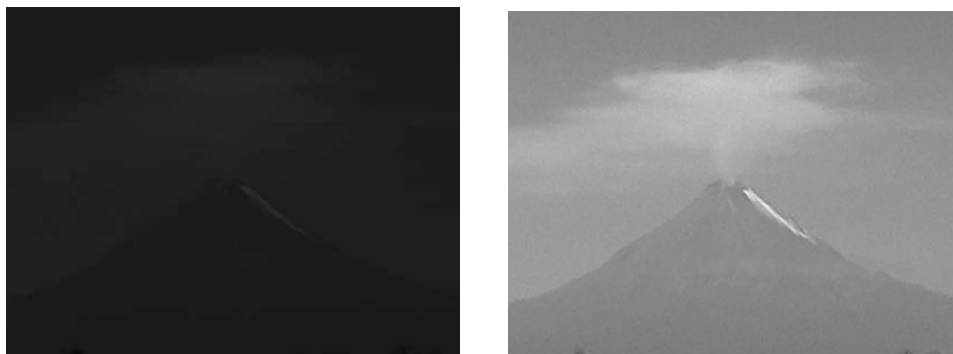


Figura 2.3. Ejemplo de procesamiento de imágenes

2.1.2. Reconocimiento de objetos

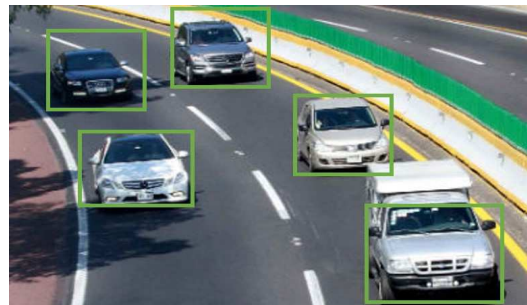
La mayoría de las tareas de la VC están dirigidas en reconocer y localizar objetos o lugares; sus aplicaciones son diversas, tales como la minería de datos de videos, identificación de objetos domésticos implementados en robots de servicio y en la búsqueda de contenido alfanumérico en imágenes [Grauman y Leibe, 2010]. De acuerdo con [Szeliski, 2010], el análisis de una escena y el reconocimiento de objetos por el cual está constituida una imagen es uno de los mayores desafíos que puede desarrollar una computadora dentro de la VC. En la actualidad el reconocimiento de objetos no es capaz de identificar la mayoría de elementos por el cual está conformada una imagen.

Por otra parte, una de las tareas principales para el reconocimiento de objetos es definir si el reconocimiento es específico o general. Para [Grauman y Leibe, 2010], el reconocimiento de objetos está dividido en dos tipos: el caso específico y el caso general. El caso específico se enfoca en identificar características de un objeto o lugar en particular. En el caso contrario, el reconocimiento de categoría general busca reconocer elementos de una misma clase conceptual.

En la figura 2.4 se puede observar un ejemplo del reconocimiento específico en el inciso a), donde el reconocimiento se centra en identificar a tres persona, y en el inciso b) se puede observar un ejemplo del reconocimiento general, que tiene como objetivo el reconocer automóviles.



(a) reconocimiento específico



(b) reconocimiento general

Figura 2.4. Ejemplos de reconocimiento específico y general

2.1.3. Reconocimiento de rostros

El reconocimiento de rostros es un tema derivado del reconocimiento de objetos y se considera del tipo general que tiene como meta identificar rostros en imágenes de acuerdo a sus características. Por otra parte, se considera que en los últimos años el reconocimiento de rostros mediante (VC) tiene mejor desempeño cuando las imágenes son frontales y con una iluminación uniforme. Para [Szeliski, 2010], el reconocimiento de rostros se utiliza en diferentes aplicaciones como identificación de personas, interacción humano computadora (IHC), monitoreo de pacientes, entre otras.

Para [Baggio, 2012], los sistemas de reconocimiento de rostros están formados por 4 pasos:

- **Detección de rostro:** localizar la región donde se encuentra la cara en una imagen
- **Procesamiento del rostro:** Proceso de ajuste de la cara que le permita obtener las características principales de cada rostro
- **Colección y aprendizaje de rostros:** recolectar un número determinado de rostros y luego aprender a reconocer cada uno de ellos

- **Reconocimiento de rostros:** verificar la similitud entre los rostros que se encuentran guardados y las imágenes obtenidas en tiempo real por la cámara

En la figura 2.5 se observa un ejemplo de la detección de rostros usando OpenCV.¹

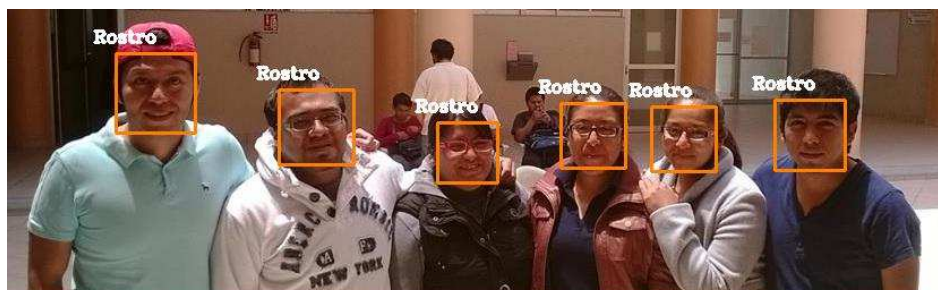


Figura 2.5. Detección de rostros usando OpenCV

2.2. Métodos de extracción de características

La extracción de características permite buscar atributos que representen de mejor forma un conjunto de datos. En otro contexto, [Nombela Escobar, 2011], define a la extracción de características como la técnica de obtener información asociada a los atributos de un conjunto de datos. En este proceso se generan todas las posibles combinaciones de atributos que permitan una mejor predicción y se eliminan los atributos que forman ruido en su predicción (atributos redundantes o irrelevantes se eliminan). Eliminar atributos irrelevantes o redundantes evita la llamada maldición de dimensionalidad, esta surge cuando el número de atributos es extenso en proporción a las instancias y esto genera muchos grados de libertad.

Para el reconocimiento de objetos y en particular para el reconocimiento de rostros, es importante la extracción de características debido a que éstas permitirán diferenciar entre los que es un rostro y lo que no. Así, la obtención de características o también conocidos como métodos de caracterización, son tareas que apoyan al proceso de clasificación, por este motivo se considera que el éxito del proceso de clasificación depende de las características obtenidas por los métodos de caracterización. A continuación se describen los métodos de caracterización implementados para este proyecto como LBP, LBP Uniforme, PCA y filtros de Gabor.

¹Opencv: una biblioteca de programación de uso libre que permite realizar tareas de visión por computadora.

2.2.1. Patrones locales binarios

[Pietikäinen *et al.*, 2011], describe a los patrones locales binarios como un método que en sus inicios fue utilizado como un operador de texturas fácil pero eficiente. Este método permite la etiquetación de píxeles de una imagen por vecindades de umbralización de cada pixel y considera un resultado como número binario. Posiblemente, la propiedad más importante dentro de su implementación en aplicaciones del mundo real para los patrones locales binarios o (LBP por sus siglas en inglés local binary patterns), es la invariancia en tonalidades de grises respecto a los cambios de iluminación como se puede ver en la figura 2.6 y la simplicidad computacional.

El funcionamiento del método LBP se centra en dividir una imagen en vecindades, usando el pixel central de cada vecindad como umbral. Para el LBP básico las vecindades esta formadas por 8 píxeles, lo cual presenta una combinación de $2^8 = 256$, de tal manera que los patrones locales binarios o códigos LBP se encuentran dentro del rango de 0 a 255. En cada vecindad se realiza una comparación entre el pixel central y sus vecinos, si la intensidad del pixel central es mayor o igual su vecino, el valor asignado es 1, de lo contrario el valor es 0; al final de compararse con todos los vecinos se obtiene un código LBP.

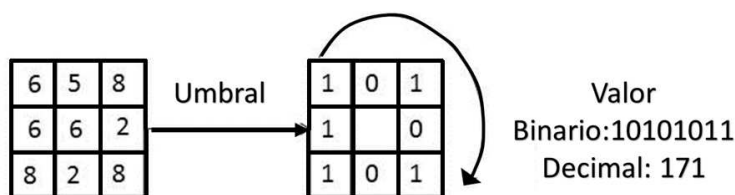


Figura 2.6. Ejemplo del operador LBP básico

Después de obtener los códigos LBP, se crea un histograma con respecto a las intensidades de los códigos LBP; como se observa en la figura 2.7, el histograma tiene un tamaño de 0 a 255 valores correspondiente al rango de los códigos LBP y el rango de intensidad está relacionada al número de píxeles o al tamaño de cada imagen procesadas por LBP. Por otra parte, LBP presenta diversas modificaciones para el análisis de un rostro. De acuerdo con [Pietikäinen *et al.*, 2011], la imagen de un rostro debe ser seccionada en regiones independientes donde cada región tenga un histograma, posteriormente se debe crear un histograma general con la concatenación de los histogramas independientes, tal como se puede observar en la figura 2.8.

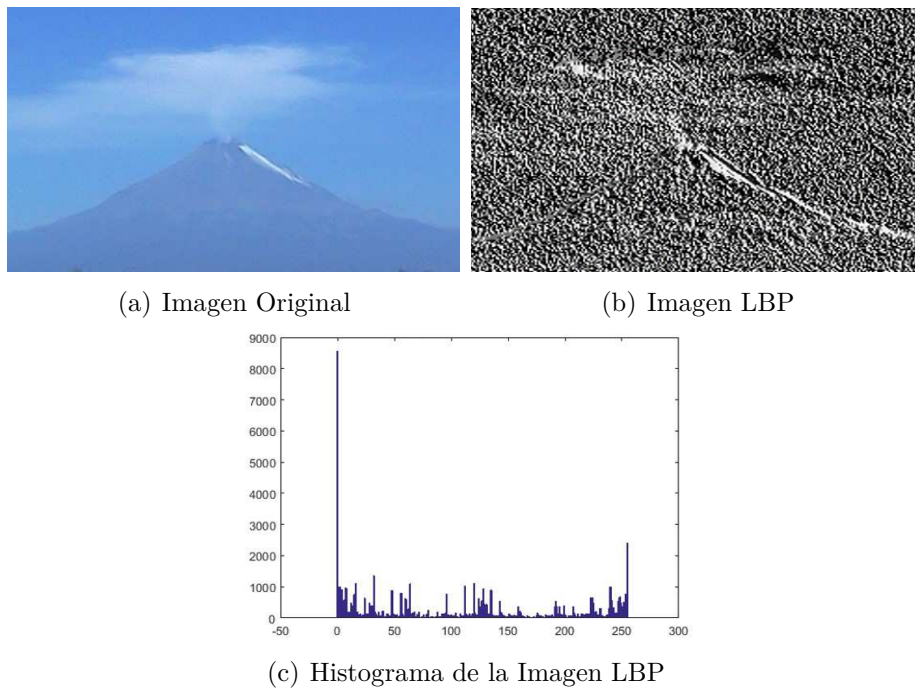


Figura 2.7. Método LBP e histograma de código LBP

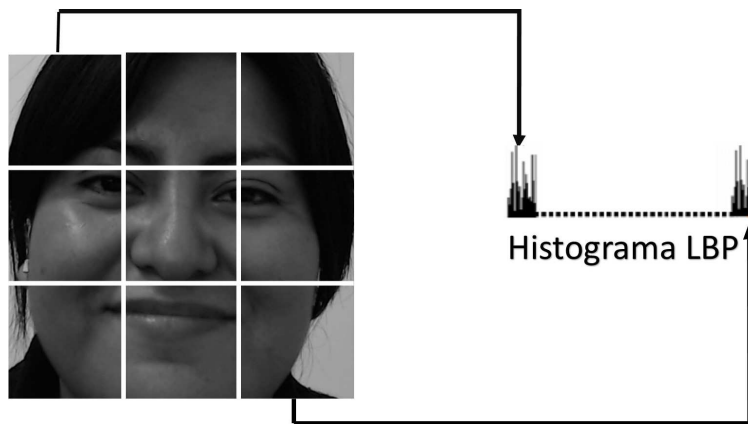


Figura 2.8. Descripción grafica del método LBP aplicado a imágenes con rostros

2.2.1.1. LBP Uniforme

LBP Uniforme es una variante del método LBP que permite la reducción del número de patrones de 256 a 59. Para [Pietikäinen *et al.*, 2011], la implementación de LBP uniforme considera una medida de uniformidad (U); esta se define como el número de transiciones $0 \leftrightarrow 1$ en la representación binaria de cada vecindad. La figura 2.9 muestra tres ejemplos de vecindades LBP con una representación de la medida de uniformidad, donde el inciso a) contiene una $U=4$, el inciso b) presenta una $U=2$ y por último el

inciso c) tiene una $U=0$.

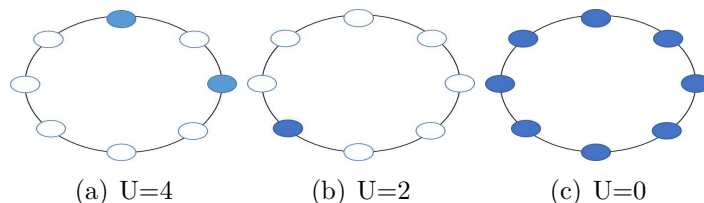


Figura 2.9. Ejemplos de medición de uniformidad

Para considerara que un código LBP es uniforme se toma en cuenta las siguientes premisas:

- Los patrones con $U=0$ o $U=2$, se les asigna un código individual (figura 2.10)
- El resto de los patrones que no cumpla con la descripción del punto anterior se les asigna un mismo código (son excluidos).

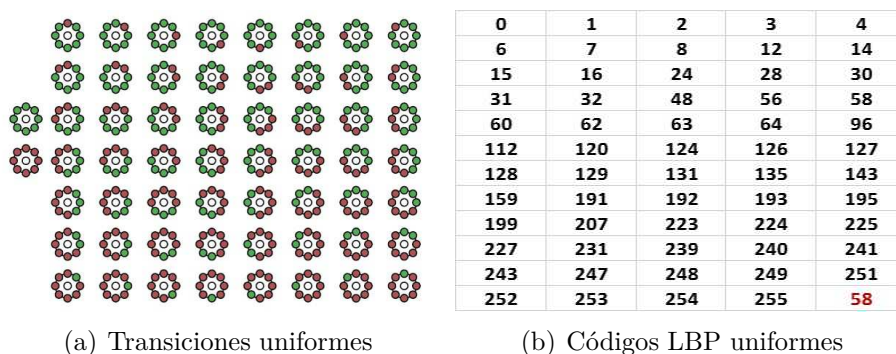


Figura 2.10. LBP uniforme

2.2.2. Filtros de Gabor

En la tesis desarrollada por [Mora Lazarini, 2008] , describe que las funciones de Gabor se ajustan al tipo de transformación producidas por algunas células de córtex visual humano. Por otra parte, Los filtros de Gabor son considerados una variante de la transformación de Fourier y se han implementado en segmentación de texturas, identificación retinal, detección de bordes, codificación de imágenes y representación de imágenes. Un filtro de Gabor de dos dimensiones se expresa como:

$$g(x, y) = s(x, y)w_r(x, y) \tag{2.1}$$

De acuerdo a la ecuación 2.1, donde $s(x, y)$ es una función sinusoidal compleja conocida como portadora y se expresa como se puede observar en la ecuación 2.2; mientras que la función $wr(x, y)$ es considerada como la envolvente gaussiana y se expresa como se muestra en la ecuación 2.3

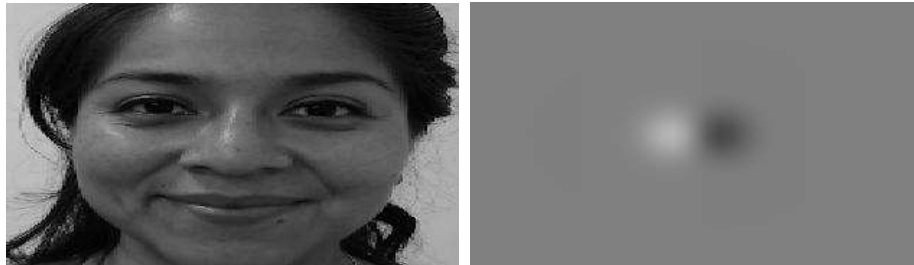
$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) \quad (2.2)$$

$$wr(x, y) = K \exp\left(-\pi\left(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2\right)\right) \quad (2.3)$$

La ecuación 2.4 muestra la forma general de un filtro de Gabor de dos dimensiones con respecto a la transformada de Fourier.

