PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

DETECCIÓN DE EMOCIONES DEL USUARIO

ERASMO JESÚS MARÍN GONZÁLEZ

Profesor Guía: **Cristian Alexandru Rusu**

Profesor Co-referente: **Silvana Roncagliolo De La Horra**

Carrera: **Ingeniería Civil Informática**

DICIEMBRE 2014

Este trabajo es dedicado a mi familia, a mis queridos padres y hermanas que siempre creyeron en mí, cuyo ejemplo de esfuerzo y sacrificio me impulsaron siempre a seguir sus pasos y dar lo mejor a pesar de las dificultades. Agradezco también a mi profesor guía y a todos aquellos que brindaron su ayuda, compañeros y amigos, los tendré siempre en mi corazón.

Resumen

El rol del reconocimiento automático de emociones está creciendo de forma continua actualmente. A medida que los computadores se vuelven más y más sofisticados, ya sea a nivel profesional o social, se vuelve más importante que estas sean capaces de interactuar de forma natural. Se considera que las máquinas deben incluir este tipo de inteligencia de forma de reconocer el estado afectivo dado ciertas señales sicológicas.

Diversos autores tratan este tema, y la investigación en este campo ha sido numerosa en la última década con distintos niveles de éxito. Ha habido diversos esfuerzos desde distintos puntos de vista. Se han descrito técnicas para el análisis de la voz humana o de las expresiones faciales por ejemplo, sin embargo la aplicación de estas tecnologías en sistemas reales sigue siendo escasa.

En este trabajo se busca implementar un sistema que permita detectar el estado emocional del usuario a partir de las expresiones faciales y validarlo. La validación se realiza con una base de datos creada para este fin.

Palabras claves: Reconocimiento de Expresiones Faciales, Reconocimiento de Emociones, Análisis de Expresiones Faciales, Interacción Afectiva.

Abstract

The role of automatic emotion recognition is growing continuously. As computers become more and more sophisticated, either at professional or social level, it becomes more important that they are able to interact naturally. It is considered that the machines should include this kind of intelligence to recognize the affective state as certain psychological signals.

Several authors address this issue, and research in this field has been numerous in the last decade with varying degrees of success. There have been various efforts from different points of view. Techniques that involve the analysis of the human voice or facial expression has been described, but the application of these technologies in real systems is still scarce.

This work describes the implementation of a system prototype to detect the user's emotional state from facial expressions. A method based in Principal Component Analysis was employed. The prototype was validated using a faces database created for this purpose.

Keywords: Facial Expressions Recognition, Emotion Recognition, Analysis of Facial Expressions, Affective Interaction.

Índice

Lista de figuras

Lista de tablas

1. Introducción y descripción del problema

El rol del reconocimiento automático de emociones está creciendo de forma continua actualmente. Esto se debe a que se ha aceptado la importancia que tiene la reacción a los estados afectivos del usuario en la interacción persona-computador.

A medida que los computadores se vuelven más y más sofisticadas, ya sea a nivel profesional o social, se vuelve más importante que estas sean capaz de interactuar de forma natural, o sea, de forma similar a como se interactúa con otros agentes humanos. La característica más importante de la interacción humana que garantiza que el proceso se haga de forma natural, es el proceso por el cual podemos inferir el estado emocional de otros. Esto permite ajustar los patrones de comportamiento y respuestas, optimizando el proceso interactivo, tal cual se señala en [1]**.**

La interacción de la persona con el computador puede ser mejorada de gran manera si se toma en cuenta el estado emocional del ser humano, haciéndola más cercana y natural; elementos ausentes en la gran mayoría de los sistemas actuales. Diversos autores tratan este tema, y la investigación en este campo ha sido numerosa en la última década. En el trabajo de *Picard et all* [2] se dice que el reconocimiento de los estados afectivos del usuario es un problema muy importante en el ámbito de la interacción humano-computador. En este mismo trabajo, se dice que "se ha argumentado que la inteligencia emocional humana es incluso más importante que la inteligencia matemática o verbal". Es por este motivo, que se considera que las máquinas deben incluir este tipo de inteligencia de forma de reconocer el estado afectivo dado ciertas señales sicológicas.

Respecto a este tema, ha habido diversos esfuerzos desde distintos puntos de vista. Se han descrito técnicas para el análisis de la voz humana o de las expresiones faciales por ejemplo, sin embargo la aplicación de estas tecnologías en sistemas reales sigue siendo escasa.

Uno de los trabajos pioneros en este campo, *Ekman and Friesen* (1975) [3] sugieren que existen seis expresiones faciales prototípicas básicas reconocidas universalmente. Estas son: enojo, disgusto, miedo, felicidad, tristeza y sorpresa. El reconocimiento automatizado de estos seis estados básicos es el primer paso para implementar una solución, que a diferencia de otras, es universal, ya que es común a todos los seres humanos.

El principal desafío o problema en este ámbito no es sólo la búsqueda de una solución algorítmica, sino que buscar la forma de mejorar la interacción persona-computador con estas nuevas tecnologías. Se busca encontrar una solución a este problema. Para esto se usarán técnicas de análisis de expresiones faciales en tiempo real y análisis de los cambios en los estados emocionales del usuario. Luego, como objetivo secundario, se hará la integración en un sistema de recomendaciones híbrido basado en contexto de contenido multimedia, de forma que el sistema pueda recomendar contenido usando como parámetro la emoción detectada.

2. Definición de objetivos

2.1 Objetivo general

Desarrollar un prototipo de sistema que permita la detección de emociones del usuario analizando las expresiones faciales en tiempo real.

2.2 Objetivos específicos

1. Establecer los principales conceptos y fundamentos involucrados en la detección facial, de expresiones faciales y de detección de emociones.
2. Desarrollar una herramienta que permita establecer los estados afectivos del usuario a partir de las expresiones faciales en tiempo real.
3. Validar la herramienta a través de pruebas con datos de prueba y mediante pruebas con usuarios reales.

2.3 Plan de trabajo

Para este plan de trabajo, mostrado en la tabla 2.1, se especifican distintas tareas relativas a la investigación, al desarrollo de la herramienta y a su validación. El desarrollo se inició con dos estados afectivos básicos, y se le agregaron posteriormente el resto, aumentando su complejidad de manera progresiva e incremental. Vale mencionar que durante el desarrollo de los prototipos, se experimentó con diversos enfoques, que se describirán también.

Tabla 2.1: Plan de Trabajo

Actividad	Inicio	Fin
Análisis de las tecnologías y métodos disponibles a utilizar	Marzo del 2014	Marzo del 2014
Reconocimiento del rostro en imágenes estáticas	Marzo del 2014	Marzo del 2014
Desarrollo de prototipo funcional básico que detecte estados afectivos básicos (feliz, sorpresa) en imágenes estáticas	Abril del 2014	Junio del 2014
Recolección y creación de una base de datos de pruebas	Mayo del 2014	Julio del 2014
Reconocimiento de estados afectivos a través de una cámara (Felicidad, Enojo, Neutral, Tristeza, Sorpresa y Disgusto)	Junio del 2014	Noviembre del 2014
Prueba con usuarios	Octubre del 2014	Noviembre del 2014
Pruebas finales	Noviembre del 2014	Noviembre del 2014

3. Computación afectiva

Una interacción persona-computador efectiva es aquella en la cual la información emocional es comunicada por el usuario en una manera natural y cómoda, reconocida por la computadora y luego, usada para mejorar la interacción [4]. Antes que el computador pueda adaptar su interacción para servir mejor a un usuario, es necesario que la retroalimentación proveída por el usuario, sea asociado con acciones de la máquina, de forma de identificar si una acción del computador agradó o no al usuario, e identificar elementos que podrían haber frustrado al usuario.

Desde el comienzo de la computación afectiva, los investigadores han buscado formas de permitir que un computador pueda ser capaz de entender y responder adecuadamente a las emociones del usuario. Se han construido prototipos que han sido capaces de detectar y clasificar distintos aspectos de las expresiones emocionales humanas, respondiendo a la frustración de los usuarios, y demostrando habilidades emocionales humanas, tales como la empatía, la escucha, y la simpatía, de forma de satisfacer en parte las necesidades emocionales del usuario. El tener una idea de estas necesidades es el primer paso para poder entender el rol que podría tener un computador en la satisfacción de estas necesidades. A continuación analizamos estas necesidades.

3.1 Necesidades emocionales del ser humano

Cuando hablamos de las necesidades emocionales del ser humano, lo hacemos desde el área de la sicología y de otras ciencias sociales. *Picard et al.* [5] señalan en su investigación, que existen dos tipos de necesidades emocionales. Las primeras están relacionadas con la inteligencia emocional, a la cual llama, necesidad de habilidades emocionales. El segundo grupo de necesidades está compuesto de las necesidades de experiencias emocionales.

3.1.1 Necesidades de habilidades emocionales

Se refiere a un grupo de habilidades necesarias para entender y reaccionar a las emociones de otros y de uno mismo. Estas habilidades son importantes en toda persona humana, y la presencia o carencia de estas habilidades puede tener repercusiones importantes en la persona, tal como dice *Picard*, puede ser la diferencia entre una vida de éxito o de miseria, ya que estas habilidades permiten mantener relaciones sanas y muchas veces significan éxito personal y laboral.

Ejemplos de estas son:

- **Autoconciencia emocional:** La habilidad de evaluar y expresar de forma precisa lo que uno siente.

- **Gestionar las emociones:** Se refiere a un mecanismo de autocontrol de forma de regular los sentimientos y emociones de forma que tengan cabida en una cultura y un contexto en particular.

- **Automotivación:** Aprovechamiento de las emociones en función de un objetivo. Un ejemplo de esto puede ser una recompensa tras cumplir una tarea complicada.

- **Percepción afectiva:** La habilidad de poder entender lo que otro está sintiendo mediante la observación de sus expresiones tanto verbales como no verbales, y el razonamiento asociado a su situación.

- **Empatía:** Es la apreciación de lo que otro siente, y comunicar este entendimiento de forma precisa y adecuada a la persona.

3.1.2 Necesidades de experiencias emocionales

Las necesidades de experiencias tienden a ser sociales por naturaleza, ya que son satisfechas mediante la interacción con otras personas. Cuando estas necesidades no son satisfechas, pueden afectar la calidad de vida degradándola.

Ejemplos de estas necesidades son:

- Necesidad de atención, la cual es muy fuerte y constante durante la niñez disminuyendo y moderándose en la adultez, sin embargo, no desaparece por completo.

- El sentir que el estado emocional de uno es entendido por el resto. Esto es particularmente más importante si el estado emocional es intenso.

- El sentir que las respuestas emocionales de uno son aceptadas por otros.

- El sentir que las experiencias y respuestas emocionales de uno se consideran apropiadas y adecuadas para una situación o contexto dado.

- El sentirse conectado a otros.

- Necesidad de compañerismo.

- Necesidad de seguridad.

3.2 Computadores y necesidades emocionales de los usuarios

La primera pregunta que puede surgir al respecto del rol que podrían jugar las máquinas en un futuro es si la interacción emocional humano - máquina puede seguir la misma lógica y las mismas dinámicas que la relación humano – humano. O más bien, antes que eso, si las necesidades humanas pueden ser satisfechas solamente entre humanos o pueden haber otros tipos de interacción que si satisfagan estas necesidades. Una respuesta a la segunda pregunta se puede dar con un ejemplo.

No hay duda alguna del rol que juegan las mascotas en la vida del ser humano. Muchas de estas mascotas, tales como los perros, tienen habilidades emocionales que le permiten interactuar con los seres humanos. *Picard et al.* [5] ejemplifica esto con el escenario mostrado en la figura 3.1.

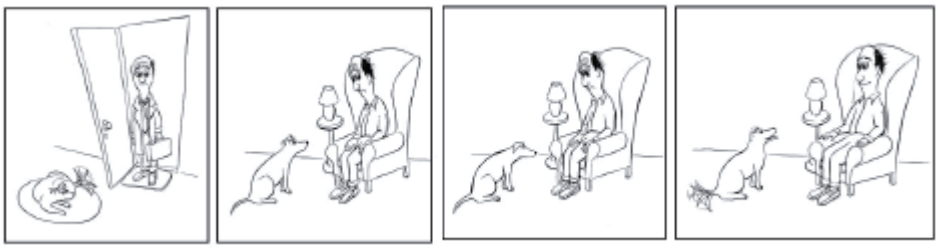


Figura 3.1: Interacción entre un perro y su amo según Picard et al.

En la secuencia de la figura 3.1 se muestra a un hombre llegando a su hogar tras un día de trabajo, con un aspecto claramente de desánimo, tal vez tras haber tenido un mal día. El perro se levanta y siente empatía por su amo, expresándolo con su cabeza gacha y pensando quizás "algo no está bien". Posterior a esto, el perro intenta alegrar a su amo moviendo la cola.

El perro no parece poseer una gran inteligencia, sin embargo, es capaz de entender lo que su amo siente y concretar una interacción exitosa con él en menos tiempo de lo que toma describir el acto, generando un impacto positivo inmediato en su amo. Eso no quiere decir que el objetivo del perro sea sólo alegrar a su amo, el perro podría estar perfectamente pensando "espero que me alimente".

Entonces, si las reglas de interacción entre un perro y un humano son similares a la que se da entre dos seres humanos, ¿podría esto extenderse a los computadores? Para responder esto, se considera la referencia a "*The Media Equation Theory*" de *Reeves y Nass* (1996) [6]. Esta teoría, la cual ha sido corroborada por varias investigaciones, indica que las reglas de interacción entre humanos también aplican entre humanos y computadores, ya que la interacción es inherentemente natural y social. También sostiene que las personas tienden a tratar a los computadores tal como si se tratara de gente real o lugares reales. Esto significa que la gente tiende a responder a los computadores tal como lo harían con otras personas o lugares, por ejemplo, atribuyéndole características de personalidad (tal como

humor, agresividad o incluso género) o respondiendo de forma educada o cooperativa. Este tipo de reacción es automática, inevitable y ocurre más frecuentemente de lo que la gente se da cuenta.

"*The Media Equation Theory*" tiene implicancias profundas para la interacción persona computador y cómo se piensa el diseño de sistemas afectivos. Con esta teoría en mente, los diseñadores deberían considerar la interacción humano-humano como la analogía más cercana a la interacción humano-computador para su evaluación. Sin embargo, esto no significa que los computadores deban copiar la forma en que dos seres humanos interactúan, como se verá a continuación.

3.3 Imitando la interacción humana en la interacción humano-computador

Algunos podrían sostener que la interacción humano-humano es la más compleja y por tanto superior, por lo que los sistemas afectivos deberían copiar este comportamiento, sin embargo, esto no es conveniente ni práctico. Una interacción humano-computador en la cual el computador responde como humano puede ser irritante. Que un sistema se comporte como humano no lo hace mejor. Esto es un fenómeno conocido, un ejemplo es la interacción de humanos con humanoides realistas, la cual resulta muchas veces perturbadora. Este fenómeno es conocido como el Valle Inquietante. A medida que el robot humanoide se hace más realista, se vuelve más incómodo, por lo que la respuesta positiva de las personas disminuye. Cuando se grafica este fenómeno, se origina un valle; este es, el Valle Inquietante, ejemplificado en la figura 3.2.

