PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

PROTOTIPO COMPUTACIONAL PARA LA DETECCIÓN Y CLASIFICACIÓN DE EXPRESIONES FACIALES MEDIANTE LA EXTRACCIÓN DE PATRONES BINARIOS LOCALES

Tesis para optar por el Título de Ingeniero Informático, que presenta el bachiller:

Yulian André Cama Castillo

ASESOR: Dr. César Beltrán Castañón

Lima, Febrero del 2015

Resumen

La expresión facial es uno de los medios más comunes y naturales que tiene el ser humano, para transmitir información sobre sus emociones e intenciones. Su análisis es un área de investigación activa desde el trabajo realizado por Charles Darwin en 1872 y recientemente, su reconocimiento de forma automatizada, ha tenido un gran desarrollo gracias a los avances en áreas como visión computacional y aprendizaje de máquina.

A pesar de lo mencionado anteriormente, uno de los principales retos que se tiene por resolver, para lograr un sistema robusto, radica en el modo en que se extraen las características faciales; es decir, el modo en que el computador representará el rostro, que facilite la distinción de las expresiones. Factores como la iluminación de la imagen, la cercanía o lejanía del rostro en la imagen, o incluso el ángulo del rostro (oclusión) pueden afectar la correcta extracción de las características por lo que deben ser abordados para lograr de forma ideal el reconocimiento de las expresiones faciales.

Este proyecto de investigación se enfoca en el estudio de la aplicación del descriptor LBP, como método basado en apariencia, para describir las expresiones en el rostro y así poder clasificarlas entre las emociones básicas mediante el uso de técnicas Boosting de aprendizaje de máquina.

Dedicatoria

A Dios, por la vida.

A mis amados padres Manuel y Lorena,

por su apoyo incondicional, guía y el gran esfuerzo que han demostrado por el desarrollo y superación de sus hijos en todo este tiempo.

Agradecimientos

Al Dr. César Beltrán, mi asesor, por su apoyo y guía para encaminar y realizar la idea que se tenía inicialmente como tesis de grado.

A Filomen Incahuanaco, mi co-asesor, por ayudarme a entender muchos de los temas que inicialmente no comprendía, así como por su apoyo constante a lo largo de todo el proyecto.

A ambos muchas gracias por el tiempo, conocimientos y paciencia brindados durante todo el desarrollo de mi tesis de grado.

Índice de Contenido

Índice de Figuras

Índice de Tablas

CAPITULO 1

1 Generalidades

1.1 Problemática

Las personas, como seres sociables, han desarrollado con el tiempo medios verbales como no verbales para poder comunicarse y brindar información [4]. Entre estos últimos se encuentran los gestos [5, 6], mediante los cuales una persona puede proporcionar distintas formas de información y así, por ejemplo, brindar la ubicación de un lugar mediante su señalización con el dedo índice; mostrar la emoción del enojo o ira a través del fruncimiento del ceño o quizás indicar el estado de una situación al mostrar el pulgar levantado.

De estos gestos, aquellos que les son más comunes y naturales a las personas se dan a través del rostro y son conocidos como expresiones faciales [5, 7, 8]. Las expresiones faciales han sido y son uno de los temas de estudio y análisis más tratados en diversos campos, como la psicología, antropología, sociología, fisiología, neurociencia, neuropsiquiatría, entre otros [7-10], debido a que transmiten información sobre las emociones e intenciones de las personas.

Su reconocimiento automático ha tomado gran impulso gracias al rápido avance en áreas como la visión por computador, interacción humano-computador, aprendizaje de máquina y el conocimiento humano [11-14]; sin embargo, a partir de las revisiones dadas sobre el estado del arte [15-19], se observó que existen diversas dificultades para que el computador pueda interpretar correctamente la información del rostro. Por lo que las investigaciones en el tema tratan de solucionar las distintas dificultades a través de diferentes métodos y enfoques sin poder llegar a una propuesta única de solución.

Una de las principales fuentes de esta dificultad es la detección del rostro, ya que es la etapa inicial a través de la cual se obtiene la región a analizar en la imagen dada al computador [14]; sin embargo, los problemas se manifiestan también una vez ya adquirida la región del rostro en la etapa de extracción de características; la iluminación, la sombra, el brillo, la distancia y la pose [18] son elementos que afectan el correcto funcionamiento de estas etapas. Si la pose del rostro no permite

mostrar todas las características en él, un software que base su identificación en el reconocimiento de ambos ojos fallará al encontrar un rostro inclinado o casi de perfil en la imagen; asimismo, si en la imagen una persona se encuentra muy alejada el rostro puede ser confundido con otro objeto o simplemente con una mancha, dependiendo de la resolución con que la imagen fue capturada o si esta presenta mucha sombra o iluminación que afecten la caracterización de la misma. Así los métodos clásicos para el procesado de la imagen serán inútiles al tratar de encontrar el rostro o representarlo [18], debido a la sensibilidad que presentan frente a estos elementos. En consecuencia, sin buenos métodos o algoritmos que tengan en cuenta dichas variables la detección y caracterización de un rostro será imprecisa o simplemente no ayudará al reconocimiento de la expresión.

Por otro lado, se encuentran los problemas generados por las diferencias individuales de las personas, diferencias que, como la forma del rostro, el color de la piel, la textura, el uso de barba, de anteojos o hasta el modo en que traen el cabello, pueden resultar como interferencia al tratar de obtener la información clave del rostro [14]. Esto debido a que mientras algunos de estos elementos ocultan las secciones importantes del rostro como los ojos, la nariz y la boca, otros marcan diferencias que complican la forma estandarizada en la extracción de sus características.

Así mismo, un problema que aporta otro punto de dificultad en los sistemas de reconocimiento facial es la clasificación o reconocimiento de las expresiones en los estados emocionales. Esta al ser la última etapa [14] realiza la comparación entre las características previamente extraídas de la imagen y las características provenientes de una base de datos con expresiones etiquetadas, aquí es donde surgen problemas de confusión entre expresiones similares que reflejan estados emocionales diferentes. Un ejemplo de esto, es cuando el software analiza una imagen y determina erróneamente que la persona presenta signos de enojo mientras que realmente experimenta asco. Este problema puede ocurrir debido a factores como una mala caracterización de la expresión, la intensidad de la expresión, expresiones espontáneas o deliberadas, o incluso debido a la transición de imágenes al expresar una emoción [14]. Estos aspectos afectan la clasificación debido a que, de forma general, no son comparadas con una expresión facial plena o debido a la brevedad de la expresión, por lo que la clasificación debería de darse en un nivel más detallado para poder reconocer los múltiples cambios en el rostro,

así como lo hacen las distintas variaciones de las unidades de acción en el sistema de codificación facial propuesto por Ekman y Friesen (FACS).

Finalmente, se hace alusión a un problema que de forma general complica el reconocimiento directo de la emoción a través del medio visual. El hecho de que las emociones sean multimodales [8, 15]; es decir, que su interpretación conlleve no solo el análisis de las expresiones faciales de un individuo, sino también al análisis de otros aspectos como el tono de la voz, la reacción corporal, entre otros. En este sentido, mientras el computador realice un análisis unimodal de la persona, éste solo podrá estimar su estado emocional, más no asegurarlo.

A partir de la variedad de problemas encontrados se puede concluir que el problema central es la dificultad en el reconocimiento automático de las expresiones de emociones en las imágenes digitales.

En consecuencia, formulado ya el problema central se define que las dificultades esenciales a tratar en este proyecto de fin de carrera se direccionan hacia la extracción de características del rostro y la clasificación de las expresiones. Así se plantea como objetivo el desarrollo de un prototipo software para la clasificación de expresiones faciales básicas a través del uso del método de descripción de patrones binarios locales, que ya ha aportado buenos índices de rendimiento para tareas de reconocimiento facial, robustez frente a cambios de iluminación, además de simpleza computacional [2], con el fin aportar más información respecto al comportamiento, capacidad y uso de este descriptor para tareas de reconocimiento de expresiones faciales.

1.1.1 Objetivo general

Implementar un prototipo computacional de análisis y procesamiento de imágenes que permita caracterizar el rostro humano mediante la extracción de patrones binarios locales (LBP) a fin de estimar las expresiones de emociones globales en personas (felicidad, tristeza, enojo, asco, miedo y asombro) propuestas por Ekman y Friesen.

1.1.2 Objetivos específicos

- (O1) Detectar y obtener de forma automática la región de interés (rostro humano) dentro de una imagen digital.
- (O2) Caracterizar el rostro obtenido mediante la extracción de las características globales relevantes aplicando patrones binarios locales.
- (O3) Clasificar la imagen del rostro caracterizado entre 6 expresiones faciales (felicidad, tristeza, enojo, asco, miedo y asombro) que representan las emociones globales según Ekman y Friesen.
- (O4) Establecer métodos de validación y métricas que permitan indicar y medir la eficiencia lograda en la etapa de entrenamiento y la etapa de clasificación respectivamente.

1.1.3 Resultados esperados

Para O1:

- (RE1) Realización del modelo de detección del rostro humano en las imágenes digitales a analizar que muestren una proporción completa de la cara.

Para O2:

- (RE2) Extracción de las características globales del rostro mediante la técnica LBP (Local Binary Pattern) sobre la región de interés previamente detectada que describe el rostro.

Para O3:

- (RE3) Realización y entrenamiento de un modelo computacional que discrimine entre 6 expresiones globales del rostro (felicidad, enojo, tristeza, miedo, asco y sorpresa).

Para O4:

- (RE4) Realización y configuración del método “Cross Validation” para la determinación de la eficiencia en la etapa de entrenamiento.
- (RE5) Métricas que permitan indicar la eficiencia lograda por el modelo en la etapa de clasificación.

1.2 Herramientas, métodos y metodologías

1.2.1 Introducción

Esta sección tiene como propósito dar a conocer de qué forma se tratará de llegar a la solución del problema planteado, es decir al objetivo general del proyecto de fin de carrera, mediante la definición de las herramientas, métodos y metodología seleccionados.

A continuación se muestra de forma resumida en la Tabla 1.1 el mapeo de las herramientas y métodos a usar en el proyecto alineados a cada resultado esperado en el mismo.

Resultados esperado	Herramientas a usarse
(RE1) Realización del modelo de detección del rostro humano en las imágenes digitales a analizar que muestren una proporción completa de la cara.	• OpenCV • Visual Studio/ QT • Framework Viola-jones
(RE2) Extracción de las características globales del rostro mediante la técnica LBP (Local Binary Pattern) sobre la región de interés previamente detectada que describe el rostro.	• OpenCV • Visual Studio/ QT • Método Local Binary Pattern(LBP)
(RE3) Realización y entrenamiento de un modelo computacional que discrimine entre 6 expresiones globales del rostro (felicidad, enojo, tristeza, miedo, asco y sorpresa).	• OpenCV • Visual Studio/ QT • Método Boosting
(RE4) Realización y configuración del método “Cross Validation” para la determinación de la eficiencia en la etapa de entrenamiento.	• Visual Studio /QT • Método Cross Validation
(RE5) Métricas que permitan indicar la eficiencia lograda por el modelo en la etapa de clasificación.	• Visual Studio/QT

Tabla 1-1 *– Mapeo de Resultados vs. Herramientas a utilizarse.*

1.2.2 Herramientas

1.2.2.1 OpenCV

OpenCV es una librería de código abierto que ayuda en el tratamiento de imágenes al proporcionar una infraestructura de visión por computador que simplifica y acelera la construcción de aplicaciones de visión sofisticadas. Esta cuenta con más de 500 funciones que cubren varias áreas del proceso de visión, como la detección de objetos, calibración de cámara, seguimiento en tiempo real, etc.

La librería está implementada en C/C++ y puede ejecutarse bajo sistemas operativos como Linux, Windows y Mac OS X. Está diseñada para la eficiencia computacional, así su implementación es optimizada y puede tomar ventaja de los procesadores multi-core [20].

Su elección para el desarrollo del prototipo, se basa principalmente por la facilidad que ofrece para implementar la visión artificial del computador, la cual es necesaria en el proyecto, sobre todo para su etapa inicial, la detección y procesamiento de la imagen; es decir permitirá reducir el tiempo de desarrollo.

1.2.2.2 Visual Studio

Es un conjunto completo de herramientas de desarrollo para sistemas operativos Windows que permite desarrollar aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Por ello permite escribir, compilar, depurar y ejecutar aplicaciones en todas las plataformas de Microsoft incluidos teléfonos, equipos de escritorio, tabletas, servidores y la nube. Soporta múltiples lenguajes de programación como C++, C#, Visual Basic .Net, F#, Java, Python, Ruby y PHP [21, 22].

Esta herramienta es seleccionada debido a las facilidades que ofrece al ser usado junto al OpenCV. Principalmente por el libro introductorio del OpenCV [20] y las múltiples fuentes de información que vinculan el uso del Visual Studio y la librería OpenCV, que servirán para el aprendizaje y uso de las herramientas para la implementación, prueba, seguimiento del código y depuración de errores del prototipo.

1.2.2.3 Qt

Qt es un entorno de desarrollo de código abierto para la construcción de aplicaciones multiplataforma con interfaces gráficas de usuario a través de las diversas herramientas que provee, así como también sin interfaz gráfica; es decir, desarrollos para línea de comandos y consolas. Este entorno provee una adecuación ideal para los desarrollos basados en C++ y además puede ser utilizado con diversos lenguajes de programación a través las ligaduras que provee [23].

Esta herramienta fue seleccionada debido principalmente a la facilidad que provee la generación de una interfaz gráfica cuando el código base usado es c++. Además de que provee una fácil integración con la librería OpenCV y sus características como la representación de imágenes. Estas características permitieron ver esta herramienta como necesaria para el desarrollo del prototipo de manera gráfica.

1.2.3 Métodos

1.2.3.1 Framework Viola-Jones

Es un marco de trabajo desarrollado por los ingenieros Paul Viola y Michael Jones que utiliza una serie de algoritmos e ideas para una robusta y extremadamente rápida detección visual. El marco cuenta de tres principales contribuciones: un entrenador de clasificadores basado en el algoritmo de aprendizaje AdaBoost, un algoritmo para la detección de objetos que utiliza la clasificación en “cascada” y una nueva representación de la imagen denominada “imagen integral”.

Inicialmente para la detección del objeto se hace uso de la “imagen integral”, que permite que la evaluación de las características sea mucho más rápida; esta imagen se obtiene a través la realización de unas pocas operaciones por pixel y que al finalizar permiten que la búsqueda de características en subregiones se transforme en una tarea de tiempo constante sin importar su escala en la subregión o posición de la misma.

Una vez obtenida la imagen el algoritmo la divide en subregiones de distintos tamaños y utiliza una serie de clasificadores, cada uno con un conjunto de

características visuales, para discriminar si en la imagen se encuentra el objeto o no [24].

Cada uno de estos clasificadores funciona como una etapa de evaluación por la que tiene que pasar la subregión, si el primer clasificador determina que es aceptado como el objeto, pasa al siguiente clasificador, más riguroso, y así sucesivamente para determinar que realmente es el objeto; en caso contrario, la subregión es descartada. A este tipo de clasificación se le denomina en "Cascada", ya que filtra y analiza de forma efectiva subregiones que pueden contener el objeto buscado, descartando y ahorrando recursos en subregiones que con certeza no lo contendrán.

1.2.3.2 Local Binary Pattern (LBP)

Es un método originalmente diseñado para la descripción de texturas, que realiza el etiquetado de cada pixel de un vecindario respecto al pixel umbral (U) en el mismo. Este genera un histograma de las etiquetas que es considerado como el descriptor de la textura. No trabaja directamente con la textura sino con una representación de la misma en escala de grises, lo que le permite obtener solo la información que requiere de la misma de forma más eficiente. De esta forma el histograma de las etiquetas puede ser considerado como un descriptor de la textura [2, 25].

Inicialmente, el método se basó en un vecindario de 3x3 pixeles, donde se plantea que para cada pixel "U" de la imagen se examinan sus 8 vecinos, es decir los 8 pixeles que se encuentran a su alrededor. Para cada uno se determina si su intensidad en la escala de grises es mayor o menor que la intensidad del pixel "U", en caso sea mayor se le asignará al pixel el valor de "1" y en caso contrario "0". Una vez asignados los valores de la vecindad se obtiene un número binario que es la concatenación de los valores asignados a los pixeles vecinos, comenzando de forma horaria desde el pixel superior izquierdo. Finalmente la transformación a decimal de dicho número binario es utilizada para etiquetar el pixel "U" dado, conocida también como el operador BLP [26].

La Figura 1.1 muestra cómo de un vecindario de 3x3 pixeles se construye el descriptor del pixel central; se obtiene el valor de la intensidad atribuido a este pixel y a partir de este se analiza a cada pixel vecino para umbralizar el vecindario a través de valores binarios. Finalmente se forma un número binario tras su

concatenación, el cual al ser transformado será la etiqueta de nuestro pixel central dado.

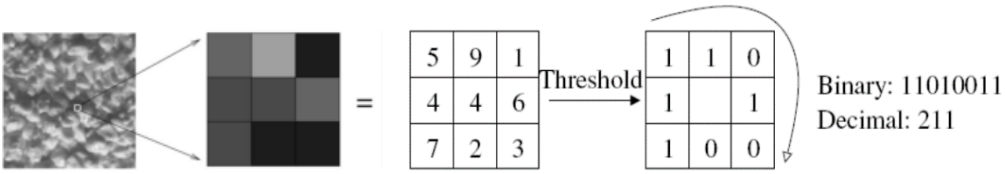

***Figura 1.1**- Ejemplo de aplicación LBP original. Imagen adaptada de [2, 3].*

La descripción formal del operador LBP [26] puede expresarse como:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

Donde "(x_c, y_c)" es el pixel central del vecindario con intensidad "i_c" en la escala de grises y "i_n" es la intensidad del pixel vecino. "s" es la función definida como:

$$S(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

1.2.3.3 Support Vector Machines (SVM)

Las máquinas de soporte vectorial o SVM son un grupo de métodos de aprendizaje supervisados que pueden ser aplicados para la clasificación o la regresión de elementos en un espacio. Formalmente la SVM construye un híper-plano o conjunto de híper-planos en un espacio de dimensiones muy altas con el fin de lograr una separación lineal de las distintas clases de elementos. Mediante su algoritmo, intenta definir estos híper-planos de tal forma que maximicen la separación entre las distintas clases de elementos en la dimensión [20, 27-30].

Las SVM se basan en el principio de inducción de la minimización del riesgo estructural (SRM) que se fundamentan en el hecho de que el error de generalización está limitado por la suma entre el error de entrenamiento y un término de intervalo de confianza que depende de la dimensión de Vapnik-

Chervonenkis (Cardinalidad del mayor conjunto de puntos que el modelos puede separar), por lo que se minimiza el límite superior del error de generalización [29].

En la figura 6 se observa un conjunto de elementos de dos clases distintas en una dimensión determinaba, separada por un híper-plano que distancia de forma óptima (con el máximo margen) todos los conjuntos de elementos de cada clase.

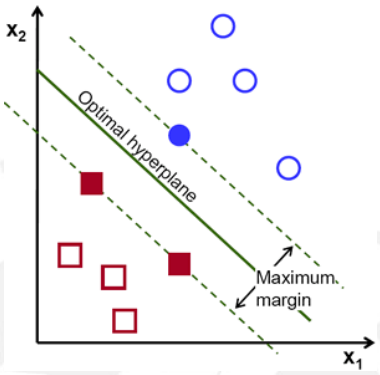

***Figura 1.2**- Híper-plano óptimo determinado en la dimensión. Imagen extraída de [30].*

1.2.3.4 Boosting Classification

El Boosting es un método de combinación dirigido hacia el aprendizaje de máquina basado en la idea de crear un predictor o clasificador robusto con una alta precisión a través de la combinación de varios predictores o clasificadores aparentemente débiles y no tan precisos. Por lo que en teoría este método reduce el error de aprendizaje al utilizar clasificadores débiles, que tienen una tasa de precisión de poco más del 50%, que al ser combinados forman un clasificador fuerte, que tendría una tasa de precisión mucho mayor [1, 31].