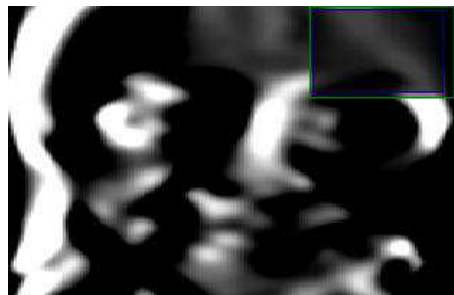
$$g(x, y) = \frac{K}{ab} \exp(j(-2\pi(x_0(u - u_0) + y_0(v - v_0)) + P)) \exp\left(-\pi\left(\frac{(u - v_0)_r^2}{a^2} + \frac{(u - v_0)_r^2}{b^2}\right)\right) \quad (2.4)$$

En la figura 2.11 se observa el funcionamiento de un filtro de Gabor de dos dimensiones aplicado a una imagen. Para implementar el método de Gabor en la clasificación de imágenes se crea un banco de filtros; el tamaño clásico de un banco es de 40 filtros, formado por 5 frecuencias y 8 orientaciones como se puede observar en la figura 2.12. De acuerdo con [Verma *et al.*, 2013], los parámetros de mayor importancia para un filtro de Gabor son el ángulo y la frecuencia.



(a) Imagen Original

(b) Filtro de Gabor



(c) Implementación del filtro de Gabor

Figura 2.11. Filtro de Gabor de 2 dimensiones aplicado a una imagen

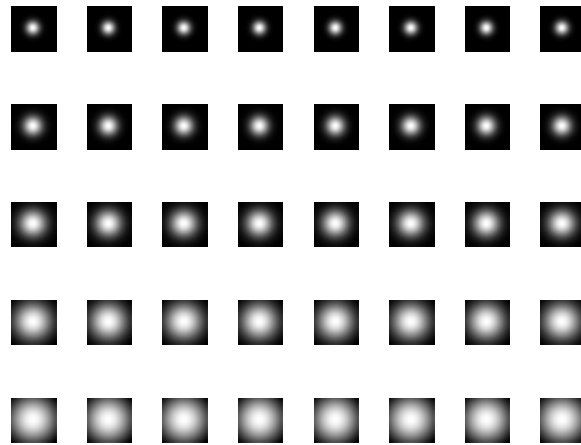


Figura 2.12. Banco de filtros clásico de Gabor

El espacio dimensional generado cuando se aplica los 40 filtros de Gabor a una imagen depende del tamaño de la misma. Si la imagen mide 80 x 80 pixeles se dice que tenemos 1600 características para dicha imagen, si se aplican los 40 filtros a la imagen, entonces el espacio dimensional crece a un total de $(1600 * 40)$ características.

2.2.3. Análisis de componentes principales

El método de Análisis de componentes principales o (PCA por sus siglas en inglés *Principal Component Analysis*) fue propuesto por Karl Pearson en el año 1901 y Harold Hotelling en el año 1933, este método permite la reducción dimensional de un conjunto de datos haciendo que la pérdida de información sea mínima, en otras palabras, PCA permite la reducción dimensional eliminando información irrelevante para la representación de ese conjunto.

La metodología de PCA se divide en 4 pasos. El primer paso es calcular el promedio de cada dato de un conjunto de entrenamiento y almacenarlo en un vector. Posteriormente se sustrae la medida de cada dato y se almacena en un vector. Después se calculan los eigenvalores y eigenvectores utilizando los vectores del paso 1 y 2. Como último paso, los eigenvectores o componentes principales son ordenados de acuerdo a los eigenvalores, entre mayor sea su valor mayor es la pérdida de información

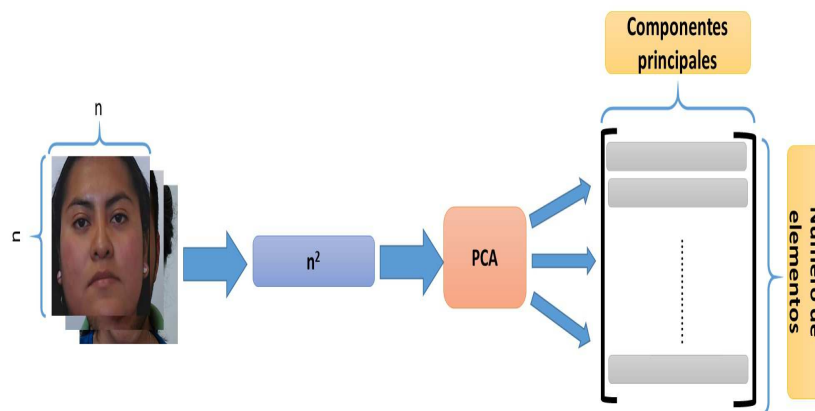


Figura 2.13. Funcionamiento general del método PCA

2.3. Aprendizaje Automático

El aprendizaje es una característica que los seres humanos tienen desarrollada y se enfoca en adquirir información para posteriormente convertirla en conocimiento, habilidades y descubrir hechos nuevos. De acuerdo con [Mitchell, 1997], estos procesos son estudiados y modelados en computadoras por el aprendizaje automático, que se considera como una disciplina derivada de la inteligencia artificial donde el objetivo es desarrollar programas computacionales que puedan aprender a realizar una tarea automáticamente y mejoren su desempeño con respecto a la experiencia.

El aprendizaje automático está dividido en dos clases: supervisado y el no supervisado. Para el aprendizaje supervisado se tiene un conjunto de características almacenadas en un vector con sus respectivas clases o etiquetas, estas son consideradas como el conjunto de entrenamiento. Por otro lado, el aprendizaje no supervisado contiene un conjunto de características sin referencia sobre sus clases. Es decir, el no supervisado tiene que definir las clases existentes dentro del conjunto de características y clasificarlo. En las siguientes subsecciones se describen de manera breve los algoritmos de aprendizaje automático utilizados para el desarrollo de este trabajo de investigación.

2.3.1. Máquinas de vectores de soporte

El algoritmo de máquinas de vectores de soporte o (SVM por sus siglas en inglés *support vector machines*) pertenece a la familia de métodos de *kernel machines* y se enfoca en construir un hiperplano en un espacio de alta dimensionalidad que permita separar las clases existentes [Shan *et al.*, 2009]. Para construir el hiperplano de separación correcto, SVM debe encontrar el margen máximo entre las clases a partir de vectores de soporte, estos vectores están compuestos por los puntos que mantiene la separación del plano. Todo elemento que no pertenezca a los vectores de soporte no

tiene relevancia sobre la definición de las fronteras del margen. La figura 2.14 presenta un ejemplo del método SVM para un problema linealmente separable.

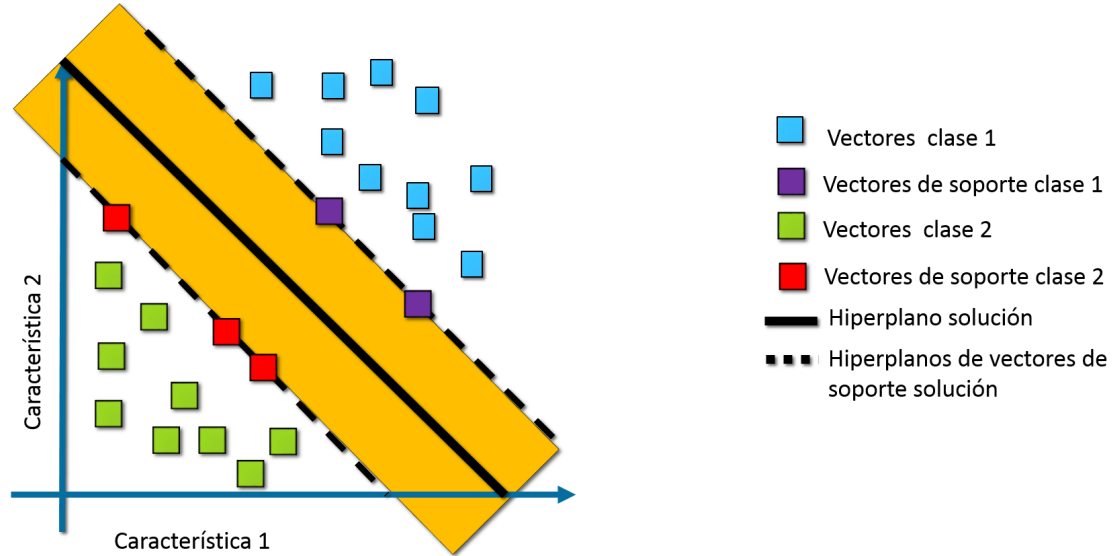


Figura 2.14. Ejemplo de SVM en un problema linealmente separable

Para cualquier algoritmo de entrenamiento supervisado es necesario utilizar un conjunto de entrenamiento y de acuerdo con SVM un conjunto de entrenamiento se define como:

$$L = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad (2.5)$$

Cada elemento del conjunto L es considerado un par (x_i, y_i) , donde:

$$(x_i, y_i) = \begin{cases} x_i = \mathbb{R}^n \\ y_i = \{1, -1\} \end{cases} \quad (2.6)$$

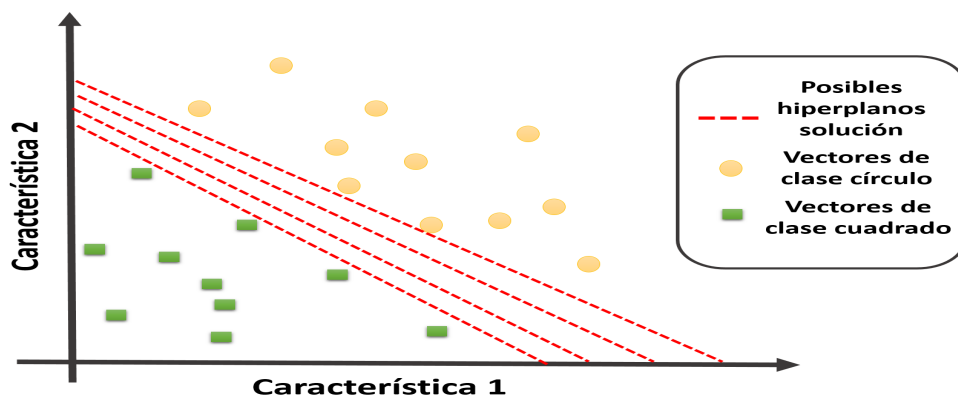
Por otra parte, se considera que el objetivo de SVM es encontrar un hiperplano solución, un hiperplano es representado como:

$$w \cdot x + b = 0 \quad (2.7)$$

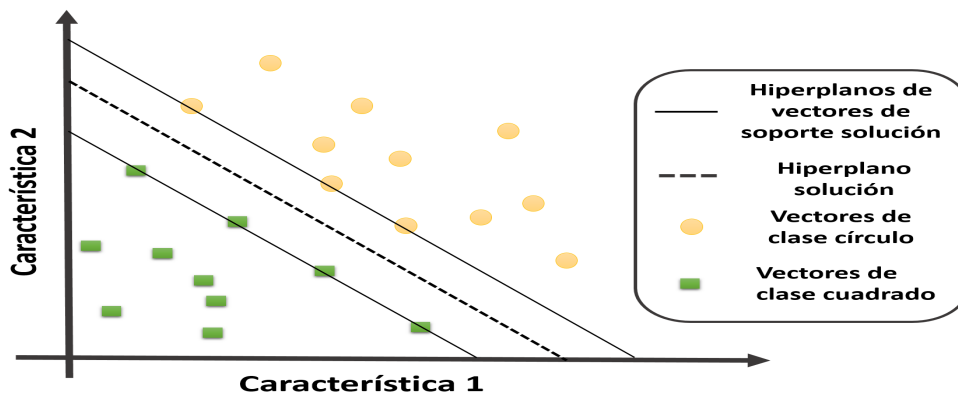
De tal manera que se considera al producto punto ($w \cdot x$) como un vector de pesos y (b) se define como un umbral. El producto punto se describe como:

$$w \cdot x = \sum_{i=1}^n w_i x_i \quad (2.8)$$

SVM puede resolver dos tipos de problemas, los linealmente separables y los no linealmente separables, los primeros tienen un número infinito de hiperplanos que pueden separar las clases, sin embargo solo existe un hiperplano con margen máximo entre el hiperplano solución y los hiperplanos de los vectores de soporte (ver figuras 2.15). Por otra parte, los no linealmente separables no tienen un hiperplano solución, para estos problemas es necesario utilizar una función de kernel como la ecuación (2.10), que permite llevar los datos a un espacio de alta dimensionalidad como se puede observar en la figura 2.16, esto con el objetivo de poder encontrar un hiperplano solución que tenga un margen máximo entre los hiperplanos de los vectores de soporte.



(a) Ejemplo de posibles hiperplanos solución



(b) Representación de hiperplano solución con vectores de soporte

Figura 2.15. Ejemplo de la clasificación linealmente separable utilizando SVM

$$K(x_i, x_j) = \phi(x_i)^t * \phi(x_j) \quad (2.9)$$

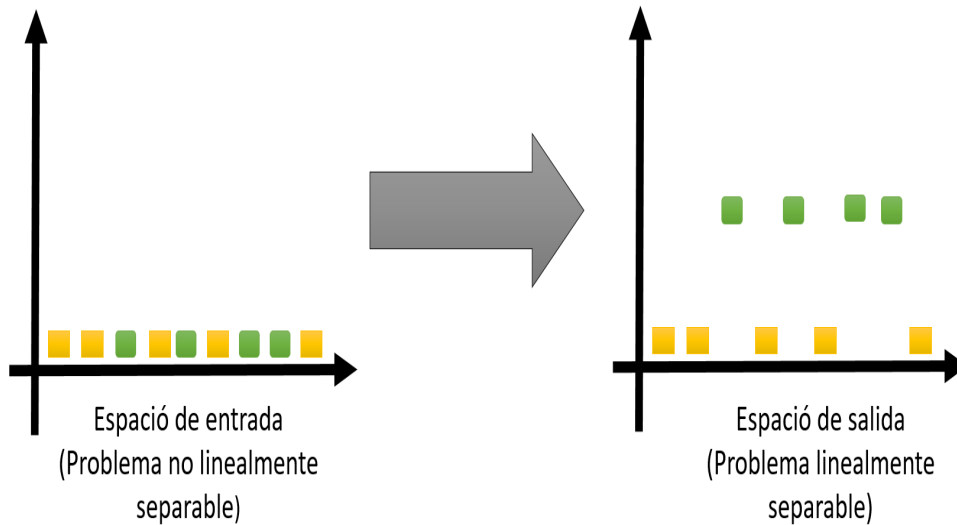


Figura 2.16. Ejemplo de SVM en un problema no linealmente separable

2.3.2. Redes neuronales artificiales

Las redes neuronales artificiales (RNAs) son influenciadas por los sistemas biológicos neuronales, los cuales están formado por cuatro elementos principales: dendritas, cuerpo celular, axón y sinapsis. Estos elementos se conectan para formar una red de neuronas que través de sus dendritas y axones se trasmite información por medio de la sinapsis. De acuerdo con [Mitchell, 1997], las RNAs están compuestas por unidades interconectadas, las cuales tiene un número de entradas y una sola salida. Las entradas de cada unidad pueden ser salidas de otras unidades y la salida de cada unidad puede ser la entrada de otra unidad.

Para conectar la unidad x a la unidad y con el objetivo de propagar la actividad ax entre las unidades. Cada enlace de entrada se le asocia un peso descrito como w_{xy} , las unidades y calculan la sumatoria de los pesos para cada una de sus entradas:

$$\sum_{x=1}^n w_{x,y} a_x \quad (2.10)$$

Posteriormente se aplica una función de activación (g) a la sumatoria para obtener un resultado. La función (g) se puede definir como una función sigmoidea representada como:

$$g(x) = \frac{1}{1 + e^{-z}} \quad (2.11)$$

Para [Mitchell, 1997], Las RNAs pueden ser formadas por diferentes topologías, entre las más populares se encuentran *feed forward* y *recurrent networks*. Para el primer tipo, la red está estructurada por diferentes capas (entrada, ocultas y salida), el segundo tipo puede generar topologías arbitrarias, se alimenta de sus resultados de cada entrada y los resultados de la red están ligados al estado inicial, las cuales dependen de las entradas anteriores.