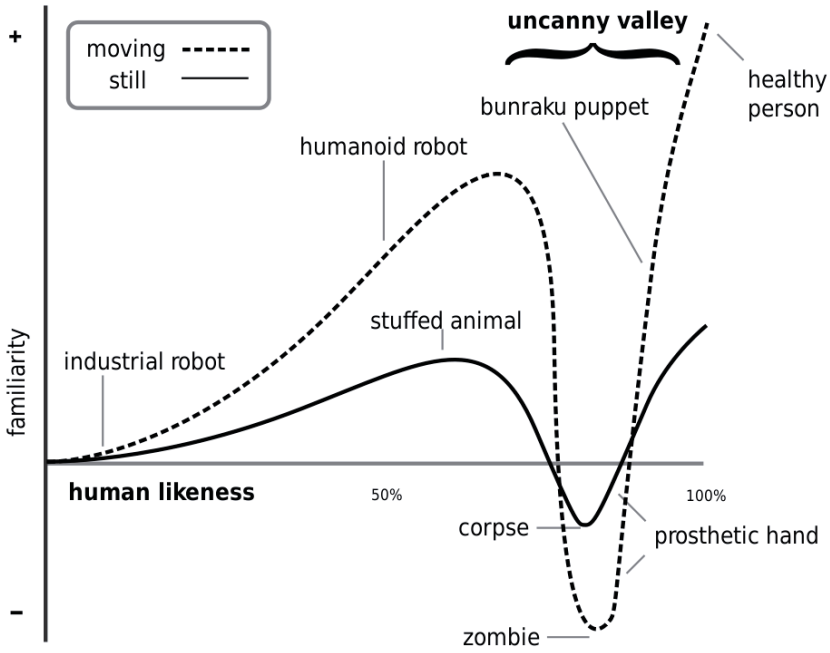


Figura 3.2: Gráfico del Valle Incómodo

La tarea de los computadores es simplificar la vida de las personas y extender sus capacidades, no sólo copiarlas. Un ejemplo de esto puede ser los algoritmos de visión por computador, o “computer vision”. Estos algoritmos no permiten a los computadores ver tal cual lo hace un ser humano, sin embargo, pueden ser diseñados para “ver” cosas que nosotros no podemos ver, tal como fallas en un producto en una línea de ensamblaje.

4. Trabajo relacionado

Las expresiones faciales suministran importantes pistas a cerca de las emociones. Debido a esto, se han desarrollado diversos métodos para clasificar los estados afectivos del ser humano. Estos métodos generalmente se basan en la posición espacial local o el desplazamiento de puntos específicos y regiones de la cara, o en algoritmos de machine learning.

Existen varios estudios sobre el reconocimiento de las emociones basado en el uso de expresiones faciales. Algunos de estos son unimodales, es decir, utilizando sólo el reconocimiento de expresiones faciales, otros tratan de utilizar más de un tipo de entrada, como el procesamiento del habla, el movimiento corporal, gestos con las manos, etc.

Busso et al. (2004) [7] idearon un sistema para el reconocimiento de emociones bimodal, basado tanto en el análisis de reconocimiento de expresiones faciales como el timbre de voz. En este trabajo se analizan las fortalezas y limitaciones de los sistemas unimodales, es decir aquellos que sólo ocupan expresiones faciales o información acústica, y discute dos enfoques usados para fusionar las dos modalidades. Se utilizó un set de datos construido a partir de grabaciones de una actriz. De dicho set de datos, se clasificaron cuatro emociones: tristeza, enojo, felicidad y neutral.

El sistema basado en el habla, utilizó características prosódicas a nivel global ("global-level prosodic features"), tales como estadísticas derivadas del tono del sonido o la intensidad. Por otro lado, el sistema basado en expresiones faciales, se utilizan marcadores faciales, y luego se utiliza Análisis de Componentes Principales (PCA) para reducir el número de características por cada frame. Los marcadores del área de la boca no se consideran, debido a que estos podrían confundir al clasificador haciendo que la expresión se clasifique como felicidad mientras la persona habla.

En este trabajo se concluye que ambos métodos son complementarios y que la precisión y robustez aumenta de forma medible cuando ambos métodos se combinan. El clasificador acústico alcanzó un 64% de precisión, mientras que el clasificador basado en expresiones faciales logró un 85% de precisión. Al combinarse ambos métodos, la precisión aumentó a un 89%. El autor recalca que estos resultados se obtuvieron con un único sujeto de pruebas, por lo que se necesitan más experimentos.

Fraponogos y Taylor (2005) [1] abordan los fundamentos teóricos utilizados en el proyecto ERMIS (emotionally rich man-machine intelligent system), un proyecto colaborativo para crear un sistema bimodal de reconocimiento de emociones del cual formaron parte. En este trabajo, dicen que hay básicamente dos enfoques para la detección de emociones en base a las expresiones faciales. En la primera, el análisis de la expresión facial se realiza en forma estadística en relación a los extremos de una expresión, lo cual requiere encontrar pistas en las arrugas, las posiciones y formas de la cara para ayudar a inferir el estado emocional de una persona, sin embargo, pocos trabajos basados en esta técnica tuvieron éxito, debido principalmente gran gasto computacional que requiere. En la segunda, más extendida y utilizada, el análisis está orientado gestos, para lo cual se extraen

varios *frames* sucesivos de la expresión facial, los cuales son analizados para encontrar variaciones, y de esta forma, se asignan a estados emocionales previamente definidos.

Ioannou et al. (2005) [9] aborda la problemática de la robustez de los clasificadores al extraer características de diferentes usuarios. En este trabajo, desarrollaron un sistema que extrae parámetros de animación facial (PAF) de la que se crea una red neuronal que utiliza lógica difusa con normas basadas en el análisis de los PAF y permite al computador aprender y adaptar las características de las expresiones faciales específicas de cada usuario. El sistema logra un rendimiento del 78%.

Hammal et al. (2005) [10] construyeron un sistema de clasificación basado en la Teoría de la Creencia, la cual puede cuantificar el nivel de fiabilidad de reconocimiento de una expresión y la intensidad de valencia con el usuario demuestra cada uno expresión. El sistema desarrollado detecta los bordes de las expresiones faciales de forma automática, y extrae cinco características basadas en distancias. Estas características luego forma parte de un vector de características que el clasificador utiliza para clasificar entre 4 emociones (felicidad, sorpresa, disgusto y neutral), considerando un quinto estado como desconocido.

Para el análisis de la emoción, este trabajo propone un sistema similar al MPEG-4, en el cual diferentes estados se combinan para formar una expresión. Debido a que sólo se utilizan cinco características, pueden ocurrir estados de indecisión o duda. Características adicionales son añadidas cuando el clasificador se encuentra en este estado.

Finalmente, se utiliza un clasificador bayesiano junto a Modelos Ocultos de Markov (HMM) con el fin de comparar el clasificador basado en la teoría de la creencia. Se utiliza una distribución gaussiana, asumiendo independencia entre las clases y un HMM de 5 estados, uno para cada emoción. Se calcula un puntaje de probabilidad, y la clase con puntuación más alta es elegida. Finalmente, se llega a la conclusión de que el clasificador que utiliza la Teoría de la Creencia logra mejores resultados que el Bayesiano y que HMM.

Ioannou et al. (2007) [11] presentaron un sistema para el reconocimiento de expresiones faciales basado en una fusión de diferentes técnicas, basadas en el principio de que no se puede controlar el medio ambiente humano en términos de luz o de la calidad del color, así como la expresividad y movimiento humano, permitiendo el reconocimiento de expresiones faciales y emociones en entornos difíciles o inusuales. El sistema detecta el rostro y corrige su posición, luego genera máscaras para los ojos, boca, cejas y detecta la nariz. Se hace la extracción de puntos característicos junto a una evaluación antropométrica. A partir de los dos pasos anteriores el sistema extrae loas parámetros de animación facial (FAP) realizando una comparación con los parámetros faciales para la expresión neutral, y finalmente efectúa el reconocimiento de la emoción. El método es bastante similar al mostrado en [9].

Castellano et al. (2008) [12] usaron un enfoque multimodal para el reconocimiento de ocho emocione, utilizando expresiones faciales, habla, gestos y movimientos corporales. El uso de clasificadores bayesianos para cada modalidad y luego, unión de datos a nivel de la toma de decisión que fue capaz de aumentar en un 10 % la potencia de un reconocimiento general de las emociones. La recolección de datos multimodales constó de

10 sujetos actuando la expresión, habla y gestos asociados de acuerdo a una pauta. Se realizó extracción de características de la cara, el cuerpo y del habla. Para la cara se utilizaron los FAPs, mientras que para el cuerpo se utilizó la librería EyesWeb. Para la voz se utilizó la intensidad, el tono, Coeficientes Cepstrales en las Frecuencias de Mel (MFCC), Bandas Espectrales de Barc, características de segmentos sonoros y duración de las pausas.

Se construyó un clasificador bayesiano para cada modalidad (rostro, gestos y habla). Los datos fueron normalizados y las características fueron discretizadas para reducir la complejidad de aprendizaje. El clasificador basado en expresiones faciales logró un rendimiento del 48,3%, mientras que el basado en gestos alcanzó un total de 67,1%. Finalmente el basado en el habla logró un total de 57.1%.

Utilizando unión de datos a nivel de las características permitió obtener un rendimiento del 78,3%, mientras que al hacerlo a nivel de decisión se obtuvo un resultado no tan bueno alcanzando un rendimiento del 74,6%.

Anisha Halder et all. (2011) [24] propusieron un método para reconocimiento de emociones basado en el posicionamiento de puntos de acción facial y Análisis de Componentes Principales. Dichos puntos se posicionaron manualmente utilizando marcadores de colores. Se creó una matriz de características, consistente en todas las combinaciones posibles entre las distancias euclidianas entre todos los puntos. Estas distancias pasaron a formar parte del vector de características, para luego aplicar Análisis de Componentes Principales a dichos puntos.

Se utilizaron un total de 36 puntos, generando una matriz de 36x36, y se seleccionaron 10 sujetos, de los cuales se extrajeron fotografías en 6 emociones distintas: neutral, enojo, disgusto, miedo, felicidad y sorpresa. La comparación de PCA frente a otros algoritmos, tales como SVM, Back-propagation, MLP y RBF, utilizando un test de Friedman modificado, arrojó que PCA era el algoritmo con mayor precisión, con un 92.25%, mientras que el test de McNemar arrojó que el algoritmo propuesto supera a los otros algoritmos con los que se comparó.

Gosavi and Khot (2013) [26] utilizan Análisis de Componentes Principales para la extracción de características y un clasificador basado en la distancia euclidiana entre los puntos proyectados. También se desarrolló un algoritmo para el cálculo de la intensidad de la emoción basado en la distancia euclidiana entre el promedio de las caras neutrales proyectadas y la imagen a clasificar.

Para el experimento se utilizó la Base de Datos de Expresiones Faciales Femeninas Japonesa (JAFFE), la cual contiene un total de 213 imágenes de 7 expresiones faciales (felicidad, disgusto, neutral, tristeza, enojo, sorpresa y miedo), incluyendo neutral, proveniente de 10 modelos japonesas. Se creó un set de entrenamiento y de prueba con 70 imágenes cada una, y se calculó la precisión y la exactitud. El sistema logró una precisión del 72,82%, mientras que la exactitud llegó al 91,63%. La tasa de reconocimiento promedio fue de un 67,14%.

En [27] *Gosavi and Khot* continúan su trabajo e implementan una técnica híbrida de reconocimiento de expresiones faciales utilizando Análisis de Componentes Principales junto a Descomposición en Valores Singulares (SVD). A diferencia de su trabajo anterior, crearon una base de datos generada por ellos. Se implementó el reconocimiento de 5 emociones: felicidad, disgusto, tristeza, enojo y sorpresa junto a neutral, logrando una precisión del 89,7% y una tasa de reconocimiento promedio del 65,42%.

Milborrow and Nicolls (2014) [15] desarrollaron un método para posicionar puntos en rostros frontales usando descriptores SIFT en un ASM modificado, demostrando mejores resultados en comparación a otras técnicas existentes. El código fuente se distribuye de manera gratuita en modo de una librería llamada STASM.

5. Detección y análisis del rostro

El reconocimiento de expresiones faciales se debe realizar en dos pasos. El primero, consiste en la detección y normalización de la imagen de la cara. La segunda parte, consiste en el reconocimiento de la expresión recibiendo como entrada la imagen de la cara ya normalizada. La detección de rostro es el proceso por el cual se encuentran las coordenadas de la cara en una imagen. El proceso de normalización posterior, es el proceso por el cual se procesa la imagen de la cara detectada para su correcta detección.

En este trabajo se utiliza un método para la detección de las expresiones faciales, que utiliza Análisis de Componentes Principales para la clasificación. La detección de rostro se efectúa utilizando el algoritmo de *Viola/Jones*. El proceso de entrenamiento es supervisado, esto quiere decir que el algoritmo requiere una preparación de los datos de entrenamiento mediante ejemplos para aprender. El clasificador busca la imagen más parecida en el set de entrenamiento y realiza la predicción de acuerdo a eso.

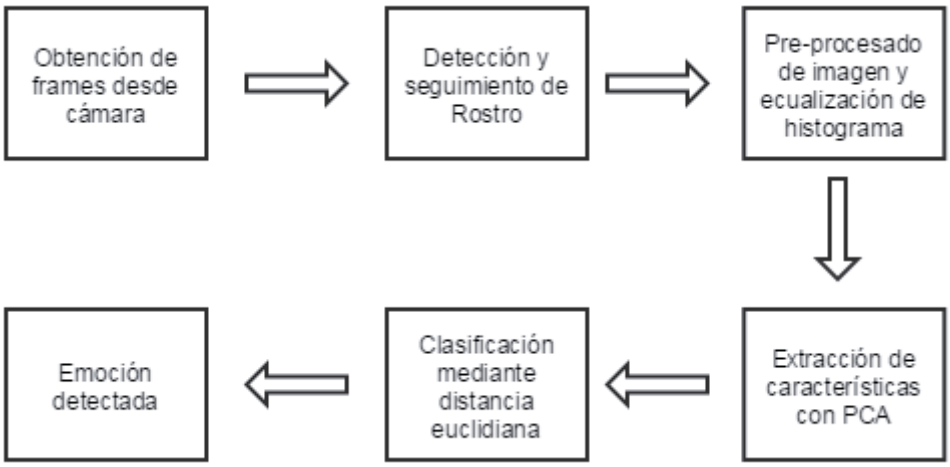


Figura 5.1: Proceso de detección de emoción propuesto

El Análisis de Componentes Principales es una técnica conocida en el procesamiento de imágenes. Se ha utilizado ampliamente en el reconocimiento facial y en otras áreas del conocimiento, tal como la medicina y la neurociencia. En este caso se utiliza la distancia euclidiana para medir la cercanía entre dos imágenes. El proceso general se muestra en el diagrama de la figura 5.1.