El algoritmo entrena iterativamente una serie de clasificadores débiles para crear clasificadores base respecto a un conjunto de muestras de entrenamiento que tiene una distribución de pesos específica. Tras cada iteración estos pesos son actualizados dependiendo de las muestras clasificadas erróneamente a través del clasificador base anterior, por lo que el siguiente clasificador débil pondrá mayor énfasis en el entrenamiento de estas muestras generando otro clasificador base con un criterio distinto, como se muestra en la Figura 1.3. Finalmente, una vez

terminadas las iteraciones se combinan los clasificadores teniendo un clasificador más preciso respecto al universo de muestras de entrenamiento.

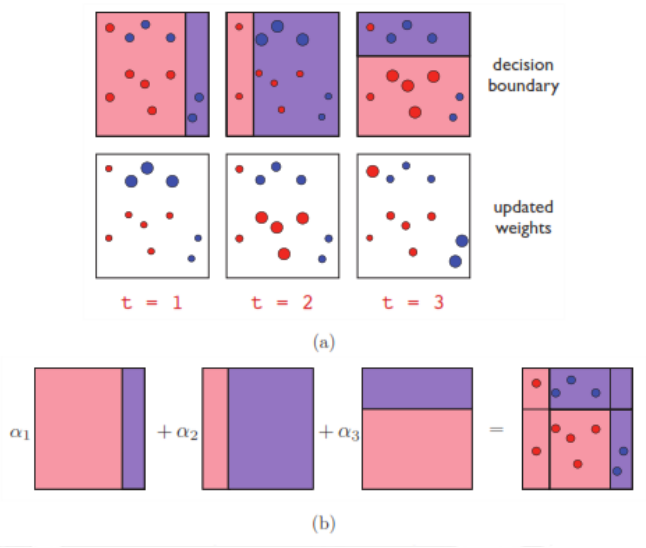

Figura 1.3 *- Ejemplo gráfico del funcionamiento de AdaBoost en 3 iteraciones. Imagen extraída de [1]*

Las variaciones del método existentes se dan en función del clasificador débil que se utilice, de las distribuciones de entrenamiento y del modo usado para realizar su combinación, siendo el algoritmo AdaBoost el primer algoritmo de aprendizaje que ponía en práctica este concepto.

1.2.3.5 Cross Validation (CV)

La validación cruzada (CV) es un método de análisis para estimar el performance de un modelo predictivo. Permite evaluar como los resultados de un análisis estadístico se generalizarán a un conjunto de datos específico (datos de entrenamiento), por lo que ayuda a ajustar el modelo en desarrollo.

En un problema de predicción, un modelo cuenta principalmente con un conjunto de datos conocidos, con los cuales se entrena (datos de entrenamiento), y con un conjunto de datos desconocidos, con los que el modelo prueba la calidad de su predicción (datos de prueba). Una ronda de la validación cruzada particiona todo el conjunto de datos obtenidos para el modelo en subconjuntos complementarios, uno que permita entrenar el modelo de predicción y otro que valide el análisis del modelo; pero, para reducir el error de variabilidad de la evaluación se realizan

múltiples rondas con diferentes particiones. Así, finalmente se promedian las evaluaciones de cada ronda para determinar la eficacia del modelo [32, 33].

1.2.4 Metodología y plan de trabajo

Para la correcta culminación del presente proyecto de fin de carrera se ha seguido la siguiente estructura de trabajo:

Estudio de la literatura

El primer paso del proyecto es la etapa de formación en la que se obtienen los conocimientos necesarios para su posterior desarrollo. En el caso de este proyecto se ha realizado el estudio de cada término asociado al reconocimiento de expresiones faciales que no se vea comúnmente relacionado al campo de la informática para un entendimiento más profundo del tema, además del estado del arte actual de las investigaciones que se centran en este ámbito, así como de ciertas técnicas que son usadas para tratar de solucionar las dificultades planteadas en la problemática.

Adquisición de la base de datos de expresiones faciales

Previo al inicio del desarrollo del prototipo es necesario considerar sobre qué base de datos se realizará la implementación, esto debido a que al tratarse de un problema de clasificación, la solución propuesta presentará un modelo discriminador entrenado que tendrá como soporte una base de datos de imágenes previamente seleccionada.

Desarrollo del prototipo de reconocimiento

La etapa central del proyecto es la del desarrollo del prototipo, que se espera permita lograr todos los objetivos necesarios para la discriminación de las expresiones faciales. El prototipo incluye las siguientes etapas:

1. **Adquisición**, fase inicial en la que el prototipo obtiene una imagen que servirá para la evaluación de la expresión facial, en caso se detecte un rostro.

2. **Pre-procesado,** fase de preparación en la imagen adquirida, permite que su manipulación sea más sencilla, eficiente y menos propensa a errores. Consta de:
   a) **Transformación a escala de grises,** excluye información poco relevante de la imagen (color).
   b) **Ecualización del histograma,** mejora la intensidad y calidad de una imagen por los efectos negativos del brillo y el poco contraste.

3. **Detección,** fase en la que se inicia la búsqueda del área de interés. Cuenta esencialmente de:
   a) **Detección del rostro,** localiza y recortar el área de interés a analizar.

4. **Extracción,** fase en la que la información del rostro es preparada, sustraída y caracterizada. Consta de:
   a) **Recorte,** secciona el rostro para eliminar las áreas que no aportan información.
   b) **Escalado,** redimensiona la imagen del rostro a un tamaño estandarizado.
   c) **Caracterización**, representa la imagen del rostro en un formato simplificado que permita facilitar la diferenciación entre expresiones.

5. **Clasificación,** fase final en la que la imagen caracterizada es clasificada dentro de un conjunto de posibles clases.

Experimentación para la optimización del prototipo

Con el fin de mejorar los resultados obtenidos luego del desarrollo del prototipo inicial se realizan una serie de experimentos en la extracción del área de interés y entrenamiento del modelo para tratar de obtener mejores resultados en la clasificación de las expresiones faciales.

Evaluación del prototipo y conclusiones obtenidas

Documentación del trabajo realizado

De forma paralela al desarrollo se procede a documentar el trabajo realizado dando la respectiva descripción de los pasos seguidos para que pueda ser entendible y

reproducible, realizando los análisis y evaluaciones pertinentes a los resultados reales y finalmente brindar el análisis de los trabajos futuros respecto a las investigaciones que no pudieron ser cubiertas con el proyecto.

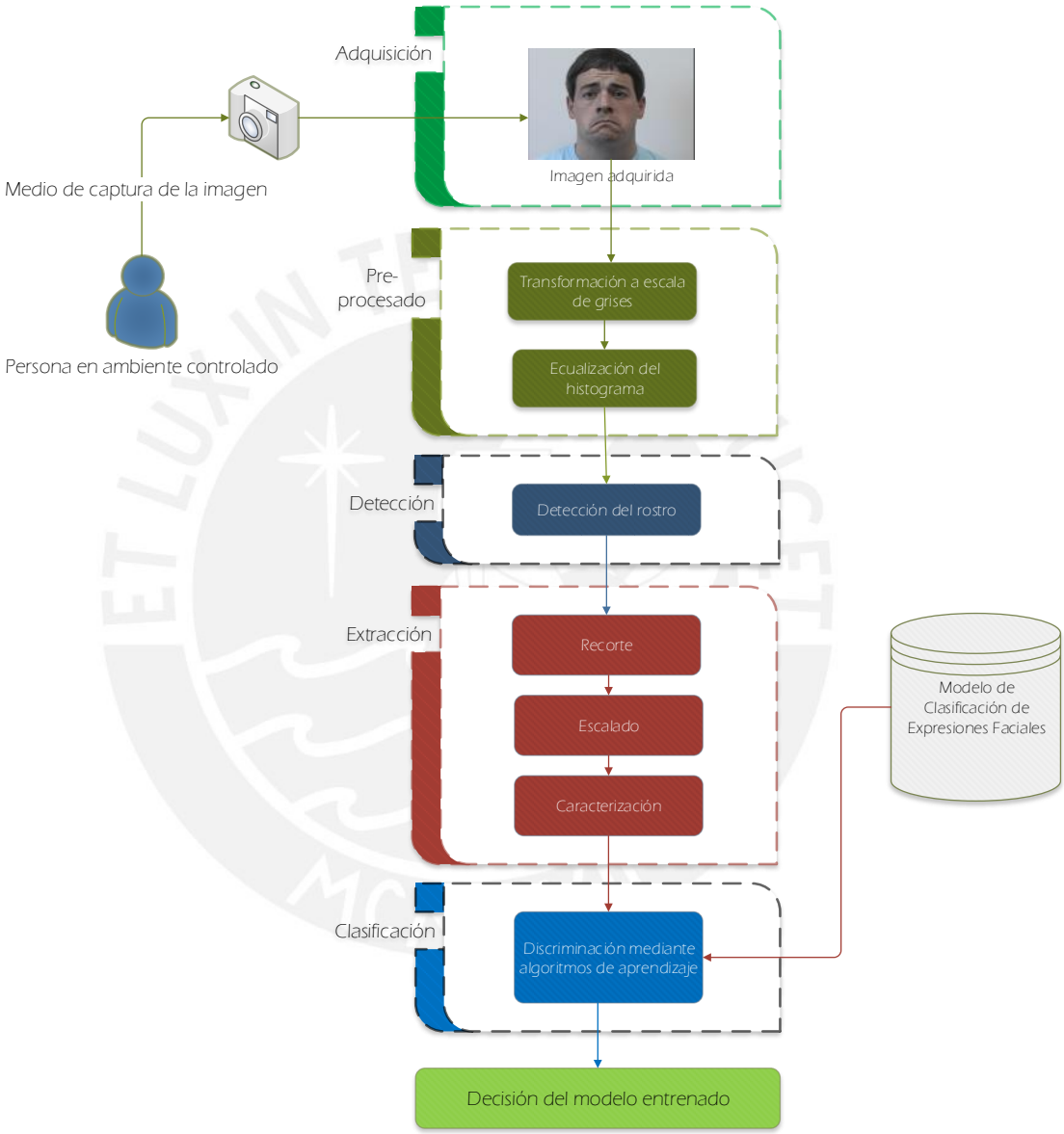

Figura 1.4 *- Diagramas de etapas del prototipo. Imagen de autoría propia.*

1.3 Delimitación

1.3.1 Alcance

El presente proyecto, ubicado dentro del área de *Ciencias de la Computación,* se encuentra orientado a la investigación aplicada e implica el desarrollo de un prototipo software que clasifique, a través del análisis de la expresión facial en una imagen, qué emoción es atribuida a dicha expresión.

El prototipo realizará la clasificación en base a seis expresiones de las emociones determinadas como globales o básicas por los psicólogos Ekman y Friesen: Alegría, tristeza, enojo, asombro, miedo y asco [34]. Se están considerando como relevantes solo estas seis debido, principalmente, a la presencia de un mayor registro de investigaciones, estudios e información sobre la universalidad de estas expresiones, así como bases de datos de imágenes que facilitarán el entrenamiento del modelo computacional que clasificará dichas expresiones.

Debido a la limitación del tiempo para el desarrollo y evaluación del prototipo software (dos semestres académicos), este realizará la estimación solo en imágenes digitales y no en secuencias de video. Inicialmente la idea se denotó mediante la clasificación de las expresiones en las secuencias de video; sin embargo, debido a la curva de aprendizaje necesaria para la compresión de los métodos y el buen uso de las herramientas para el desarrollo, además de la complejidad adicional que supondría el seguimiento del rostro en las secuencias de video se delimitó el objetivo del proyecto a imágenes digitales, lo que permite lograr un mayor enfoque en las etapas de extracción de características y clasificación de la expresión.

Así mismo, se hace mención que se hará uso de imágenes en ambientes controlados. Lo que permitirá evitar los problemas que conllevan las imágenes obtenidas en ambientes no controlados, como son la visibilidad del rostro en la imagen, la excesiva o mínima iluminación de la misma o el ángulo facial que puede evitar su detección; esto con el fin de no centrar la implementación en la detección precisa del rostro o pre-procesado de la imagen. El ambiente controlado brindará de facilidades para la mejor obtención de las características del rostro (expresiones faciales) y por ende un análisis más efectivo para la clasificación de la expresión facial, que es el fin de este proyecto.

1.3.2 Obstáculos: riesgos y problemas externos

A continuación se describen los riesgos y problemas identificados en el proyecto que podrían afectar la continuidad del mismo, además de la valoración del impacto y medidas correctivas que logren mitigar cada uno de estos.

Riesgo identificado	Impacto	Medidas correctivas para mitigar
Fallo inesperado del computador en el que se realiza la investigación y avance del proyecto.	Alto	Usar de forma permanente backups de respaldo para la investigación (nube / disco externo) y fijar el tiempo de actualización de estos.
Probabilidad de escases en bases de datos de imágenes de expresiones faciales específicas que compliquen el entrenamiento del clasificador.	Medio	De darse el caso en el que la expresión de alguna de las 6 emociones determinadas no se encuentre muy difundida en las bases de datos se procederá a realizar la captura de imágenes de autoría propia para el entrenamiento.
Dificultad de acceso a las implementaciones de algoritmos de aprendizaje para la última fase del prototipo, la clasificación.	Medio	Uso de algoritmos disponibles que brinden la mejor precisión posible, o re-implementación de algoritmos que se encuentren en un lenguaje diferente al usado, en caso no encontrar alguno disponible.

Tabla 1-2 *- Riesgos y medidas correctivas*

1.4 Justificación y viabilidad

1.4.1 Justificación

En este proyecto de fin de carrera se pretende implementar un prototipo que, a través de conceptos de visión computacional y aprendizaje de máquina, permita clasificar expresiones faciales en ciertos estados emocionales determinados como globales. Si bien a lo largo de la carrera los temas de visión computacional no fueron tocados, la investigación y el auto-aprendizaje fomentado y adquirido a lo largo de la misma son puntos clave para poder desempeñar este tipo de proyecto.

El área hacia el cual va dirigido el proyecto, es un área en creciente desarrollo por lo que cada investigación brinda información clave a tomar en cuenta para futuras investigaciones en el mismo ámbito. Y que en el caso del proyecto, permitirá brindar más información respecto al análisis de las expresiones a través de un enfoque holístico sobre el rostro, específicamente a través de la técnica de patrones binarios locales que de por sí ya brinda buenos resultados para el reconocimiento facial, así como la especificación de algoritmos de aprendizaje de máquina que favorecen la clasificación y análisis en este tipo de representación.

Finalmente, cabe destacar que la automatización del análisis de las expresiones faciales puede ser aplicado a múltiples y distintas áreas con el fin de facilitar o mejorar su trabajo [10, 14, 35]. Áreas como son la interacción humano computador (HCI), al permitir una comunicación más fluida entre un usuario y el computador a través de los gestos faciales; en psicología y psiquiatría, al permitir realizar diagnósticos más precisos cuando se analiza a un paciente; en la vigilancia, al permitir detectar más fácilmente comportamientos sospechosos en las personas; en investigaciones policiales, al aumentar la probabilidad de detección de mentiras en los individuos intervenidos al ser adicionado a los otros métodos de detección; o incluso en la robótica, al permitir una comprensión de los estados emocionales de las personas y así mejorar la interacción con las mismas.

1.4.2 Viabilidad

Esta sección tiene como propósito demostrar que el proyecto de fin de carrera planteado es viable en términos de ejecución, económicos, de tiempo y de acceso a la información necesaria para su realización y culminación plena.

En primer lugar, haciendo referencia a la viabilidad técnica, se puede mencionar que los métodos elegidos provisionalmente cuentan con el soporte de diversas investigaciones en el campo que validan su uso en el proyecto [19, 26-28, 36]. Se aclara además que a pesar de la falta de conocimiento pleno de dichos métodos, algoritmo y herramienta (OpenCV) por parte del autor del proyecto, se cuenta con una alta predisposición para aprender sobre estos, ya que el tema es de gran interés en el autor.

El proyecto es viable en términos económicos, debido a que no se tiene la necesidad de invertir por la adquisición de software necesario para su implementación. Las herramientas a utilizar se pueden adquirir de forma gratuita y legal. En el caso del OpenCV, por ser de software libre [20, 30] y en el caso del Visual Studio, por haberse adquirido de forma gratuita gracias al convenio que tiene la universidad con Microsoft DreamSpark.

Respecto a que la viabilidad temporal del proyecto (dos semestres académicos), se confirma que esta es viable debido al análisis previo realizado sobre el tiempo necesario para el aprendizaje, desarrollo y pruebas del prototipo, lo que permitió delimitar el proyecto hacia el enfoque en las etapas críticas de este, reduciendo el tiempo que hubiera sido necesario para enfocarse en etapas que no están directamente relacionadas al fin del proyecto. Ver **Anexo 1** - Diagrama de Gant.

Finalmente, respecto a la viabilidad del acceso a la información, necesaria para la investigación, se afirma que el acceso a ella es viable. Esto se da debido al abundante material bibliográfico sobre los métodos y algoritmos, necesarios a estudiar para la realización de cada una de las etapas del proceso de estimación, encontrados de forma virtual, así como diversos artículos de investigación (papers/ surveys) que sirven de orientación para el desarrollo del modelo.

CAPITULO 2

2 Marco conceptual

2.1 Conceptualización

2.1.1 Interacción humano-computador (IHC)

La evolución del computador, desde sus inicios, como herramientas de operaciones de cálculo hasta las que se conocen hoy en día para los procesamientos gigantescos de datos y obtención de información, muestra el gran interés del ser humano en la mejora constante de esta herramienta con el fin de facilitar su trabajo [37]. La gran difusión presente desde los años iniciales de su distribución hacia la población no científica ha permitido generar un nuevo campo de estudio en la informática denominado Interacción humano-computador [11], cuyo propósito principal es el de facilitar el uso de los computadores a la cada vez más creciente población.

Para la compresión plena de la IHC se debe conocer el significado de "Interacción", que según Alan Dix, "Es el proceso de transferencia de información" [38]. Con este concepto se puede entender que este campo de la informática trata de mejorar el proceso de transferencia de información entre el usuario y el computador; información que será procesada e interpretada por este último y que emitirá una respuesta adecuada mediante la realización de una determinada tarea o acción.

La información enviada por el usuario es obtenida por diversos dispositivos del computador, lo que revela la existencia de diferentes tipos de datos que pueden ser interpretados por el mismo [38].

Inicialmente, la información que una persona quisiera transferir a un computador debía de ingresarse de una forma completamente física (uso de tarjetas perforadas); sin embargo, los avances producidos desde su aparición permitieron grandes cambios en el modo de interacción. Siendo algunos de los iniciales:

- La aparición de la línea de comandos.
- La introducción de interfaces gráficas para usuario.

Es por ello que a través de los años se fueron introduciendo distintos dispositivos para interpretar las instrucciones de los usuarios y a la par para facilitarles su uso. Actualmente en este campo, los temas relacionados al reconocimiento e interpretación de voz, de ojos y gestos están siendo investigados y desarrollados para el futuro de la interacción humano-computador. En la **Tabla 2.1** se muestra una serie de dispositivos introducidos para el proceso de transferencia de información.

Dispositivo de Entrada	Tipo de Información
Teclado	Entrada de texto/Ejecución de comandos
Pantalla táctil	Desplazamiento e interacción en interfaz/ Entrada de texto
Mouse	Desplazamiento e interacción en interfaz
Touchpad	Desplazamiento e interacción en interfaz
Trackball	Desplazamiento en interfaz
Thumbwheel	Desplazamiento en interfaz
Keyboard nipple	Desplazamiento e interacción en interfaz
Joystick	Desplazamiento e interacción en interfaz
Cámara	Desplazamiento e interacción en interfaz
Micrófono	Entrada de texto/Ejecución de comandos

Tabla 2-1 *- Algunos dispositivos de interacción e información que pueden brindar [38].*

2.1.2 Emoción

Dentro del marco de la comunicación, las personas trasmiten unas a otras cierta información de su estado de ánimo, tanto de forma voluntaria como involuntaria a través de las emociones. Pero ¿qué se entiende por emociones y de dónde provienen?