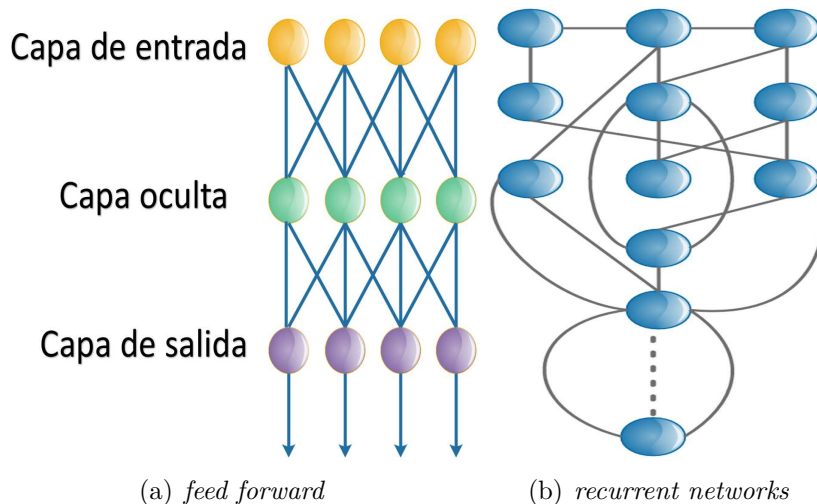


Figura 2.17. Ejemplo de redes neuronales

2.3.3. K-vecinos más cercanos

[Mora-Florez *et al.*, 2008], define a k-vecinos más cercanos o (Knn por sus siglas en inglés *k-Nearest Neighbors*), como un método fácil y práctico para predecir o clasificar un dato nuevo basado en observaciones previas y antiguas. Esta técnica se enfoca en recordar todos los datos que ve en la etapa de entrenamiento, cuando un nuevo dato se presenta, se clasifica según la distancia del vecino o vecinos más cercanos hacia él [Aha *et al.*, 1991].

Como ejemplo, se describe el siguiente caso, se tiene un conjunto de entrenamiento de triángulos y cuadrados como se puede observar en la figura 2.18, se requiere conocer

un elemento nuevo dibujado como una x . Entonces el método se enfoca en buscar un elemento que se encuentre más cerca de la x para asignarle la etiqueta correspondiente, que en este caso el elemento x se puede considerar como un triángulo (ver figura 2.19).

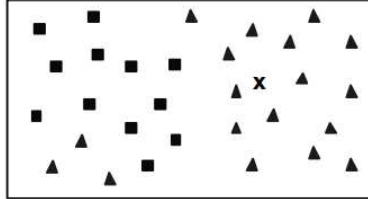


Figura 2.18. Ubicación de un dato nuevo entre datos conocidos [Aha *et al.*, 1991]

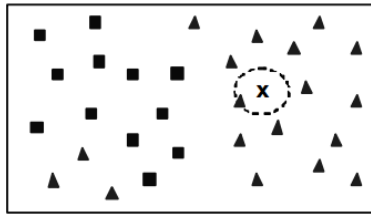


Figura 2.19. Predicción de la clase de un dato nuevo con respecto al vecino más cercano [Aha *et al.*, 1991]

Las muestras con ruido suelen ser el caso donde la selección del vecino más cercano no siempre es el más representativo para el elemento nuevo. En la figura 2.20, se aprecia el ejemplo de una muestra con ruido.

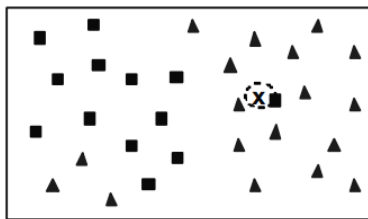


Figura 2.20. Predicción de la clase de un dato nuevo con respecto al vecino más cercano, los datos conocidos contienen ruido [Aha *et al.*, 1991]

Se puede observar que debido al ruido, el elemento se clasifica como cuadrado. Para considerar el problema del ruido basta con tomar un número mayor de uno e impar de vecinos y así generar etiquetas correspondientes para los elementos con mayor cercanía. Esto es considerado como k -vecinos más cercanos, donde K es el número de vecinos más cercanos. En la figura 2.21 se observa un ejemplo donde los vecinos más cercanos son cinco. Cuatro de los cinco vecinos son triángulos y solo uno es un cuadrado, lo cual define la clasificación de x como un triángulo.

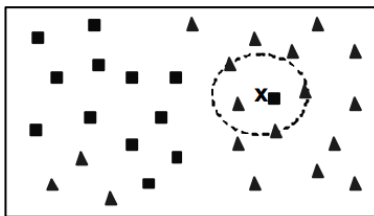


Figura 2.21. Predicción de la clase de un dato nuevo con respecto a los cinco vecinos más cercanos, los datos conocidos tienen ruido [Aha *et al.*, 1991]

2.4. Métricas de evaluación

Las métricas de evaluación permiten medir el desempeño de un algoritmo de clasificación. Para la comparación de los algoritmos de clasificación que se utilizan en las experimentaciones de este proyecto de tesis, se implementan algunas métricas como: accuracy, precision, f-measure y recall.

2.4.1. Matriz de confusión

Antes de poder calcular las métricas descritas anteriormente es necesario calcular una matriz de confusión. Esta matriz contiene la información de la clasificación de un conjunto de datos, las columnas son el número de predicciones de una clase y las filas representan el número de instancias de una clase. En la figura 2.22, se puede observar la composición de una matriz de confusión.

- **TP** (True Positive, traducido como verdaderos positivos) número de predicción correctas de un ejemplo positivo
- **FP** (False Positive, traducido como falsos positivos) número de predicciones incorrectas de un ejemplo positivo
- **TN** (True Negative, traducido como verdadero negativo) número de predicciones correctas de un ejemplo negativo

- **FN** (False Negative, traducido como falso negativo) número de predicciones incorrectas de un ejemplo negativo

		Predicción	
		Manzanas	Platanos
Clase actual	Manzana	8 (TP)	2 (FN)
	Platanos	1 (FP)	9(TN)

Figura 2.22. Matriz de confusión de la clasificación entre manzanas y plátanos

Después de calcular la matriz de confusión se puede obtener las siguientes métricas de evaluación:

Accuracy(exactitud): El porcentaje del total de instancias clasificadas correctamente entre el total de las instancias

$$accuracy = \frac{TN + TP}{TN + FP + FN + TP} \quad (2.12)$$

Precision (Precisión): Instancias clasificadas como positivas de acuerdo con el criterio del algoritmo de clasificación

$$precision = \frac{TP}{TP + FP} \quad (2.13)$$

Recall (Recuerdo): Fracción de instancias relevantes clasificadas correctamente

$$Recall = \frac{TP}{FN + TP} \quad (2.14)$$

f-measure (Medida F): métrica basada en la precisión armónica, esta combina el porcentaje de precision y recall

$$f - measure = 2 \cdot \frac{P \cdot Recuerdo}{P + Recuerdo} \quad (2.15)$$

2.5. OpenCV

OpenCV es una biblioteca de funciones de programación libre enfocado a la visión artificial, creada en sus inicios por Intel, la primera versión alfa fue presentada en enero de 1999, su aplicación en diversos proyectos fue en sistemas de seguridad con detección de movimiento, sistemas de reconocimiento de objetos, entre otros. OpenCV cuenta con una licencia BSD, que permite su uso libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas [Wikipedia, 2014a]. OpenCV está escrito en C y C++ y tiene la capacidad de ejecutarse en Linux, Windows y Mac OS X. El objetivo de OpenCV es proporcionar una infraestructura de VC que permita el desarrollo de aplicaciones sofisticadas rápidamente [Olmedo *et al.*, 2011].

Según [Bradski y Kaehler, 2008], la librería de OpenCV contiene más de 500 funciones enfocadas en la VC y está formado por tres módulos, el primero ésta dirigido al procesamiento de imágenes, el segundo se centra en el aprendizaje automático y el último es usado en manejo de imágenes, videos y funciones para crear interfaces gráficas de usuario. Para [Olmedo *et al.*, 2011], existen funciones en OpenCV para el procesamiento de imágenes que utilizan filtros y máscaras que pueden ser aplicados en una imagen para mejorar la calidad de dicha imagen, encontrar información que no es visible o que no está disponible en primera instancia. Estas funciones son utilizadas para sistemas de seguridad, calibración de cámaras, imágenes médicas, entre otras.

2.6. Expresiones faciales y emociones

De acuerdo a [RAE, 2014], una emoción es “la alteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somática ”. Según [las-emociones.com, 2014], las emociones tienen un papel importante en las comunicaciones humanas y tienen una gran importancia sobre la percepción y la toma de decisiones. Paul Ekman, uno de los psicólogos con más influencia en la investigación de las emociones y sus expresiones, realizó un estudio transcultural en individuos destacados de la tribu de Papúa Nueva Guinea [Ekman y Oster, 1981], basado en la teoría universal donde las expresiones faciales de los individuos están relacionada con las emociones. A partir de ese estudio estableció una lista de seis emociones básicas: alegría, enojo, miedo, tristeza, sorpresa y repugnancia.

Las expresiones faciales crean un lazo clave para el entendimiento y transmisión de emociones. Algunos estudios han demostrado que la interpretación de emociones puede alterar el sentido de una conversación [Ratliff y Patterson, 2008]. También, Las emociones tienen un papel importante en la comunicación humana y tienen un efecto dominante en la toma de decisiones. Esto ha generado un interés por reconocimiento de expresiones faciales, donde algunas de sus aplicaciones fueron dirigidas a la atención médica y en la interacción humano computadora [Xu *et al.*, 2013].

2.7. Experiencia del usuario y usabilidad

La experiencia del usuario o (UX por sus siglas en inglés *User Experience*), es descrita por [uxpa.org, 2014], como cada aspecto en la interacción de un usuario con un producto, empresa o servicio que conforman las percepciones de los usuarios de todo un conjunto. La UX involucra las emociones de una persona sobre el uso de un servicio, producto o sistema. También, incluye la utilidad, la facilidad de uso y la eficiencia de un sistema. Principalmente UX tiene como objetivo el crear una experiencia a través de un dispositivo, un producto, un sistema o aplicación. La UX se centra en tener un profundo conocimiento de las necesidades, lo que valoran, habilidades y limitaciones de los usuarios.

Una parte de la UX se centra en estudiar el diseño de información, la cual es considerada como la capacidad de organizar y presentar datos de forma que los humanos puedan usarlos eficiente. Es decir, el diseño de información tiene como objetivo organizar y presentar datos, transformándolos en información valiosa y significativa. Para [Martín, 2005] el diseño de información está formado por diferentes áreas como la comunicación visual, la representación de la información, el diseño de interfaces entre otras. Para poder evaluar el diseño de información de un sistema o software es necesario realizar algunas pruebas de usabilidad.

Antes de describir las pruebas de usabilidad es necesario definir que es usabilidad y donde se utiliza. Para [journal, 2014], la usabilidad o la calidad de uso, es un concepto que tiene un relación directa con la interacción humano computadora. El término de usabilidad tiene un origen inglés y su definición se refiere al grado de eficacia y satisfacción con la que usuarios específicos puede lograr objetivos específicos, en un ambiente específico. La usabilidad no se puede ver como un atributo de la calidad de un producto, sino como un resultado, una metodología de diseño y evaluación. En [Vos, 2005], menciona que la usabilidad tiene como principal trabajo centrarse en la experiencia y el rendimiento del usuario. Desde el punto del punto de vista del diseño de una interfaz, el propósito es simple: recopilar la información del usuario respecto al sistema y ponerla a disposición de aplicaciones u otros usuarios.

Por otra parte, existen métodos para medir la usabilidad conocidos como pruebas de usabilidad, estas se dedicada a descubrir a que grado un sistema está cumpliendo con el propósito para el que fue creado. De acuerdo con [Hassan Montero *et al.*, 2003], una prueba de usabilidad se encarga de analizar y observar un conjunto de usuarios mientras utilizan un sistema que recopila errores que serán solucionados posteriormente. Algunos ejemplos de este tipo de pruebas son: pruebas beta, prueba A/B, pruebas moderadas, por mencionar algunas.

Para este proyecto de tesis las pruebas elegidas como objetivo son las pruebas moderadas. Estas evalúan elementos del diseño que permiten entregar datos para la mejora

de la arquitectura, contenido y diseño de un sistema. En la implementación de este tipo de pruebas de acuerdo a [SpA, 2015], se crea una lista de actividades que el usuario debe realizar mientras se analizan sus acciones. También, existe un moderador que guía a los usuarios durante la prueba, en algunos casos la interacción usuarios y moderador es personal.

Durante la prueba el moderador y los observadores deben estar atentos al comportamiento del usuario, tanto en el lenguaje corporal o comportamiento anormal, las pruebas suelen ser ejecutadas en laboratorios especializados donde existe una cámara de Gessel, esta permite mantener ocultos a los observadores, en otras palabras, los observadores en la mayoría de los casos no tienen relación personal con los usuarios. En cada prueba realizada para un sistema o software existe un registro audiovisual frontal, de contexto y un video de la pantalla, esto con la intención de permitir su análisis posterior para la creación de un reporte final.

2.8. Estado del Arte

En la investigación [Xu *et al.*, 2013], desarrollaron un sistema de reconocimiento de expresiones faciales multimodal, donde se enfocaron en la detección de expresiones faciales y del habla. Implementaron un algoritmo de triangulación y otro de iteración para adquirir una imagen reconstruida que es similar a la imagen original y al mismo tiempo se obtiene la localización de puntos característicos faciales. Para el reconocimiento de expresiones implementaron un modelo de apariencia activo (AAM), donde los parámetros de animación faciales (FAP) son calculados para identificar los cambios de expresiones. Por otra parte, la extracción de características del habla emocional fueron obtenidas mediante la extracción de la frecuencia fundamental, que es el formato de frecuencia y un promedio de energía de corto plazo por cada (*frame*).

Teniendo las características de las emociones faciales y del habla, éstas posteriormente son unidas y clasificadas por el método de *redes neuronales artificiales*. Los resultados obtenidos para el sistema utilizando únicamente las características faciales, muestran la efectividad para reconocer la felicidad en 63%, tristeza en 89%, enojo en 58.9% y repugnancia en 75%. Los resultados para el sistema utilizando las características faciales y del habla muestran la efectividad para reconocer la felicidad en 95.2%, tristeza en 92.9%, enojo en 93.3% y repugnancia en 80.6%.

Un año después [Martinez y Du, 2012], propusieron un sistema de reconocimiento de expresiones faciales básicas. El rostro y las formas de los principales componentes faciales son identificados automáticamente, incluyendo delimitación como las cejas, boca, los ojos, la nariz y la mandíbula. La forma de cada elemento muestra puntos emblemáticos igualmente espaciados. Se calcula el centro de masa de estos puntos y forma un vector de características de dos dimensiones. Como método de clasificación utilizaron

SVM lineal y las pruebas se realizaron con *leave-one-out* (dejar uno afuera), los resultados presentan que la expresión feliz se clasificó correctamente en 99 % de las veces, la sorpresa y el disgusto 95 % de las veces, la tristeza y la ira 90 % al 94 %, mientras que el miedo se clasifica con éxito en el 92 %.

Cuatro años antes, en la investigación de [Shan *et al.*, 2009], realizaron una evaluación de la representación facial basada en *Patrones locales binarios* (LBP) y los métodos de aprendizaje automático utilizados fueron *template matching* (comparación de plantillas), SVM, análisis lineal discriminante o LDA por sus siglas en inglés *Linear Discriminant Analysis* y técnica de programación lineal. Además, propusieron Boosted-LBP, que es la unión del método LBP y *AdaBoost*, la implementación de Boosted-LBP mejoró el rendimiento de diferentes clasificadores.

Los resultados de la investigación describen que la comparación de plantillas con características de LBP, lograron un 79.1 % de precisión para la clasificación de expresiones faciales (enojo, repugnancia, miedo, feliz, tristeza, sorpresa, neutro) denominado tarea 7 y un 84.5 % de precisión para la clasificación de (enojo, repugnancia, miedo, feliz, tristeza, sorpresa) llamado tarea 6. También [Shan *et al.*, 2009], describen que feliz y la sorpresa presentan una precisión de un 90- 92 %, mientras que la enojo y el miedo suelen confundirse fácilmente con las demás. Con el método de SVM usando una función principal *Radial Basis Function* (RBF) y con las características de LBP, los resultados para repugnancia, feliz, sorpresa y neutro tiene una precisión de 90-98 %, por otra parte, para el miedo y la tristeza los porcentajes de precisión son de 68 % al 69 %.