Para el desarrollo se ha seleccionado la librería OpenCV. Entre los motivos de esta elección está que se trata de una librería open-source liberada bajo la licencia BSD. OpenCV es desarrollada por Intel, es multiplataforma, y dispone de interfaces para su uso con distintos lenguajes de programación (C, C++, Python y Java). Además, es ampliamente usada tanto en sistemas comerciales como en investigación, debido a que su licencia lo permite.

OpenCV dispone de más de 500 funciones, implementando una gran cantidad de algoritmos en el área de la computación visual. Permite manipular imágenes a bajo y alto nivel. Dispone de una comunidad de más de 47 mil personas y ha alcanzado la no despreciable suma de 7 millones de descargas. OpenCV incluye algoritmos de detección facial y de machine learning, como Haar Cascades, AdaBoost y SVM, y gran cantidad de funciones que permite la manipulación de matrices para la implementación de estos algoritmos.

Para la obtención de *frames* desde la cámara y la detección del rostro se utilizan librerías que facilitan este proceso. La extracción de las características por otro lado debe ser implementada. Esta extracción sirve como *input* para el clasificador. La técnica de clasificación debió ser desarrollada.

Para el proceso de entrenamiento, se creó una base de datos propia. La base de datos consta de fotografías tomadas a 18 alumnos de la escuela de Ingeniería Informática de la Pontificia Universidad Católica de Valparaíso, tanto de pregrado como de postgrado. En total se tomaron 8 fotografías por emoción. Ya que el sistema implementa 6 emociones, la base de datos completa consta con un total de 864 fotografías ordenadas y clasificadas.

El mayor esfuerzo de implementación está en la fase de integración de los distintos módulos que conformarán el sistema, en el desarrollo y entrenamiento del clasificador, la preparación y programación de pruebas y en la creación de una interfaz que permita acceder al predictor desde otra aplicación por vía de comandos.

5.1 Detección del rostro

En el presente trabajo se utiliza el algoritmo de *Viola/Jones* para la detección del rostro. Este algoritmo es una técnica que utiliza clasificadores *Haar* en cascadas. Permite la detección de objetos en tiempo real debido a su gran rapidez. Fue propuesto en 2001 por *Paul Viola y Michael Jones*. Es un enfoque basado en machine learning. Consiste en el entrenamiento de una función cascada con ejemplos positivos y negativos del objeto que se desea detectar. El entrenamiento es lento, pero los resultados son muy buenos. Este algoritmo se utiliza para la detección del rostro, el primer paso antes de aplicar PCA.

Cuando se ocupa en detección de caras, el algoritmo necesita muchos ejemplos positivos (imágenes con caras) y negativos (imágenes sin caras) para entrenar el clasificador. Luego de esto, se deben extraer las características desde las imágenes. Para este proceso se ocupan *Haar features*. Cada característica se obtiene restando la suma de pixeles bajo un rectángulo blanco y la suma de pixeles bajo un rectángulo negro, siendo estos rectángulos adyacentes. El resultado es un valor único.

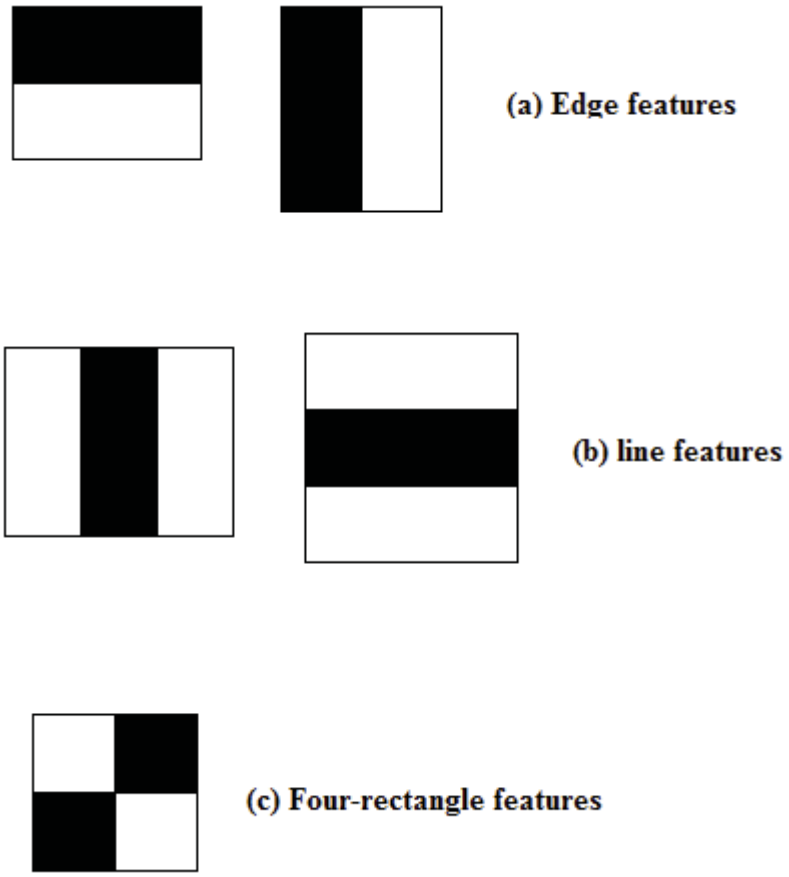


Figura 5.2: Features usadas en el algoritmo de Viola/Jones

Para la extracción de características se necesita calcular una para cada posibilidad, para distintos tamaños de rectángulos y distintas posiciones. Los 3 tipos de características que se usan se muestran en la figura 5.2. La extracción de características requiere mucho poder de cálculo, por lo cual se utilizan imágenes integrales para hacer el proceso más rápido.

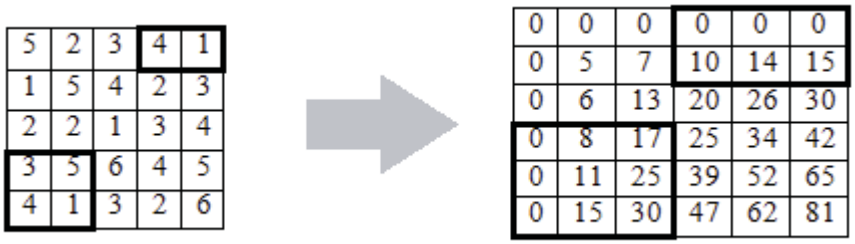


Figura 5.3: Conversión de una imagen original a su imagen integral

Las imágenes integrales son una conversión matricial de una imagen. Esta conversión permite sumar todos los píxeles de una imagen utilizando sólo cuatro píxeles de la imagen integral. Un ejemplo de dicha conversión se muestra en la figura 5.3.

Las imágenes integrales son la matriz integral de la representación matricial de una matriz. Para una matriz de dimensiones n x m, la matriz integral tendrá dimensiones n+1 x m+1. La primera fila y la primera columna de la matriz integral se llena con ceros. Para obtener el valor de las casillas en la matriz integral, se suman los valores en el rectángulo para el cual la casilla a calcular está en la esquina inferior derecha.

Ahora, para calcular la suma de casillas (pixeles) en una submatriz, se hace una operación equivalente en la submatriz correspondiente en la imagen integral de la siguiente forma, siendo S el valor de la suma, tal como se muestra en la ecuación (1).

$$(1)\ S = bottom\ right + top\ left - top\ right - bottom\ left$$

El uso de cuatro operaciones en lugar de lo que podrían ser cientos o miles de operaciones aritméticas, simplifica enormemente el cálculo de las características, lo cual hace viable el algoritmo para detección de rostro en tiempo real, no requiriendo gran poder de cálculo para realizar la identificación. Librerías como OpenCV incluyen esta funcionalidad, e incluyen sus propios clasificadores ya entrenados.

5.2 Reconocimiento de emociones a partir de expresiones faciales

Según el trabajo de *Paul Ekman* en la década de los 70, *"Unmaskink the face, a Guid to Recognizing emotions from facial clues"* [3], se sabe que existen seis emociones básicas: felicidad, tristeza, sorpresa, miedo, enojo y disgusto. Esta conclusión fue producto de una larga investigación realizada en distintas naciones. En esta investigación se concluyó que al hacer la asociación entre la expresión facial y la palabra que describe la emoción; hubo acuerdo sólo en seis. *Paul Ekman* dice que existen otras expresiones, tales como la de vergüenza o la de excitación, pero que estas expresiones no han sido fuertemente establecidas en la cultura, a diferencia de las anteriores. También menciona que existen muchas expresiones derivadas, pero que estas serían sólo combinaciones de las seis principales. Este trabajo utiliza esa clasificación, omitiendo miedo para incluir neutral, en una clasificación que se ha utilizado en otros trabajos revisados. El rostro "neutral" es aquel en el cual los músculos de la cara están relajados.

Si bien se encontró una correspondencia entre la expresión facial y la emoción, es importante recalcar que las expresiones faciales tienen significado según el contexto. Un ejemplo es la risa. ¿Cuántas veces hemos visto gente riendo cuando en realidad están nerviosos o angustiados? Otro ejemplo es el caso cuando las personas fingen una emoción, como el caso en que una persona demuestra una expresión seria cuando en realidad está aguantando la risa. De esta forma, se puede hacer una asociación entre un rostro y una

emoción, sin embargo, se debe hacer la advertencia de que esa asociación siempre estará ligada a un contexto en particular. En un caso parecido, podría ser normal que una persona riera mucho viendo una comedia junto a sus amigos, sin embargo, la reacción podría variar drásticamente si esa comedia la ve solo.

Existe una forma de clasificar las emociones a partir de expresiones faciales, y esto es posible gracias al uso del sistema de Codificado de Acciones Faciales (FACS). Este sistema consiste en taxonomizar los movimientos faciales humanos por su apariencia en la cara. Fue originalmente desarrollado por el anatomista sueco *Carl-Herman Hjortsjö* [21]. Luego, fue adoptado por *Ekman et al.* (1978) [22]. Estas expresiones físicas pueden ser luego categorizadas en emociones por la combinación de los movimientos musculares.

5.3 Emoción espontánea versus emoción actuada

La mayoría del trabajo existente ha construido sus bases de datos de prueba pidiendo a sujetos que explícitamente posen la expresión facial pedida. Sin embargo, dichas expresiones faciales exageradas rara vez ocurren en la vida real. Recientemente la atención se ha centrado en el análisis de las expresiones faciales espontáneas, y se han construido bases de datos recolectando dichas expresiones. El tema es complejo, ya que la mayoría de los modelos computacionales han tenido dificultad para capturar las complejas fronteras que separan una expresión espontánea de una actuada.

Las expresiones faciales espontáneas se diferencian de las expresiones actuadas en términos de los músculos que se mueven y cómo se mueven de forma dinámica. Las expresiones faciales espontáneas suelen ser más rápidas y comienzan más suavemente. Por ejemplo, ha sido observado que las sonrisas actuadas tienen una amplitud mayor a las espontáneas, y tienen una relación menos consistente entre la amplitud y la duración de las sonrisas espontáneas.

En este trabajo se utilizan expresiones faciales actuadas debido a la alta complejidad de construir una base de datos de expresiones espontáneas, sin embargo, los sujetos de prueba utilizados no han sido actores, y tampoco se les ha pedido exagerar las expresiones, haciendo el proceso lo más natural posible.

5.4 Bases de datos existentes

Durante la investigación previa realizada, se ha investigado las bases de datos de expresiones faciales existentes. Algunas de ellas están disponibles para su uso, mientras que otras son de muy difícil acceso. La tabla 5.1 muestra un resumen con las principales bases de datos encontradas. Se ha incluido información sobre la cantidad de sujetos usados para la construcción de la base de datos, la cantidad de expresiones incluidas, si las expresiones fueron actuadas o espontáneas, y si vienen previamente clasificadas o no.

Tabla 5.1: Resumen de las bases de datos de expresiones faciales existentes

Base de datos	Sujetos	Expresiones	Tipo	Clasificada
JAFFE Database	10	7	Actuada	Si
Cohn-Kanade Database	100	Gran rango	Actuada	Si
MMI Database	53	Gran rango	Actuada/Espontánea	Si
FGNet Database	18	7	Espontánea	Si
Authentic Expression Database	28	4	Espontánea	Si
UTDallas-HIT	284	11	Espontánea	Si
RU-FACS	100	Gran rango	Espontánea	Si
MIT-CBCL	12	9	Espontánea	Si
BU-3DFE Database	100	7	Actuada	Si
FABO Database	23	Gran rango	Actuada	Si

Muchas de estas bases de datos no trae una cantidad de expresiones faciales fijas, o no todos los sujetos aparecen con distintas expresiones, sobre todo aquellas que son espontáneas. Otras incluyen la intensidad e incluso las distintas etapas de la expresión, desde su comienzo hasta su fin.

La cantidad de sujetos utilizados varía mucho, desde 10, la más pequeña, hasta 284 la más grande, y la diferencia de expresiones clasificadas muestra que no existe realmente un estándar de cuántas deberían ser la cantidad de expresiones mínimas disponibles.

En este trabajo se optó por construir una base de datos utilizando 18 sujetos de prueba y 6 emociones en total. Se utilizó a la vez una resolución relativamente baja con el objetivo de simular las imágenes tomadas por una webcam promedio integrada.

5.5 Elementos faciales de interés

Por componentes faciales entendemos los órganos, o elementos constituyentes, de una cara humana normal: cejas, ojos, boca, nariz, etc. Tampoco existe una elección única y universal de los componentes faciales de interés, sino que diferentes métodos trabajan con distintos elementos, descartando otros.

Cuanto mayor nivel de detalle requiera una aplicación, más componentes pueden considerarse. Por ejemplo, para un interface perceptual puede ser suficiente con conocer la posición media de ambos ojos, sin embargo, un sistema de análisis del punto de mirada debe distinguir entre: globo ocular, párpado, iris y pupila.

Entre los elementos faciales con mayor relevancia en la literatura, podemos mencionar los siguientes:

- **Ojos:** Son los componentes básicos, y a veces los únicos que son tratados. La localización suele darse con la posición media; pero es difícil encontrar una definición precisa, y no siempre está claro si se refiere al centro del globo ocular o de la pupila. Por ejemplo, ¿cuál es la posición teórica cuando el ojo está cerrado? Cuando hablamos de "ojo izquierdo" nos referimos al que aparece más a la izquierda en la imagen, que será normalmente el ojo derecho de la persona y viceversa. Otra cuestión importante relacionada con los ojos es la orientación de la cara en la imagen. La orientación, o inclinación, se suele definir como el ángulo de la recta que pasa por ambos ojos, respecto del eje horizontal.

- **Boca:** Existen varias formas comunes de especificar la localización de la boca. La más sencilla es mediante la posición media del conjunto boca/labios. Lógicamente, esa elección presenta las mismas ambigüedades que para los ojos, y especialmente cuando la boca está abierta. Algunos trabajos buscan también la extensión horizontal de la boca. Por su parte, la extensión vertical está relacionada con su grado de apertura.