Las emociones están usualmente ligadas a los eventos externos, los cuales son percibidos como estímulos [8]. Por ello, las personas suelen mostrar diversas

emociones al percibir estímulos distintos, estímulos, como son ciertos objetos, lugares, situaciones, recuerdos o ciertas personas, que podrían representar un hito importante en su vida. De igual forma, las reacciones o actitudes que la persona tome frente a dicho estímulo están ligadas a la emoción que se experimente [39]; así por ejemplo, el sentirse alegre y sonreír al revivir un buen recuerdo o entristecerse y llorar al conocer sobre el fallecimiento de un familiar.

Asimismo, estos eventos o estímulos experimentados por las personas suelen generar diversas reacciones en sus cuerpos [8], como son los cambios corporales, entre ellas la sudoración, tensión, movimientos faciales, movimientos involuntarios; variaciones en la entonación de la voz; aceleración del ritmo cardiaco, etc., las cuales han sido motivo de estudio para la comprensión de la emoción.

Componente del episodio de la emoción	Teoría
Experiencia consciente	Teoría de los sentimientos
Cambios en el cuerpo y rostro	Teoría somática
Tendencias de acción	Teoría del comportamiento
Modulación de los procesos cognitivos	Teoría del modo de procesamiento
Pensamientos	Teoría de la cognición pura

Tabla 2-2 *- Teorías que identifican a las emociones con distintos componentes en episodios de la emoción. Tabla adaptada de [8]*

A través de la historia diversos psicólogos y científicos [8, 9] han propuesto teorías o modelos con el propósito de definir y comprender todos los aspectos que conllevan a la experimentación de emociones. En la **Tabla 2.2**, se puede observar la clasificación de las teorías más relevantes, según Jesse Prinz, sobre la emoción y los componentes que le fueron atribuidos para su compresión.

A través de la historia diversos psicólogos y científicos [8, 9] han propuesto teorías o modelos con el propósito de definir y comprender todos los aspectos que conllevan a la experimentación de emociones. En la tabla 2.1, se puede observar la

clasificación de las teorías más relevantes, según Jesse Prinz, sobre la emoción y los componentes que le fueron atribuidos para su compresión.
Cada teoría plantea distintas formas de ver y comprender las emociones mediante lo que se denominan componentes, sin embargo sin llegar a una concepción general y clara sobre las emociones. Según Prinz dichos cambios corporales (en los sistemas nerviosos somático y autónomo), actitudes, acciones, pensamientos y sentimientos son causa y efecto de las emociones [8].

En este sentido, se puede concluir que las emociones se manifiestan como un conjunto de reacciones complejas en el cuerpo tanto fisiológicas como psicológicas que determinan una conducta o postura ante los eventos externos como respuesta.

2.1.2.1 Expresión facial

Como se mencionó anteriormente, las emociones se manifiestan de diversas formas en las personas. Una de estas manifestaciones alude a cambios que se presentan en el cuerpo. El rostro, como parte del cuerpo, presenta cambios al experimentarse una emoción. Estos cambios en el rostro son causados por los distintos músculos alojados en él, como respuestas expresas que el sistema nervioso somático envía al experimentarse la emoción [8]. Estos cambios en el rostro son denominados expresiones faciales.

Las expresiones faciales pueden brindar información relevante sobre la comunicación entre las personas. Ya que el rostro es uno de los medios del lenguaje corporal, que permite una comunicación a nivel no verbal [5, 6]. Sin embargo, desde sus inicios en las investigaciones y estudios no hubo respuestas claras a ciertas preguntas [7], las cuales resultaron motivos importantes de estudio.
Dichas preguntas según diversos autores que analizaron el tema fueron [7]:

- ¿Es exacta?
- ¿Es universal?
- ¿Es innata?

El primer motivo de estudio hace referencia a qué tan exacta es la información que se puede obtener de una expresión facial. Si bien una expresión facial muestra el estado emocional de una persona, está podría estar condicionada por la cultura o

podría ser forzada en el intento de expresar una emoción que no es la genuina y así transmitir una información falsa.

En este sentido, la siguiente pregunta se relaciona con la anterior, ya que, si una expresión puede ser condicionada por la cultura o sociedad habría que preguntarse si es posible que existan expresiones indistintas de aspectos culturales y ambientales; es decir, expresiones universales. Si una persona de occidente expresa la emoción de la felicidad realizando cierta deformación en su rostro, entonces, si es universal, una persona del oriente expresará dicha emoción con la misma deformidad en su expresión.

El último motivo, pone en duda el hecho de que una persona, desde el nacimiento, pueda expresar sus emociones mediante expresiones faciales, sin la necesidad de que estas sean aprendidas durante su crecimiento y por ello condicionadas por la cultura.

Charles Darwin sugirió en [39], basado en la teoría de la evolución, que las expresiones son innatas y por lo tanto parte de nuestra biología. La afirmación tenía fuertes bases teóricas, sin embargo, no tenía pruebas que lo corroboraran. Aproximadamente en el año 1968, Paul Ekman y Wallace Friesen conducen sus estudios e investigaciones hacia la confirmación de lo innato y universal de las expresiones [34].

	Grupo Participante				
Expresión	**Estados Unidos**	**Chile**	**Brasil**	**Argentina**	**Japón**
Felicidad	97	90	92	94	87
Miedo	88	78	77	68	71
Desagrado	84	85	86	79	82
Enojo	68	76	82	72	63
Sorpresa	91	88	81	93	87
Tristeza	87	91	82	88	80
Promedio	86	85	83	82	78
Nota. Todos los valores listados son el porcentaje de participantes quienes correctamente juzgaron la expresión emocional indicada.					

Tabla 2-3 *- Precisión en el reconocimiento de expresiones faciales de población americana (Estudio de cinco culturas por Ekman 1972). Tabla adaptada de [40]*

En la **Tabla 2.3**, se puede observar los resultados obtenidos al realizar su investigación (mostrar fotografías, con ciertas expresiones faciales, a estudiantes universitarios de distintos países para que pudieran determinar la emoción que mostraba la imagen).

Los resultados a pesar de confirmar parcialmente, con buenos resultados, la universalidad de las expresiones, no fueron clave para reforzar completamente su validez [40]. Debido, principalmente, a que aquellas personas podían tener conocimientos de la cultura americana y ver su percepción influenciada. Por otro lado, las culturas aborígenes que tienen poco o ningún contacto con la cultura occidental podrían brindar una información más confiable y verídica dada su percepción natural y no influenciada. La investigación tomó ese rumbo y se llevó acabo con dos culturas alejadas de la sociedad, concluyéndose efectivamente que las expresiones de las emociones básicas son universales [34] y que por tanto parte de nuestra biología (innatas).

Respecto a la exactitud de la información que brindan las expresiones faciales sigue siendo tema de investigación, debido a factores que, en las investigaciones, no se han podido controlar [7]. Según mencionan Ekman y Oster en [7], las correctas interpretaciones sobre la experiencia emocional de una persona no tendrían que atribuirse el cien por ciento de las veces a las expresiones faciales, ya que también podrían interpretarse por otros índices como son los movimientos corporales bruscos, la postura u otros.

2.1.2.1.1 Sistema de codificación de acción facial (FACS) y las unidades de acción (UA)

Recapitulando, las expresiones son resultado de los movimientos musculares en el rostro, sin embargo antes de la publicación del trabajo realizado por Ekman y Friesen en 1978 [41] la compresión del comportamiento facial se basaba solo en la observación, como el trabajo de Darwin en [39], lo que no constituyó una base sólida para el reconocimiento de las expresiones.

Luego fue desarrollado el sistema de codificación de la acción facial (FACS, por sus siglas en inglés) con el objetivo principal de crear un sistema parametrizado para la compresión de cada posible movimiento facial observable, y todo esto basado en los músculos que los producen [41].

El sistema identifica los músculos en el rostro, que de forma individual o grupal, causan cambios en el comportamiento facial, cada uno de estos cambios es denominado unidad de acción (AU, por sus siglas en inglés) [13]. Una expresión en el rostro podría ser generada entonces por una unidad o por varias de ellas combinadas; en el caso de las combinaciones, las unidades de acción pueden ser aditivas cuando no se pierde la esencia del AU o no aditivas cuando la presencia de otros músculos altera el AU [13, 41, 42].

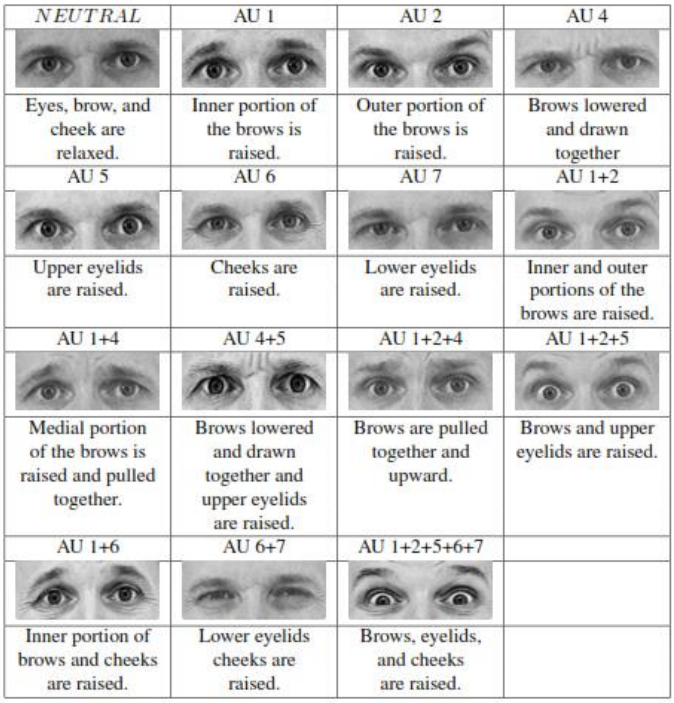

Figura 2.1 *- Unidades de acción de la parte superior de la cara y algunas combinaciones. Imagen extraída de [42].*

2.1.3 Visión por computador

Para los seres humanos la obtención de información respecto a lo que ven está dada de forma natural por los sistemas complejos que estos poseen; por ejemplo, una persona común y corriente, sin dificultades visuales, que desee coger un vaso de agua, en primer lugar observará donde se encuentra el vaso, luego se aproximará a este y finalmente estirará su brazo y lo alcanzará. Este proceso que sucede de forma natural para el ser humano es un conjunto de interpretaciones y acciones tomadas por el cerebro respecto a lo que este ve, como son la identificación del objeto y el cálculo de la distancia entre el observador y el objeto.

La visión por computador es un campo de la informática que intenta emular este comportamiento, percepción visual [12].

Según Hanson y Riseman en [43], " La meta de las investigaciones en la visión por computador es la comprensión de los complejos procesos visuales y de la construcción de un efectivo sistema visual basado en la computadora". Esto quiere decir que su principal objetivo es la extracción de información de los objetos que pueda identificar en una imagen o secuencia de imágenes obtenidas de su entorno. En la **Tabla 2.4** se muestran ejemplos de algunas aplicaciones basadas en la visión por computador.

Dominio	Objetos	Modalidad	Tareas	Fuentes de conocimiento
Robótica	Tridimensional ▪ Escenas en exteriores ▪ Escenas en interiores Partes mecánicas	Luz Rayos-X Luz Luz estructurada	Identificar o describir objetos en una escena Tareas industriales	Modelos de objetos Modelos de la reflexión de la luz en objetos
Imágenes aéreas	Tierra Edificios, etc.	Luz Infrarrojo Radar	Mejora de imágenes Análisis de recursos Predicción del clima Espionaje Direccionamiento de misiles Análisis táctico	Mapas Modelos geométricos de formas Modelos de la formación de la imagen
Astronomía	Estrellas Planetas	Luz	Composición química Mejora de imágenes	Modelos geométricos de formas

Médico	Órganos del cuerpo	Rayos-X	Diagnóstico de anormalidades	Modelos anatómicos
Macro		Ultrasonido Isotopos Calor	Planeación del tratamiento y operación	Modelos de la formación de la imagen
Micro	Células Cadena de proteínas Cromosomas	Microscopía electrónica Luz	Patología, citología, cario tipificación.	Modelos de forma
Químico	Moléculas	Densidades electrónicas	Análisis de la composición molecular	Modelos químicos Modelos estructurados
Neuroanatomía	Neuronas	Luz Microscopía electrónica	Determinación de la orientación espacial	Conectividad neuronal
Física	Rastros de partículas	Luz	Encontrar nuevas partículas Identificar rastros	Física atómica

Tabla 2-4 *- Aplicaciones clásicas y actuales. Tabla adaptada de [12].*

Se puede decir entonces, de forma general, que las investigaciones y avances en este campo tratan de brindar capacidad de análisis visual a los computadores para la realización de determinadas tareas. Pero, ¿cuál sería el proceso que sigue el computador para el análisis de las imágenes?

Según J. Velez el computador debe seguir una serie de cuatro fases principales para dicho análisis [44]:

- **Captura**

  En esta fase el computador obtiene la imagen o imágenes digitales de su entorno mediante algún sensor.

- **Pre-procesamiento**

En esta segunda fase el computador trata digitalmente la imagen; es decir, toma la imagen y reproduce una versión modificada con el fin de mejorarla para facilitar la extracción de información posterior; es decir, trata de disminuir todos los defectos que la imagen pueda tener que impidan una correcta obtención de datos.

- **Segmentación**

En esta tercera etapa, se trata de aislar los elementos de la imagen por las características que estos posean en la misma. Características como color, brillo o densidad, las cuales permitirán tener una clasificación de cada pixel y obtener una imagen dividida por regiones comunes. Esta etapa facilita el análisis y reconocimiento de los objetos en la imagen.

- **Reconocimiento o Clasificación**

En esta última fase se trata de identificar a los objetos segmentados a través del análisis de las características extraídas previamente. Es decir identificarlos o clasificarlos dentro de un universo predefinido.

2.1.4 Conclusión

En conclusión, dado que el proyecto de fin de carrera tiene como objetivo estimar emociones, en primer lugar, es necesario entender a qué llamamos emociones para saber qué información estimar. La información será obtenida mediante imágenes, por lo cual es necesario conocer qué analizar de la imagen, es por ello que se detalla el concepto de expresión facial y su relación con las emociones para así conocer cómo obtener dicha información del rostro. Finalmente, se habla de las áreas de la informática relacionadas con el concepto del análisis automático de las expresiones observándose disciplinas como interacción humano-computador y visión por computador; todos estos conceptos permitirán al lector comprender el panorama de la problemática y el contexto en general del documento.

2.2 Estado del arte

2.2.1 Método usado en la revisión del estado del arte

Para la revisión del estado del arte fue utilizada la revisión sistemática y para ello se realizaron las búsquedas a través de las bases de datos siguientes:

- IEEE Explore
- Scholar Google

2.2.1.1 Formulación de la pregunta

La pregunta de investigación formulada fue: ¿Qué temas han tratado las investigaciones relacionadas con el reconocimiento automático de las expresiones surgidas en los últimos años? Los términos usados para resolver esta pregunta fueron: “face”, “facial”, “emotion”, “expression”, “recognition”, “estimation”, ”classification” “detection”, “visual”, “images” y “image”.

2.2.1.2 Selección de las fuentes

A partir de combinación de los términos listados anteriormente y haciendo uso de conectores lógicos “AND” y “OR” se obtuvieron las siguientes sentencias:

	Cadenas generales básicas de búsqueda
1	“facial expression recognition” AND (images OR image)
2	“face estimation” AND images
3	“emotion recognition” AND visual
4	“facial detection”
5	“facial expression” AND classification

Tabla 2-5 *- Cadenas generales básicas de búsqueda.*

Uno de los criterios de exclusión usados fue respecto la fecha de publicación. Se determinó que la obtención de información se daría desde el año 2008 hasta la actualidad. Además de realizar la búsqueda de las cadenas clave solo en los títulos de los artículos. De igual forma se limitó la búsqueda en documentos de revisión publicados en conferencias o simposios IEEE, excluyendo publicaciones de otras

fuentes como revistas o journals. Estos criterios de exclusión fueron usados en todas las búsquedas realizadas. Cabe resaltar que de cada una de las búsquedas especificadas a continuación, por cada cadena formulada, se extrajo un documento para su revisión.

Primero, al realizar la búsqueda con la cadena 1 en la base de datos IEEE se encontraron 32 resultados relacionados y al realizar la búsqueda en Scholar Google se encontraron 62 resultados.

Luego, la búsqueda con la cadena 2 en la base de datos IEEE mostró 14 resultados, mientras que al realizar la misma búsqueda en Scholar Google se encontraron 35 resultados.

Mediante la búsqueda con la cadena 3 en la base de datos IEEE se encontraron 11 resultados y mediante Scholar Google, la búsqueda mostró 67 resultados.

Posteriormente, la búsqueda con la cadena 4 en la base de datos IEEE mostró 147 resultados y en Scholar Google se mostraron 584 resultados.

Finalmente, la búsqueda con la cadena 5 en la base de datos IEEE mostró 8 resultados y en Scholar Google se mostraron 29 resultados.

2.2.2 Investigaciones en el tema

En esta sección se hará mención de las investigaciones que se han dado en los últimos años respecto al tema del proyecto de fin de carrera.

2.2.2.1 Reconocimiento de la emoción audio-visual usando modelos de mezcla gaussianos para el rostro y la voz.

Esta investigación centra su trabajo en la hipótesis que el uso combinado de las modalidades facial y vocal de estimación emocional, logran una mayor precisión respecto al uso individual. Inicialmente, cada tipo de entrada, visual y sonora, se modela mediante el modelo de mezcla gaussiano. Luego para la combinación de las modalidades se usaron técnicas de combinación de clasificación Bayesiana por esquema de ponderación y de clasificación de vectores de soporte (SVC por sus siglas en inglés) que usa una precisa post clasificación. Los resultados proveyeron

de una mayor precisión y rendimiento a la combinación de los clasificadores de las modalidades, ya que cuenta con mayor información para la clasificación en los casos en los que una de las modalidades no representa tanto una emoción o esta es confusa [15].

2.2.2.2 Reconocimiento de la expresión facial en secuencia de imágenes basado en puntos característicos y correlaciones canónicas

Esta investigación se basa en el reconocimiento de la expresión facial mediante el análisis del desplazamiento de ciertos puntos faciales característicos del rostro a través de una secuencia de imágenes desde un estado neutral hasta uno pico en la expresión. Para realizar el seguimiento o rastreo de estos puntos se usa el flujo óptico en la secuencia de imágenes. Luego, se extraen tanto el desplazamiento normalizado de los puntos, respecto a su estado neutral, como ciertas distancias geométricas estandarizadas de una imagen facial alineada para formar un vector propio; debido a la secuencia de imágenes se forman varios vectores y se ordenan en
una matriz denominada *FEAD* (Facial-Expression-Arising-Dataset), la cual representa la expresión. Finalmente, para clasificar dicho FEAD se utilizan correlaciones canónicas para comparar su similitud con FEADs ya establecidos, que representan las seis expresiones básicas, y determinan la expresión; además de usar una función discriminante lineal para mejorar el proceso de aprendizaje y reconocimiento [16].