También, realizaron una comparación de los métodos LBP y filtros de Gabor, para el reconocimiento de expresiones faciales, usando como clasificador SVM. [Shan *et al.*, 2009], describen que LBP presenta una ventaja en la extracción de características con respecto al costo de memoria, LBP con 2478 características en memoria tiene un tiempo de ejecución de 0.03 segundos mientras que Gabor con 42.65 características en memoria, presenta un tiempo de ejecución de 30 segundos. Por otra parte, la implementación de SVM usando RBF e implementando LBP presenta un rendimiento de $92.6\% \pm 2.9$ para la tarea 6 y $88.9\% \pm 3.5\%$ para la tarea 7 y Gabor presenten un rendimiento del $89.8\% \pm 3.1\%$ para la tarea 6 y $86.8\% \pm 3.6\%$ para la tarea 7.

Para la experimentación con LDA donde utilizaron PCA, el 98 % de los datos se mantuvo, formando vectores con un rango de 358 a 378 características para la tarea 6 y 405 a 431 características para el problema de la clase 7. Después se usó KNN como clasificador, logrando un 73,4 % de eficiencia para el reconocimiento de la tarea 7 y 79,2 % para el reconocimiento de la tarea 6. En otro orden de ideas, la implementación de SVM lineal tomando como entrada las proyecciones de PCA presenta un rendimiento de $80.2\% \pm 4.9\%$ para la tarea 7 y $87.7\% \pm 4.7\%$ para la tarea 6. Por último, la técnica de programación lineal obtuvo un rendimiento de 82.3 % para el reconocimiento de la tarea 7 y el 89.6 % para el reconocimiento de la tarea 6. Sin embargo, SVM lineal

presentó un rendimiento de $86.0\% \pm 3.3\%$ para la clase 6 y $90.4\% \pm 3.9\%$.

En el siguiente año [Ratliff y Patterson, 2008], desarrollaron un sistema para el reconocimiento de emociones faciales, que utiliza las distancias euclidianas simples para obtener los parámetros faciales y representarlos en conjunto de un modelo de apariencia activa (AMM), como método clasificación utilizaron SVM. Como resultado, el sistema clasifica correctamente las expresiones faciales (miedo, feliz, sorpresa, enojo, repugnancia, tristeza y una expresión neutral) en un rango general de 60 - 100% para cada persona, la mayoría de las personas que realizó la prueba se encontraba entre 80 - 90% de efectividad y sólo algunas personas presentaron un rendimiento pobre dentro de la experimentación. Las expresiones que presentaron mayor error fueron la enojo y el miedo con un 63.9% para ambos.

En otro proyecto, [Zhao y Pietikainen, 2007] realizaron un trabajo para identificar expresiones faciales mediante métodos de visión por computadora y aprendizaje automático. Como métodos de caracterización utilizaron a LBP, VLBP y LBP-TOP y como método de clasificación implementaron SVM. Las expresiones faciales en la que se enfocaron fueron la sorpresa, felicidad, tristeza, miedo, enojo y repugnancia. Los mejores resultados obtenidos en este trabajo fueron los siguientes: El método $LBP-TOP_{4,4,4,3,3,3}^{u2}$ obtuvo un 91.18% de efectividad en la clasificación de las expresiones faciales. Para el método $LBP-TOP_{8,8,8,3,3,3}^{u2}$ presento un 94.38% de efectividad y el mejor resultado fue para la combinación de métodos $LBP-TOP_{8,8,8,3,3,3}^{u2} + VLBP_{3,2,3}$, obteniendo un 95.19% de efectividad

Durante el año 2005 existen cuatro trabajos enfocados al reconocimiento de expresiones faciales implementando técnicas de visión por computadora y aprendizaje automático. [Bartlett *et al.*, 2005] desarrollo un sistema de reconocimiento de expresiones faciales basado en técnicas de aprendizaje automático y visión por computadora. Para la implementación del proyecto utilizaron los métodos de clasificación SVM y LDA, como métodos de caracterización utilizaron a Adaboost y Gabor. Los mejores resultados se obtuvieron mediante la selección de un subconjunto de filtros de Gabor utilizando AdaBoost seguido de la clasificación con SVM. El sistema opera en tiempo real y tiene un 93% de precisión en la clasificación de expresiones faciales como: felicidad, tristeza, sorpresa, repugnancia, miedo, enojo y neutral. El conjunto de imágenes fue recopilados por [Bartlett *et al.*, 2005], de videos recodificados en analógico S-video.

[Shan *et al.*, 2005], realizaron un estudio para reconocimiento de expresiones faciales implementado el método LBP para la extracción de característica y como clasificador utilizaron a SVM. Las emociones faciales reconocidas son enojo, repugnancia, miedo, felicidad, tristeza y sorpresa. Dentro de la experimentación LBP+SVM, fue evaluado con una generalización de rendimiento lineal, polinomial y RBF para el método SVM. Los resultados obtenidos muestran que LBP+SVM con un rendimiento lineal tiene una exactitud de 87.2%, con un rendimiento polinomial la exactitud es de 88.4% y por último, SVM con RBF tiene como resultado de 87.6% de exactitud.

Para concluir los proyectos del año 2005, [Guo y Dyer, 2005], desarrollaron una investigación enfocada al reconocimiento de expresiones faciales. Como método de extracción de características implementaron los filtros de Gabor y como métodos de clasificación utilizaron SVM y AdaBoost. Las emociones detectadas por el sistema son: felicidad, tristeza, sorpresa, enojo, repugnancia y el miedo, el conjunto de prueba está conformado por 213 imágenes de mujeres japonesas. El resultado obtenido para AdaBoost es de un 71.9% de exactitud con 80 características, para SVM con una función lineal el resultado es de 92.4% de exactitud con 912 características y para una función no lineal, la exactitud es de 91.9% con 612 características.

Dentro del año 2004 la investigación por [Busso *et al.*, 2004], se enfocó en el análisis del reconocimiento de expresiones faciales y tono de voz. Para la extracción de características de expresiones faciales usaron marcadores geométricos, por lo tanto no es necesario utilizar algoritmos de detección y seguimiento. Es importante mencionar que dividieron los elementos de la cara (frente, cejas, ojos, mejillas) y cada elemento tiene un vector de características. Antes de clasificar los vectores se implementó un análisis de componentes principales (ACP) y realizaron un pre-procesamiento utilizando el clasificador K-NN con un número de vecinos igual a tres, reduciendo el nivel de diez a cuatro características y posteriormente implementaron el clasificador SVM. Como resultados de la experimentación el enojo obtuvo un 85% de exactitud, la repugnancia un 79%, la tristeza 81%, la felicidad un 100% y la expresión neutra un 81%. Es importante mencionar que las pruebas fueron realizadas únicamente para una persona.

Por último, [Zhang *et al.*, 1998], desarrollaron un estudio dirigido al reconocimiento de expresiones faciales comparando los métodos de extracción de características geométricas y el método de Gabor, como clasificador utilizaron a SVM. Las expresiones fáciles clasificadas fueron: felicidad, tristeza, sorpresa, enojo, repugnancia, miedo y neutra. La experimentación se realizó con 213 imágenes de mujeres y los resultados que obtuvieron son los siguientes: utilizando las características geométricas la efectividad fue de un 73.3% y utilizando el método de Gabor obtuvieron un 92.2% de efectividad.

Trabajo relacionado	N° imágenes	N° emociones	Método de extracción de características	Método de clasificación	Resultados
[Xu <i>et al.</i> , 2013]	?	4	AAM	Redes neuronales artificiales	71.475 %
[Zhang <i>et al.</i> , 1998]	320	6	LBP	template matching	79.1 %
[Zhang <i>et al.</i> , 1998]	320	7	LBP	template matching	84.5 %
[Zhang <i>et al.</i> , 1998]	320	6	Geometric feature	TAN	73.2 %
[Zhang <i>et al.</i> , 1998]	320	6	LBP	SVM(RBF)	92.6 %
[Zhang <i>et al.</i> , 1998]	320	7	LBP	SVM(RBF)	88.9 %
[Zhang <i>et al.</i> , 1998]	320	6	Gabor	SVM(RBF)	89.8 %
[Zhang <i>et al.</i> , 1998]	320	7	Gabor	SVM(RBF)	86.8 %
[Ratliff y Patterson, 2008]	500	7	AMM	SVM	80 - 90 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP^{u2}$	SVM	91.18 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP_{4,4,4,3,3,3}^{u2}$	SVM	91.18 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP_{8,8,8,3,3,3}^{u2}$	SVM	94.38 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP_{8,8,8,1,1,1}^{u2}$	SVM	90.37 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP_{16,16,16,3,3,3}^{u2}$	SVM	88.77 %
[Zhao y Pietikainen, 2007]	?	6	$VLBP_{3,2,3}$	SVM	91.18 %
[Zhao y Pietikainen, 2007]	?	6	$LBP - TOP_{8,8,8,3,3,3}^{u2} + VLBP_{3,2,3}$	SVM	95.19 %
[Martinez y Du, 2012]	?	6	Landmark points	SVM	94.16 %
[Bartlett <i>et al.</i> , 2005]	?	6	Adaboosts	SVM(lineal)	93.3 %

Tabla 2.1. Trabajos relacionados

Capítulo 3

Metodología

La metodología ejecutada para el desarrollo del sistema de reconocimiento expresiones faciales esta compuestas por cuatro etapas: 1) Obtención de imágenes (Conjunto de entrenamiento) con expresiones faciales, 2) procesamiento de imágenes, 3) Extracción de características implementando los métodos de LBP, LBP Uniforme con aplicación para rostros, PCA y filtros de Gabor, 4) Clasificación de características utilizando métodos de aprendizaje automático como: SVM, RNA y vecinos más cercanos. La figura 3.1 muestra un diagrama de la metodología general para el desarrollo del proyecto. En las siguientes subsecciones se describirán cada etapa de forma detalla.

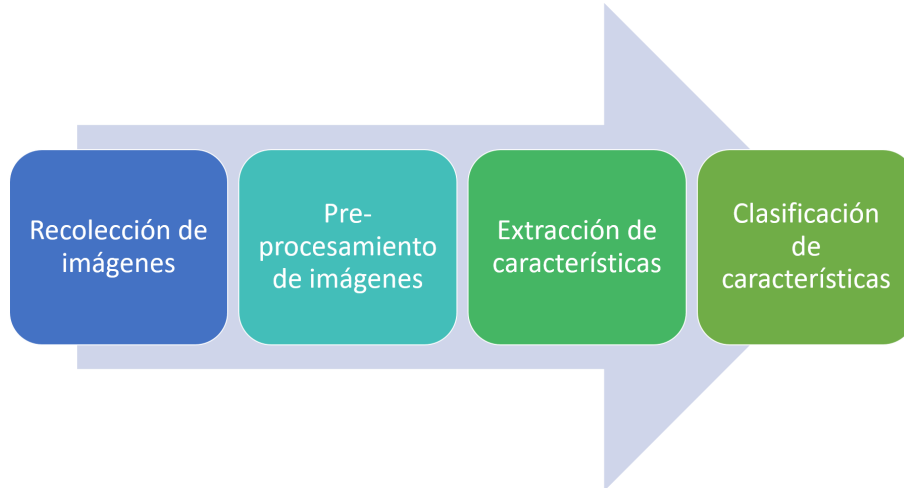


Figura 3.1. Metodología general

3.1. Recolección del conjunto de imágenes

Para crear el conjunto de datos de imágenes, se les solicito a alumnos de la Universidad Politécnica de Puebla su apoyo para participar en una sesión de fotos, donde cada

alumno tenía que imitar diferentes expresiones faciales. Así, un total de 109 alumnos participaron en la creación de esta base de imágenes. El procedimiento de la sesión de fotos fue el siguiente: A cada alumno se le proyectó una presentación con las seis expresiones faciales básicas (feliz, enojado, repugnancia, triste, miedo y sorpresa), por cada expresión proyectada, se solicitaba que imitara la expresión mientras se tomaban las fotos.

A pesar de que fueron 109 alumnos que participaron en la sesión de fotos y que las expresiones faciales proyectadas fueron 6, No todos pudieron imitar las expresiones faciales con éxito, por esta razón el total de imágenes con expresiones faciales recolectadas fueron: 109 neutras, 91 felices, 42 con expresión de enojo y 52 de repugnancia, quedando fuera las expresiones de miedo, sorpresa y tristeza.

3.2. Pre-Procesamiento de imágenes

El desarrollo de esta etapa consistió en crear una herramienta que permitiera buscar rostros en el conjunto de imágenes obtenido previamente. Para lograr el objetivo se implementó uno de los módulos de reconocimiento de rostros en OpenCV, el módulo seleccionado utiliza como caracterizador a LBP y el clasificador de cascada, la programación de la herramienta está hecha en el lenguaje C++. En la figura 3.2 se puede observar el funcionamiento general de la herramienta, que consiste en identificar un rostro dentro de una imagen, después se obtiene una sub-imagen con el rostro, creando un nuevo conjunto de entrenamiento. El objetivo de seleccionar únicamente los rostros y crear un nuevo subconjunto, permite eliminar información innecesaria para la clasificación de las imágenes.

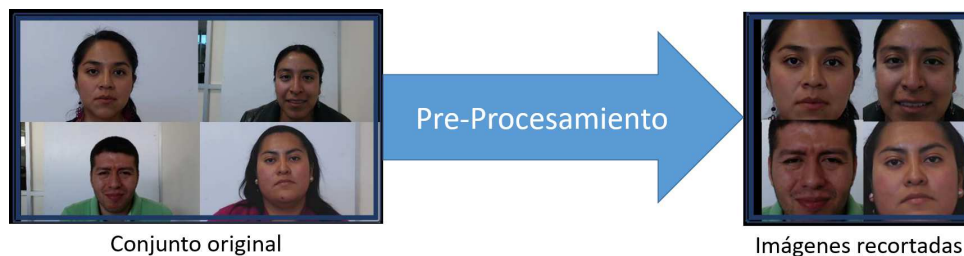


Figura 3.2. Detección y recorte de rostros del conjunto de imágenes creado

3.3. Extracción de características

Para el desarrollo de esta fase se implementaron los siguientes métodos: LBP, LBP uniforme con aplicación para rostros, PCA y filtros de Gabor. Estos métodos ya fueron

descritos en la sección anterior, Sin embargo es importante mencionar que el método de LBP Uniforme con aplicación para rostros, surge de la unión del método LBP Uniforme y la modificación de LBP empleada para el análisis de rostros descrita en la sección 2.2.1. Desde este punto se hará referencia a este método como LBP_{Ur}. La implementación tanto para LBP y LBP_{Ur}, consiste en crear un histograma con respecto a los códigos LBP obtenidos, Los histogramas son guardados en una matriz de entrenamiento con el tamaño de $(N \times M)$, donde N es igual al número de imágenes y M al tamaño del histograma, para el caso de LBP la $M = 256$ y para LBP_{Ur} $M = 531$; (ver figura 3.3).