- **Nariz:** La localización de la nariz suele tener una importancia relativa muy inferior, ya que dispone de reducida movilidad, y casi siempre asociada al movimiento de la boca, y tiene menor influencia en la expresión facial. Cuando se incluye, la posición buscada es típicamente la punta de la nariz o los orificios nasales.

5.6 Detección de componentes faciales claves

La localización de componentes faciales presenta similares desafíos a la detección de caras humanas. Pero surgen algunas cuestiones específicas que resulta conveniente identificar. Por un lado, el problema se simplifica al asegurarse la existencia de una cara, con sus dos ojos, una nariz y una boca. Pero, por otro lado, al trabajar con objetos de tamaño más reducido se multiplican las dificultades debido a la variación de apariencia de los componentes.

Muchas son las dificultades que pueden surgir al intentar localizar los componentes faciales claves, a continuación se nombran algunas:

- **Escasa resolución.** Cuando el tamaño de las caras se aproxima al mínimo detectable, la distinción de los componentes faciales por separado es simplemente imposible. Por ejemplo, en una cara de 24x30 píxeles, los ojos no ocupan más de 3*x*3 píxeles. En esas circunstancias, se hace evidente que la localización sólo puede tener lugar dentro de la estructura global de la cara.

- **Expresión facial.** El efecto de las expresiones sobre la apariencia del rostro es mucho más drástico cuando analizamos los elementos faciales individuales. Además, esta situación ocurre con mucha frecuencia.

- **Oclusión y elementos adicionales.** Una cara con oclusión parcial puede tener algunos componentes ocluidos total o parcialmente. La oclusión de un elemento facial provocará que la hipótesis de partida –en relación a que "existen dos ojos, una nariz y una boca"– no sea necesariamente cierta en todos los casos. El problema puede tener diferentes causas: existencia de elementos faciales (como gafas, bigote, barba), superposición de objetos externos, y la desaparición de algunos componentes con giros grandes de la cara. Los primeros, además, dificultan la localización aunque la oclusión no tenga lugar.

- **Sombras**. El efecto de las sombras puede hacer que los ojos o la boca, no sean más que manchas indistinguibles de píxeles oscuros, incluso con resoluciones elevadas. En muchos de estos casos, la sombra es producida por la propia cara.

- **Estructura facial.** Otro aspecto que conviene tener en cuenta es la estructura propia del rostro humano. Los distintos constituyentes de la cara no aparecen en posiciones aleatorias, sino que deben presentar una estructura coherente: los ojos están encima de la boca, las cejas están sobre los ojos, los ojos y la boca forman un triángulo isósceles, etc. Por lo tanto, el localizador debe garantizar la coherencia del resultado.

Durante el desarrollo de este trabajo se exploró la posibilidad de utilizar un sistema de detección de puntos faciales. Existen diversas alternativas para realizar este proceso. El uso de puntos faciales para la detección de la expresión y los FACS no es nuevo, y los resultados son buenos si se logra detectar dichos puntos con precisión.

El conjunto de métodos que utilizan puntos faciales se denominan "Métodos Basados en Análisis Estructural". Los métodos de localización estructural persiguen no sólo encontrar la posición de los elementos faciales, sino también describir su contorno o su forma. En su mayoría se trata de técnicas basadas en modelos, que pueden ser genéricos o específicos del dominio de las caras.

De esta manera, el proceso de localización se puede entender como una búsqueda del mejor ajuste del modelo a la instancia actual, a través de la optimización de una función objetivo. Podemos distinguir dos subcategorías: las técnicas orientadas al análisis de elementos individuales (*snakes*, patrones deformables), y las que tratan la cara de manera global (modelos de forma activa y apariencia activa). Tanto los Modelos de Forma Activa (ASM) como los modelos de Apariencia Activa (AAM) se basan en PDM. A continuación se explicarán los más conocidos.

5.6.1 Snakes y patrones deformables

Los snakes, conocidos también como contornos activos, son una técnica genérica de modelado de contornos que ha sido usada para obtener el perímetro de la cara. Un snake es

una curva cerrada compuesta por trozos de curvas, que se sitúan inicialmente en una posición de la imagen y que se adapta progresivamente al contenido de esta.

Existe el concepto de energía E_{snake} que controla la manera en que el contorno se ajusta a la imagen, y suele tener la forma:

$$(2) E_{\text{snake}} = E_{\text{interna}} + E_{\text{externa}}$$

El término de energía interna, E_{interna} controla la evolución natural de la curva. La elección más habitual consiste en definirla proporcionalmente a la distancia entre los puntos de control del *snake*. Esto le otorga un comportamiento elástico, que hace que tome formas compactas. Por otro lado, la energía externa suele estar relacionada con el gradiente de la imagen, haciendo que la curva se sitúe próxima a los bordes más destacados.

5.6.2 Modelos de distribución de puntos (PDM)

Los modelos de distribución de punto, o *"Point Distribution Model"* (PDM) fueron propuestos por *Cootes y Taylor* [13] como una manera compacta de describir los posibles modos de variación de un conjunto de puntos asociados a distintas partes de la cara. La localización facial se hace de forma global, no independiente, para cada componente.

El método hace Análisis de Componentes Principales (PCA) sobre las variaciones que se observan en una serie de puntos característicos. Para poder realizar este análisis, se deben posicionar un conjunto de puntos etiquetados manualmente en posiciones determinadas en una serie de imágenes, generalmente entre 60 y 200. La detección de estos puntos, por tanto, depende mucho de las imágenes seleccionadas para el entrenamiento.

5.6.3 Modelos de forma activa (ASM)

PDM no es completo ni suficiente, ya que para una imagen nueva no se conocen los puntos característicos. Los modelos de Forma Activa [13, 14, 16, 17], o *"Active Shape Model"* (ASM) definen una forma de ajustar los parámetros de posición y forma del PDM, añadiendo información a los puntos del modelo. El modelo se inicializa en una posición cercana a la cara y luego tiene lugar un proceso iterativo de ajuste.

Los modelos de forma activa tienen ventajas a otros enfoques de localización de componentes. La más importante es que se aprovechan tanto la forma global como las propiedades locales. Por otro lado, los resultados obtenidos ofrecen una descripción detallada de las posiciones de interés definidas en el proceso de entrenamiento. Entre las desventajas e inconvenientes de los ASM es que se requiere un entrenamiento del modelo y que el proceso de ajuste es un proceso costoso.

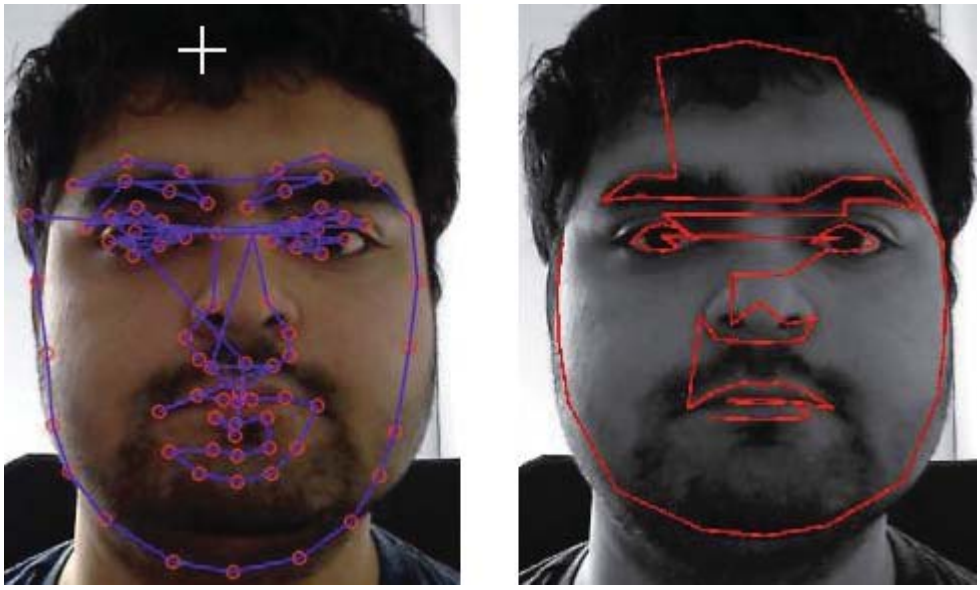


Figura 5.4: Detección de componentes con asmlib (izquierda) y Stasm (derecha)

En la figura 5.4 se muestran los resultados de la detección de puntos en una cara utilizando asmlib para OpenCV (a la izquierda) y la librería Stasm [15] (a la derecha), dos librerías disponibles para uso libre.

Ambas librerías mostraron un comportamiento poco satisfactorio. Stasm logró un mejor desempeño, sin embargo, al estar optimizada para expresiones faciales neutras, su uso para la detección de expresiones faciales no es una buena idea.

Stasm es una librería escrita en C++ y que permite la localización de puntos de referencia en una cara. Stasm recibe una imagen conteniendo una cara, y retorna un vector con las marcas detectadas. Stasm está diseñada para funcionar con vistas frontales y con poca variación de ángulo de la cara.

Entre las ventajas que Stasm posee, es su interface simple y versátil, la posibilidad de captar errores y las funciones de utilidades que permiten convertir el modelo default de 77 puntos a otro formato y forzar el posicionamiento de puntos fuera de los límites de una imagen (por ejemplo, si la cabeza aparece cortada).

5.6.4 Modelos de apariencia activa (AAM)

Los modelos de Apariencia Activa [18, 19, 20], o *"Active Appearance Model"* (AAM) son una extensión de los ASM. La principal diferencia es que agrega información de textura al modelo de puntos. Al igual que los PDM, los distintos modos de variación de la textura se obtienen aplicando PCA sobre los niveles de intensidad de los pixeles.

Los modelos AAM mejoran el uso de la información en comparación a ASM, sin embargo presentan problemas similares, tales como la ineficiencia. A pesar de esto, los AAM han sido usados en enfoques similares, tal como los modelos deformables 3D, los cuales trabajan en un espacio tridimensional de forma explícita.

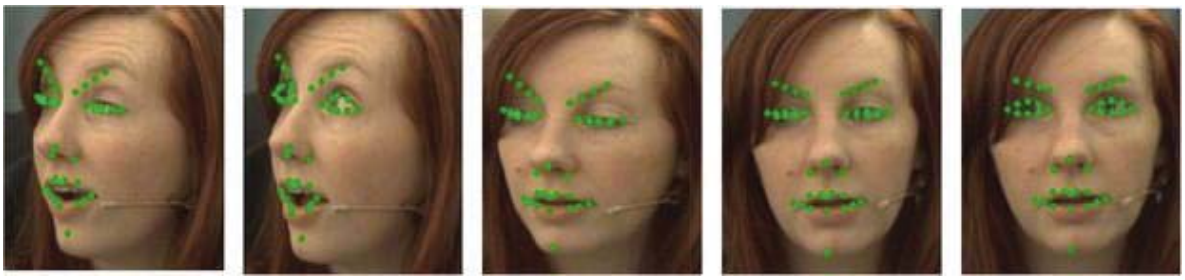


Figura 5.5: Facial Tracker, software privativo que utiliza un modelo AAM

En la figura 5.5 se muestra un ejemplo de implementación de AAM. Como se puede observar, la principal diferencia es la presencia de una textura que se va adaptando a los cambios en las expresiones. En el fotograma, la persona cambia de postura lentamente. La textura se va adaptando a la forma de la cara.

A pesar de sus ventajas, no se encontraron herramientas libres que utilizaran este enfoque, o las que se encontraron no estaban debidamente documentadas o actualizadas para su utilización, lo cual implica que si se desea utilizar AAM, se debería considerar construir el modelo desde cero, lo cual agrega complejidad que pone en riesgo el éxito del proyecto. AAM también tiene la desventaja de que presenta complicaciones con caras no vistas anteriormente. Dicho problema se conoce como el problema de "Generic AAM".

5.7 Clasificadores

Un clasificador, en el ámbito del reconocimiento de patrones y la inteligencia artificial, es un algoritmo que se utiliza para asignar a un objeto de entrada una categoría concreta y conocida. Para implementar un clasificador, se requiere seleccionar las características y una fase de entrenamiento. A continuación se detalla la teoría de algunos clasificadores estadísticos importantes.

5.7.1 Clasificador Bayesiano

El Clasificador Bayesiano es un clasificador probabilístico que se fundamenta en el teorema de Bayes.

Dato un conjunto de *n* atributos característicos definido como $X = \{X_1, X_2, \dots, X_n\}$ y un conjunto de *m* clases posibles $c_{1,}c_{2,} \dots, c_m$, el problema consiste en asignar a un objeto descrito por dicho conjunto de características tal que la probabilidad de la clase se

maximiza, como se muestra en (3). Para resolver el problema enterior, se puede usar el teorema de Bayes, como se muestra en (4). Finalmente, reemplazando (4) en (3) se tiene la ecuación (5).

$$(3)\ Arg_C[MaxP(C|X)]$$

$$(4)\ P(C|X) = P(C)P(X|C)/P(X)$$

$$(5)\ Arg_C\left[Max[P(C|X) = P(C)P(X|C)/P(X)]\right]$$

Cabe señalar que el denominador, *P(X)* no varía para las diferentes clases, por lo que se puede considerar como constante. Finalmente, para resolver un problema de clasificación con el clasificador bayesiano, se requerirá la probabilidad a priori de cada clase *P(C)*, y la probabilidad de cada una de las características para cada clase *P(X|C)*, conocida como verosimilitud y que se lee como la probabilidad de *X* dado *C*.

La complejidad de este clasificador está en que si se aplica la ecuación (4) directamente, resulta en un sistema extremadamente complejo y de un alto gasto computacional, debido a que a mayor número de atributos, la expresión $P(X|C)$ crece exponencialmente.

Para solucionar este problema, se suelen ocupar simplificaciones. Una de las simplificaciones más comunes consiste en considerar todas las características como independientes dada la clase, este clasificador es conocido como el Clasificador Bayesiano Ingenuo. Reescribiendo la ecuación (4) considerando estas restricciones, se tiene la ecuación (6).

$$(6)\ P(C|X) = P(C)\prod_{i=1}^{n} P(X_i|C)$$

Para encontrar $P(X_i|C)$ se suele asumir una distribución normal de los datos, por lo que se utiliza la función de densidad de dicha distribución, como se muestra en la ecuación (7). Esto requiere por lo tanto, calcular previamente los promedios y las desviaciones estándar del conjunto de datos de entrenamiento para cada clase.

$$(7)\ P(X_i|C) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1(x_i-\mu)}{2\ \sigma}}$$

5.7.2 Vecino más cercano (KNN)

Este clasificador es muy simple e intuitivo, y se basa en tomar la información que se desea clasificar y calcular las distancias de entre todos los objetos del conjunto de entrenamiento. Luego, para clasificar el objeto, se busca un número K de vecinos cuyas distancias son menores al objeto a clasificar. De esa forma se determina la clase a la que pertenece el objeto. Entre sus características principales están:

- No requiere la construcción de un modelo.
- Utiliza los datos originales sin requerir una representación especial de estos.
- Realiza su predicción en base a la información local (cuando se aplican heurísticas de búsqueda).
- Puede producir resultados equivocados si el cálculo de la distancia no es adecuado.
- A mayor K, mayor probabilidad de una predicción correcta.