2.2.2.3 Detección de puntos faciales usando regresión potenciada y modelos gráficos

Este trabajo centra su objetivo en la mejora de la detección de puntos claves en el rostro, importante para el análisis de expresiones basado en características geométricas. Su estudio presenta un método basado en la combinación entre vectores de soporte de regresión y los cambios aleatorios de Markov. Las regresiones aprenden un mapeo entre la apariencia de la zona que rodea un punto y la posición del punto; mientras que los cambios aleatorios de Markov reducen el área de búsqueda de los puntos debido al aprovechamiento de las constelaciones que se pueden formar con los puntos faciales. Todo esto con el fin de reducir de forma drástica el tiempo de localización de cada punto e incrementar la precisión de detección y la robustez del algoritmo [17].

2.2.2.4 Estimación de la pose del rostro en tiempo real de imágenes de rango individuales

Este trabajo presenta un algoritmo para estimar la pose en 3D de un rostro desde una imagen de rango individual (imágenes con profundidad por píxel). Basado en la identificación de la nariz en la imagen de rango, se generan candidatos para su posición y luego se generan y evalúan hipotéticas poses en paralelo. Luego mediante una función se compara la imagen adquirida (imagen de rango) con las distintas poses pre computadas y finalmente se calcula la pose del rostro estimando su rotación y ángulo formado [18].

2.2.2.5 Sistema de clasificación de expresión facial basado en el modelo de forma activa y máquinas de vectores de soporte

Esta investigación basó su desarrollo en el uso de componentes faciales (ojos, cejas, nariz y boca) para la localización de texturas dinámicas del rostro, como son las líneas de expresión, arrugas de la nariz, patrones y pliegues naso labiales, y lograr mediante estas la clasificación de las expresiones faciales. En la fase inicial se hizo uso del Modelo de Forma Activa (ASM por sus siglas en inglés) junto con el algoritmo de aprendizaje Adaboost utilizando características Haar-like para detectar de forma precisa la cara y adquirir regiones importantes de características faciales. Para la siguiente fase se hizo uso de Filtro de Gabor y Laplaciano de Gauss para extraer la información de la textura de las regiones adquiridas. La información extraída se almacena en vectores de características de textura que representan los cambios de la textura del rostro de una expresión a otra y mediante el uso Maquinas de Vectores de Soporte (SVM por sus siglas en inglés) se clasifica dicho conjunto de vectores entre las seis expresiones básicas de las emociones [19].

2.2.3 Productos desarrollados

Esta sección dará a conocer algunos productos existentes en el mercado, su relación respecto al tema y las cualidades que otorgan.

2.2.3.1 Cámara digital Cyber-Shot

La cámara digital Cyber-shot DSC T200 fue el primer lanzamiento que hizo Sony en el que incorporó software de visión por computador. Este producto permitió la detección de rostros así como la detección de sonrisas; de este modo se capturaba de forma automática una foto en el instante en el que se detectasen sonrisas. A partir de esa serie se incorporó este software, al cual se le denominó Smile Shutter, en las series de cámaras digitales Cyber-Shot [45].

Figura 2.2*- Detección de sonrisa en cámara Cyber-Shot. Imagen extraída de [25].*

2.2.3.2 FaceReader

FaceReader es una herramienta software desarrollada por *Noldus Information Technology* que provee de un sistema robusto para el reconocimiento de la expresión facial basado en las emociones básicas propuestas por Ekman y Friesen (felicidad, tristeza, sorpresa, desagrado, enojo y miedo) además de un estado neutral en el rostro. Este permite el análisis de video en tiempo real a través de webcam, así como el análisis de imágenes [46].

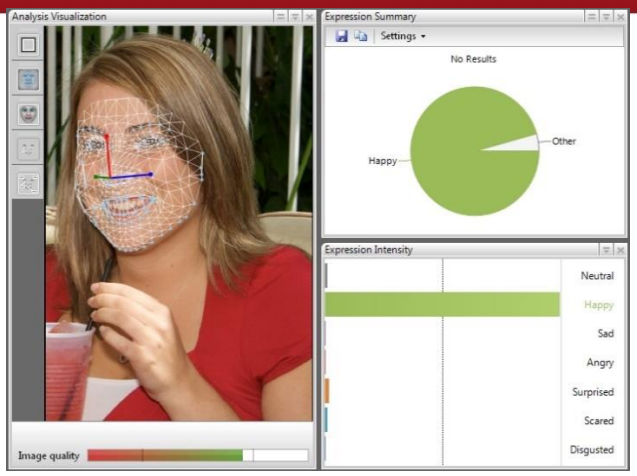

Figura 2.3- *Análisis de imagen en FaceReader. Imagen extraída de [46].*

2.2.3.3 Emotient API

Emotient API es un software desarrollado por la compañía Emotient para el análisis y el reconocimiento de la expresión facial. Este da la posibilidad de analizar las respuestas emocionales de los usuarios en tiempo real, al detectar y rastrear expresiones de emociones básicas, como la alegría, sorpresa, ira, asco, tristeza, desprecio y miedo; emociones avanzadas, como la frustración y confusión; sentimientos generales, como los positivos, negativos y neutros; la combinación de dos o más emociones y diecinueve unidades de acción faciales [47].

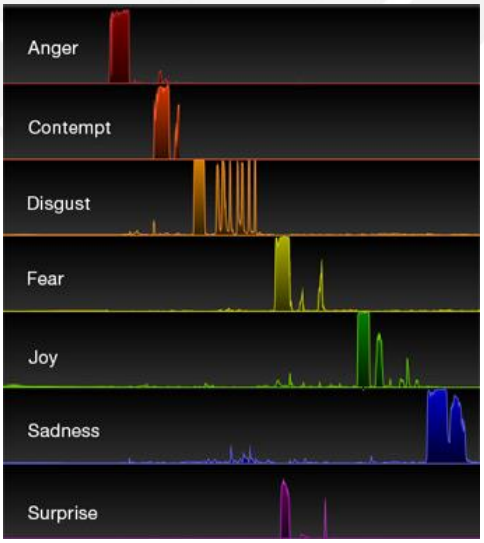

Figura 2.4*- Cuadro de análisis de 7 emociones en tiempo real. Imagen extraída de [47].*

2.2.4 Resumen del estado del arte

2.2.4.1 Investigaciones en el tema

Investigación	Métodos de reconocimiento	Problema que intenta solucionar	Base de Clasificación	Bases de datos usadas	Tasa de Reconocimiento promedio
Reconocimiento de la emoción audio-visual usando modelos de mezcla gaussianos para el rostro y la voz.	Modelos de mezcla Gaussiano + Clasificador de Vectores de Soporte (SVC) + Clasificación Bayesiana por esquema de ponderación.	Emociones multimodales	Felicidad, enojo, tristeza y neutralidad del rostro.	Interactive Emotional Dyadic Motion Capture database (IEMOCAP)	75.50%
Reconocimiento de la expresión facial en secuencia de imágenes basado en puntos característicos y correlaciones canónicas	Flujo Óptico + Clasificador por correlaciones canónicas.	Extracción de características del rostro	6 emociones básicas expresadas en el rostro.	Cohn-Kanade Facial Expression Database	90.70%
Detección de puntos faciales usando regresión potenciada y modelos gráficos	Regresión de Vectores de Soporte (SVR) + Cambios aleatorios de Markov.	Extracción de características del rostro	-	FERET and MMI-Facial Expression databases	-

Estimación de la pose del rostro en tiempo real de imágenes de rango individuales	Clasificador de pose a través de imágenes de rango.	Detección del rostro	-	Base de datos propia	-
Sistema de clasificación de expresión facial basado en el modelo de forma activa y máquinas de vectores de soporte	Modelo de Forma Activa (ASM) + Maquinas de Vectores de Soporte (SVM).	Clasificación de las expresiones en estados emocionales	6 emociones básicas expresadas en el rostro.	Cohn-Kanade Facial Expression Database + Base de datos propia	91.70%

Tabla 2-6 *- Resumen de las investigaciones seleccionadas.*

2.2.4.2 Productos desarrollados

Producto	Presentación	Características	Basado en
Cámara digital Cyber-Shot	Hardware y Software	Detección automática del rostro. Reconocimiento solo en video.	Expresión de alegría-felicidad
FaceReader	Software	Detección automática del rostro. Reconocimiento y clasificación en imágenes y video.	6 emociones básicas y el estado de neutralidad.
Emotient API	Software	Detección automática del rostro. Reconocimiento y clasificación en imágenes y video. Reconocimiento de múltiples rostros.	19 unidades de acción faciales. 7 emociones básicas. 2 emociones avanzadas. Sentimientos positivos, negativos y neutros.

Tabla 2-7 *- Resumen de los productos desarrollados.*

2.2.5 Conclusiones sobre el estado del arte

Luego de la revisión dada a los documentos seleccionados y según la pregunta de revisión planteada inicialmente se puede concluir que, las investigaciones realizadas en el campo, en estos últimos años, centran su atención en el desarrollo, adaptación o combinación de distintos métodos algorítmicos para atacar las diversas dificultades encontradas en la estimación de emociones en imágenes digitales. Mediante los documentos mostrados se pueden conocer los distintos ángulos por los cuales se está tratando de atacar este problema, ya sea a través de la mejora de la detección del rostro y estimación de la pose, de la extracción de información del mismo, o de, finalmente, su clasificación en los estados emocionales definidos; todo esto para llegar a una solución óptima, eficiente y precisa.

Por otro lado, también se observan los productos lanzados al mercado, que aunque reducidos, brindan directa o indirectamente soluciones parciales al problema de la estimación de la emoción. Indirectas como la cámara Cyber-Shot, cuyo fin no tiene relación directa con la estimación de emociones pero si con la detección de sonrisas, la cual alude a la emoción de la felicidad; o directas como los software “FaceReader” y “Emotient API” que permiten reconocer una cantidad específica de emociones basadas en las investigaciones hechas respecto a la relación expresión facial-emoción y que continúan en procesos de mejora constante.

CAPITULO 3

3 Modelo para la detección de la región de interés

3.1 Introducción

El objetivo principal de este primer resultado esperado es permitirle al prototipo computacional la adquisición, preparación y detección de cierta región en la imagen sobre la cual basará su análisis. Es por ello que se implementó una fase en el modelo que permita la detección del rostro, en una imagen adquirida, y así se obtenga la fuente principal de información que el prototipo requiere para la estimación, ya que, la base de su discriminación es el rostro y las características presentes en él. El presente capítulo detallará el proceso empleado para la detección del rostro, así como los procesos realizados de forma previa para su correcta determinación, además de la estructura de datos en la que se representará la imagen.

3.2 Descripción

La etapa de detección es la fase primordial para la obtención de la información como se mencionaba previamente; sin embargo, para realizar la detección del rostro en el prototipo se requieren de ciertos procesos previos. Todo este conjunto de procesos estará inmerso en el módulo de procesamiento que se verá completado en el capítulo siguiente.

Figura 3.1 *- Secuencia de procesos para la detección. Imagen de autoría propia.*

3.2.1 Adquisición

La **"fase de adquisición"** es una de las fases iniciales que el prototipo requiere para la obtención de la fuente de información, "las imágenes".

A través de la librería OpenCV se logra mapear la imagen dada en una clase denominada "Mat". Esta clase es una matriz o arreglo bidimensional que contiene la

información en formato numérico de todas las intensidades de los píxeles de la imagen y que, mediante ciertos atributos de la misma, facilita su recorrido y manipulación.

4	5	63	9	1	6
10	12	57	5	2	1
3	6	35	8	0	6
3	5	22	7	0	7
3	3	54	7	2	5
3	3	40	13	0	2

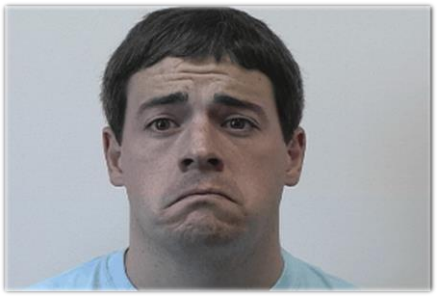

Figura 3.2 *– Izquierda, ejemplo de estructura de la imagen en la clase "Mat". Derecha, imagen adquirida para procesamiento. Imagen extraída de [48].*

A continuación se muestra la construcción usual de una instancia de la clase "Mat" a través de ciertos atributos necesarios para su definición.

```
Mat img(int rows, int cols, int type);

rows; //Indica el número de filas de la matriz(pixeles de largo).
cols; //Indica el número de columnas de la matriz(pixeles de ancho).
type; //Indica el tipo de dato que manejará la matriz y el número de
      canales de la imagen, que en casos normales es de 1 o 3
      canales(BGR).
```

3.2.2 Pre-procesado

Adquirida la imagen y previo a la detección del rostro, la imagen pasa por la "**fase de pre-procesamiento"**, en la que se adecua para una mejor detección de la región de interés. En esta fase la imagen pasa por dos sub-procesos.

El primer sub-proceso aplicado a la imagen es la **transformación a la escala de grises**, que reduce la información de la imagen al representarla en un solo canal (gris) a comparación de una imagen a color que es representada con 3 canales de información (rojo, verde y azul). Lo que permite resaltar su luminosidad, además de permitir el uso del método de detección de rostros a usar [49]. La Figura 3.3

muestra a la imagen adquirida previamente luego de la transformación a una escala de grises.

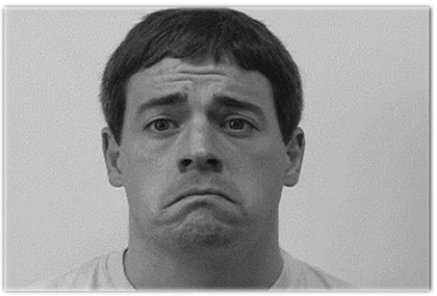

Figura 3.3 *– Imagen transformada en escala de grises. Imagen de autoría propia.*

En este caso, a través de la librería OpenCV, se hace uso de la función "**cvtColor(Mat src, Mat dst, int code)**" que permite realizar la conversión al espacio de colores deseado, que este caso es la escala de grises.

```
src;  //Matriz de la imagen de entrada (a color).
dst;  //Matriz de la imagen de retorno buscada(en escala de grises).
code; //Código de conversión al espacio de color deseado.
```

De forma posterior a la transformación, se realiza el sub-proceso de **ecualización del histograma**, que estandariza la distribución de la intensidad de los pixeles en la imagen mejorando su contraste [49]. Este proceso es necesario debido a que las imágenes son tomadas en distintas condiciones que presentan variaciones en la iluminación y contraste pueden afectar la detección correcta del rostro. La Figura 3.4 muestra la imagen previa luego del proceso de ecualización del su rango de intensidades.

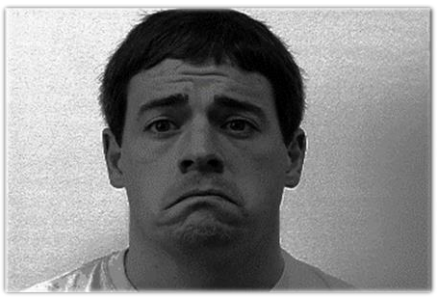

Figura 3.4 *- Imagen ecualizada. Imagen de autoría propia.*

Para este procedimiento de igual forma se hizo uso de la librería OpenCV a través de la función "`equalizeHist( Mat src, Mat dst)`" que permite realizar la estandarización de las intensidades de una matriz de entrada hacia una matriz destino.

En la Figura 3.5 se puede observar de forma más detallada la diferencia entre la imagen original en escala de grises y la posterior imagen con el histograma ecualizado. (a) Presenta un rango de niveles de gris concentrado al lado derecho del histograma, lo que puede darle poco contraste a la imagen y dificulte el proceso de detección, por otro lado, (b) presenta un histograma con valores de intensidad más distribuidos o uniformes a lo largo del rango de valores de intensidad, con lo que el contraste de la imagen queda mejorado y la detección menos propensa a errores.

a) b)

***Figura 3.5**- a) Histograma de la imagen en escala de grises. b) Histograma de la imagen ecualizada. Imágenes de autoría propia.*

3.2.3 Detección

Una vez pre-procesada la imagen, se procede con la "**fase de detección"**. La fase de detección del rostro le proporcionará al computador la base de la información que necesita para el posterior análisis y clasificación de las expresiones faciales, por lo que un error en esta fase implicará el error en las fases posteriores. Por ello para el siguiente modelo se ha implementado la detección del rostro a través del framework Viola-Jones debido a las altas tasas de detección que proporciona, así como la poca carga computacional que implica su uso.

El método propuesto por Viola-Jones[24] permite la detección de rostros en una vista frontal y a diferencia de otros métodos procesa la información presente en una imagen en escala de grises, ya que, no requiere información adicional brindada en

las imágenes a color. El uso innovador de imágenes integrales junto con algoritmo de aprendizaje AdaBoost, sumado a la estructura de clasificación en cascada que usa permiten que el método brinde altas tasas de detección positiva, así como rapidez en el cálculo a un costo computacional bajo. Lo que le brinda cabida a su uso frente a detecciones de objetos en tiempo real.

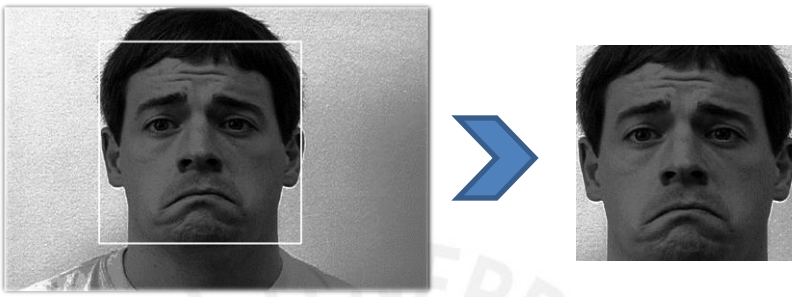

Figura 3.6 *- Detección de la región del rostro en la imagen usando el framework Viola-Jones. Imagen de autoría propia.*

La librería OpenCV permite usar una implementación del framework Viola-Jones a través de clasificadores pre-entrenados para la detección de rostros [20]. Dicha implementación permite variar tanto el tamaño mínimo que podrá tener el rostro detectado (ventana de detección), así como el escalado que se aplicará a la imagen para reducir el área en el cual el framework actuará, lo que se traduce en menor coste computacional debido a que la búsqueda, con la misma ventana de detección, será realizada en un menor espacio.

Para esta fase se generó una función que brinda información del rostro y ojos, sus coordenadas de inicio en la matriz fuente, además de la longitud y achura de dichas regiones inscritas en la clase “Rect”, y cuyo resultado da conocer si se detectó o no el rostro. A continuación se detallan ciertos aspectos de la función.

```
bool face_eyes_detection(Mat& src, Rect& face, Rect& r_eye, Rect&
l_eye);

src;  //Indica la imagen fuente ya procesada donde se realizará la
      detección.
face; //Variable que contendrá la información del rostro detectado.
r_eye;//Variable que contendrá la información del ojo derecho
      detectado.
l_eye;//Variable que contendrá la información del ojo izquierdo
      detectado.
Rect  //Clase OpenCV que contiene coordenadas de ejes Y, X y
      dimensiones Ancho y Largo como atributos.
```

CAPÍTULO 4

4 Descriptor de características relevantes en el rostro

4.1 Introducción

En el presente capítulo se explicará el proceso empleado para la extracción de características relevantes en el rostro, así como los procesos previos utilizados para mejoran la caracterización dada en este prototipo; además, se detallará la estructura final del módulo de procesamiento.

4.2 Descripción

Luego de realizada la detección de la región del rostro, el prototipo requiere que dicha información sea descrita de forma sencilla y única, y que, además, permita conservar de forma eficiente la información relevante en ella para facilitar la labor final de clasificación. Esta etapa denominada de extracción de características se encarga principalmente de obtener los valores que se consideran relevantes en la imagen y tratar de descartar, en lo más posible, el resto de valores que no aporta información sobre el rostro y que pueden interferir negativamente en la labor de clasificación.