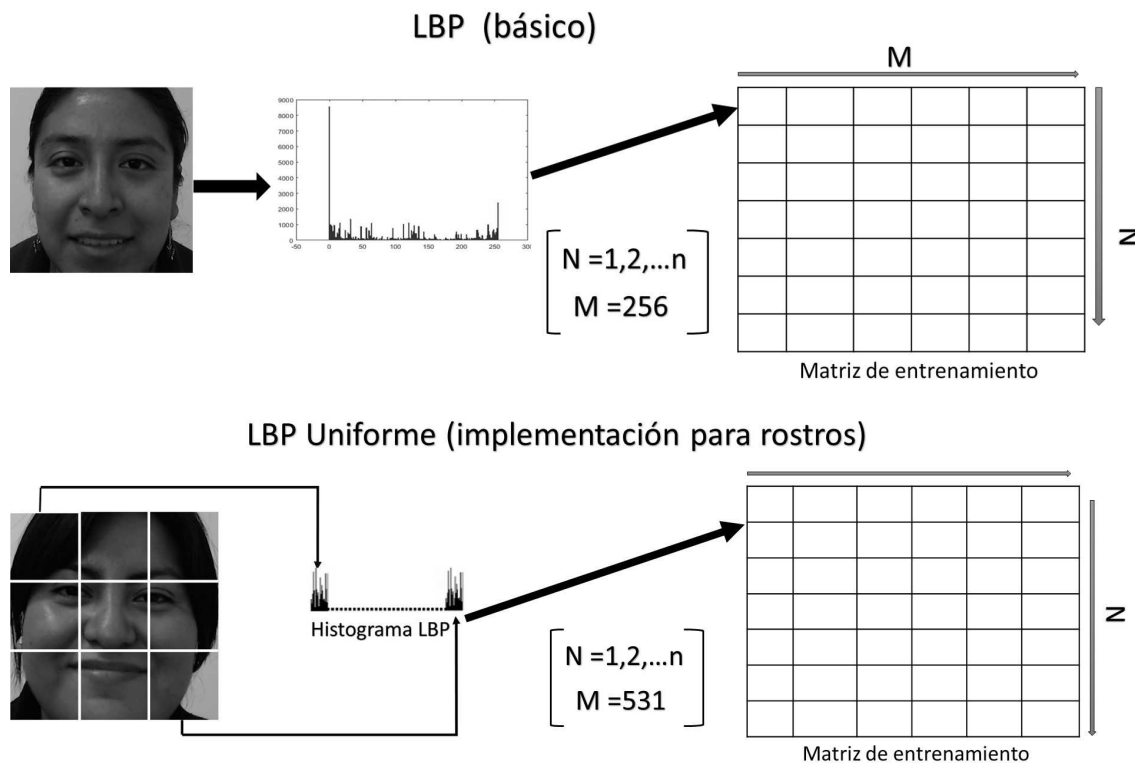


Figura 3.3. Etapas de los Métodos LBP y LBP_{Ur}

Por lo contrario, la implementación del método PCA consiste en representar cada imagen del conjunto como un vector unidimensional el cual está formado por los componentes principales (Ver imagen 3.4). Es importante mencionar que a mayor número de componentes principales menor la pérdida de información por cada imagen.

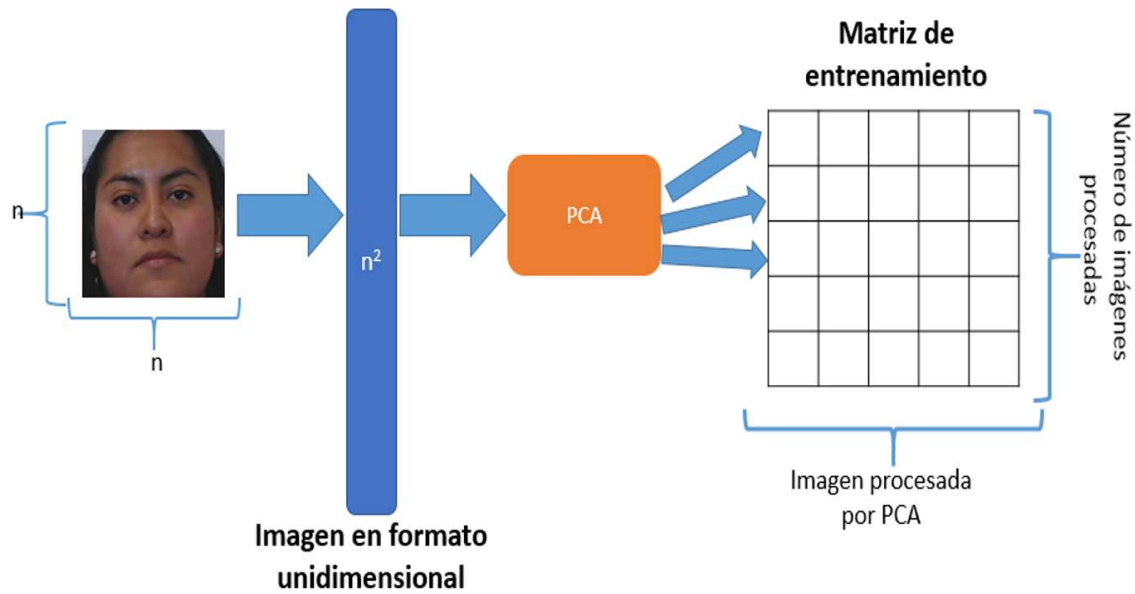


Figura 3.4. Procesamiento de imágenes con PCA

Finalmente, el método restante es Gabor, este consiste en tres periodos: 1) Crear un filtro o un banco de filtros, 2) Aplicar los filtros a cada imagen y 3) obtener y crear una matriz de aprendizaje para guardar la información por cada filtro aplicado. El tamaño de columnas de la matriz de aprendizaje dependerá del número de filtros aplicado y del tamaño de la imagen (ver figura 3.5).

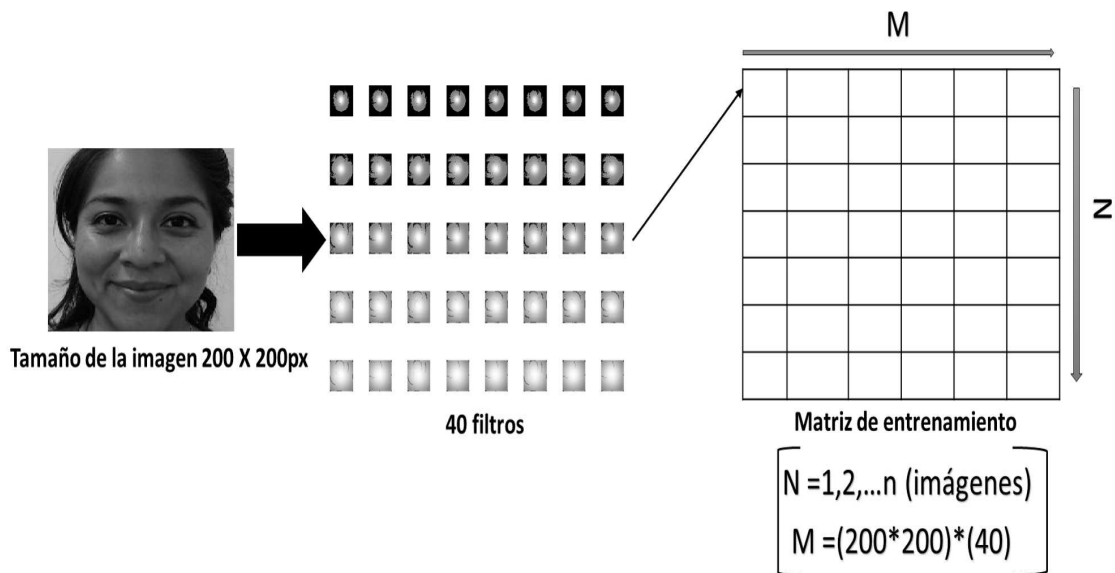


Figura 3.5. Etapas del método de Gabor

3.4. Clasificación de expresiones usando aprendizaje automático

Para el desarrollo de esta etapa fueron aplicados los algoritmos de clasificación descritos en la sección 2.3. Los algoritmos de clasificación tienen como datos de entrada una matriz de aprendizaje, en el caso de los métodos LBP y LBPUr esta matriz está compuesta por histogramas. (ver figura 3.3). Por otro lado, en PCA se utilizan los componentes principales para llenar la matriz de aprendizaje (ver figura 3.4). Ahora bien, para los filtros de Gabor la matriz de aprendizaje está conformada por los datos de cada filtro aplicado al conjunto de imágenes.

Capítulo 4

Resultados Experimentales

En este capítulo se describen los experimentos realizados, los cuales fueron divididos en dos etapas, la primera etapa se enfocó en la comparación de los algoritmos de clasificación junto con los métodos de caracterización descritos en el capítulo 2, la segunda etapa fue dirigida a la implementación de los métodos para el desarrollo del sistema de reconocimiento de expresiones faciales. En las siguientes subsecciones se mostraran los resultados obtenidos en cada experimentación realizada en las dos diferentes etapas.

4.1. Resultados experimentales de la primera etapa

Para esta etapa la experimentación utilizo cuatro conjuntos binarios y un conjunto multiclase, los conjuntos binarios están formados por imágenes de dos tipos y fueron nombrados como C1,C2,C3 y C4 los cuales son descritos en la tabla 4.1. El conjunto multiclase está formado por imágenes de tres diferentes tipos de expresiones faciales, este fue nombrado como M1 y sus características se describen en la tabla 4.2. La implementación de los métodos LBP básico y Filtros de Gabor se realizó en MatLab y el método de LBP Uniforme con implementación para rostros se realizó en OpenCV.

Por otra parte, Los algoritmos de clasificación como: K-vecinos más cercanos, RNAs y SVM implementados en Weka, fueron utilizados en esta etapa y con el objetivo de obtener mejores resultados para cada experimentación, se modificaron diversos parámetros para cada algoritmo. Asimismo, los resultados reportados se obtuvieron mediante el promedio de ejecutar 5 veces cada algoritmos del clasificación aplicando *10-cross-validation* y las métricas *Accuracy*, *Presition*, *F-measure* y *Recall* para evaluar el desempeño de cada ejecución.

Imágenes	C1	C2	C3	C4	Etiqueta
Normal	109	109	109	109	0
Feliz	36	91	-	-	1
Repugnancia	36	-	-	52	
Enojo	37	-	42	-	
Total de Imágenes	218	200	151	161	-

Tabla 4.1. Descripción general de los conjuntos de entrenamiento binarios.

Imágenes	M1	Etiqueta
Normal	109	0
Feliz	91	1
Repugnancia	52	2
Total de Imágenes	252	-

Tabla 4.2. Descripción general del conjunto de entrenamiento multiclase.

4.1.1. Resultados de la clasificación utilizando LBP básico

Tal como fue descrito en la sección 2.2.1, los histogramas creados en esta experimentación tienen un tamaño de 256 valores y el rango de intensidades es de 40,000 valores, el cual está ligado al tamaño de las imágenes que en este caso fueron de 200 pixeles de alto por 200 pixeles de ancho. Por otra parte, los parámetros modificados para cada algoritmo de clasificación en relación a cada conjunto de entrenamiento se pueden observar en las tablas 4.7. De la misma forma en la tabla 4.4 se muestran los resultados obtenidos para cada conjunto con respecto a los algoritmos de clasificación y las métricas utilizadas.

	K-vecinos más cercanos	Redes neuronales artificiales		Máquinas de Vectores de soporte
	KNN	<i>Momentum</i>	<i>Training Time</i>	<i>Exponent</i>
C1	9	-	100	2
C2	9	0.5	100	2
C3	9	0.5	100	2
C4	9	0.8	100	2

Tabla 4.3. Configuración de los algoritmos de clasificación para la experimentación con el método LBP básico

Algoritmos de clasificación	<i>Accuracy</i>				<i>Prestition</i>				<i>F-measure</i>				<i>Recall</i>			
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>
Conjuntos de entrenamiento																
<i>K</i> -vecinos más cercanos	54.31%	57.60%	78.98%	64.10%	54.32%	57.20%	70.42%	57.86%	54.28%	57.14%	64.02%	58.80%	54.32%	57.60%	78.98%	64.10%
Redes neuronales artificiales	69.27%	73.10%	76.42%	72.67%	69.28%	73.14%	75.18%	71.78%	69.24%	73.12%	75.48%	71.94%	69.26%	73.10%	76.42%	72.66%
Máquinas de vectores de soporte	69.91%	74.90%	75.63%	70.19%	69.91%	74.90%	74.40%	69.16%	69.90%	74.84%	74.76%	69.48%	69.92%	74.84%	75.64%	70.18%

Tabla 4.4. Promedios de métricas utilizando LBP básico aplicado a los conjuntos C1, C2, C3 y C4

4.1.2. Resultados de la clasificación utilizando PCA

Tal como fue descrito en la sección 2.2.3, PCA permite reducir el espacio dimensional en un conjunto de datos extrayendo los componentes principales, para esta experimentación el espacio dimensional fue reducido de 40,000 valores a 43 componentes principales por cada imagen, Los parámetros modificados de cada algoritmo de clasificación se muestran en la tabla 4.5, los resultados experimentales en relación a los conjuntos de entrenamiento y los algoritmos de clasificación se presentan en la tabla 4.6.

4.1.3. Resultados de la clasificación utilizando Filtros de Gabor

Para el desarrollo de la experimentación se utilizaron 40 filtros de Gabor por cada imagen, tal como se mencionó en la experimentación anterior, las imágenes utilizadas para esta etapa son del tamaño de 200 por 200 pixeles, de tal manera que el espacio dimensional para cada imagen utilizando los filtros de Gabor es de 1,600,000 atributos; resultado obtenido de la multiplicación del tamaño de la imagen y los 40 filtros. Para reducir el espacio dimensional se utilizó una técnica llamada *Downsampling* (disminución de resolución) la cual permitió reducir los atributos de 1,600,000 a 1000 por imagen. En otro orden de ideas, los parámetros modificados para cada algoritmo de clasificación se pueden observar en la tabla 4.7, de la misma forma, los resultados obtenidos de esta experimentación se muestran en la tabla 4.8.

	K-vecinos más cercanos	Redes neuronales artificiales		Máquinas de Vectores de soporte
	KNN	<i>Momentum</i>	<i>Training Time</i>	<i>Exponent</i>
C1	3	0.5	10	-
C2	3	0.5	10	-
C3	3	0.5	10	-
C4	3	0.5	10	-

Tabla 4.5. Configuración de los algoritmos de clasificación para la experimentación con el método PCA

Algoritmos de clasificación	<i>Accuracy</i>				<i>Precision</i>				<i>F-measure</i>				<i>Recall</i>			
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>
Conjuntos de entrenamiento																
<i>K</i> -vecinos más cercanos	58.72%	48.90%	69.40%	62.86%	58.82%	47.84%	65.86%	60.38%	58.56%	47.94%	66.78%	61.22%	58.72%	48.90%	69.42%	62.86%
Redes neuronales artificiales	76.24%	87.20%	74.44%	80.37%	76.30%	87.22%	73.24%	80.22%	76.22%	87.20%	73.56%	80.26%	76.24%	87.20%	74.42%	80.38%
Máquinas de vectores de soporte	75.14%	86.80%	74.17%	80.25%	75.36%	87.06%	71.60%	79.78%	75.06%	86.72%	71.58%	79.74%	75.14%	86.80%	74.18%	80.22%

Tabla 4.6. Promedios de métricas utilizando PCA aplicado a los conjuntos C1, C2, C3 y C4

	K-vecinos más cercanos	Redes neuronales artificiales			Máquinas de Vectores de soporte	
	<i>KNN</i>	<i>Momentum</i>	<i>Training, Time</i>	<i>Hidden-Layer</i>	<i>Exponent</i>	<i>C</i>
C1	9	0.5	100	-	2	-
C2	9	-	200	13	2	2
C3	9	0.5	100	20	2	-
C4	9	0.8	100	-	2	-

Tabla 4.7. Configuración de los algoritmos de clasificación para la experimentación con el método filtros de Gabor

	<i>Accuracy</i>				<i>Precision</i>				<i>F-measure</i>				<i>Recall</i>			
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>
<i>Algoritmos de clasificación</i>																
<i>Conjuntos de entrenamiento</i>																
<i>K-vecinos más cercanos</i>	71.10%	71.50%	76.42%	73.66%	73.54%	77.10%	82.24%	74.04%	70.30%	68.94%	69.28%	69.32%	71.12%	71.50%	76.42%	73.66%
<i>Redes neuronales artificiales</i>	85.14%	89.90%	87.28%	94.66%	85.18%	89.92%	87.00%	94.70%	85.14%	89.90%	87.02%	94.62%	85.14%	89.90%	87.28%	94.66%
<i>Máquinas de vectores de soporte</i>	84.04%	89.90%	86.75%	94.53%	84.04%	89.90%	86.44%	94.52%	84.02%	89.90%	86.38%	94.50%	86.20%	89.90%	86.78%	94.54%

Tabla 4.8. Promedios de métricas utilizando filtros de Gabor aplicado a los conjuntos C1, C2, C3 y C4

4.1.4. Resultados de la clasificación utilizando LBPUr

Para el desarrollo de esta experimentación el método LBPUr se programó en C++ utilizando Qt como IDE de programación y se utilizaron librerías de OpenCV que apoyaron el desarrollo del método en la parte de visión por computadora, la clasificación se realizó con los algoritmos implementados en Weka ¹. Como se explicó en la sección 2.2.1, el método contiene un histograma general que tiene un tamaño de 531 valores por cada imagen. Para esta experimentación las imágenes tienen un tamaño de 200 pixeles de alto por 200 pixeles de ancho. Los conjuntos utilizados para el desarrollo de la experimentación son los descritos en las tablas 4.1 y 4.2; en la tabla 4.9 se muestran las configuraciones para cada algoritmo de clasificación en relación a los conjuntos de entrenamiento. Por último, los resultados experimentales en relación a los conjuntos de entrenamiento y los algoritmos de clasificación se muestran en la tabla 4.10