La elección del K a utilizar es muy importante, ya que los resultados pueden variar mucho, por este motivo, es importante conocer bien la información y cómo se calculará la distancia. Se suelen definir heurísticas de búsqueda. Por ejemplo, si se utilizan distancias euclidianas, se pueden establecer límites, "encerrando" el objeto en un cierto radio. Otro punto importante es la selección del conjunto de entrenamiento. La elección de casos representativos es importante para una buena predicción. Una cantidad muy grande produciría un gasto computacional excesivo.

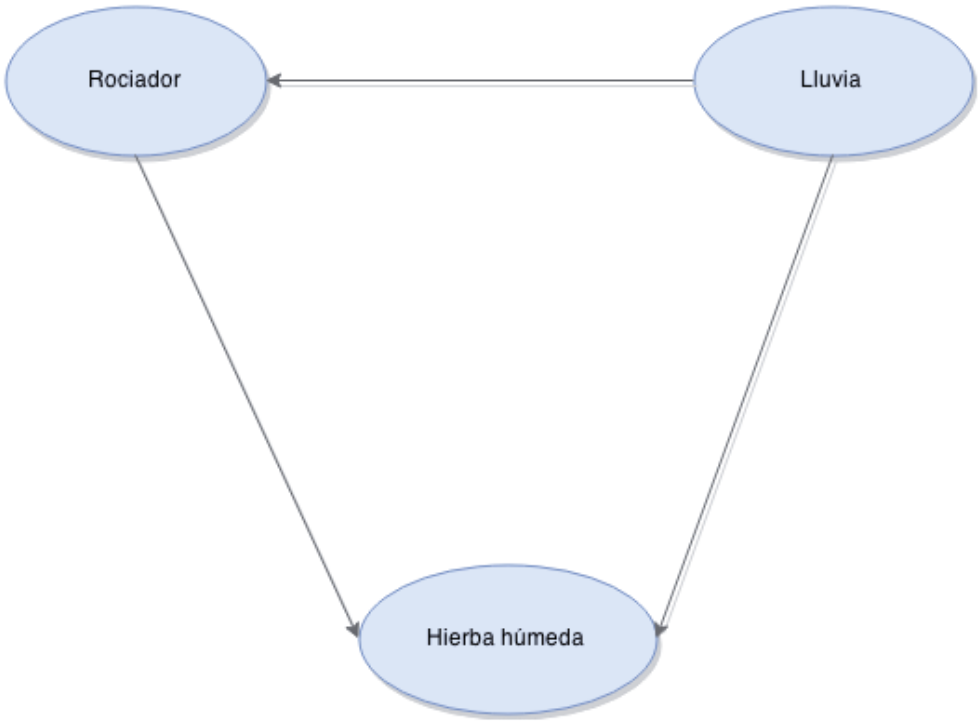
5.7.3 Redes Bayesianas

Las redes bayesianas se construyen con grafos que buscan modelar un fenómeno utilizando para ello un conjunto de variables y relaciones probabilísticas entre dichas variables. Existen dos formas de construir una red bayesiana, la primera consiste en hacerla manualmente. Para ello se requiere un experto sobre el tema que pueda plasmar el conocimiento en la red. La segunda forma es utilizar un algoritmo que sea capaz de encontrar las relaciones existentes entre los datos, utilizando algoritmos de búsqueda y la entropía.

Una red bayesiana consta de un grafo acíclico dirigido, esto quiere decir que para cada vértice v, no existe un camino directo que empiece y termine en v. Las aristas del grafo representan la dependencia entre el conjunto de variables, mientras que los vértices representan al dato. Para cada nodo, a su vez, se puede construir una tabla de probabilidad.

Lluvia	Rociador	
	T	F
F	0.4	0.6
T	0.01	0.99

Lluvia	
T	F
0.2	0.8



Rociador	Lluvia	Hierba	Húmeda
		T	F
F	F	0	1
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Figura 5.6: Ejemplo de Red Bayesiana

Un ejemplo de red bayesiana junto a sus tablas de probabilidad se muestra en la imagen 5.1. En ella se establecen tres variables en el modelo, el rociador, la lluvia y la hierba húmeda. Es sabido entonces por nosotros que si hay lluvia, entonces la hierba estará húmeda, lo mismo si el rociador está prendido, y que difícilmente el rociador estará

encendido si existe lluvia. Construyendo tablas de probabilidad para cada suceso, se puede utilizar esta información para responder preguntas del tipo de probabilidad condicional.

Para responder las preguntas se debe construir la función de probabilidad conjunta. En este caso, tendríamos que para la imagen 5.1, dicha función sería:

$$P(G,S,R) = P(G|S,R)P(S|R)P(R), \quad (8)$$

G: Hierba húmeda
S: Rociador prendido
R: Lluvia

Un ejemplo de utilización de red bayesiana para el reconocimiento de emociones se da en el caso de utilización del Sistema de Codificación de Acciones Faciales (FACS) o de los Parámetros de Acción Facial (PAF), en donde cada acción facial o parámetro pasa a ser un nodo del grafo que llevan en su conjunto a un nodo final que es la emoción. Un ejemplo que muestra esto, utilizando FACS, se muestra en la imagen 5.2.

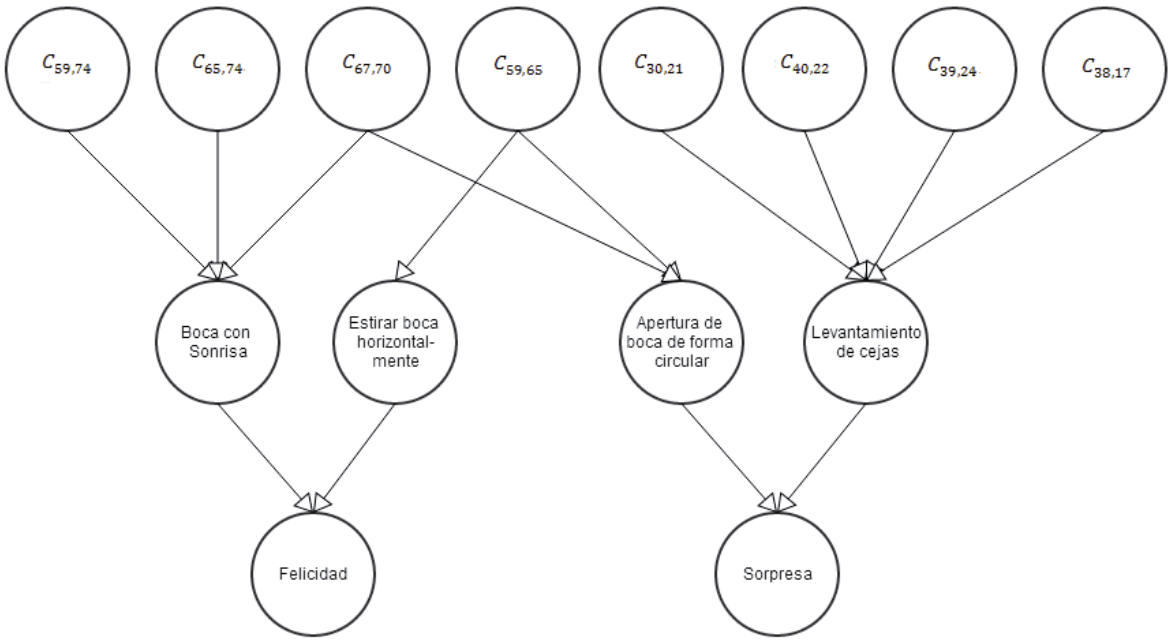


Figura 5.7: Red Bayesiana para emociones felicidad y sorpresa utilizando el sistema de Codificación de Acciones Faciales (FACS)

5.7.4 Arboles de decisión (Id3)

Los árboles de decisión son un tipo de clasificador que consiste en un conjunto de reglas del sipo "si", "sino", y "entonces", que poseen la ventaja de otorgar un modelo visual

y su costo computacional no es alto. Este tipo de clasificador tiene la bondad de que permite determinar los atributos que aportan información relevante al sistema, estableciendo jerarquías de importancia entre dichos atributos. Además, es rápido clasificando nuevos elementos y la creación del modelo tiene un bajo costo computacional. Una vez creado, puede ser modificado por un experto, al igual que en las redes bayesianas, ya que es entendible e interpretable.

A pesar de sus ventajas, también tiene desventajas, tales como que favorece a los atributos multivariados, tiene poca tolerancia al ruido de los datos, usualmente requiere discretizar atributos continuos y puede llevar a la creación de árboles muy grandes para la explicación del ruido.

Usualmente se construyen utilizando fórmulas que nos permiten indicar si un atributo debe ser usado o no. Se suele utilizar la entropía y el índice de Gini. Para la entropía, o medida de incertidumbre, se suele usar su forma binaria, la cual se muestra en la ecuación (9), mientras que el índice de Gini, que representa la medida de desigualdad del sistema, se suele usar como se muestra en la ecuación (10). En ambas, $p(i|t)$ representa la probabilidad de que cierto caso c ocurra o no dada cierta evidencia. Para cada caso existirá un $p(i|t)$ distinto, y la suma de ambos será 1.

$$(9)\ -\sum_{i=0}^{c-1}\big(p(i|t)\big)log_2\big(p(i|t)\big)$$

$$(10)\ 1-\sum_{i=0}^{c-1}(p(i|t))^2$$

Lo que se busca es encontrar los atributos que generen menor entropía, o en otras palabras, que no tiendan al desorden. Mientras menor es la entropía, más alto en el árbol quedará el atributo.

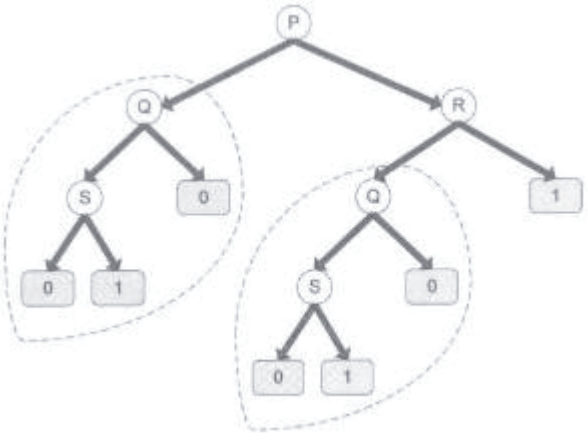


Figura 5.8: Árbol de decisión con ramificaciones duplicadas

En la figura 5.8 se muestra un árbol de decisión. Si se hubiese utilizado el método de la entropía, significaría que el atributo P presentó una menor entropía que los demás, siendo los nodos Q y R del segundo nivel posibles valores para P. Luego, dado Q o dado R se vuelve a iterar buscando el atributo con menor entropía, para Q fue S, mientras que para R fue Q. Las hojas del árbol corresponde a la decisión que debe tomar el clasificador, en este caso, 0 o 1, que representan generalmente una decisión de si o no.

En la figura se muestra también ramificaciones duplicadas. Estas ramificaciones pueden ser podadas de forma de hacer el árbol más eficiente, existiendo algoritmos para ello.

5.7.5 Clustering y algoritmo K-means

El clustering es una técnica utilizada para generar agrupaciones entre los datos de forma de obtener información de estos. Se busca determinar la similitud o diferencia de los elementos que confirman los datos y en base a esto tomar una decisión. Para esto se ocupan diversos algoritmos de agrupamiento. La decisión de qué algoritmo utilizar, dependerá de la característica de los datos.

Existen básicamente 8 formas de formar los grupos de datos, por particiones, jerárquicos, difusos, grupos bien separados o definidos, basados en prototipos, basados en conexiones, basados en densidad y basados en forma. Cada uno de estos se explica a continuación:

- **Por particiones:** Cada grupo se define de forma que cada elemento puede pertenecer a un grupo únicamente.

- **Jerárquicos:** Se definen jerarquías, es decir, que dentro de un grupo pueden existir subgrupos, y así sucesivamente, pudiendo un dato pertenecer a un grupo y a un subgrupo al mismo tiempo.

- **Difusos:** Un elemento puede tener un grado de pertenencia a más de un grupo.

- **Grupos bien separados definidos:** Esto ocurre cuando dos elementos son difíciles de confundir ya que tienen características particulares que permiten distinguirlos casi sin error.

- **Basados en prototipos:** Se establece la pertenencia al grupo basado en la distancia a alguna medida estadística, como la mediana o el promedio, o a un elemento muy representativo de este.

- **Basados en conexiones:** Se utilizan para agrupar elementos que se conectan entre si pero que están desconectados de otros grupos. Esto permite agrupar elementos en un grupo con formas inusuales o poco comunes.

- **Basados en densidad:** Agrupa a los elementos basándose en las regiones con mayor densidad o más pobladas. Los elementos que no pertenecen a estas zonas son ignoradas generalmente.

- **Basados en forma:** En este tipo de agrupamiento, se tiene un alto conocimiento del comportamiento de los datos, lo cual permite definir grupos con determinada forma basados en alguna fórmula.

Dentro de los algoritmos de clustering más usados y conocidos, se encuentra el algoritmo K-means, el cual es un algoritmo de clustering basado en prototipos. Cuenta de 2 pasos básicamente los cuales se explican a continuación:

- **Paso 1:** Establecer estimaciones de las medias $m_1, m_2, \dots, m_k$
- **Paso 2:** Mientras no existan cambios en las medias, hacer:
    - Usar las medias estimadas para clasificar el elemento en un grupo.
    - Desde **i=1** hasta **i=k** hacer
        - Reemplazar $\boldsymbol{m_i}$ con el promedio de todos los elementos del cluster $\boldsymbol{i}$
    - Fin Desde
- Fin Mientras

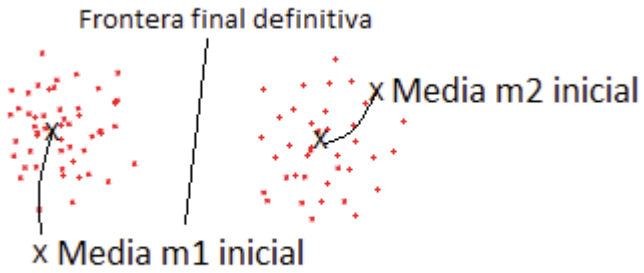


Figura 5.9: Ejemplo que muestra cómo dos medias se mueven hacia los centros de sus clusters

La figura 5.7 muestra cómo las medias se desplazan desde su posición inicial estimada hasta la posición final tras el proceso iterativo de ajuste para k=2. En este caso se pueden distinguir los clusters claramente, y es evidente que el k óptimo es 2, sin embargo, muchas veces el k no está claro y debe ser determinado utilizando alguna regla, como el criterio de Schwarz.

5.7.6 Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial, más conocidas como SVM de su nombre en inglés, Support Vector Machines, son un método de aprendizaje computacional relativamente nuevo usado para la clasificación binaria introducida por *Vladimir Vapnik* en 1995.