Sin embargo, considerando que las detecciones no se dan exactamente en las mismas condiciones, estas pueden brindar regiones de rostros con distintas dimensiones y alineaciones que afecten de facto su caracterización y su reconocimiento. Por ello la imagen obtenida debe pasar por dos procesos previos a la caracterización. De igual forma, todos estos procesos se encontrarán en módulo de procesamiento.

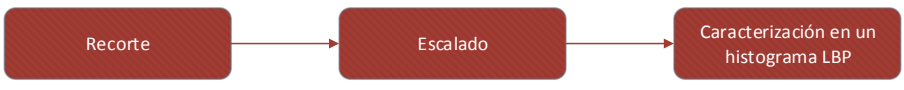

Figura 4.1 *- Secuencia de procesos para la caracterización. Imagen de autoría propia.*

4.2.1 Recorte y Escalado

Una vez que se adquiere la sección del rostro a través del método Viola-Jones, se puede observar que en ella se presentan características que resultan irrelevantes, pues no brindan información alguna de la expresión facial, como el pelo y las orejas; por ello, se debe de tratar de descartar estas secciones.

A través del proceso de recorte se eliminan dichas regiones, reduciendo el área que será analizada y caracterizada. Para el prototipo, se realizó el cálculo de la zona central del rostro, así como el cálculo del desplazamiento horizontal y vertical que se tomará a partir del centro previamente calculado de la imagen. Pudiéndose lograr una ventana mayor de la proporción del rostro o una menor dependiendo del porcentaje que se le atribuya, como se puede observar en la Figura 4.2. El prototipo emplea el recorte con medidas de desplazamiento de 65% horizontal y 85% vertical, ya que estos valores permiten abarcar la región justa que se quiere analizar para el caso de las imágenes de rostros frontales en la base de datos CK.

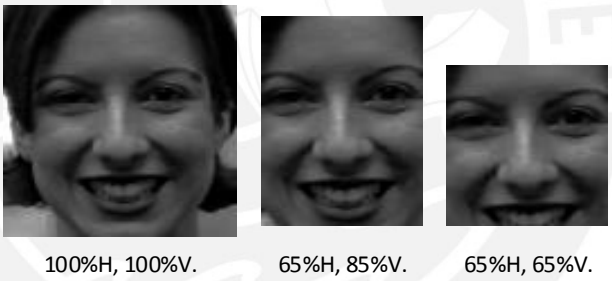

Figura 4.2 *- Proporciones del rostro con sus respectivas medidas. Imagen de autoría propia.*

A continuación se muestra el método generado que realiza este proceso.

```
void crop_face(Mat& image, double offsetx, double offsety);

image;  //Indica la matriz que contiene el rostro a recortar.
offsetx;//Factor de visibilidad del rostro en el eje X.
offsety;// Factor de visibilidad del rostro en el eje Y.
```

Finalmente, a través del proceso de escalado se estandarizan las dimensiones de la imagen del rostro ya recortado (resolución) para que esta cumpla con las mismas características de los rostros de entrenamiento, mejorando así las tasas de reconocimiento en la etapa de clasificación. El prototipo permite variar dichas

dimensiones estándar para verificar con que valores se mejora el reconocimiento, sin embargo, la resolución por defecto es la de 70 x 90 pixeles.

Figura 4.3 *- Rostros recortados y escalados (70 x 90). Imagen de autoría propia. El método contiguo es el que realiza dicho proceso a través de interpolación bilineal brindada por OpenCV.*

```
void resize_image(Mat& image, double rows, double cols);

image;  //Matriz que contiene el rostro recortado a estandarizar.
rows;   //Variable que especifica la cantidad de pixeles de largo.
cols;   //Variable que especifica la cantidad de pixeles de ancho.
```

4.2.2 Caracterización

Para la describir la imagen del rostro ya alineado y estandarizado, la implementación tomó como base la extracción de las características a través de los patrones binarios locales (LBP) en la imagen y su representación en un vector de baja dimensionalidad. La simpleza del algoritmo permite que la transformación de la imagen sea rápida y ayude en la reducción del costo computacional.

La idea del algoritmo es la de resumir la estructura local de una imagen mediante la comparación de cada pixel con su vecindario. El procedimiento básico para la extracción de los patrones binarios locales, basado en un vecindario de 3x3 pixeles, es el siguiente:

Se hace un recorrido pixel por pixel de la matriz de la imagen, solo incluyendo en el proceso los pixeles que tengan un vecindario completo. Se toma dicho píxel central como umbral del vecindario y se analizan los 8 pixeles a su alrededor para formar un código, comúnmente denominado operador LBP. En el análisis se realizan comparaciones de la intensidad, en la escala de grises, de los pixeles; si la intensidad del pixel vecino es mayor o igual que la intensidad del pixel umbral se denota dicho pixel como 1; caso contrario, se denota como 0 [26]. Luego de

realizada la comparación, se tendrá un número binario de 8 bits (formado por la concatenación, en sentido horario, de cada de los valores asignados a los pixeles vecinos, comenzando desde el pixel superior izquierdo) que al ser transformado a decimal sustituirá el valor del pixel umbral tomado, resumiendo así la estructura de dicho vecindario. Finalmente salta al siguiente pixel y se realiza el mismo procedimiento. La Figura 4.4 muestra el procedimiento del algoritmo explicado previamente en seudocódigo.

```
Inicio
   Entrada matriz;
   Generar matriz_lbp [filas_en_matriz-2, columas_en_matriz-2];
   Para f = 2 hasta f = (filas_en_matriz – 1) Hacer
      Para  c = 2 hasta c = (columnas_en_matriz -1) Hacer
         centro = elemento_ en_ matriz[f , c];
         código_lbp = 0;
         Si ( elemento_en_matriz[f-1, c-1]   > centro) Entonces { código_lbp = código-lbp + 2^7}
         Si ( elemento_en_matriz[f-1, c]     > centro) Entonces { código_lbp = código-lbp + 2^6}
         Si ( elemento_en_matriz[f-1, c+1]   > centro) Entonces { código_lbp = código-lbp + 2^5}
         Si ( elemento_en_matriz[f, c+1]     > centro) Entonces { código_lbp = código-lbp + 2^4}
         Si ( elemento_en_matriz[f+1, c+1]   > centro) Entonces { código_lbp = código-lbp + 2^3}
         Si ( elemento_en_matriz[f+1, c]     > centro) Entonces { código_lbp = código-lbp + 2^2}
         Si ( elemento_en_matriz[f+1, c-1]   > centro) Entonces { código_lbp = código-lbp + 2^1}
         Si ( elemento_en_matriz[f, c-1]     > centro) Entonces { código_lbp = código-lbp + 2^0}
         elemento_en_matriz_lbp[f, c] = código_lbp;
      Fin Para
   Fin Para
   Salida matriz_lbp;
Fin Inicio
```

Figura 4.4 *- Seudocódigo para la determinación los patrones binarios locales. Imagen de autoría propia.*

Un ejemplo es mostrado en la Figura 4.5, donde se analiza un vecindario de 3x3 pixeles para la obtención del operador LBP, donde el pixel umbral es el pixel central con intensidad; luego se umbraliza la información y se obtiene el código binario al realizar la concatenación en sentido horario desde el pixel superior izquierdo, resumiendo dicho vecindario en el valor decimal del código binario obtenido.

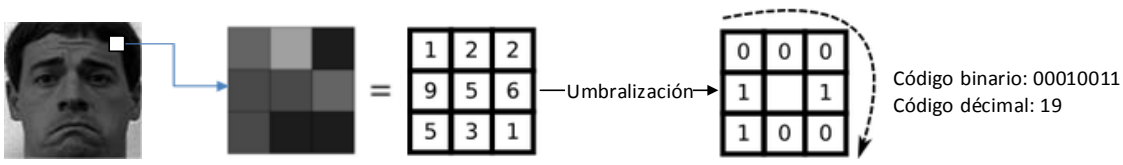

Figura 4.5- *Proceso para la obtención del operador LBP en un vecindario 3x3. Imagen adaptada de*

Una vez terminada la operación en toda la imagen, el resultado es como el mostrado en la Figura 4.6, en el cual se aprecia la retención e intensificación de ciertas características del rostro y la omisión de otras.

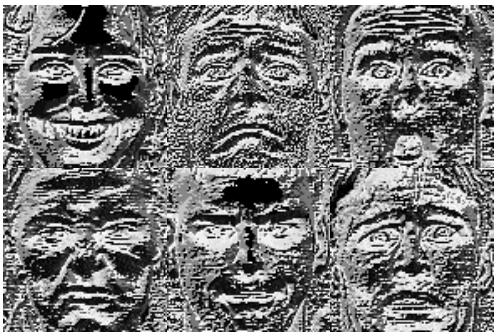

Figura 4.6 *- Imágenes transformadas mediante el algoritmo LBP original. Imágenes de autoría propia.*

La matriz que contendrá la imagen LBP mantendrá en formato decimal los operadores LBP obtenidos en el recorrido de toda la imagen. Además se destaca que esta nueva matriz se verá reducida en dos filas y dos columnas respecto a la dimensión de la imagen dada, debido a que solo se mantiene la información de los pixeles centrales, como se muestra en la Figura 4.7, y que además reduce el tamaño de los posibles valores que tendrá cada celda al depender esta del tamaño del vecindario, que en el caso original (8 elementos) son de 256 posibles valores.

4	5	63	9	1	6
10	12	57	5	2	1
3	6	35	8	0	6
3	5	22	7	0	7
3	3	54	7	2	5
3	3	40	13	0	2

Región gris: pixeles de la imagen descartados como umbrales en el algoritmo LBP.

Región blanca: pixeles umbrales que serán analizados por el algoritmo LBP y que generaran operadores LBP.

La imagen o matriz LBP resultante del algoritmo mantendrá la dimensión de la región blanca.

Figura 4.7 *– Región de reducción en una imagen LBP. Imagen de autoría propia.*

Sin embargo, el algoritmo usado en el prototipo es su versión extendida (ELBP), que permite lograr la misma idea base pero a través de vecindarios de radio mayores que uno y con un número de vecinos mayores que ocho. Lo cual trae consigo dos efectos:

1) Al incrementar el radio del vecindario, se logra suavizar la imagen transformada y capturar detalles más grandes en la imagen.

2) Al incrementar el número de vecinos o puntos de muestreo, se tendrá más poder discriminativo, puesto que se codificarán más patrones al incrementarse la cantidad de bits del operador LBP, pero a cambio de costo computacional y un mayor rango de posibles valores.

Esto le permite al prototipo experimentar con la caracterización a distintos grados de descripción y permitir observar con que valores de radio y puntos de muestreo logra una mejor discriminación.

Transformada ya la imagen en una estructura resumida, solo hace falta traspasar toda la información en un vector. Para ello se puede volcar dicha información en un único histograma, sin embargo, en dicho proceso se descarta la información espacial que es esencial para tareas ligadas al reconocimiento; por lo que, se usó la propuesta dada por Ahonen et al. en [2], donde se procede a dividir la imagen LBP en cuadrículas, realizar el cálculo de histogramas de cada una y proceder a concatenarlos en secuencia, reteniendo de esta forma la información espacial de la imagen y generando un vector de histogramas como si fuese un histograma único, tal como se puede apreciar en la Figura 4.6.

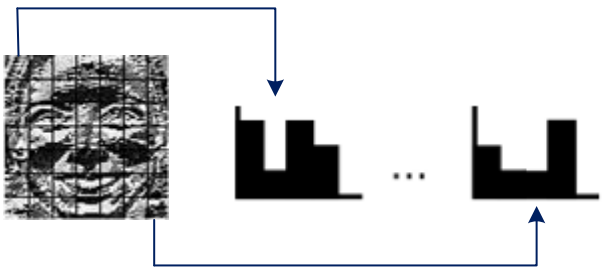

Figura 4.8 *- Concatenación de histogramas LBP calculados de cada cuadrante. Imagen de autoría propia.*

Su generación se puede observar en el seudocódigo mostrado en la Figura 4.9, donde en primer lugar se recorre la imagen obteniéndose cada cuadrícula como una matriz por separado, y una vez con ella se procedió con el cálculo de su histograma. Aquí se recorre cada celda de dicha matriz y por cada una se obtiene su valor de intensidad, valor que es tomado como posición dentro una matriz unidimensional (histograma) para incrementar el número de incidencias de dicha posición o valor de intensidad; este proceso se repite por cada celda hasta completar la dimensión de esa cuadrícula y haber calculado todo su histograma. Finalmente, obtenidos todos los histogramas, se genera una matriz unidimensional con la longitud del histograma de cada cuadrícula multiplicada por el número de cuadriculas generadas en la imagen, y se procede a copiar en secuencia cada valor de los histogramas obtenidos. El siguiente método es el que se encarga de la generación del vector característico o matriz unidimensional.

```
Mat spatial_histogram(const Mat& src, int numPatterns, int gridx,
int gridy);

Src;        //Indica la matriz fuente.
numPatterns;//Indica la cantidad de elementos de cada histograma.
gridx;      //Indica la cantidad de cuadrículas en el eje X.
grixy;      //Indica la cantidad de cuadrículas en el eje Y.
```

Debido a las características de la función el prototipo también permite variar la cantidad de divisiones que se hacen a la imagen con el fin de lograr una mayor experimentación de la caracterización.

```
Inicio
    Entrada matriz;
    Entrada num_patrones;
    Entrada columnas_cuadrícula,filas_cuadrícula;
    Generar vector_de_histogramas;
    Generar histograma [1 , num_patrones];
    Para c = 0 hasta c = (columnas_matriz - columnas_cuadrícula) Hacer
        Para  f = 0 hasta f = (filas_matriz – filas_cuadrícula) Hacer
            matriz_cuadrila = obtener_cuadrícula(c,f,columnas_cuadrícula,filas_cuadrícula,matriz);
            histograma = calcular_histograma (matriz_cuadrícula);
            Agregar histograma A vector_de_histogramas;
            incrementar f en filas_cuadrícula;
        Fin Para
        incrementar c en columnas_cuadrícula;
    Fin Para

    Generar histograma_concat [1 , num_histogramas * num_patrones];
    Para h = 0 hasta h = número_histogramas_en_vector Hacer
        Para  i = 0 hasta i = columnas_histograma Hacer
            y = h * columnas_histograma + i;
            elemento_en_histograma_concat[0 , y] = vector_de_histogramas[h].elemento[i];
        Fin Para
    Fin Para
    Salida histograma_concat;
Fin Inicio
```

Figura 4.9 *- Seudocódigo de la generación del histograma completo (concatenado). Imagen de autoría propia.*

En el caso por defecto del prototipo, la división dada es de 3 x 4 cuadrículas lo que permite que toda la información de los histogramas concatenados brinden un vector de características de 3072 valores, siendo este el resultado de la cantidad de histogramas formados en la imagen (12 cuadrículas) multiplicado por el rango de valores de intensidad que tiene cada histograma (256 valores – caso LBP original). Sin embargo, dado que la dimensión del vector aún es alta, se implementó el método Uniform Pattern para reducir el tamaño de los histogramas a 59 valores (caso LBP original) y por ende reducir la dimensión del vector quedando con 708 valores característicos en el prototipo por defecto.

El método Uniform Patterns señala que en un histograma LBP ciertos patrones (intensidades) retienen más información que otros [50]. Es por ello que es posible utilizar solo un subgrupo de los patrones usados en el LBP para describir la textura. Ojala et al. [50] denominó a este subgrupo de patrones como uniform patterns, en el cual un patrón binario local es uniforme si contiene como máximo 2 transiciones de 1 a 0 o viceversa. Mediante esta idea entonces se puede generar un histograma que solo contenga los patrones uniformes y un contenedor adicional para el resto de patrones no uniformes. Así para el caso LBP original con 256 posibles patrones

existen solo 58 que cumplen dicha regla, generándose un histograma de 58+1 valores. La Figura 4.8 muestra una imagen LBP antes y después de aplicación del método de patrones uniformes, en la que se puede apreciar una reducción en la intensidad de la imagen, debido a la menor gama de intensidades que maneja el histograma, sin perder la definición de las características en esta.

Figura 4.10 *- Imagen original LBP (izquierda) - Imagen LBP con Uniform Patterns (derecha). Imagen de autoría propia.*

En el prototipo se hace uso del siguiente método que implementa el algoritmo LBP y que adiciona el método Uniform Patters para reducir la dimensionalidad del vector característico.

```
Mat up_elbp(const Mat& src, bool uniform, vector<int> lookup, int
radius = 1, int neighbors = 8);

src;        //Matriz fuente.
radius;     //Radio que tomará el algoritmo LBP.
neighbors;  //Número de vecinos que tomará el algoritmo.
uniform;    //Indica si se hace uso del método uniform patterns.
lookup;     //Variable que tiene todos los patrones uniformes
            identificados y  enumerados que pueden ser generados
            con un vecindario específico.
```

4.3 Módulo de procesamiento

Una vez definido todo el flujo de procesamiento que debe seguir el prototipo, se puede especificar su integración a la estructura del prototipo. Este módulo representado específicamente a través de la clase "**Processor**" adiciona todos los procesos vistos previamente como métodos propios para conseguir la extracción de las características. A continuación se presenta un breve vistazo de los atributos más importantes de esta clase.

```
class Processor{

private:

    //Información de la caracterización
    int  gridx;           //Número de divisiones en el eje x.
    int  gridy;           //Número de divisiones en el eje y.
    int  radius;          //Radio usado en el cálculo LBP.
    int  neighbors;       //Número de vecinos.

    bool uniform;         //Condicional de uso de patrones uniformes.
    int   num_uniforms;   //Número de patrones uniformes.
    vector<int>
         uniform_lookup;  //Vector de mapeo de patrones uniformes y no uniformes.

    double offsetx;       //Factor de visibilidad del rostro en el eje X.
    double offsety;       //Factor de visibilidad del rostro en el eje Y.

    double standardDimx; //Ancho estándar del rostro a caracterizar.
    double standardDimy; //Largo estándar del rostro a caracterizar.
         .
         .
         .
}
```

Dicha estructura permite mantener los aspectos más relevantes para la extracción de las características del rostro, además de proveer una capacidad alta para la variación de la caracterización del rostro a través de la técnica LBP.

CAPITULO 5

5 Modelo de clasificación de expresiones faciales y validación del modelo

5.1 Introducción

En el presente capítulo se detallarán los procesos envueltos alrededor de la generación del modelo de clasificación (predicción y entrenamiento), así como su integración con los procesos vistos anteriormente en el módulo de procesamiento; se presentará el set de datos usado para el entrenamiento y la prueba del modelo; así como las especificaciones del método de validación usado en el modelo generado.

5.2 Descripción

La fase de clasificación, incluida en el módulo con el mismo nombre, viene a ser la última etapa del reconocimiento cuando se intenta discriminar la expresión de un rostro en una imagen. Para que esta se logre es necesario que el modelo base su capacidad de discriminación respecto a cierto aprendizaje previo recibido y que, en el caso del prototipo, es brindado a través del entrenamiento mediante imágenes con expresiones previamente etiquetados.

Es por ello que se encuentran dos elementos esenciales que intervienen directamente en la tarea de la clasificación:

- La base de datos o el set de datos de imágenes con expresiones faciales, que permitirán brindar la información para el entrenamiento y prueba del prototipo.
- El modelo computacional entrenado, que mediante el método de aprendizaje de máquina usado e integrado con la detección del rostro y la extracción de características permite que se logre la clasificación buscada por el prototipo.

Una vez con el modelo entrenado, el prototipo debe brindar la certeza sobre los resultados de predicción que brinda, por lo que, como etapa final del prototipo, se realiza la validación del modelo generado, de igual forma incluido en el módulo de clasificación.