¹herramienta de licencia libre desarrollada por la Universidad de Waikato, para mayor información ir: <http://www.cs.waikato.ac.nz/ml/weka/>

	K-vecinos más cercanos	Redes neuronales artificiales			Máquinas de Vectores de soporte			
	<i>KNN</i>	<i>Learning rate</i>	<i>Momentum</i>	<i>Training Time</i>	<i>Exponent</i>	<i>C</i>	<i>Kernel</i>	<i>Gamma</i>
C1	5	-	-	100	2	-	-	-
C2	9	-	0.5	10	2	-	-	-
C3	9	-	0.5	10	-	-	-	-
C4	9	-	-	10	2	-	-	-
M1	9	0.5	-	10	-	9	RBF	0.01

Tabla 4.9. Configuración de los algoritmos de clasificación en Weka para los conjuntos C1, C2, C3, C4, M1

Algoritmos de clasificación	<i>Accuracy</i>					<i>Presition</i>					<i>F-measure</i>					<i>Recall</i>				
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>M1</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>M1</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>M1</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>M1</i>
Conjuntos de entrenamiento																				
<i>K</i> -vecinos más cercanos	100%	96.60%	88.74%	92.42%	78.65%	100%	96.82%	90.30%	93.20%	80.24%	100%	96.60%	87.70%	92.16%	81.00%	100%	96.60%	88.72%	92.38%	78.68%
Redes neuronales artificiales	100%	99.40%	98.68%	98.14%	91.75%	100%	99.40%	98.70%	98.23%	91.84%	100%	99.40%	98.66%	98.14%	91.78%	100%	99.30%	98.66%	98.14%	91.78%
Máquinas de vectores de soporte	100%	99.90%	100%	99.38%	92.06%	100%	99.90%	100%	99.40%	92.16%	100%	99.90%	100%	99.40%	92.10%	100%	99.90%	100%	99.40%	92.10%

Tabla 4.10. Promedios de métricas utilizando LBPUr aplicado a los conjuntos C1, C2, C3 y C4

4.2. Resultados experimentales de la segunda etapa

En esta etapa, el desarrollo del sistema de reconocimiento de expresiones faciales se realizó en lenguaje C++ bajo el IDE de programación Qt. Para la implementación de los métodos de caracterización y clasificación dentro de C++ se utilizó OpenCV. De acuerdo con los resultados obtenidos en la etapa 1 y la información recopilada en el estado del arte (ver la sección 2.8). SVM fue el algoritmo de clasificación seleccionado para el desarrollo del sistema, de la misma forma los métodos de caracterización seleccionados son LBP Uniforme con implementación para rostros y filtros de Gabor. Cabe mencionar que OpenCV no cuenta con un método de downsampling (disminución de resolución) similar al implementado en MATLAB lo que limitó seleccionar únicamente un filtro para caracterizar las imágenes utilizando el método de Filtros de Gabor

En cuanto a los conjuntos de imágenes, estos fueron modificados con el objetivo de tener conjuntos balanceados y con un mayor número de imágenes. En esta segunda etapa se utilizó un conjunto nombrado como Ms1, el cual está descrito en la tabla 4.11. Para poder crear los conjuntos balanceados se incluyeron imágenes de la base de datos del (MIT por sus siglas en inglés Massachusetts Institute of Technology). De tal forma que para el conjunto neutro se incluyeron 11 imágenes y para el conjunto de repugnancia se incluyeron 48 imágenes.

Imágenes	Cantidad	Etiqueta
Normal	100	0
Feliz	100	1
Repugnancia	100	2
Total de imágenes	300	-

Tabla 4.11. Descripción general de conjunto Ms1

Con respecto a la configuración de SVM, en las tablas 4.12 y 4.13 se puede observar la configuración implementada para cada experimentación. Por otro lado, se utilizó 10-cross-validation para evaluar el desempeño de cada experimentación y se utilizaron las métricas *Accuracy*, *Precision*, *F-measure* y *Recall*. Cabe destacar que fueron programadas desde cero las métricas y la técnica de *10-cross-validation*, ya que OpenCV no cuenta un módulo que permita el cálculo de estas. Los resultados experimentales para esta etapa se muestran en la tabla 4.14.

Configuración de SVM utilizando LBP Uniforme con aplicación para rostros

Elementos	Valor
Tipo de SVM	NU_SVC
Tipo de kernel	RBF
Tolerancia Rango	(1e7, 1e-6)
Número de iteraciones	10000000
<i>Epsilon</i>	2.22045E-16
<i>Gamma</i>	0.50625
<i>nu</i>	0.01

Tabla 4.12. Configuración de SVM utilizada en OpenCV para la experimentación de LBPUr**Configuración de SVM utilizando Filtros de Gabor**

Elementos	Valor
Tipo de SVM	C_SVC
Tipo de kernel	RBF
Tolerancia Rango	(1e7, 1e-6)
Número de iteraciones	10000000
<i>Epsilon</i>	2.22045E-16
<i>Gamma</i>	0.00001
<i>C</i>	312.5

Tabla 4.13. Configuración de SVM utilizada en OpenCV para la experimentación de Filtros de Gabor

Método de caracterización	<i>Accuracy</i>	<i>Precision</i>	<i>F-measure</i>	<i>Recall</i>
LBP Uniforme con aplicación para rostros	86.53 %	86.53 %	86.44 %	86.45 %
Filtros de Gabor	83.47 %	83.48 %	83.41 %	83.58 %

Tabla 4.14. Resultados experimentales de SVM utilizando LBPUr y Filtros de Gabor

4.3. Comparación de resultados

En esta subsección fueron seleccionados los mejores resultados obtenidos de cada experimentación de acuerdo con la métrica *accuracy* (exactitud), con el objetivo de realizar una comparación entre los métodos de caracterización y los algoritmos de clasificación, La comparación de resultados se realizó de acuerdo a cada conjunto de entrenamiento.

En la figura 4.1 se puede observar como el método con mayor afectividad fue LBPUr utilizando SVM como algoritmo de clasificación, el cual obtuvo un 100 % de *accuracy*

seguido de un 85.14 % de accuracy utilizando filtros de Gabor y RNAs. Por último, el peor resultado fue para el método LBP el cual presento un 69.91 % de accuracy, utilizando como clasificador SVM.

Tal como se puede observar en la figura 4.2, para el conjunto C2 el método que presento la mejor exactitud de acuerdo a la clasificación de sus características es LBPUr, el cual obtuvo un 100 % utilizando SVM como clasificador. El segundo mejor resultado fue obtenido por los filtros de Gabor con un 89.90 % de exactitud utilizando SVM. Por otra parte, el peor resultado fue de 74.90 % de exactitud utilizando LBP y SVM.

Para el conjunto C3, la figura 4.3 muestra una gráfica donde se puede observar que el método con mayor exactitud fue LBPUr el cual presento un 100 % utilizando como clasificador a SVM. El segundo mejor resultado fue de 87.28 % utilizando filtros de Gabor y RNAs. Por último, el peor resultado fue de 74.44 % de exactitud por el método PCA utilizando como clasificador SVM.

En la figura 4.4 se puede observar los mejores resultados experimentales para la clasificación del conjunto C4 y el método que obtuvo el mejor desempeño entre ellos fue LBPUr con un 99.38 % de exactitud, utilizando como clasificador SVM. El método con el segundo mejor resultado fue filtros de Gabor utilizando como clasificador RNAs, el cual presento un 94.66 % de exactitud. Por último, el método con el peor desempeño fue PCA utilizando como clasificador a las RNAs, el cual presento un 80.37 % de exactitud.

Como fue descrito en las secciones 4.1.4 y 4.2, el conjunto M1 fue caracterizado únicamente con LBPUr y el conjunto Ms1 fue caracterizado con los métodos LBPUr y filtros de Gabor pero sólo utilizo a SVM como algoritmo de clasificación. Los resultados para el conjunto M1 se pueden observar en la gráfica de la figura 4.5, donde el mejor resultado fue de 92.06 % de exactitud utilizando a SVM, seguido de un 91.75 % de exactitud utilizando como clasificador a las RNAs y el peor resultado fue de 78.65 % de exactitud utilizando Knn como algoritmo de clasificación.

Por otra parte, los resultados para el conjunto Ms1 son representados en una gráfica de la figura 4.6, donde se muestra que el método que obtuvo mejor desempeño fue LBPUr con un 86.53 % de exactitud, además para este experimento la expresión feliz fue reconocida en un 99.4 %, también la expresión normal fue reconocida en un 80.2 % seguida de la expresión de repugnancia con un 80 % . Para los filtros de Gabor se obtuvo un 83.47 % de exactitud, donde la expresión feliz fue reconocida en un 88.8 %, la expresión normal en un 85.4 % y la expresión de repugnancia presento un 76.2 %.

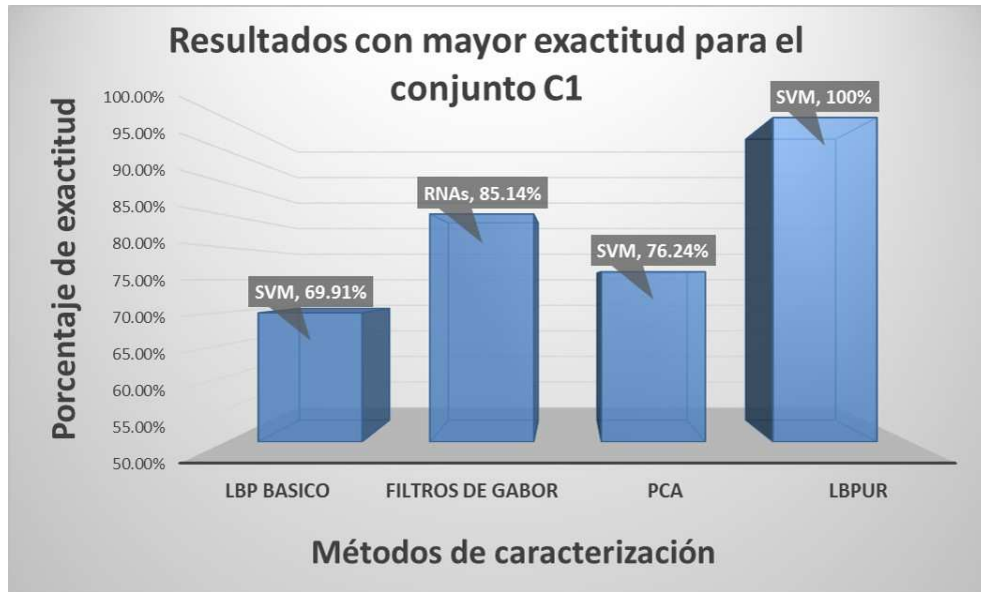


Figura 4.1. Comparación de los mejores resultados para la clasificación del conjunto C1.



Figura 4.2. Comparación de los mejores resultados para la clasificación del conjunto C2



Figura 4.3. Comparación de los mejores resultados para la clasificación del conjunto C3



Figura 4.4. Comparación de los mejores resultados para la clasificación del conjunto C4

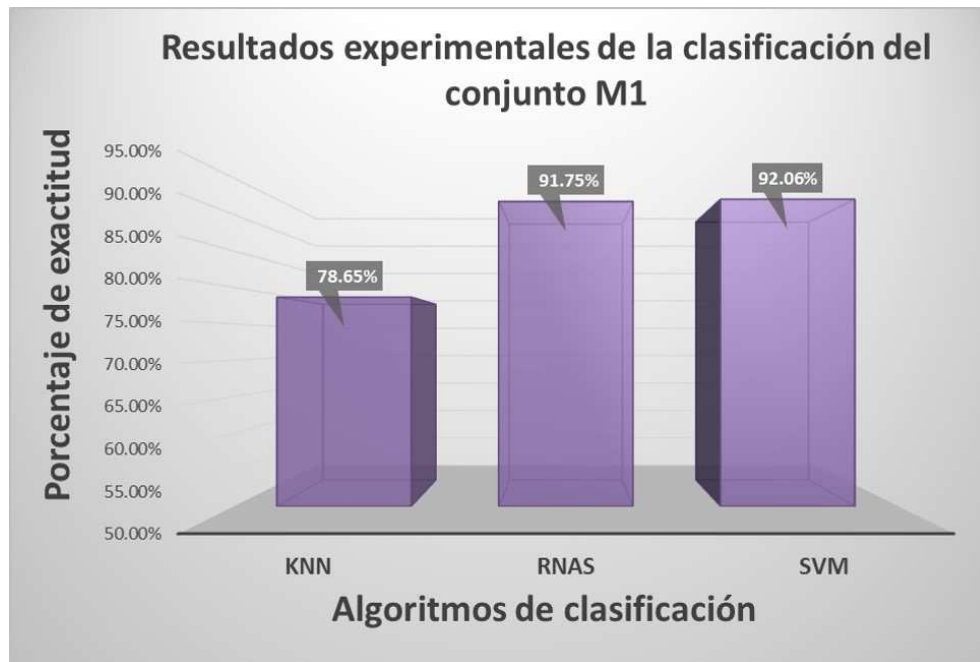


Figura 4.5. Resultados experimentales de la clasificación del conjunto M1

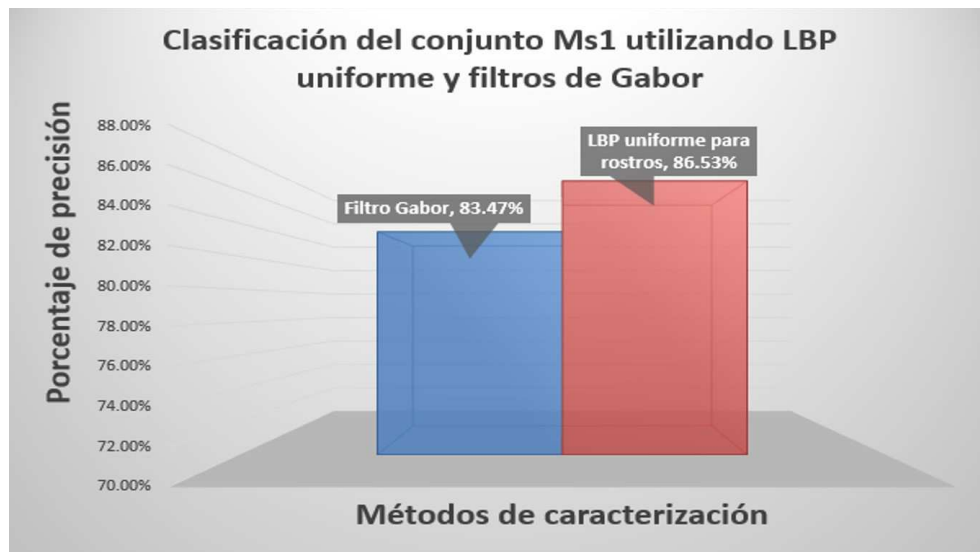


Figura 4.6. Resultados experimentales de la clasificación del conjunto Ms1

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo de tesis se propuso el desarrollo de un sistema de reconocimiento de expresiones faciales utilizando técnicas de visión por computadora y aprendizaje automático que permitiera apoyar las pruebas de usabilidad moderada. Para desarrollar el sistema se experimentó con diferentes métodos de caracterización de imágenes como LBP, LBPUr, PCA y Filtros de Gabor. También, se utilizaron diferentes algoritmos de clasificación como KNN, RNAs y SVM. Los datos de entrenamiento fueron imágenes de rostros con expresiones faciales procedentes de alumnos de la Universidad Politécnica de Puebla, estas imágenes fueron divididas en diferentes conjuntos de acuerdo a las expresiones faciales feliz, repugnancia y neutral. Para la mayoría de los conjuntos creados, los métodos de caracterización que presentaron mayor rendimiento para la clasificación fueron LBPUr y filtros de Gabor utilizando como algoritmos de clasificación a SVM y RNAs, por este motivo el sistema utiliza LBPUr y filtros de Gabor como métodos de caracterización y como clasificador utiliza el método de SVM.