La idea principal es encontrar un hiperplano que separa la información multidimensional perfectamente en dos clases. Sin embargo, debido a que los datos no son generalmente linealmente separables, SVM introduce la noción de espacio característico inducido por kernel, el cual transforma la información a un espacio dimensional más alto donde los datos si son separables.

Generalmente, el transformar los datos a un espacio multidimensional más alto generaría problemas computacionales debido a su alto costo. La clave de SVM es que este espacio de dimensiones superiores no necesita ser tratado de forma directa, eliminando el problema inicial.

5.8 Análisis de componentes principales

Estas técnicas fueron en un comienzo desarrolladas por *Pearson* a fines del siglo XIX y luego fueron estudiadas por *Hotelling* en los años 30 del siglo XX. Sin embargo, no fue hasta la aparición de los computadores que se empezaron a popularizar. Para estudiar las relaciones que se presentan entre n variables correlacionadas que miden información común, se puede transformar el conjunto original de variables en otro conjunto de nuevas variables no correlacionadas entre sí (que no tenga repetición o redundancia en la información) llamado conjunto de componentes principales.

Las nuevas variables son combinaciones lineales de las anteriores y se van construyendo según el orden de importancia en cuanto a la variabilidad total que recogen de la muestra. De modo ideal, se buscan $m < n$ variables que sean combinaciones lineales de las n originales y que estén no correlacionadas, recogiendo la mayor parte de la información o variabilidad de los datos. Si las variables originales están no correlacionadas de partida, entonces no tiene sentido realizar un análisis de componentes principales. El análisis de componentes principales es una técnica matemática que no requiere la suposición de normalidad multivariante de los datos, aunque si esto último se cumple se puede dar una interpretación más profunda de dichos componentes.

El análisis de componentes principales (conocido por sus siglas en inglés PCA), es una técnica bastante usada en el reconocimiento de patrones y procesamiento de señales para reducción de la dimensionalidad de los datos. Los datos generalmente contienen información redundante, por lo que utilizar un vector de características permite preservar la

información intrínseca de los datos eliminando en gran parte la redundancia que no contribuye a entregar nueva información. El Análisis de Componentes Principales permite reducir el tamaño de los datos, haciéndolos manejables. Además permite explicar las causas de la variabilidad de los datos y ordenarlas por importancia. En este trabajo se utiliza PCA para extraer las características de las distintas emociones detectadas, basado en el enfoque utilizado en [26]. Otros trabajos similares que utilizan en este ámbito se pueden ver en [24] y [27].

El análisis de componentes principales es una técnica multivariante que trata de reducir el número de variables originales $(X_1, X_2, \dots, X_{n)}$ a un número menor de variables $(CP_1, CP_2, \dots, CP_p)$, denominadas componentes principales. Dichas variables son una combinación lineal de las variables iniciales, y sintetizan la mayor parte de la información contenida en los datos originales.

5.8.1 Cálculo de componentes principales

Se considera una serie de variables $(x_1, x_2, \dots, x_p)$ sobre un grupo de objetos, y se trata de calcular un nuevo conjunto de variables $y_1, y_2, \dots, y_p$ no correlacionadas entre sí y cuyas varianzas disminuyan progresivamente.

Cada y_j para $j = 1,\dots, p$, es una combinación lineal de las variables originales $(x_1, x_2, \dots, x_p)$, como se muestra en la ecuación (11), Donde $a_j' = (a_{1j},\ a_{2j}, \dots, a_{pj})$ un vector de constantes, y $x = \begin{pmatrix} x_1 \\ \cdots \\ x_p \end{pmatrix}$.

$$(11) \quad y_1 = a_{j1}x_1 + a_{j2} + \cdots + a_{jp}x_p = a_j'x$$

Para mantener la ortogonalidad del vector, el módulo del vector a_j' debe ser 1, como se muestra en la ecuación (12).

$$(12) \quad a_j'\, a_j = \sum_{k=1}^{p} a_{kj}^2 = 1$$

Luego para la elección del primer componente se elige un $a_1$de modo que $y_1$tenga la mayor varianza, y cumpla que $a_1'a_1 = 1$. La variable a_2 se elige de modo que la variable y_2 no esté correlacionada con y_1, y así sucesivamente hasta $y_p$de modo que las variables obtenidas tengan cada vez menor varianza.

En este trabajo, el cálculo de los componentes principales se a partir de la matriz de correlaciones. Para hacer esto, se calculan los componentes sobre las variables estandarizadas, es decir, se hace la media 0 y la varianza 1. Esto implica que se toman todos los componentes principales de la matriz de correlaciones en lugar de la matriz de covarianza, ya que cuando las variables se estandarizan, las covarianzas y las correlaciones son iguales. De esta forma, los componentes son autovectores de la matriz de correlaciones y son distintos de la matriz de covarianza. Al hacer esto, se le da igual importancia a todas las variables originales, sin embargo es posible ordenarlas por importancia de forma posterior a su obtención.

5.8.2 PCA y Kernel PCA

En el PCA aplicado a imágenes, el objetivo es transformar las imágenes de rostros en un grupo pequeño de características, denominadas autocaras o "*eigenfaces*", correspondientes a los componentes principales del entrenamiento inicial del conjunto de las imágenes. Cuando se proyecta una nueva imagen a este subespacio formado por las autocaras, es posible comparar su posición en el espacio con las autocaras ya clasificadas y conocidas. Esta aproximación tiene como ventajas su simplicidad, velocidad, capacidad de aprendizaje y su relativa insensibilidad a pequeños cambios en la imagen del rostro.

Por otro lado, los métodos que utilizan kernel, presentan alternativas en diferentes tareas para el análisis y procesamiento de señales, tales como la eliminación de ruido, reducción de la domensionalidad y reconocimiento de patrones.

La función kernel $k(\cdot,\cdot)$ permite una generalización no lineal de la mayoría de los algoritmos lineales, tales como máquinas de soporte vectorial, métodos de agrupamiento, análisis de componentes independientes kernel y kernel PCA.

Al combinar el método de kernels con el método de *eigenfaces*, se origina el método conocido como Kernel PCA, o KPCA. Los datos de entrada son mapeados del espacio característico original a uno de mayor dimensionalidad, convirtiendo un problema no lineal en uno lineal más sencillo de resolver.

Un Kernel k se presenta como una medida de similitud. Esta puede ser vista como un producto punto en un espacio característico H. Utilizando un mapeo φ para un conjunto X de entrada, se tiene:

$$(13)\varphi: X \rightarrow H$$

$$x \rightarrow \varphi(x)$$

Donde la función kernel correspondiente es:

$$(14)\ k(x, x') = \langle \varphi(x), \varphi(x') \rangle$$

Esta técnica puede aumentar el rendimiento de PCA por sí solo, por lo que queda planteada la inquietud de una implementación utilizando este método.

6. Trabajo realizado

Para la implementación del clasificador se utilizan las herramientas que OpenCV incluye por defecto, entre dichas funcionalidades se encuentran las funciones básicas para operar con matrices, tales como calcular la traspuesta o los vectores propios.

Durante la primera etapa del proyecto se construyó un prototipo utilizando un clasificador bayesiano y un modelo ASM. En la segunda etapa, esta idea fue dejada de lado para pasar a una solución con PCA. Ambos enfoques son explicados a continuación.

6.1 Implementación de prototipo usando clasificador Bayesiano y modelo ASM

En esta etapa se usaron Puntos de Referencia del rostro para la extracción de características. Los Puntos de Referencia ya han sido usados antes para la definición de características, por ejemplo, en [24] se definen 36 puntos faciales. En dicho trabajo se utilizan las distancias euclidianas entre todos los puntos, por lo que se construye una matriz de 36x36 que luego sirve como conjunto de entrada al modelo de predicción.

Para este prototipo, se definieron un conjunto de puntos, 24 en total, correspondientes a puntos predefinidos del modelo *Muct77* que incorpora la librería Stasm por defecto, y que se compone de 77 puntos en total.

Dado un punto $P_n\,[x_n, y_n]$ del modelo Muct77, donde $0 \leq n \leq 76$ y corresponde al índice del punto en el modelo que se puede observar en la figura 5.3, x es la coordenada del punto en el eje x, e y a la coordenada del punto en el eje y. Podemos expresar la distancia euclidiana entre dos puntos $P_n\,[x_n, y_n]$ y $P_m[x_m, y_m]$ como:

$$(15)\; d_E(P_n\,, P_m\,) = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}$$

Se utiliza la distancia euclidiana debido a que esta no se ve alterada con el ángulo de la cabeza, sin embargo, no es suficiente. Debido a que la cercanía o lejanía del rostro respecto de la cámara afectan estos valores, se debe encontrar un método que sea insensible a la distancia del rostro. Para ello se utiliza la distancia euclidiana de dos puntos constantes $P_k\, y\, P_l$ para cada característica. Posibles valores para k y l pueden ser 0 y 12 respectivamente, que corresponde a puntos que describen el ancho del rostro. En base a lo anterior se define una característica $C_{n,m}$ insensible a la distancia del rostro en la ecuación (16). Notar que el uso de raíces cuadradas no es necesario, por lo que para ahorrar gasto computacional y evitar problemas de exactitud no se utilizan. Por ello las distancias se muestran al cuadrado. Este cambio constituye una diferencia con el trabajo hecho en [24].

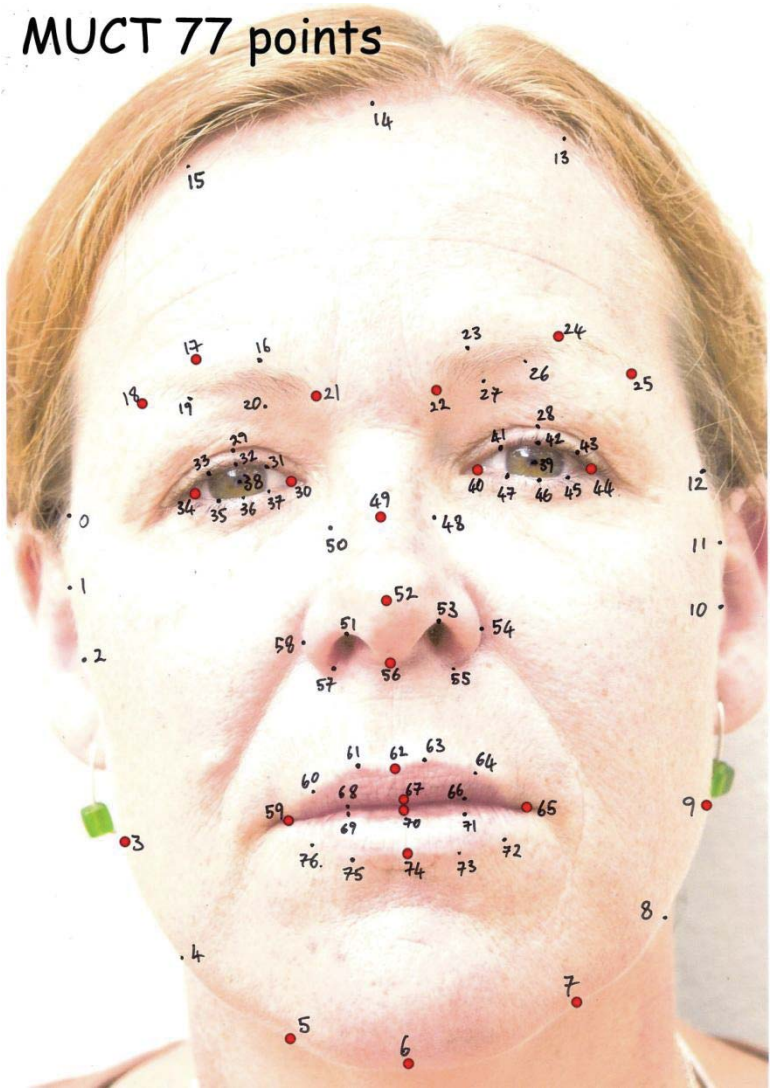


Figura 6.1: Puntos del modelo Muct77 según documentación de Stasm. Los puntos destacados corresponden a los puntos usados en este prototipo.

$$(16) C_{n,m} = \frac{d_E(P_n, P_m)^2}{d_E(P_k, P_l)^2}$$

De los 77 puntos definidos en Muct77, no todos ellos contribuyen a una expresión facial en particular. Debido a que existen puntos que no son representativos y se mantienen constantes, como aquellos del contorno del rostro, se han eliminado. Producto de ello, la cantidad de puntos totales que se utilizaron son 24.

Las características, extraídas a partir de los puntos detectados, son ingresadas a un clasificador bayesiano, el cual utiliza el algoritmo del clasificador bayesiano explicado en la sección 5.7. La distribución de probabilidad utilizada fue la Gaussiana.

Los resultados de este clasificador fueron buenos para dos a tres emociones, sin embargo, al aumentar el número de clases, el porcentaje de acierto baja drásticamente. Esto fue la causa que se descartara este prototipo y se implementara otro método. La principal causa del problema es que la librería Stasm no es tan exacta como se quisiera y tiende a ajustarse a una expresión facial neutral.

6.2 Algoritmo propuesto utilizando PCA y distancia euclidiana

En esta sección se explica el conjunto de características utilizado, los algoritmos implementados, tanto como para el entrenamiento como para el clasificador, el cálculo de la emoción, las restricciones del modelo propuesto, y se incluye documentación sobre la utilización del prototipo.

6.2.1 Conjunto de características

Una imagen de un rostro en dos dimensiones con tamaño NxN también puede ser considerada como un vector de largo N^2. Cada uno de los vectores de características es de largo N^2, y es una combinación linear de la imagen original de la cara. Dichos vectores corresponden a los vectores propios de la matriz de covarianza correspondiente a las imágenes originales. Cuando dichos vectores son renderizados como una imagen, tienen una apariencia similar a una cara, es por eso que se les llama "caras propias", o en inglés, *eigenfaces*. Mientras más vectores son utilizados, mayor es la exactitud del sistema, pero la complejidad computacional es incrementada.

6.2.2 Algoritmos de entrenamiento y de clasificación

El enfoque utilizado en este trabajo utiliza PCA para la clasificación en distintas emociones. Se cuenta con un set de imágenes de entrenamiento de donde se extraen las características y se efectúa PCA. El clasificador compara el rostro a clasificar buscando el más cercano. El indicador de similitud entre dos imágenes está dado por un indicador basado en la distancia euclidiana existente entre los vectores proyectados al espacio de caras de la imagen correspondiente al set de entrenamiento y la imagen a clasificar.

Debido a que las personas tienen distintas características, el utilizar una única matriz de proyección para todas las imágenes no da buenos resultados siempre. Para ello se usa en su lugar, una matriz de proyección para cada sujeto de prueba. Luego, la imagen a comparar se transforma usando la matriz de cada sujeto, buscando el mayor grado de cercanía.