5.2.1 Base de datos de imágenes Cohn-Kanade (CK)

Como se mencionó previamente la base de datos usada para el entrenamiento del modelo es uno de los aspectos fundamentales para las tareas de clasificación o reconocimiento, esto debido principalmente a que el "aprendizaje" que realizará el modelo tomará como base la información determinada en los set de datos. En este caso, el prototipo hace uso del set de imágenes brindados por la base de datos de imágenes CK, en su versión inicial [51] y extendida [48], debido a la serie de características que lo convierten en un set de datos ideal para el entrenamiento de expresiones, así como también por ser unas de las bases de datos más usadas en investigaciones de la comunidad de reconocimiento de expresiones faciales.

A continuación se presentan algunas características de la base de datos CK:

- De uso libre para la investigación en el reconocimiento automático facial.
- Contiene imágenes con una alta variedad de expresiones faciales. Ver la Figura 5.1.
- El set total contiene un número de 123 sujetos.
- Las imágenes toman la secuencia de cada expresión, desde un estado neutral hasta el estado pico de la expresión. Ver la Figura 5.2.
- Cada secuencia de expresión registrada esta codificada en el sistema FACS.
- Los sujetos mantienen una edad entre los 18 y 30 años.
- 65% de las muestras pertenecen a mujeres.
- Las imágenes tienen una vista de pose frontal.
- Las imágenes tienen una dimensión de 640x490 pixeles.
- Contiene información adicional respecto a las AU mapeadas en cada expresión facial.

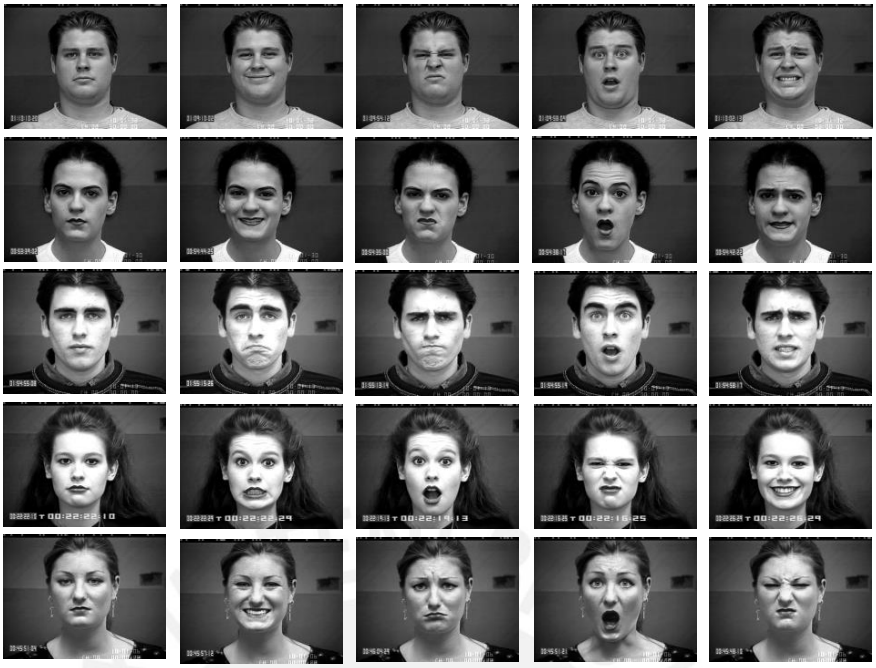

Figura 5.1 *- Ejemplos de expresiones faciales en la base de datos CK. Imagen adaptada de [48].*

Figura 5.2 *- Secuencia de una expresión desde un estado neutral hasta el pico de la expresión. Imagen adaptada de [48].*

Dado que la base de datos CK presenta una serie de expresiones, secuencias y poses que no son necesarias para entrenar el modelo se hizo una selección manual de las imágenes que serían de utilidad para el mismo. A continuación se presenta una serie de criterios tomados para la selección realizada.

- La base de la selección son las 6 emociones globales: Felicidad, tristeza, enojo, asco, miedo y asombro, además del estado neutral.
- Se generan 7 clases de emoción, cada una con 55 imágenes de expresiones faciales pertenecientes a una misma clase que brinda un total de 385 imágenes.
- Cada imagen en el subgrupo es de un sujeto diferente.
- Cada imagen representa una expresión en su estado pico.

- Se utiliza solo el estado pico de la expresión para disminuir la probabilidad de error en la clasificación que puede dar una expresión en proceso.
- La base de la selección para determinar que expresión alude a tal emoción fue dada respecto a la descripción de las emociones en términos de unidades acción facial junto a los criterios de discriminación propios del autor.

La Tabla 5-1 muestra los criterios respecto al sistema de codificación de acción facial para determinar cada emoción, en esta se observan las combinaciones de distintas unidades de acción que formarían las expresiones representativas de una u otra emoción.

Emoción	Criterio
Felicidad	AU12 debe estar presente.
Tristeza	AU1+4+15 debe estar presente o AU11. Una excepción es AU6+15.
Enojo	AU23 y AU24 deben estar presentes en una combinación.
Asco	AU9 o AU10 debe estar presente.
Miedo	Combinación de AU1+2+4 debe estar presente, a menos que AU5 sea de intensidad E en ese caso AU4 puede estar ausente.
Asombro	AU1+2 o AU5 debe estar presente y la intensidad de AU5 no debe ser mayor que B.

Tabla 5-1 *- Descripción de la emoción en términos de unidades de acción faciales [48].*

Dentro del prototipo se mantiene la base de datos de imágenes seleccionada como recurso base para la generación de los distintos modelos de predicción. Esta base de datos está dividida en agrupaciones por cada tipo de expresión facial. Además, el prototipo mantiene un archivo en formato .CSV que mapea todos los nombres de los recursos de las imágenes, sus carpetas o agrupaciones, así como la etiqueta que indica a qué clase de expresión pertenece.

Expresión	Etiqueta
Enojo	0
Asco	1
Felicidad	2
Tristeza	3
Miedo	4
Asombro	5
Neutral	6

Tabla 5-2 *- Representación numérica de cada expresión.*

El archivo separado por comas (CSV) cuenta con la siguiente estructura:

Ruta	Nombre de imagen	Formato de imagen	Etiqueta

Finalmente, en la Figura 5.3 se puede observar las agrupaciones formadas dentro de la base de datos, donde cada una contiene 55 imágenes que representan las expresiones en las que están catalogadas.

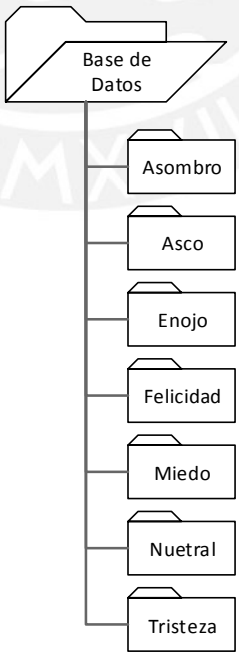

Figura 5.3 *- Agrupaciones de la base de datos de imágenes.*

5.2.2 Flujos del modelo de clasificación

La clasificación es la última fase por la que debe pasar el elemento que quiere ser discriminado, lograr esto requiere que el prototipo maneje cierto “conocimiento” que le brinde dicha capacidad. Este “conocimiento” es brindado a través del entrenamiento de un modelo generado por un método de aprendizaje que lo construye basándose en la información que se le brinde; por ello, para que sea adquirido de forma adecuada, la información, que la entrenará y generará, debe de ser preparada. A continuación se presentan tanto el flujo de entrenamiento como el de clasificación por los que pasará el prototipo.

En primer lugar para el entrenamiento, de igual forma que en la clasificación o predicción, es necesaria la adquisición de imágenes, pero a diferencia de esta, se cargan todas aquellas que entrenarán al modelo. En el prototipo esta se realiza a través de la lectura del archivo .CSV, mencionado anteriormente, con la que se cargan dos vectores, un vector que guarda la imágenes en formato “Mat” y otro que guarda sus etiquetas en un vector de enteros. De forma posterior, se comunica con el módulo de procesamiento para realizar las fases de pre procesado, detección y caracterización de cada imagen en el vector con el fin generar un conjunto de data estandarizada de entrenamiento. Finalmente, el método de entrenamiento recibe el conjunto de caracterizaciones junto con sus respectivas etiquetas y genera un modelo que logrará discriminar dichas caracterizaciones.

En segundo lugar para la clasificación, se lleva acabo el mismo flujo para obtener la caracterización pero esta vez de solo una imagen. Una vez obtenida, se procede a enviarla al método de predicción del modelo previamente entrenado, para que finalmente emita su resultado de predicción basándose en toda la data de entrenamiento que se le brindo.

La Figura 5.4 muestra el flujo descrito previamente, para el entrenamiento y clasificación. La línea azul muestra el flujo de la clasificación y la roja del entrenamiento.

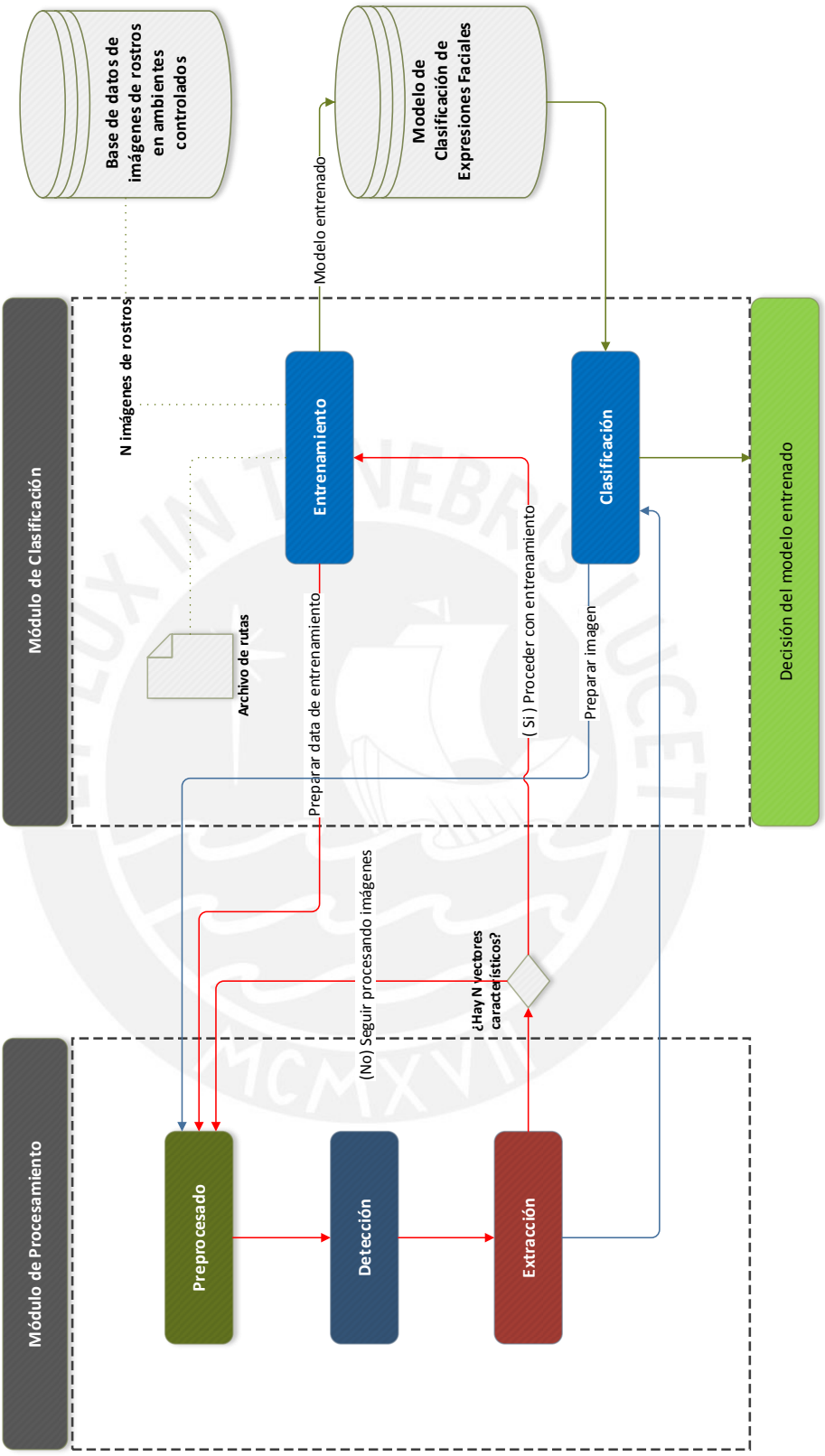

Figura 5.4 *– Flujo de procesos de clasificación y entrenamiento. Imagen de autoría propia.*

5.2.3 Generación del modelo de clasificación

Para la generación del modelo de clasificación en el prototipo se utilizan métodos de aprendizaje por **Boosting** que por definición utilizan de forma iterativa clasificadores débiles (algoritmos de aprendizaje más simples) para generar un clasificador mucho más robusto y acertado.

En el prototipo, la integración del método AdaBoost se da a través de la librería OpenCV que permite construir correctamente el modelo de aprendizaje de máquina necesitado. OpenCV brinda la clase "CvBoost" que maneja métodos para la generación de modelos de predicción binarios a través de la especificación de ciertos parámetros, además de la especificación de los datos de entrenamiento con sus respectivas etiquetas. A continuación se muestran los parámetros más importantes que se requieren definir en CvBoost:

```
Int boost_type;   //Tipo de método Boosting a utilizar en el modelo.
Int weak_count;   //Número de clasificadores débiles que usa.
Double weight_trim_rate;//Umbral entre 0 y 1 usado para ahorrar
                  tiempo computacional en el entrenamiento.
```

Dado que en el caso del prototipo, es requerido discriminar entre 6 distintas clases de elementos, el método desarrollado para el entrenamiento genera 6 diferentes modelos binarios de predicción; vale decir, que cada modelo generado está basado en la distinción de 2 distintas clases, aquellas que pertenecen a una emoción en particular y aquellas que no (resto de emociones). En la Tabla 5-3 se presentan todas las combinaciones de clases que serán usadas para la generación de modelos.

Clase 1 (Emoción principal)	Clase 2 (Resto de emociones)	Modelo Generado
Enojo	Asco-Felicidad-Tristeza-Miedo-Asombro-Neutral	angry_model
Asco	Enojo-Felicidad-Tristeza-Miedo-Asombro-Neutral	disgust_model
Felicidad	Enojo-Asco-Tristeza-Miedo-Asombro-Neutral	joy_model
Tristeza	Enojo-Asco-Felicidad-Miedo-Asombro-Neutral	sad_model
Miedo	Enojo-Asco-Felicidad-Tristeza-Asombro-Neutral	scare_model
Asombro	Enojo-Asco-Felicidad-Tristeza-Miedo-Neutral	surprise_model

Tabla 5-3 *- Especificación de clases de datos para la generación de modelos.*

Por ello, el método desarrollado obtiene todas las imágenes que le servirán de entrenamiento con sus respectivas etiquetas, las envía al módulo de procesamiento para obtener todas sus caracterizaciones. Una vez con ellas procede a separarlas por su clase, todas las caracterizaciones de una misma clase las mantiene en una matriz, a través de la clase **Mat**, donde cada fila contendría una imagen caracterizada. Para poder acumular todas estas clases hace uso de un vector de matrices **Mat**. Finalmente, ya con las caracterizaciones divididas, procede a realizar las combinaciones de datos observados en la Tabla 5-3, cada combinación es nuevamente ingresada en una matriz **Mat** que la mapea con su clase y que es enviada al método de entrenamiento, proporcionado por la clase **CvBoost,** generando un modelo. De esta forma, se genera el resto de modelos necesarios para la predicción de las 6 expresiones globales y que en conjunto forma un modelo de predicción multi-clase.

Por ejemplo, en el caso de la expresión del enojo, el método desarrollado usa todas las caracterizaciones de entrenamiento de esta expresión y las etiqueta como clase 1, luego toma el resto de caracterizaciones y las etiqueta como clase 2. Este conjunto de datos y etiquetas al ser entrenado permite discriminar entre lo que es una expresión de enojo y lo que no es. A continuación se presenta el método que se encarga de la generación de los modelos de predicción de expresiones y que se encuentra incluido en el módulo de clasificación.

```
bool boost_train(string csv_source_path);

csv_source_path; //Variable que indica la ruta del archivo en
                 formato .CSV.
```

Para la predicción o clasificación, se desarrolló un método que unificará todos los modelos generados previamente y que a partir de ellos emita un único resultado. A continuación se presenta dicho método, que de igual forma se encuentra incluido en el módulo de clasificación.

```
int boost_predict(Mat face_characterized);

face_characterized;//Variable que indica el rostro ya caracterizado
                   en una matriz unidimensional.
```

5.2.4 Validación cruzada del modelo de clasificación

Como se mencionó de forma inicial en la descripción del capítulo, la validación del modelo generado es la última tarea que cumple el prototipo con el fin de garantizar que la tasa de precisión que brindan sus valores de predicción, sobre la base de datos, no esté sujeta a las particiones de datos con las que se entrena y prueba el modelo.

El prototipo hace uso del método de la validación cruzada, el cual esencialmente de forma iterada, y dado el conjunto de datos de entrenamiento general, particiona su entrada en datos de entrenamiento y datos de test, entrena un modelo con los datos de entrenamiento y brinda sus predicciones sobre los datos de test; así se realizan distintas variantes por cada iteración y se obtiene un porcentaje de precisión por cada una. Al finalizar el proceso se realiza la media de todos los valores de precisión obtenidos calculando un valor de precisión general que

garantiza la no dependencia de la partición de los datos de entrenamiento sobre la base de datos usada.

Con este propósito se desarrolló un método sobrecargado de validación cruzada:

```
double          cross_validation(Mat class1,Mat class2);
double          cross_validation(Mat elements);
```

Ambos realizan el mismo método de partición de entrenamiento y test pero con ciertas especificaciones respecto a la forma de obtener dichos elementos, como se puede observar en las Figuras 5.5 y 5.6.

La primera sobrecarga requiere de dos matrices “class1” y “class2” con las caracterizaciones de los elementos que le sirven de entrenamiento en sus filas pero divididas por clase; en la correspondiente figura cada circulo representa un elemento caracterizado, es decir una fila de la matriz, y cada color representa una clase, es decir cada matriz. Aquí la extracción de los elementos de test se da por cada clase, de forma proporcional y consecutiva, así se tiene siempre como conjunto de test elementos de ambas clases y el resto para el entrenamiento.

Por otro lado, en la siguiente sobrecarga solo se requiere de la matriz “elements”, que contiene todas las caracterizaciones de los elementos de entrenamiento en secuencia; es decir que contendrá todas las caracterizaciones de los elementos de la clase 1 seguido de todas las caracterizaciones de los elementos de la clase 2. En este caso la extracción de los elementos de test es de forma consecutiva, se extraen “n” elementos consecutivos de la matriz que son considerados como test, mientras que el resto de elementos es considerado de entrenamiento como se aprecia en la figura correspondiente.

En ambos casos, cada iteración actualiza la posición desde donde se debe de comenzar a obtener los elementos de test, genera una variante para el entrenamiento intermedio y calcula la precisión de esa variante respecto a los datos de test. Como resultado final retornan el promedio de precisiones obtenido por cada iteración.