El sistema está desarrollado en lenguaje C++ bajo el IDE de programación Qt, utilizando la librería OpenCV para el desarrollo de las técnicas de visión por computadora y aprendizaje automático. El usuario puede elegir entre los dos métodos de caracterización (LBPUr y filtros de Gabor). Por otra parte, el sistema es capaz de reconocer 3 expresiones faciales (feliz, normal y repugnancia), también registra la transición entre cada emoción, almacenando el segundo cuando se genera el cambio de una emoción a otra, esta información se muestra de forma gráfica para que pueda ser analizada por los observadores de las pruebas de usabilidad. El sistema presenta problemas con la iluminación, principalmente cuando la iluminación no es uniforme o cuando es excesiva en el rostro. Como trabajo extra, el sistema fue configurado para captura de video de un webcam remotamente sí es necesario, esto permite que su ejecución sea transparente para los usuarios.

Como trabajos a futuro se presenta cuatro propuestas, La primera, es agregar las expresiones faciales de miedo, tristeza, sorpresa, enojo para completar las 6 emociones básicas. La segunda propuesta es agregar un selector de atributos que permita la reduc-

ción de características en el conjunto de entrenamiento y seleccionar la información con mayor prioridad. Como tercera propuesta es crear módulos en el sistema que permitiera monitorear un mayor número de cámaras en entornos locales y remotos. Como última propuesta es realizar una comparación utilizando una modificación del método LBP nombrada como (LTP por sus siglas en ingles Local Ternary Pattern), que permitiría atacar problemas con la iluminación.

Apéndice A

Sistema de reconocimiento de expresiones faciales

El objetivo principal del proyecto fue desarrollar un sistema que permita reconocer al menos dos expresiones faciales de acuerdo a las descritas por Paul Ekman. El sistema desarrollado en este proyecto de investigación puede reconocer tres expresiones faciales: Normal, Feliz y Repugnancia. El sistema está desarrollado en lenguaje C++ utilizando como ambiente de desarrollo integrado la herramienta Qt que permite el desarrollo de aplicaciones gráficas y profesionales, también se incluyeron librerías de OpenCV. El objetivo de esta sección es describir el funcionamiento general del sistema, explicar el funcionamiento de cada ventana y elemento que conforma el entorno gráfico.

A.1. Funcionamiento General

En la figura A.1, se muestra la ventana principal del sistema donde se puede apreciar los componentes de la ventana encerrados en recuadros rojos. A continuación cada componente será descrito en relación a su funcionamiento y la unión que tienen con otras ventanas del sistema.

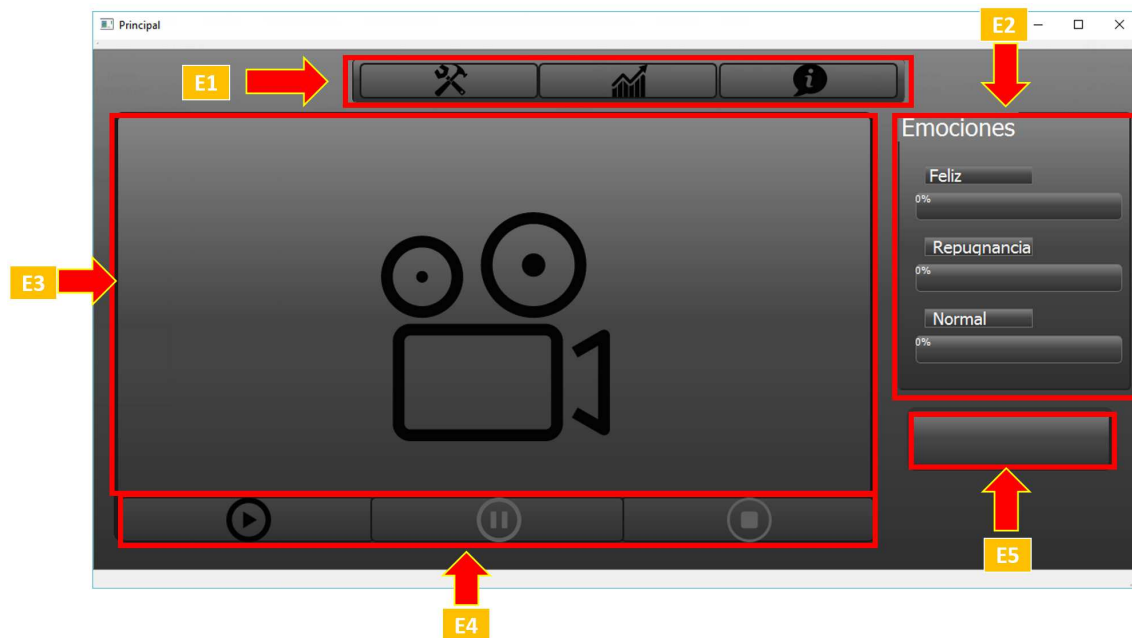


Figura A.1. Ventana principal del sistema de reconocimiento

- **E1** es menú principal el cual contiene tres botones. El primer botón de izquierda a derecha permite abrir la ventana titulada como Configuración, la cual se puede observar en la figura A.2, el siguiente botón permite visualizar y ocultar la ventana titulada como Sensor emocional, la cual contiene un gráfico lineal generado en tiempo real con respecto a las emociones identificadas durante una prueba (ver figura A.4 índice (b)), El último botón permite abrir la ventana titulada como Información general, ésta ventana contiene información genérica del sistema (ver figura A.3).
- **E2** en este elemento permite visualizar el porcentaje de expresiones identificadas en tiempo real
- **E3** Este elemento permite visualizar el video en tiempo real, se activa con los botones del elemento E4
- **E4** Este contiene tres botones los cuales pueden iniciar, pausar o detener el video mostrado en elemento E3
- **E5** En este elemento se muestra los mensajes de error y los frames por segundo a los que está trabajando el sistema.

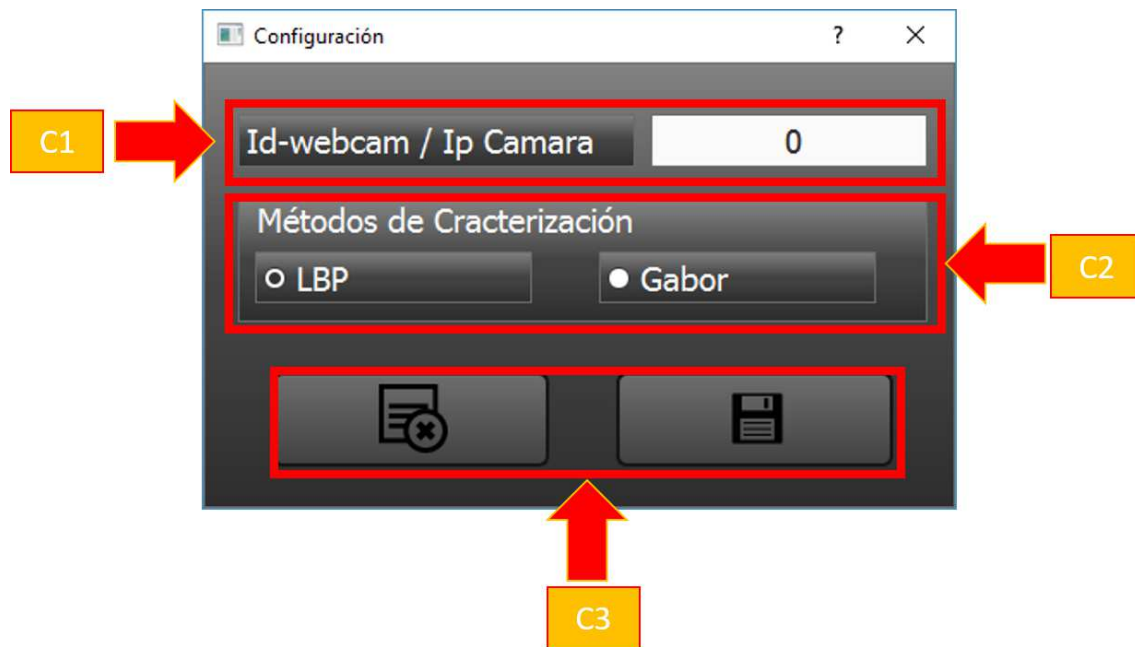


Figura A.2. Ventana de configuración

Como ya se mencionó anteriormente, la figura A.2, presenta la ventana de configuración donde el componente etiquetado como C1 permite la elección del id de la cámara, si la computadora que ejecuta el sistema contiene más de una cámara web es necesario seleccionar el id de la cámara con la que se desea trabajar, la numeración de las cámaras inicia desde 0 y por defecto el sistema está configurado con este id; también se puede acceder a cámaras IP sustituyendo el id de la cámara por la dirección de red donde se encuentra ubicada la cámara IP. El componente C2 permite seleccionar el método de caracterización (LBP y Gabor), por defecto el método seleccionado es LBP. Por último, el componente C3 contiene 2 botones; el primer botón de izquierda a derecha permite cancelar los cambios realizados en la interfaz y el siguiente botón permite guardar los cambios ejecutados. La configuración por defecto se puede apreciar en la figura A.2

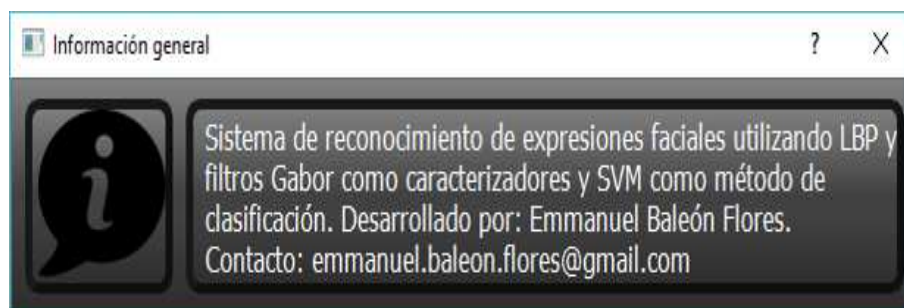
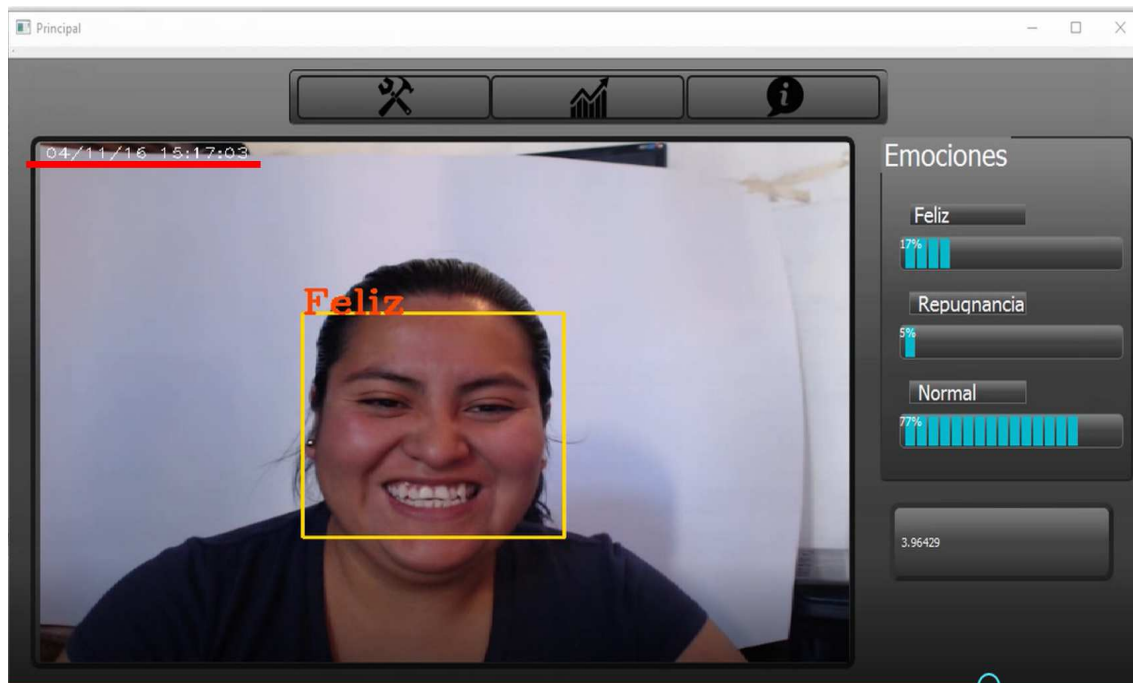


Figura A.3. Ventana de información general del sistema

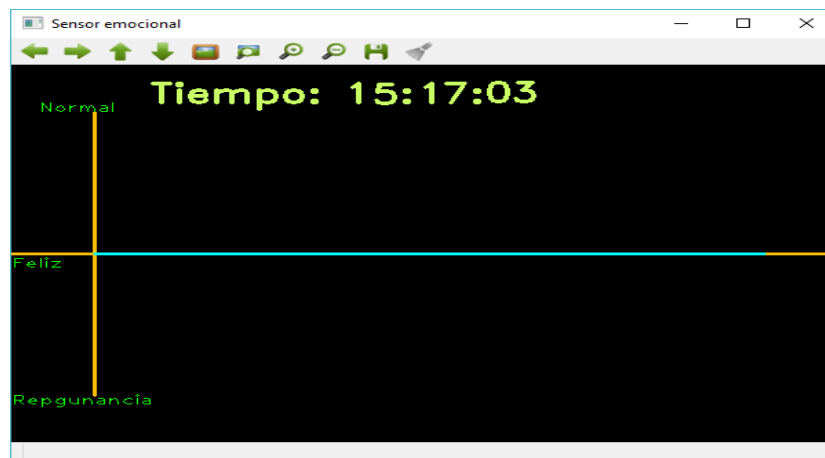
En la figura A.4 índice (b), se puede observar la ventana sensor emocional, tal como se ha mencionado anteriormente, esta genera un grafica lineal en relación a las 10 expresiones faciales detectadas por el sistema. Otro elemento que se encuentra dentro de la gráfica es la hora del sistema, el objetivo de mostrar la hora del sistema es generar una relación entre tiempo y expresiones detectadas.

La figuras A.4, A.5 y A.6 presenta el reconocimiento de expresiones durante la ejecución del sistema, también proyecta como el sistema genera un registro de cada reconocimiento. Como se puede observar dentro del índice (b) de las figuras antes mencionadas, donde se muestra la imagen de la ventana sensor emocional, en ella se presenta un grafica lineal del reconocimiento de los 10 últimos *frames*¹ detectados con una expresión facial. Por otra parte, cuando se presiona la tecla stop para detener el reconocimiento de expresiones, se genera un reporte final de todas las expresiones detectadas durante la prueba. Este reporte se presenta en la ventana titulada histograma general como se puede ver en el índice (c) de las figuras A.4, A.5 y A.6. En esta incluye una gráfica donde se registran las expresiones detectadas y el momento cuando se detectó, para que una expresión sea registrada en el histograma general es necesario detectar 8 frames con la misma expresión facial consecutivamente, esto indica que la expresión debe durar un aproximado de 2 a 2.5 segundos

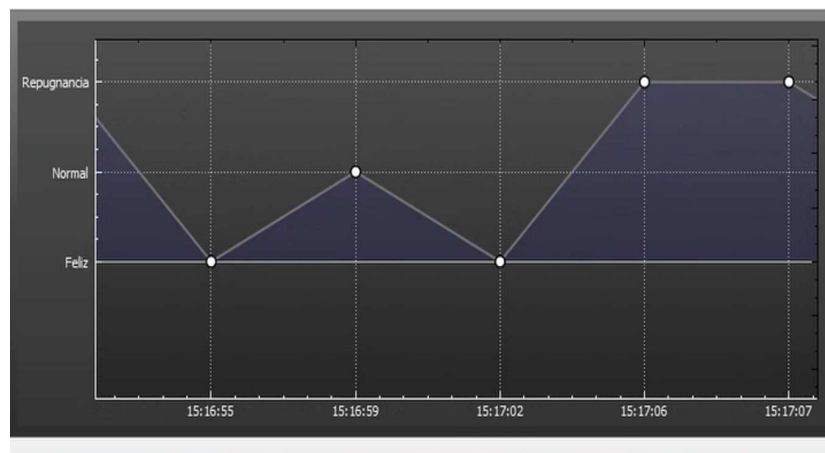
¹*frame*:imagen instantánea con las que se divide un video las cuales al ser proyectadas secuencialmente dan la sensación de movimiento.