Se implementa un algoritmo iterativo en bloque, que consiste en encontrar todos los vectores de proyección a la vez. El objetivo es calcular la matriz V (matriz de valores propios de la matriz de covarianza) y la matriz U (la matriz de transformación o proyección), sin embargo, se calcula una matriz V y una matriz U para cada sujeto de entrenamiento. El pseudocódigo de este algoritmo se muestra a continuación:

1) Desde *i = 0* hasta *i = k*, donde *k* es el número de sujetos de prueba, hacer:
2) Inicializar matriz de datos A_i con datos de prueba
3) Centrar los valores de A_i respecto al promedio.
4) Calcular la covarianza efectuando $C_i = A_i^T * A_i$
5) Obtener los valores propios de C_i y almacenarlos en V_i
6) Calcular U_i haciendo $U_i = (A_i * V_i)^T$

En el caso particular de este trabajo, la matriz de datos A contiene todas las imágenes de entrenamiento. Cada columna de la matriz A inicial corresponde a una imagen de entrenamiento en escala de grises y representada unidimensionalmente. Luego para clasificar una imagen en una clase, se sigue el algoritmo explicado a continuación:

1) Inicializar *min = 999999*
2) Inicializar *columna = -1*
3) Desde *i = 0* hasta *i = k*, donde *k* es el número de sujetos de prueba, hacer:
4) Calcular la proyección *P* haciendo $P = U_i * C_i$
5) Calcular la proyección del set de entrenamiento $T_i = U_i * A_i$
6) Desde $j = 0$ hasta $j = n$, *n* es el número de imágenes de entrenamiento por sujeto, hacer:
7) *índice* = $T_i[j]/(P - T_i[j])$, donde $T_i[j]$ es la columna *j* de la matriz T_i
8) Si *índice* < *min*, hacer *min = índice* y *columna = j*
9) Se clasifica la imagen en la clase correspondiente a la más cercana, indicada por la variable *columna*.

6.2.3 Cálculo de la intensidad de la emoción

El sistema calcula la intensidad de la emoción calculando el grado de cercanía de la emoción con la imagen encontrada. Se calcula primero el porcentaje para la emoción detectada, luego la diferencia faltante para el 100% se hace calculando la cercanía de cada emoción en porcentaje y luego reajustándola de forma que la suma total sea de 100.

El resultado es graficado por el sistema en la ventana principal en forma de gráfico de barras horizontal, actualizándose para cada cuadro.

6.2.4 Restricciones del modelo propuesto

El modelo propuesto tiene limitantes o restricciones de operación que son conocidas previa implementación, y estas se especifican a continuación:

- **Detección de rostros:** Debido a que se utiliza OpenCV para la detección de rostros, el sistema tendrá problemas con rostros con una inclinación muy pronunciada. Experimentalmente, se determinó que la detección del rostro comienza a fallar con valores cercanos a los 20° de inclinación. Por otro lado, sólo se detectan rostros frontales en condiciones de luz adecuada. No se hace análisis de perfiles.

- **Carencia de Reconocimiento Facial:** El sistema propuesto no ocupa reconocimiento facial, esto quiere decir que sólo analiza la cara detectada más cercana a la cámara. Si la persona se retira de la cámara e ingresa otra persona, el sistema no tiene cómo saber que se trata de otra persona.

- **Sólo se analiza un rostro a la vez:** Como se mencionó anteriormente, el sistema propuesto sólo analiza la cara más cercana a la cámara. Si bien es posible analizar más de un rostro, el proceso puede ser costoso computacionalmente, y se omite.

- **Los resultados dependen de las condiciones de ambiente:** Los resultados de predicción dependerán en cierto grado de las condiciones de luz y del conjunto de entrenamiento usado.

6.2.5 Sistema implementado

La implementación que finalmente se utilizó fue similar a la utilizada en [26], es decir, PCA y un clasificador basado en la distancia euclidiana entre los puntos proyectados en la fase de entrenamiento y los de la fase de clasificación.

El entrenamiento consiste en una extracción de las características de los conjuntos de datos pre-clasificados para la construcción de los vectores de características, dicha extracción se hace usando Análisis de Componentes Principales.

El prototipo fue escrito en el lenguaje C++ debido a que existe mejor documentación de OpenCV para este lenguaje. El sistema ocupa el software Cmake para automatizar el proceso de compilación y el link con OpenCV y otras librerías.

Tabla 6.1: Software utilizado en la construcción del prototipo

Sistema Operativo	Gnu/Linux 64 bits
OpenCV	OpenCV 2.4.8
Cmake	Cmake 2.8.12.2
C++	ISO/IEC 14882:2011 (C++11)
C++ compiler	GCC 4.9.1

El prototipo funcional consiste en una ventana que muestra los frames captados por la cámara y una ventana que refleja la emoción detectada con una caricatura. Cuando la emoción detectada cambia, la caricatura cambia. Además de la caricatura, la emoción detectada se puede observar en la salida del programa en la consola del sistema, y la intensidad de la emoción se dibuja en forma de gráfico en la ventaja principal.

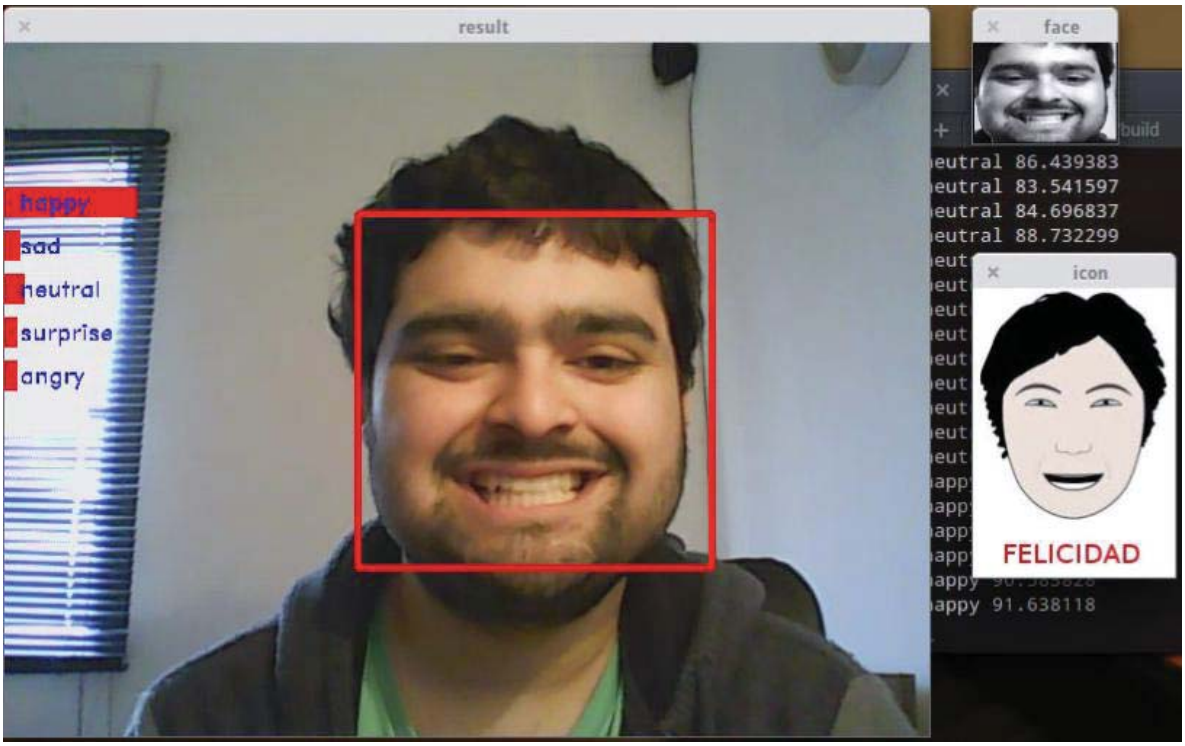


Figura 6.2: Prototipo de sistema mostrando porcentajes para cada emoción detectada

En la figura 6.2 se muestra el prototipo del sistema con el modo webcam en funcionamiento. En la ventana de la derecha se observa la caricatura mostrando la emoción detectada, mientras que en la ventana de la izquierda se muestra el flujo de la cámara web y un gráfico de barras que muestra la intensidad de la emoción detectada. La aplicación permite una serie de opciones por vía de comandos.

6.2.6 Uso del sistema

En esta documentación se explican las funcionalidades incluidas en el sistema y algunas aclaraciones adicionales.

El ejecutable compilado se encuentra en la carpeta REF/src/build. Para arrancar el programa, abra una consola del sistema, diríjase a la carpeta del ejecutable y escriba:

```
$ ./ref
```

En la primera ejecución, si usted ejecuta el programa sin ninguna opción definida, el sistema mostrará la lista de opciones disponibles. Las opciones disponibles en el sistema se explican en la tabla siguiente a modo de resumen. Explicaciones detalladas se encuentran en la tabla 6.2.

Tabla 6.2: Opciones por vía de comandos

Opción	Descripción
-f [file]	Realiza la detección del archivo de imagen especificado. Entre los formatos soportados están los archivos png, jpg, tiff y bmp.
-t [images path] [samples per class] [training subjects]	Lee los archivos previamente preparados para el entrenamiento en la carpeta especificada. El formato esperado es el siguiente: [emotion][number].jpg Por ejemplo: happy1.jpg happy2.jpg … happy20.jpg angry1.jpg angry2.jpg … angry20.jpg El sistema espera recibir la misma cantidad de imágenes de entrenamiento para cada emoción. En caso de no encontrar un rostro en una imagen, el sistema muestra un mensaje de error para dicha imagen.
-c [device number]	Inicia el sistema en modo de tiempo real, utilizando la cámara web del equipo. Se puede especificar el número del dispositivo

	de entrada a utilizar en caso que se cuente con más de una cámara. El sistema utiliza la cámara por defecto si no se especifica otra.
-p [test number]	Ejecuta una prueba predefinida. El sistema incluye 2 pruebas, por lo que los valores esperados para "*test number"* son 1 o 2.

Funcionalidades incluidas:

Modo webcam: El modo webcam es un modo en el cual el sistema captura imágenes desde la cámara web. Para iniciar este modo, debe usar la poción –c. Opcionalmente, usted puede especificar un número de dispositivo si desea ocupar una webcam distinta a la por defecto. Por ejemplo, para usar la webcam 1 en lugar de la 0, usted debe ejecutar:

```
$./ref –c 1
```

Clasificar imagen: Para clasificar una imagen, usted puede hacer uso de la opción –f, a la cual debe incluirle la ruta del archivo con la imagen. Por ejemplo:

```
$./ref –f /ruta/a/mi/imagen.jpg
```

Observación: Entre los formatos soportados se encuentran *.png, *.jpg, *.tiff y *.bmp.

Entrenamiento: Para entrenar el sistema con un conjunto de datos previamente clasificados, puede hacer uso de la opción –t. Como parámetros para esta opción se debe especificar la carpeta donde se encuentran las imágenes, la cantidad de imágenes por emoción, y la cantidad de sujetos de entrenamiento. Posteriormente al entrenamiento, se debe editar el archivo de configuración config.ini los parámetros que allí se piden de acuerdo al entrenamiento efectuado. Por ejemplo, si usted tiene un set de datos con 10 imágenes por emoción, con 2 sujetos de prueba, debe escribir:

```
$./ref –t /carpeta/con/imágenes 10 2
```

Y luego editar el archivo de configuración para que quede de esta forma:

[Trainer]

```
#La cantidad de imágenes de prueba por cada clase
samples_per_class=10
#La cantidad de sujetos de prueba
training_subjects=2
```

Observación: El sistema requiere que las imágenes estén en formato *.jpg. Debe haber la misma cantidad de imágenes por sujeto, y la misma cantidad de imágenes por emoción. El formato para cada emoción debe seguir el mismo formato que la base de datos incluida.

Ejecutar pruebas: El sistema incluye por defecto dos pruebas. Para ejecutar cualquiera de ellas, usted debe utilizar la opción –p, especificando el número de la prueba a ejecutar. Los resultados de la prueba se muestran por consola.

Tabla 6.3: Ejemplos para la ejecución de pruebas

./ref –p 1	Ejecuta la prueba número 1, con 1 sujeto de prueba y 4 imágenes por emoción.
./ref –p 2	Ejecuta la prueba número 2, con 18 sujetos de prueba y 72 imágenes por emoción.
./ref –p 5	Muestra mensaje de error, puesto que la prueba 5 no existe.

6.2.7 Integración con otros sistemas

Para integrar el sistema de reconocimiento de emociones con otros sistemas, se puso a disposición de este para la creación de un sistema que captura las imágenes vía una interfaz web y que se conecta con el sistema de reconocimiento de emociones. El sistema web recibe el *stream* de salida de la consola, que indica la emoción detectada. Los resultados de este trabajo se presentan en [28].

La figura 5.7 muestra la arquitectura del sistema web. De parte del cliente, las imágenes se capturan desde la cámara web utilizando jQuery para este cometido. La imagen captada es procesada y subida al servidor. Luego, un script escrito en python es el encargado de ejecutar el sistema de detección de emociones y enviar la respuesta de la emoción detectada.