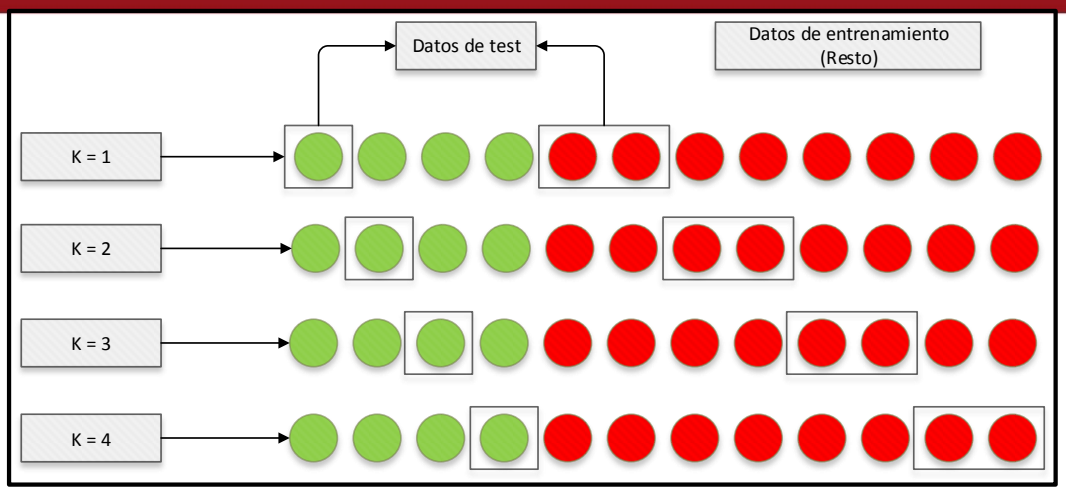

Figura 5.5 *- Validación cruzada de "k" iteraciones a través de dos matrices de datos. Imagen de autoría propia.*

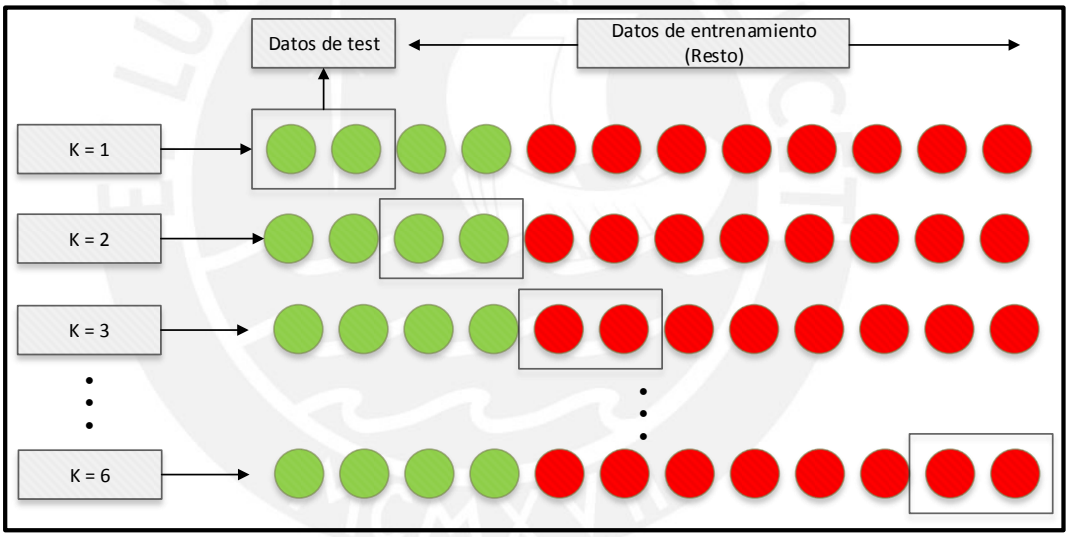

Figura 5.6 *- Validación cruzada de "k" iteraciones en una única matriz de datos. Imagen de autoría propia.*

5.2.5 Módulo de clasificación

Definido todo el flujo que sigue la clasificación tanto para su proceso de entrenamiento como para la predicción y de forma adicional su validación se puede detallar su estructura en el prototipo. Este módulo es representado a través de la clase "**Classifier**" que adiciona todos los métodos vistos anteriormente para el entrenamiento de modelos, la predicción basada en estos y su validación. A continuación se presenta la estructura principal de esta clase a través de sus atributos principales.

```
class classifier{

private:

    //Modelos de predicción
    CvBoost angry_model;
    CvBoost disgust_model;
    CvBoost joy_model;
    CvBoost sad_model;
    CvBoost scare_model;
    CvBoost surprise_model;

    bool were_trained;    //Indica si todos los modelos han sido entrenados o no.
    bool previus_models;  //Indica si existen modelos ya entrenados previamente pero no activos.
    bool previus_charact; //Indica la existencia de caracterizaciones previas.

    //Información del modelo
    int n_classes;        //Número de clases en la base de datos de imágenes.
    int elem_per_class;   //Número de elementos por clase.
    int test_elem_per_class; //Número de elementos para test por clase.

    //Información sobre los parámetros
    int boost_type;       //DISCRETE=0, REAL=1, LOGIT=2, GENTLE=3
    int weak_count;       //Indica la cantidad de clasificadores débiles en cada modelo.

   //Dependencia con el procesamiento de imágenes
    Processor proc;

    //Información para la validación
    int cv_type;          //Tipo de validación cruzada: extracción consecutiva o por cada clase.
    CvBoost cv_model;     //Modelo intermedio para las validaciones cruzadas.
    int n_groups;         //Número de elementos extraídos para test en la validación cruzada
    int k_folds;          //Número de iteraciones en la validación cruzada.
        .
        .
        .
}
```

Esta estructura permite que el modelo global maneje los 6 modelos de predicción mencionados en el flujo y emule un clasificador multi-clase, le permite realizar las configuraciones sobre el método de aprendizaje, especificar el número de elementos que serán utilizados para el test en los modelos y finalmente configurar aspectos del método de validación del modelo. Por lo que la estructura permite variar aspectos propios del entrenamiento y validación de los modelos que en el prototipo se generan.

5.2.6 Integración en el prototipo

Hasta este momento ya se han definido los dos módulos que se utilizarán en el prototipo, pero con el fin de especificar su interacción real se presente en la Figura 5.5 el diagrama de los componentes de los cuales estará compuesto. Este diagrama muestra las relaciones y dependencias que tendrá cada elemento incluido en el prototipo.

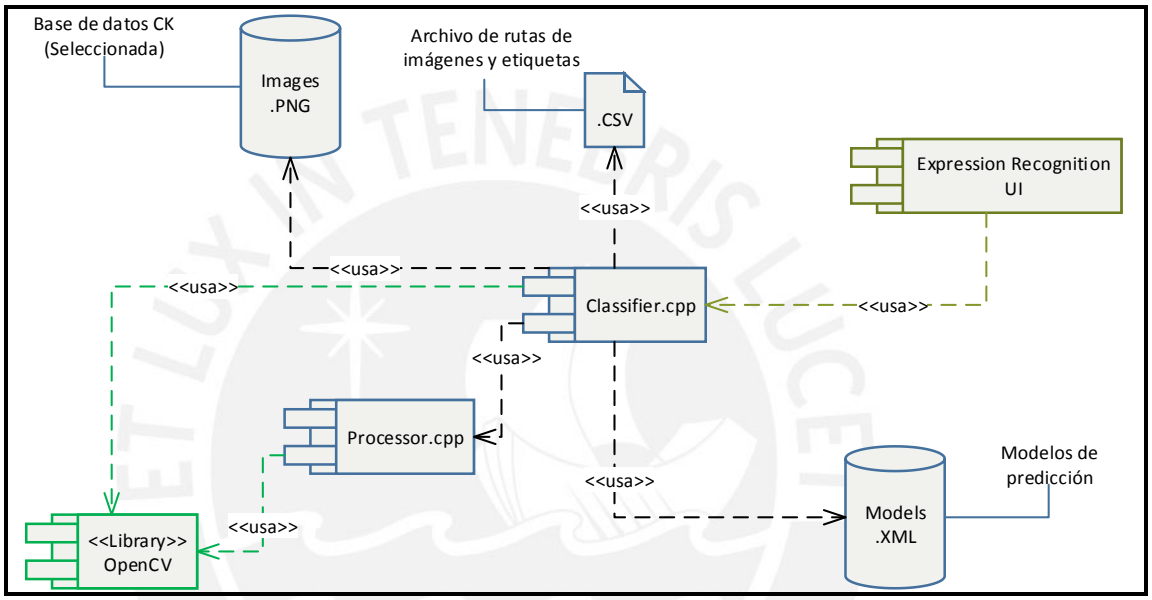

Figura 5.7 - *Diagrama de componentes del prototipo. Imagen de autoría propia.*

CAPITULO 6

6 Experimentación

6.1 Introducción

En el presente capítulo se detallan las observaciones obtenidas a lo largo del desarrollo del prototipo, el protocolo experimental usado por el prototipo, una experimentación previa para determinar el método de aprendizaje a usar por el mismo y finalmente las experimentaciones y resultados obtenidos mediante el prototipo final.

6.2 Observaciones

Dentro del trabajo de fin carrera se delimitó en sí tres fases que permitirían el cumplimiento del objetivo general (fase de detección, caracterización y clasificación), dentro de estas se hicieron ciertas acotaciones respecto a su funcionamiento para un mejor desempeño. A continuación se presentan dichas acotaciones.

En la fase de detección, al hacer uso del framework Viola-Jones, se hicieron pruebas respecto a la capacidad de detección del mismo, determinando hasta qué punto una imagen podía ser reducida y aun así detectar el rostro del individuo. A través de esto, se llegó determinar que aproximadamente el rostro debería tener una dimensión mayor a 23 x 23 pixeles para lograr ser detectado; sin embargo, si adicionalmente se quiere capturar o reconocer ciertas secciones dentro del rostro, como los ojos, a través del mismo framework, se observó que se requiere de una dimensión mayor de aproximadamente 90 x 90 pixeles.

En la fase de caracterización, una de las acotaciones más importantes puede darse respecto a la cantidad de características que brindarán la representación del rostro. Ya que se observó que al enfocar esta de una forma holística existe un sin-número de características en ella que no son relevantes y que simplemente causan ruido en su representación, por lo que se adicionó otro método que le permitió reducir las características que guardan menos información al representar el rostro mediante el método LBP.

En la fase de clasificación, se integró inicialmente el método SVM y se pudo observar que las tazas de predicción lanzadas no eran lo suficientemente altas para poder lograr una buena clasificación con la caracterización dada. Por ello se analizaron distintos métodos de aprendizaje y se observó que los resultados brindados por algunos de estos resaltan más las diferencias brindadas en el vector característico, reduciendo aún más el ruido y permitiéndoles mejorar la clasificación.

6.3 Protocolo experimental

La experimentación llevada a cabo se dio mediante el uso de las imágenes de la base de datos Cohn-Kanade (CK) [48, 51], de forma específica, con la selección de imágenes y división de las mismas en clases de expresiones detallada en el capítulo previo. Para ello se debe especificar el protocolo usado en la base de datos sobre el uso de imágenes para la experimentación en el entrenamiento del modelo y en la prueba. Dado que el universo de datos, para el prototipo, es de 385 imágenes, se han dividido en 350 imágenes para el entrenamiento del modelo y 35 imágenes para las pruebas, vale decir que de cada clase de 55 imágenes con expresiones se han tomado 5 para la prueba y 50 para el entrenamiento del modelo. De igual forma se menciona que para la generalización del performance de los modelos experimentales se ha adoptado el esquema de validación cruzada de 10 iteraciones, mostrando los valores promedio en los resultados.

6.4 Experimentos de clasificación

Para poder determinar que algoritmos de aprendizaje pueden brindar mejores resultados en la clasificación a través de la caracterización realizada y con el fin de determinar que técnica puede ser utilizada en el prototipo, se hizo uso de la herramienta “WEKA” para efectuar dicho análisis, ya que esta herramienta permite generar modelos de clasificación a través de los datos que se le brinden mediante distintas técnicas de aprendizaje de máquina.

Para ello, en primer lugar, dado que las mejores tasas de clasificación son determinadas cuando son comparadas dos clases, se procedió a hacer modelos de clasificación en pares entre toda la base de datos de expresiones obtenida la cual cuenta con 55 imágenes por cada expresión facial (felicidad, tristeza, asombro, enojo, asco, miedo, neutral).

La Tabla 6.1 muestra todas las posibles combinaciones (21) que pueden darse para clasificar 2 clases entre los estados previamente definidos, además de los mejores resultados de precisión obtenidos en promedio por ciertos algoritmos de aprendizaje que brinda WEKA. En esta herramienta, también se analizó el algoritmo SVM implementado inicialmente, sin embargo, para todos los caso brindó una tasas de precisión de 45.45%, por lo que no es considerado en la tabla.

El análisis de la tabla permite conocer que ciertos algoritmos de aprendizaje facilitan la discriminación con la caracterización dada, así como permite determinar que pares de expresiones faciales son más sencillos de clasificar al mostrar promedios altos en todos los métodos de aprendizaje.

Como resultado al análisis de la tabla anterior, se optó por el uso de métodos de aprendizaje basados en Boosting dado que proveyeron algunos de los mejores resultados para la clasificación con la caracterización realizada, además de estar presente y disponible en la librería OpenCV, que de forma adicional permite usar entre cuatro variaciones distintas de Boosting.

Tabla 6-1 *- Porcentajes de precisión de distintos algoritmos de aprendizaje.*

			ALGORITMOS DE APRENDIZAJE							
N°	CLASE 1	CLASE 2	Adaboost.M1	MultiClassClassifier	Dagging	LogitBoost	RandomSubSpace	RotationForest	ThresholdSelector	Promedio
1	Felicidad	Tristeza	95.45%	96.36%	91.82%	95.45%	93.64%	96.36%	97.27%	95.19%
2	Felicidad	Asombro	92.73%	99.09%	96.36%	95.45%	94.55%	97.27%	88.18%	94.80%
3	Felicidad	Enojo	93.64%	97.27%	94.55%	90.90%	90.90%	92.73%	91.82%	93.12%
4	Felicidad	Asco	95.45%	96.36%	93.64%	95.45%	91.82%	94.55%	86.36%	93.38%
5	Felicidad	Miedo	69.09%	75.45%	73.64%	78.18%	72.73%	75.45%	77.27%	74.54%
6	Felicidad	Neutral	94.55%	94.55%	91.82%	94.55%	94.55%	93.64%	90.91%	93.51%
7	Tristeza	Asombro	82.73%	87.27%	82.73%	80%	78.18%	80.91%	80.91%	81.82%
8	Tristeza	Enojo	72.73%	74.55%	70.91%	73.64%	73.64%	76.36%	62.73%	72.08%
9	Tristeza	Asco	87.27%	95.45%	91.82%	91.82%	90.91%	88.18%	89.09%	90.65%
10	Tristeza	Miedo	84.55%	86.36%	88.18%	87.27%	86.36%	88.18%	88.18%	87.01%
11	Tristeza	Neutral	59.09%	70.91%	60%	57.27%	60.91%	62.73%	68.18%	62.73%
12	Asombro	Enojo	89.09%	96.36%	97.27%	86.36%	89.09%	90.91%	96.36%	92.21%
13	Asombro	Asco	94.55%	94.55%	96.36%	94.55%	90.91%	95.45%	69.09%	90.78%
14	Asombro	Miedo	90.91%	90.91%	90%	90.91%	87.27%	92.73%	92.73%	90.78%
15	Asombro	Neutral	79.09%	90.90%	87.27%	80%	80.91%	82.73%	75.45%	82.34%
16	Enojo	Asco	85.45%	90.91%	92.73%	83.64%	82.73%	85.45%	83.64%	86.36%
17	Enojo	Miedo	87.27%	89.09%	89.09%	87.27%	80%	88.18%	90.91%	87.40%
18	Enojo	Neutral	73.64%	83.64%	70%	70%	72.73%	70%	76.36%	73.77%
19	Asco	Miedo	84.55%	94.55%	90%	90%	81.82%	90.91%	88.18%	88.57%
20	Asco	Neutral	88.18%	94.55%	90.91%	84.55%	90.91%	87.27%	94.55%	90.13%
21	Miedo	Neutral	82.73%	89.09%	90%	86.36%	87.27%	90.91%	83.64%	87.14%
		Promedio	84.89%	89.91%	87.10%	85.41%	84.37%	86.71%	84.37%	
			5	1	2	4	7	3	6	

Precisión > = 90%	
Precisión > = 80%	
Precisión >= 70%	
Precisión <= 69%	

6.5 Experimentación con el modelo de clasificación generado

Esta sección tiene como fin poder mostrar la capacidad de descripción y reconocimiento que aporta el prototipo generado a través de los métodos “Patrones Binarios Locales” y “Boosting”, así como observar el comportamiento que toma la precisión de sus predicciones al realizar ajustes sobre ciertos parámetros en el mismo.

A continuación se presentan una serie resultados obtenidos a través la variación de ciertos parámetros en este, tanto del módulo de clasificación como de procesamiento. Como referencia se debe aclarar que al mencionar parámetros base se hará alusión a las siguientes configuraciones:

Parámetros de procesamiento

- Radio = 1
- Vecindario = 8
- División imagen LBP = 4 x 3 cuadrículas
- Dimensión estándar = 70 * 90 pixeles
- Recorte rostro = 0.65 Horizontal x 0.85 Vertical

Parámetros de clasificación

- Tipo de Boosting = DISCRETE
- Número de clasificadores débiles = 100

Inicialmente se realizó la variación del **tipo de Boosting** con parámetros base y se pudo observar que aquellos que proveyeron una mejor tasa de reconocimiento mediante la validación cruzada fueron las variantes “DISCRETE” y “GENTLE” para cada una de las seis expresiones analizadas en el proyecto (Enojo, asco, felicidad, tristeza, miedo y asombro). Variantes que alcanzaron una máxima precisión del 89.71% sobre la expresión del asco y una mínima precisión de 80% y 75.71% respectivamente en la expresión de tristeza como se muestra en la gráfica siguiente.

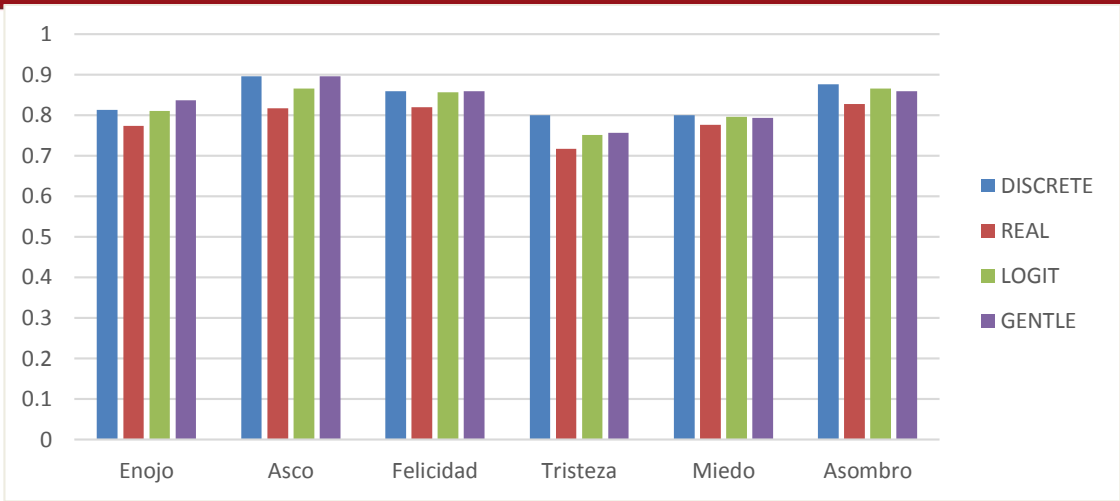

Figura 6.1 *– Tasas promedio de reconocimiento por expresión tras variar el tipo de Boosting usado en el prototipo.*

Posterior a ello, se variaron las divisiones en la imagen LBP del rostro, mediante DISCRETE Adaboost y con el resto de parámetros base, donde se observó que tomando como divisiones las variantes 3X4, 5x7 y 7x9 el reconocimiento mejoró de forma gradual cuando el número de cuadrículas se incrementó de forma intermedia, lo que representa un incremento en el tamaño del vector, pero también se notó que al incrementar más esta división el reconocimiento comenzaba a perder precisión, debido esencialmente a la sobrecarga de información en la caracterización. Por lo que en este caso la división que logro una mejor precisión en promedio fue la de 5 x 7 cuadrículas con un máximo de 91.71% sobre la expresión de asco y un mínimo de 80.85% en la expresión del miedo como se puede apreciar en el siguiente gráfico.