(a) Reconocimiento de expresión Feliz en la ventana principal

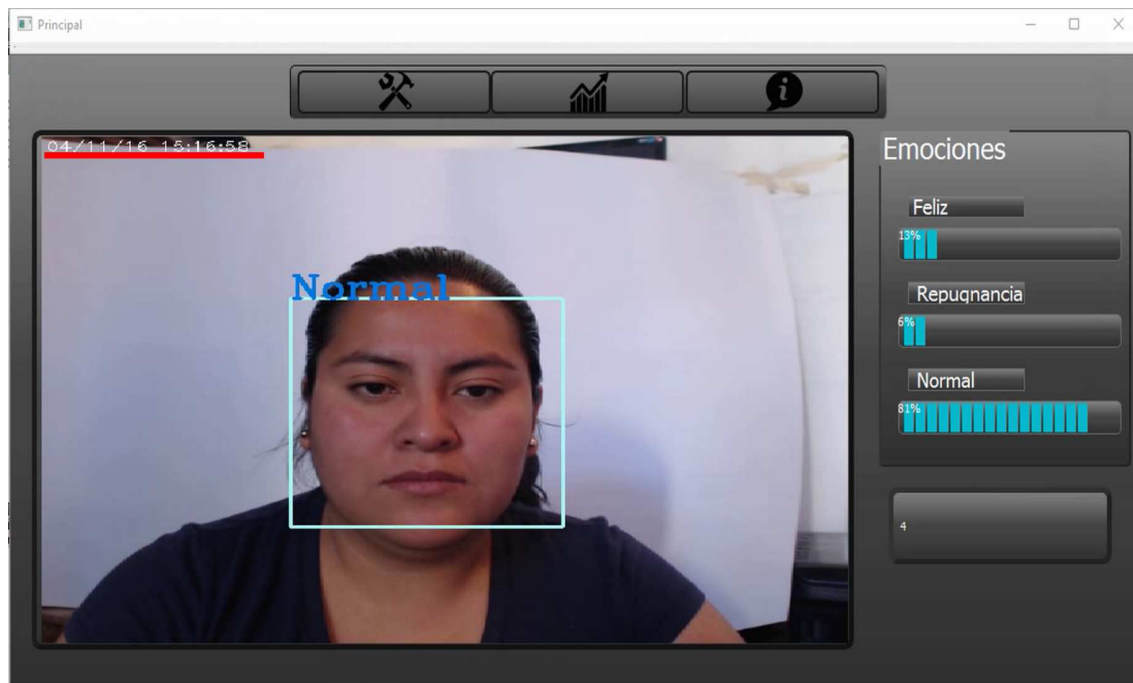


(b) Ventana sensor de emocional detectando la expresión feliz

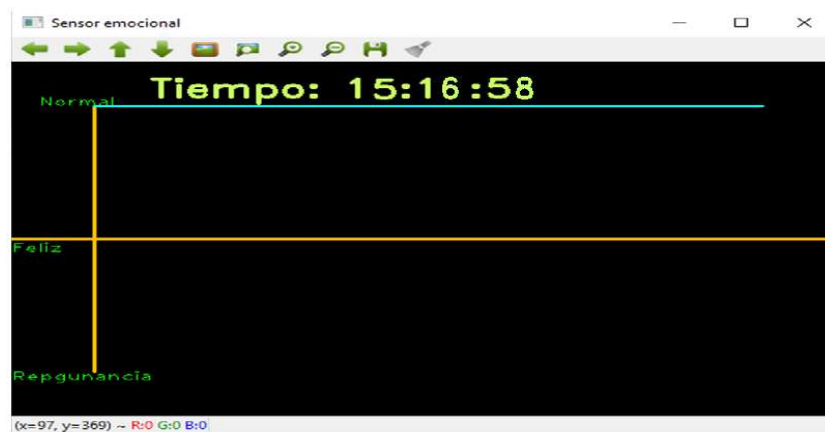


(c) Reporte desde la ventana histograma general

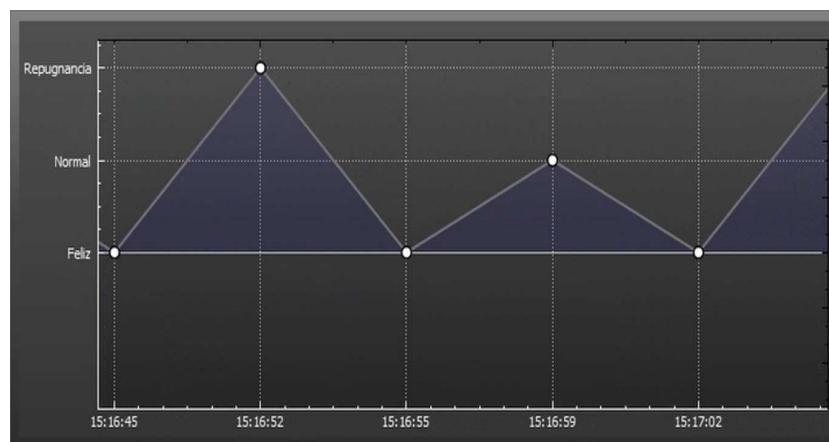
Figura A.4. Ejemplo del reconocimiento de la expresión feliz



(a) Reconocimiento de expresión normal en la ventana principal

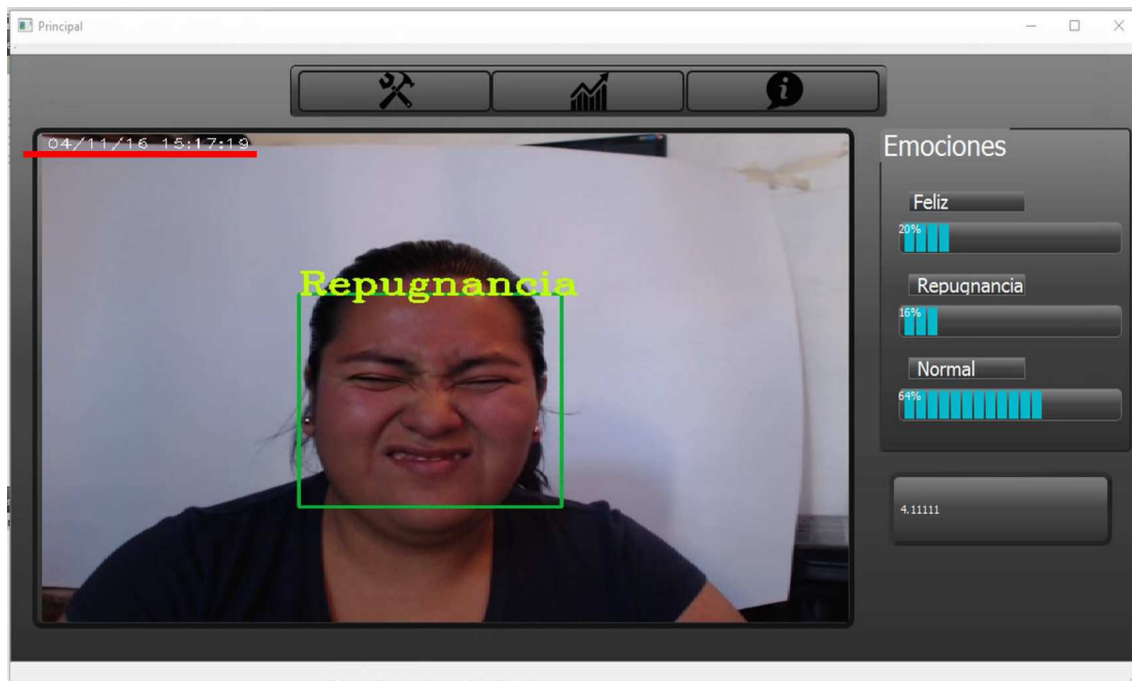


(b) Ventana sensor de emocional detectando la expresión normal

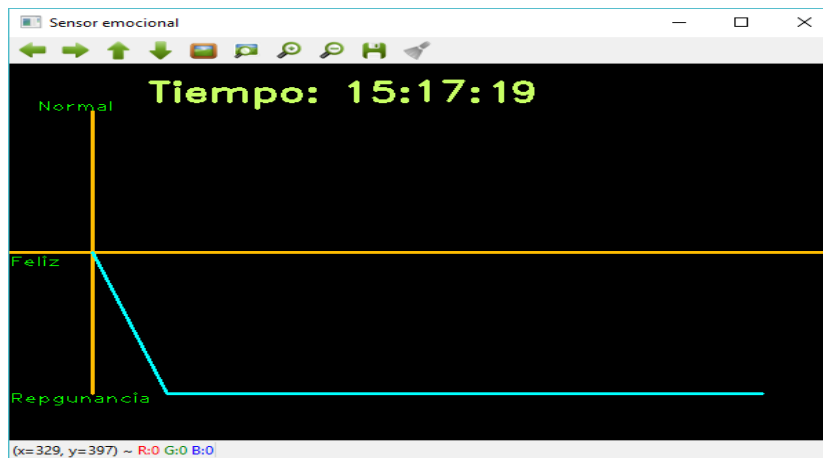


(c) Reporte desde la ventana histograma general

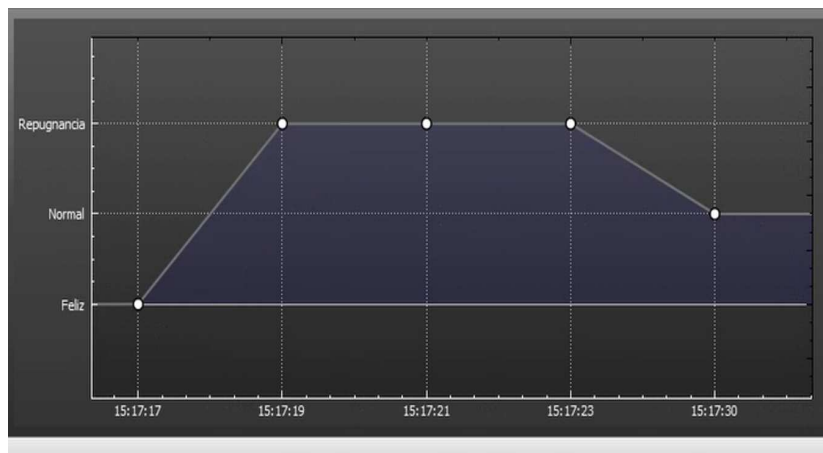
Figura A.5. Ejemplo del reconocimiento de la expresión normal



(a) Reconocimiento de expresión repugnancia en la ventana principal



(b) Ventana sensor de emocional detectando la expresión repugnancia



(c) Reporte desde la ventana histograma general

Figura A.6. Ejemplo del reconocimiento de la expresión repugnancia

Bibliografía

- [Aha *et al.*, 1991] Aha, D. W., Kibler, D., y Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.
- [Baggio, 2012] Baggio, D. L. (2012). *Mastering OpenCV with practical computer vision projects*. Packt Publishing Ltd.
- [Bartlett *et al.*, 2005] Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., y Movellan, J. (2005). Recognizing facial expression: machine learning and application to spontaneous behavior. En *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volumen 2, pp. 568–573. IEEE.
- [BMVA, 2014] BMVA (15 de Octubre del 2014). <http://www.bmva.org/visionoverview>.
- [Bradski y Kaehler, 2008] Bradski, G. y Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc."
- [Brave y Nass, 2002] Brave, S. y Nass, C. (2002). Emotion in human-computer interaction. in j. jacko & a. sears (eds.).
- [Busso *et al.*, 2004] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., y Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. En *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 205–211. ACM.
- [Ekman y Oster, 1981] Ekman, P. y Oster, H. (1981). Expresiones faciales de la emoción. *Estudios de Psicología*, 2(7):115–144.
- [García, 2008] García, I. D. S. (2008). *Visión Artificial y Procesamiento Digital de Imágenes usando Matlab*. Ibarra, Australia.
- [Grauman y Leibe, 2010] Grauman, K. y Leibe, B. (2010). *Visual object recognition*. Morgan & Claypool Publishers.
- [Guo y Dyer, 2005] Guo, G. y Dyer, C. R. (2005). Learning from examples in the small sample case: face expression recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(3):477–488.

- [Hassan Montero *et al.*, 2003] Hassan Montero, Y., Fernández, M., y J., F. (2003). Método de test con usuarios. *No sólo usabilidad*.
- [las-emociones.com, 2014] las-emociones.com (06 de Noviembre del 2014). <http://las-emociones.com/>.
- [Learned-Miller, 2011] Learned-Miller, E. (2011). Introduction to computer vision. *University of Massachusetts, Amherst, Department of Computer Science*.
- [Martín, 2005] Martín, F. J. (2005). Diseño de información. [Internet; descargado 6-noviembre-2015].
- [Martinez y Du, 2012] Martinez, A. y Du, S. (2012). A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *The Journal of Machine Learning Research*, 13(1):1589–1608.
- [Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- [Mora-Florez *et al.*, 2008] Mora-Florez, J., Morales-España, G., y Barrera-Cárdenas, R. (2008). Evaluación del clasificador basado en los k vecinos más cercanos para la localización de la zona en falla en los sistemas de potencia. *Ingeniería e Investigación*, 28(3):81–86.
- [Mora Lazarini, 2008] Mora Lazarini, A. (2008). *FUNCIONES GABOR BIDIMENSIONALES PARA EL ANALISIS Y CLASIFICACION DE TEXTURAS*. Tesis doctoral.
- [Nombela Escobar, 2011] Nombela Escobar, B. (2011). Aplicación de técnicas de aprendizaje automático para la extracción de información en textos farmacológicos.
- [Olmedo *et al.*, 2011] Olmedo, E., De La Calleja, J., Benitez, A., y Medina, M. A. (2011). Procesamiento puntual de imágenes digitales usando cómputo paralelo. *Vision Politécnica, Universidad Politécnica de Puebla, Puebla, México*.
- [Pietikäinen *et al.*, 2011] Pietikäinen, M., Hadid, A., Zhao, G., y Ahonen, T. (2011). *Computer vision using local binary patterns*, volumen 40. Springer.
- [RAE, 2014] RAE (04 de Noviembre del 2014). <http://www.rae.es/>.
- [Ratiff y Petterson, 2010] Ratiff, S. M. y Petterson, E. (2010). Emotion recognition using facial expressions with active appearance models.
- [Ratliff y Patterson, 2008] Ratliff, M. S. y Patterson, E. (2008). Emotion recognition using facial expressions with active appearance models. En *Proceedings of the Third IASTED International Conference on Human Computer Interaction, (Innsbruck, Austria)*, pp. 138–143.

- [Shan *et al.*, 2005] Shan, C., Gong, S., y McOwan, P. W. (2005). Robust facial expression recognition using local binary patterns. En *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volumen 2, pp. II-370. IEEE.
- [Shan *et al.*, 2009] Shan, C., Gong, S., y McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816.
- [SpA, 2015] SpA, I. D. A. (2015). 6 test de usabilidad fundamentales para asegurar la experiencia de usuario. [Internet; descargado 19-noviembre-2016].
- [Sucar y Gómez, 2011] Sucar, L. E. y Gómez, G. (2011). Visión computacional. *Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México*.
- [Szeliski, 2010] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer.
- [uxpa.org, 2014] uxpa.org (04 de Noviembre del 2014). <https://uxpa.org/resources/definitions-user-experience-and-usability>.
- [Vélez *et al.*, 2003] Vélez, J. F., B, M. A., Ángel, S., y Marín, J. L. E. S. (2003). *Visión por computador*.
- [Verma *et al.*, 2013] Verma, D., Dhaka, V., y Agrwal, S. (2013). An improved average gabor wavelet filter feature extraction technique for facial expression recognition. *International Journal on Innovations in Engineering and Technology*, 2:2319–1058.
- [Wikipedia, 2014a] Wikipedia (2014a). Opencv — wikipedia, la enciclopedia libre. [Internet; descargado 14-noviembre-2014].
- [Wikipedia, 2014b] Wikipedia (2014b). Paul ekman — wikipedia, la enciclopedia libre. [Internet; descargado 6-noviembre-2014].
- [Xu *et al.*, 2013] Xu, C., Du, P., Feng, Z., Meng, Z., Cao, T., y Dong, C. (2013). Multi-modal emotion recognition fusing video and audio. *Appl. Math*, 7(2):455–462.
- [Zhang *et al.*, 1998] Zhang, Z., Lyons, M., Schuster, M., y Akamatsu, S. (1998). Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. En *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 454–459. IEEE.
- [Zhao y Pietikainen, 2007] Zhao, G. y Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928.