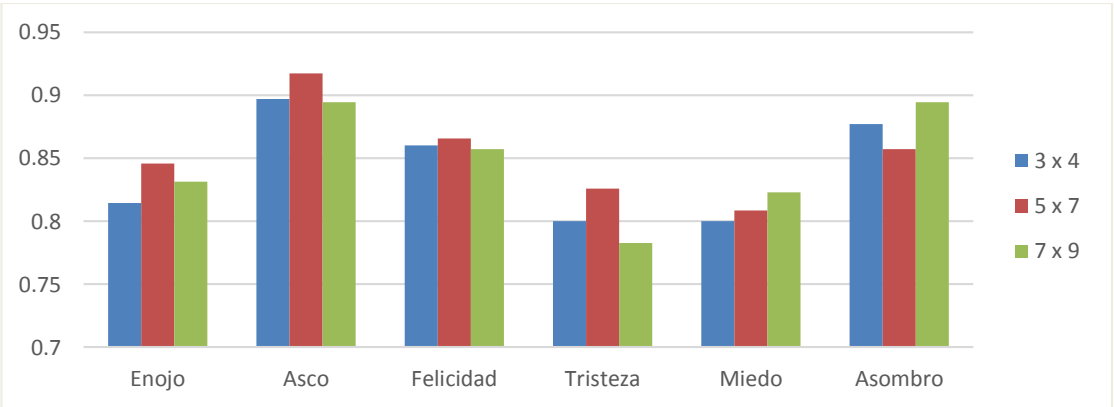

Figura 6.2 *- Tasas promedio de reconocimiento por expresión tras variar el número de cuadrículas generadas para la generación del vector característico.*

Otro análisis dado fue través de la variación del radio usado por el método LBP, mediante la división de (5 x 7) cuadrículas y DISCRETE Adaboost, aquí se observó que estas variaciones en ciertos casos acentúan más el reconocimiento de algunas expresiones pero también disminuyen otras; sin embargo, al analizar los promedios de precisión por radio usado se determinó que el máximo de estos se hallaba al usar r=4 con 86.38% frente al resto de radios r=1, r=2, r=3 y r=5 cuyos valores resultaron 85.33%, 85.66%, 85.85% y 85.14% respectivamente.

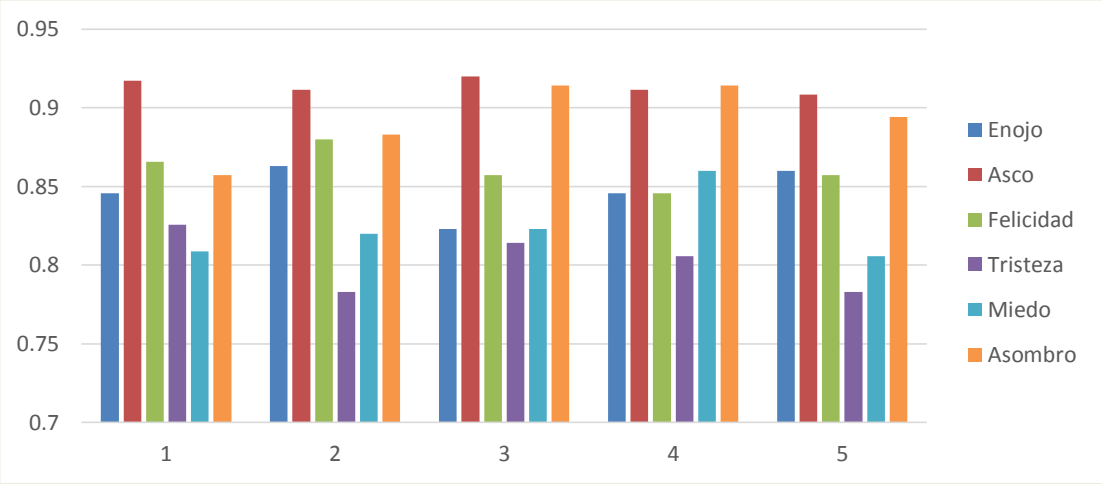

Figura 6.3 *- Tasas promedio de reconocimiento tras variar el radio en el método LBP (1-5).*

Finalmente se analizó el comportamiento del prototipo al variar el número de clasificadores débiles haciendo uso del método DISCRETE Adaboost, una división de 5 x 7 cuadrículas, radios 3 - 4 y con el resto de parámetros base. La tendencia de incremento a esta variación fue el incremento gradual de la precisión en la mayor parte de las expresiones, sin embargo se notó que la experimentación basada en LBP de radio 3 logró superar a la que se basaba en LBP de radio 4 como se puede apreciar en la última gráfica logrando un máximo de 93.14% en la expresión del asombro y un mínimo de 83.14% en la tristeza. La diferencia dada al usar de 200 clasificadores débiles a 300 para este modelo no representó gran cambio por lo que para el estado óptimo del presente prototipo se considera que 200 clasificadores débiles bastan para una mejora relevante del reconocimiento.

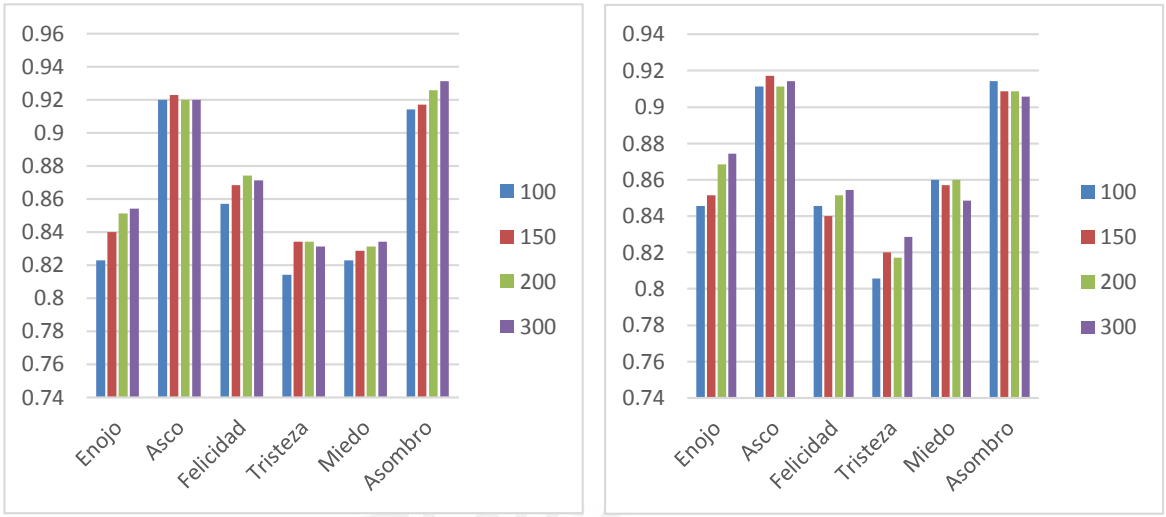

Figura 6.4 *- Tasas promedio de reconocimiento por expresión tras variar el número de clasificadores débiles en el método Boosting. A la izquierda la caracterización se basó en LBP de radio 3, a la derecha LBP de radio 4.*

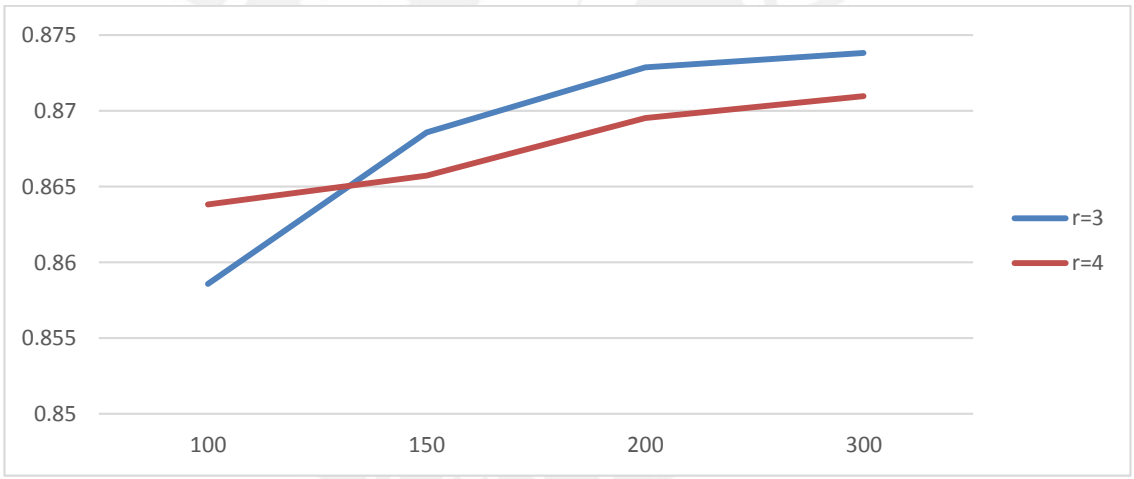

Figura 6.5 *- Tasas promedio de reconocimiento general en función de la variación de los clasificadores.*

CAPITULO 7

7 Conclusiones y trabajos futuros

7.1 Conclusiones

En el presente proyecto se ha estudiado, desarrollado, evaluado y documentado un prototipo para el reconocimiento de expresiones faciales a través de la técnica LBP, a continuación se mencionan las conclusiones obtenidas a lo largo del desarrollo de cada una de las fases del prototipo, así como conclusiones finales respecto a la efectividad lograda a través de la experimentación.

Inicialmente el uso del método desarrollado por Viola-Jones permitió que la primera fase fuera posible en el prototipo al brindar una tasa de detección del rostro alta sobre la base de datos usada, sin embargo por observación se llegó a notar que el método no tolera rotaciones (15 grados como máximo) por lo que para lograr para mejorar un sistema de reconocimiento de este tipo es necesario encontrar o desarrollar un método de detección más robusto que lo contemple.

En la siguiente fase, la caracterización, se pudo confirmar el uso del método LBP para tareas de reconocimiento como la realizada en este proyecto, ya que permite representar y acentuar efectivamente las líneas marcadas por las expresiones en el rostro. También se concluye que para llevar a cabo una mejor discriminación es recomendable el uso de métodos adicionales para reducir la dimensionalidad sobre la caracterización LBP realizada, ya que existe gran parte de la información en esta que es irrelevante y que causa ruido y mayor coste computacional, al tener una mayor dimensión que procesar; por lo que, adicionalmente se menciona, en este caso, que la implementación del método de “patrones uniformes” sobre la caracterización ayudó a la disminución de estos aspectos. Finalmente, se llega a concluir que efectivamente el método logra que el tiempo empleado sea muy bueno debido su coste computacional $(O)^2$ que junto con el bajo coste del método Viola-Jones lo que vuelve viable para su aplicación en tiempo real.

Para la última fase, de clasificación, la implementación inicial realizada en base al algoritmo de aprendizaje SVM brindó gran información respecto a la caracterización dada ya que al presentar bajos índices de reconocimiento indicó que la distribución de la información de las expresiones en el rostro nos es tan fácilmente separable tal

y como se presenta en el vector característico debido esencialmente a que hay secciones esta que son iguales y que no permiten esa diferenciación clara; por lo que al dar uso de algoritmos de aprendizaje basados en Boosting se logró una notoria diferencia debido a que este solo utiliza la información relevante que le permite al modelo realizar una buena distinción entre sus clases llegando a alcanzar porcentaje de acierto hasta de 93.14% para el asombro y con un mínimo aceptable de 84.13% en la tristeza. Por lo que se puede concluir que es un método ideal cuando se conoce a priori que la distribución de características del elemento no presenta muchas diferencias que faciliten la clasificación por la separación de todos sus elementos sino que solo algunas de estas servirán como referencia para esta evaluación.

Finalmente, por experimentación las variaciones realizadas pudieron mostrar que el modelo puede mejorar de forma gradual sus tazas de reconocimiento pero que de igual forma un exceso de estas logra su decremento. Si bien se puede concluir que las variaciones más relevantes se dieron al ajustar el número de divisiones en la imagen LBP y el número de clasificadores débiles no se descarta el radio usado por el método LBP dado a pesar que en las experimentaciones iniciales no mostró gran efecto, al complementar otras variaciones fue relevante su actuación.

7.2 Trabajos futuros

Con el fin de poder extender y mejorar los aspectos que no pudieron ser cubiertos en este proyecto, en esta sección se proponen ciertas líneas de trabajo que pueden ser tomadas para mejorar la performance de un sistema de reconocimiento de expresiones faciales.

- Desarrollo de un modelo de detección facial que considere la pose de los rostros en distintos ángulos o que presente cierta oclusión. En este proyecto se especificó que el prototipo se basa en rostros frontales para lograr un mayor enfoque en la caracterización; sin embargo, un problema común se encuentra en la detección de un rostro en una posición inclinada o con cierta oclusión, por lo que esta línea permitiría una mayor robustez en la detección del rostro en tiempo real y por consiguiente una gran adición para el ingreso de información en un sistema de reconocimiento de expresiones en tiempo real.

- Desarrollo de un modelo multimodal de reconocimiento emocional, que adicione al modelo propuesto el reconocimiento patrones en la tonalidad de la voz como medio para mejorar la predicción del estado emocional. Tal como se mencionó en la problemática si bien una expresión puede decir mucho sobre una probable emoción, esto no quiere decir que lo sea debido a su multi-modalidad; considerar los patrones que se generan al momento de conversar pueden brindar más información al sistema reconocimiento emocional, aumentando su precisión y performance en tiempo real.

- Desarrollo del modelo a través de nuevas técnicas de caracterización con el fin de poder evaluar la efectividad de cada una de ellas, tanto en tiempo de ejecución como en representación de información y de este modo poder conocer cual tiene un mejor performance en tiempo real.

Glosario de Acrónimos

API:	Application Programming Interface
AU:	Action Units
CK:	Cohn–Kanade
CSV:	Comma-Separated Values
CV:	Cross Validation
FACS:	Facial Action Coding System
LBP:	Local Binary Pattern
SVM:	Support Vector Machine
UP:	Uniform Patterns
WEKA:	Waikato Environment for Knowledge Analysis

Referencias bibliográficas

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*: MIT press, 2012.

[2] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 2037-2041, 2006.

[3] X. Zhao and S. Zhang, "Facial expression recognition based on local binary patterns and least squares support vector machines," in *Advances in Electronic Engineering, Communication and Management Vol. 2*, ed: Springer, 2012, pp. 707-712.

[4] P. Watzlawick, H. Beavin, and D. D. Jackson, "Teoría de la comunicación," *Tiempo contemporáneo,* 1971.

[5] F. Davis, *El lenguaje de los gestos: la comunicación no verbal*: Emecé, 1982.

[6] G. R. Wainwright, *El lenguaje del cuerpo*: Ediciones Pirámide, 1998.

[7] P. Ekman and H. Oster, "Expresión Facial de la Emoción," *Annual Review of Psychology,* vol. 30, pp. 527-554, 1979.

[8] J. J. Prinz, *Gut Reactions : A Perceptual Theory of Emotion*. Cary, NC, USA: Oxford University Press, USA, 2004.

[9] M. Chóliz, "Psicología de la emoción: el proceso emocional," *Universidad de Valencia, España,* 2005.

[10] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, "Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders," *Journal of neuroscience methods,* vol. 200, pp. 237-256, 2011.

[11] J. Abascal and R. Moriyón, "Tendencias en interacción Persona-Computador," *Revista Iberoamericana de Inteligencia Artificial,* vol. 6, 2002.

[12] D. H. Ballard and C. M. Brown, "Computer vision, 1982," *Prenice-Hall, Englewood Cliffs, NJ*.

[13] P. Ekman, *Facial Action Coding System: Manual*: Consulting Psychologists Press, 1993.

[14] Y. Tian, T. Kanade, and J. F. Cohn, "Facial expression recognition," in *Handbook of face recognition*, ed: Springer, 2011, pp. 487-519.

[15] A. Metallinou, S. Lee, and S. Narayanan, "Audio-visual emotion recognition using gaussian mixture models for face and voice," in *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, 2008, pp. 250-257.

[16] K. Lu and X. Zhang, "Facial expression recognition from image sequences based on feature points and canonical correlations," in *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, 2010, pp. 219-223.

[17] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2729-2736.

[18] M. D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister, "Real-time face pose estimation from single range images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.

[19] C.-C. Hsieh and M.-K. Jiang, "A Facial Expression Classification System Based on Active Shape Model and Support Vector Machine," in *Computer Science and Society (ISCCS), 2011 International Symposium on*, 2011, pp. 311-314.

[20] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*: O'Reilly Media, Inc., 2008.

[21] Microsoft. (2014, 20/08). *Introducción a Visual Studio*. Available: http://msdn.microsoft.com/es-es/library/52f3sw5c(v=vs.90).aspx

[22] Microsoft. (2014). *Visual Studio*. Available: www.visualstudio.com

[23] Q. Project. (2014, 05/11). *Qt Framework*. Available: http://qt-project.org/

[24] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision,* vol. 57, pp. 137-154, 2004.

[25] D. o. C. S. a. Engineering. (2014). *Local Binary Pattern (LBP)*. Available: http://www.cse.oulu.fi/CMV/Research/LBP

[26] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 41, pp. 765-781, 2011.

[27] O. Ivanciuc, "Applications of support vector machines in chemistry," *Reviews in computational chemistry,* vol. 23, p. 291, 2007.

[28] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*: Cambridge university press, 2000.

[29] V. N. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on,* vol. 10, pp. 988-999, 1999.

[30] OpenCV.org. (2014). *Introduction to Support Vector Machines*. Available:http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[31] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *ICML*, 1996, pp. 148-156.

[32] B. Efron and R. Tibshirani, "Improvements on cross-validation: the 632+ bootstrap method," *Journal of the American Statistical Association,* vol. 92, pp. 548-560, 1997.

[33] J. Shao, "Linear model selection by cross-validation," *Journal of the American statistical Association,* vol. 88, pp. 486-494, 1993.

[34] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of personality and social psychology,* vol. 17, p. 124, 1971.

[35] B. V. Kumar, "Face expression recognition and analysis: the state of the art," *Course Paper, Visual Interfaces to Computer,* 2009.

[36] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern Recognition Letters,* vol. 33, pp. 431-437, 2012.

[37] G. Beekman, *Introducción a la computación*: Pearson Educación, 1999.

[38] A. Dix, *Human-computer interaction*: Springer, 2009.

[39] C. Darwin, *The expression of the emotions in man and animals*: Oxford University Press, 1998.

[40] H. A. Elfenbein and N. Ambady, "Universals and cultural differences in recognizing emotions," *Current Directions in Psychological Science,* vol. 12, pp. 159-164, 2003.

[41] P. E. Group. (2014). *FACS*.
Available: http://www.paulekman.com/

[42] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 23, pp. 97-115, 2001.

[43] A. Hanson and R. Edward, *Computer vision systems*: Elsevier, 1978.

[44] J. F. Vélez Serrano, *Visión por computador*. España: Dykinson, 2004.

[45] Sony. (2014). *Smile shutter*. Available:http://www.sony.com.pe/corporate/PE/tecnologias/Camaras/Camaras-Digitales/smile-shutter-.html; http://docs.esupport.sony.com/dvimag/DSCH70_guide/eng/contents/03/04/03/03.html

[46] Noldus. (2014). *FaceReader 5.1 - Technical specifications*. Available:http://www.noldus.com/human-behavior-research/products/facereader

[47] Emotient. (2014). *Emotient API*. Available: http://www.emotient.com/products#FACETSDK

[48] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 94-101.

[49] OpenCV. (2014, Septiembre). *OpenCV Tutorials*. Available: http://docs.opencv.org/doc/tutorials/

[50] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 971-987, 2002.

[51] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 46-53.