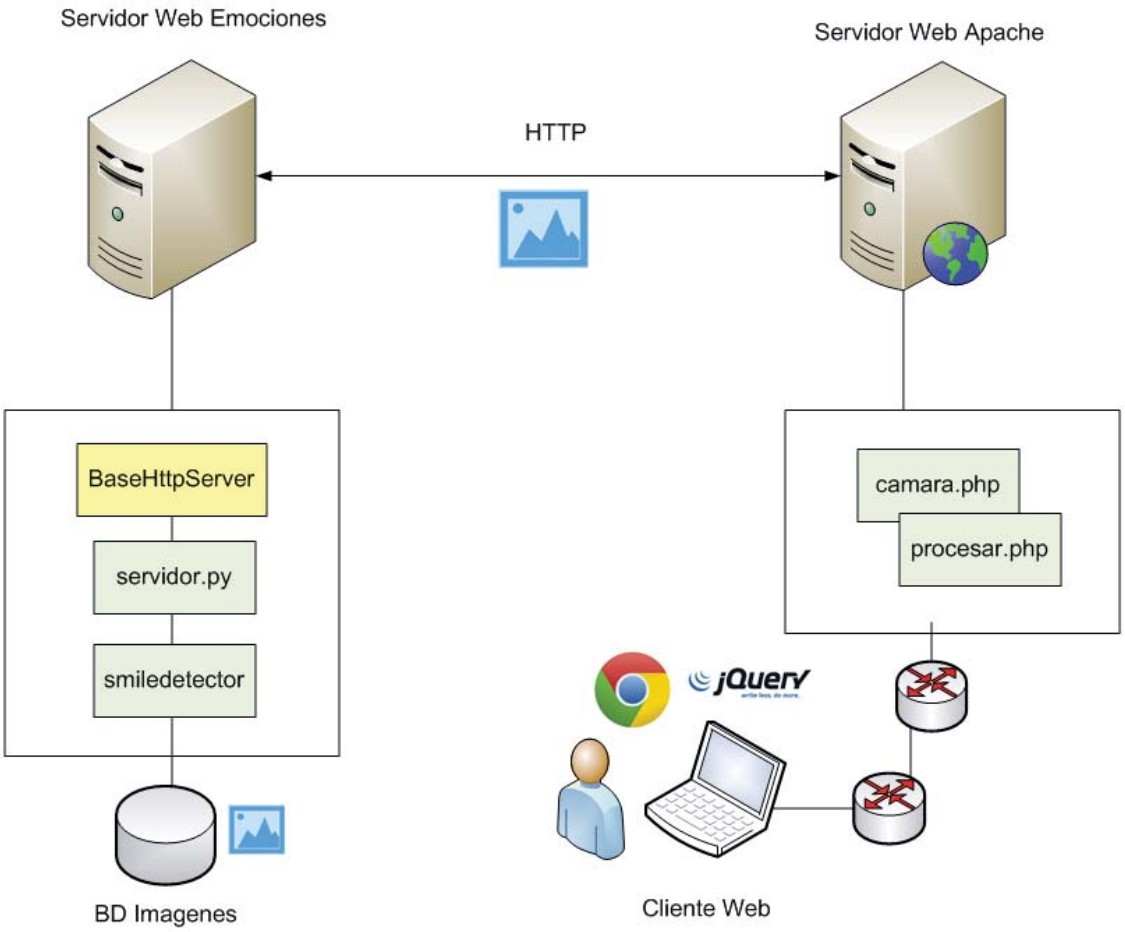


Figura 6.3: Prototipo de sistema web según *Chanchi*

En el presente prototipo se hace uso de un sistema de recomendaciones basado en contexto, el cual utiliza un listado de contenidos previamente clasificados de acuerdo a las emociones del usuario. Este listado está consignado en un documento XML, el cual fue generado a partir de la base de datos provista por la API de Echonest. Esta API permite acceder a un amplio catálogo de características de contenidos multimedia musicales, tales como: ritmo, energía, tono, tempo, duración, arousal, etc. Dichas características son usadas para determinar el estado de ánimo al cual está pertenece una determinada canción. Así mismo, la base de datos XML ha sido filtrada teniendo en cuenta la API de youtube, de tal manera que los contenidos musicales son cotejados con la base de datos de videos de youtube. De acuerdo a la emoción identificada por el módulo de detección de emociones, desde el catálogo XML de contenidos clasificados se presenta en el portal web un listado de contenidos asociados a esa emoción. Una vez el usuario escoge el contenido del listado, la librería jwplayer permite la reproducción del mismo.

Figura 6.4: Prototipo de sistema web en funcionamiento según Chanchi

7. Pruebas y análisis de resultados

A continuación se hace un análisis de los resultados obtenidos en las distintas pruebas realizadas.

7.1 Pruebas de rendimiento

Las pruebas de rendimiento efectuadas se realizaron con el hardware especificado en la tabla 7.1, mientras que los resultados se especifican en la tabla 7.2.

Tabla 7.1: Hardware utilizado para las pruebas

Procesador	Intel Core i3 2350M @ 2.30GHz
Placa de Video	2048MB ATI Radeon HD 7670M (HP)
Cámara web	Logitech C170 USB 2.0
Resolución	800x600

Tabla 7.2: Resultados pruebas de rendimiento

Tiempo promedio de detección por frame	64.93 milisegundos
Frames por segundo promedio	15.4 fps

7.2 Pruebas con imágenes

Con el fin de determinar qué tan bueno es el clasificador utilizado, se han realizado pruebas con imágenes estáticas previamente clasificadas. Se cuantifican la precisión, la exactitud y la tasa de detección por cada clase y promedio.

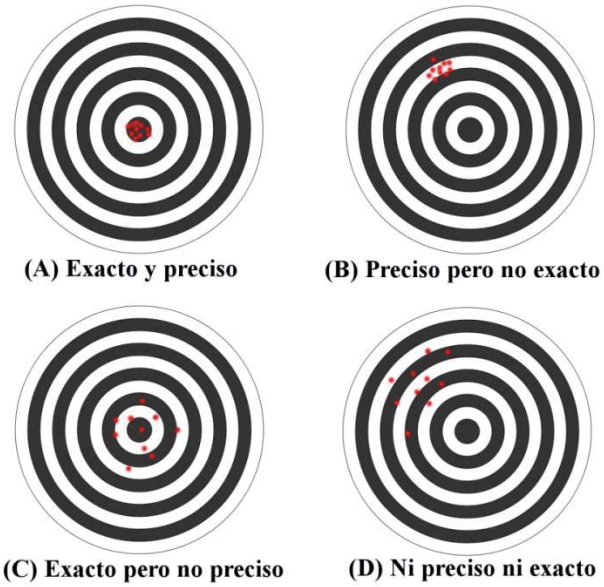


Figura 7.1: Ejemplificación gráfica de los conceptos de precisión y exactitud.

La exactitud permite medir qué tan cerca estuvo la clasificación hecha respecto de su valor real o esperado, siendo mayor cuando el valor obtenido está más cerca del esperado. Por otro lado la precisión permite establecer la dispersión de los resultados obtenidos, siendo esta mayor cuando los datos están menos dispersos. Una representación gráfica de estos conceptos se puede ver en la figura 7.1.

Para la precisión se utiliza la ecuación (17), mientras que para la exactitud se utiliza la ecuación (18). Para ello, se deben cuantificar los falsos positivos (FP), falsos negativos (FN), verdaderos positivos (TP) y verdaderos negativos (TN) para cada clase.

$$(17)\ precision = \frac{TP}{TP + FP}$$

$$(18) accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Una forma de explicar estos conceptos de forma sencilla, es con un ejemplo. Suponiendo que tenemos un clasificador que clasifica animales en perros, gatos y conejos, los valores para la clase gato serían como sigue:

- **Verdaderos positivos:** Es cuando un gato real es clasificado correctamente como gato.
- **Falsos positivos:** Son los casos donde los perros fueron incorrectamente clasificados como gatos.
- **Falsos negativos:** Los casos en los cuales los gatos fueron incorrectamente marcados como perros.
- **Verdaderos negativos:** Los casos restantes en que los animales restantes fueron correctamente clasificados como no gatos.

7.3 Recolección de datos

Para una primera versión del sistema que incluyó sólo 5 emociones, se tomaron fotografías a un total de 6 sujetos de prueba, 8 fotografías por emoción, para dar un total de 240 fotografías. Los resultados de las primeras pruebas utilizan esta base de datos previa. A uno de los sujetos, se le tomó un total de 20 fotografías por emoción extra, con motivo de ejecutar una prueba persona-dependiente, es decir, sólo se utilizaron imágenes de dicho sujeto para entrenar el clasificador.

Para las segundas pruebas con imágenes estáticas, se llevó a cabo la recolección de datos mediante la colaboración alumnos de pregrado y postgrado de la escuela de Informática de la Pontificia Universidad Católica de Valparaíso para la construcción de una base de datos definitiva. Dicha base de datos consta de un total de 18 sujetos de prueba, 8 imágenes por emoción, lo que da un total de 864 fotografías, a una resolución de 800x600.

Se pidió a cada sujeto de prueba que actuara cada emoción. Los sujetos de prueba actuaron según su parecer cada una de las emociones pedidas, sin embargo, en caso de duda, se les mostraba una imagen de referencia.

Las fotografías se tomaron bajo la misma luminosidad, simulando el uso del computador, de forma que la posición del rostro es la posición natural percibida desde el ángulo de la cámara del computador utilizado, sin embargo, no se utilizó una cámara integrada, sino que una de mayor calidad.

7.4 Resultados y análisis de resultados

A continuación se hace el análisis de resultado de los experimentos realizados. Para exponer los resultados se hace uso de una matriz de confusión. Las matrices de confusión son la forma estándar de exponer los resultados de un clasificador, ya que no sólo muestra el porcentaje de clasificaciones correctas, sino que también es posible ver las incorrectas, y dónde ocurrieron dichas clasificaciones, permitiendo observar el comportamiento del clasificador para cada clase y entre clases. Esto es importante ya que permite establecer los niveles de "confusión" del clasificador entre dos clases.

En una matriz de confusión, las filas corresponden a la clase real a la que corresponde la imagen a clasificar, mientras que las columnas indican el resultado que dio el clasificador para dicha imagen.

El primer experimento consistió en tomar 20 fotografías por emoción a un mismo sujeto, lo que da un total de 100 fotografías. Se seleccionaron 10 fotografías por emoción para el entrenamiento, y 10 fotografías para las pruebas de forma aleatoria.

Tabla 7.3: Resultados para un mismo sujeto y 5 emociones

	Felicidad	Enojo	Neutral	Tristeza	Sorpresa	Precisión	Exactitud	Tasa de detección
Felicidad	**90%**	10%	-	-	-	100%	98%	90%
Enojo	-	**60%**	20%	-	20%	85%	90%	60%
Neutral	-	-	**100%**	-	-	66%	90%	100%
Tristeza	-	-	30%	**70%**	-	100%	94%	70%
Sorpresa	-	-	-	-	**100%**	83%	96%	100%
Media	-	-	-	-	-	**86.8%**	**93.6%**	**84%**

La tabla 7.3 muestra los resultados de la primera prueba en una matriz de confusión del clasificador. El clasificador alcanza un promedio de 86.8% de precisión, mientras que la exactitud llega a un 93.6%.

Se puede apreciar que existe un problema con la detección de la emoción Neutral, ya que si bien, todas las imágenes neutrales fueron efectivamente clasificadas como neutrales, hubo una cantidad mayor de falsos positivos respecto a otras clases. Dichos falsos positivos provienen de la clase Enojo y de la clase Tristeza, que corresponden a las clases con menor tasa de detección.

Tabla 7.4: Resultados para 6 sujetos y 5 emociones

	Felicidad	Enojo	Neutral	Tristeza	Sorpresa	Precisión	Exactitud	Tasa de detección
Felicidad	**83%**	17%	-	-	-	96.6%	100%	83%
Enojo	-	**66%**	25%	-	8%	80%	50%	66%
Neutral	-	33%	**50%**	17%	-	80%	60%	50%
Tristeza	-	17%	-	**58%**	25%	86.66%	70%	58%
Sorpresa	-	-	8%	-	**92%**	93.33%	78.57%	92%
Media	-	-	-	-	-	**87.33%**	**71.71%**	**69.8%**

El segundo experimento consistió en tomar 8 fotografías por emoción a 6 sujetos en total. El set de entrenamiento fue de 4 fotografías por emoción, y el set de pruebas las 4 restantes. Se procuró que las fotografías se sacaran en distintas posiciones y con distintas intensidades para emular un uso real, de esa forma se añadió dificultad al experimento.

En esta prueba, nuevamente el clasificador logró buenos resultados clasificando Felicidad y Sorpresa, destacando por sobre las demás, mientras que se evidenció un problema con la clase Neutral y Enojo, donde el clasificador no fue tan exacto, y a la vez, presentó la precisión más baja.

Tabla 7.5: Resultados para un mismo sujeto y 6 emociones

	Felicidad	Tristeza	Neutral	Sorpresa	Enojo	Disgusto	Precisión	Exactitud	Tasa de detección
Felicidad	**90%**	-	-	-	-	10%	100%	83.33%	90%
Tristeza	-	**60%**	30%	-	-	10%	100%	83.33%	60%
Neutral	-	-	**100%**	-	-	-	58.82	76.67%	100%
Sorpresa	-	-	-	**100%**	-	-	83.33%	80%	100%
Enojo	-	-	30%	10%	**60%**	-	75%	80%	60%
Disgusto	-	-	10%	10%	20%	**60%**	75%	80%	60%
Media	-	-	-	-	-	**-**	**82.03%**	**80.56%**	**78.33%**

Tabla 7.6: Resultados para 18 sujetos y 6 emociones

	Felicidad	Tristeza	Neutral	Sorpresa	Enojo	Disgusto	Precisión	Exactitud	Tasa de detección
Felicidad	**69.44%**	6.94%	15.27%	-	5.56%	2.78%	80.64%	92.13%	69.44%
Tristeza	-	**58.33 %**	18.05%	-	22.22%	1.39%	52.50%	84.26%	58.33%
Neutral	5.56%	18.05%	**61.11 %**	1.39%	13.89%	-	53.01%	84.48%	61.11%
Sorpresa	4.17%	11.11%	1.39%	**76.39%**	6.94%	-	98.21%	95.83%	76.39%
Enojo	-	6.94%	5.56%	-	**86.11 %**	1.39%	57.41%	87.04%	86.11%
Disgusto	6.94%	9.72%	13.89%	-	15.27%	**54.17%**	90.70%	91.44%	54.17%
Media	-	-	-	-	-		**72.08%**	**89.20%**	**67.43%**

Los resultados son similares con el trabajo de *Ajit P.Gosavi and S.R. Khot* [26], sin embargo, existen diferencias, las cuales pueden ser explicadas por las particularidades propias de los sujetos de prueba y de la metodología usada donde se pidió a los sujetos que no mantuvieran la expresión fija, sino que intentaran realizarla con distintos ángulos e intensidades. Es importante señalar que en [26] se utilizó la base de datos JAFFE, la cual consta de sólo mujeres con rasgos faciales similares al ser todas las modelos de origen japonés.

Finalmente, el tercer experimento se realizó con el sistema modificado para detectar 6 emociones. La nueva emoción agregada fue disgusto. Se tomaron 4 fotografías por sujeto para el entrenamiento y 4 para la clasificación de forma aleatoria. Los resultados obtenidos, en la tabla 7.5, fueron similares en cuanto a las tasas de detección, siendo la precisión de un 72.08% y la exactitud de un 89.2%. El principal problema lo presentaron las clases Neutral y Enojo, con un número elevado de falsos positivos. Felicidad y Sorpresa volvieron a destacar, con pocos falsos positivos y con una tasa de detección superior. La clase Disgusto también se desempeñó de buena forma, aunque con una tasa de detección menor. Los resultados completos con número de ocurrencia en lugar de porcentajes, se pueden ver en el la tabla 7.7.

Se construyeron gráficos con el fin de comparar visualmente la diferencia de rendimiento del clasificador cuando se utilizó 1 sujeto y 18 sujetos de prueba y de entrenamiento. También se graficó la diferencia de rendimiento al utilizar 5 tipos de emociones o 6 tipos de emociones con cantidad de sujetos constante.

Tabla 7.7: Resultados para 18 sujetos y 6 emociones, número de casos

	felicidad	tristeza	neutral	sorpresa	enojo	disgusto	TP	TN	FP	FN	Precisión	Exactitud	Tasa de detección
felicidad	**50**	5	11	0	4	2	50	348	12	22	80.64%	92.13%	69.44%
tristeza	0	**42**	13	0	16	1	42	322	38	30	52.50%	84.26%	58.33%
neutral	4	13	**44**	1	10	0	44	321	39	28	63.01%	84.49%	61.11%
sorpresa	3	8	1	**55**	5	0	55	359	1	17	98.21%	95.83%	76.39%
enojo	0	5	4	0	**62**	1	62	314	46	10	57.41%	87.04%	86.11%
disgusto	5	7	10	0	11	**39**	39	356	4	33	90.70%	91.44%	54.17%
Media	-	-	-	-	-	-	-	-	-	-	**72.09%**	**89.20%**	**67.43%**

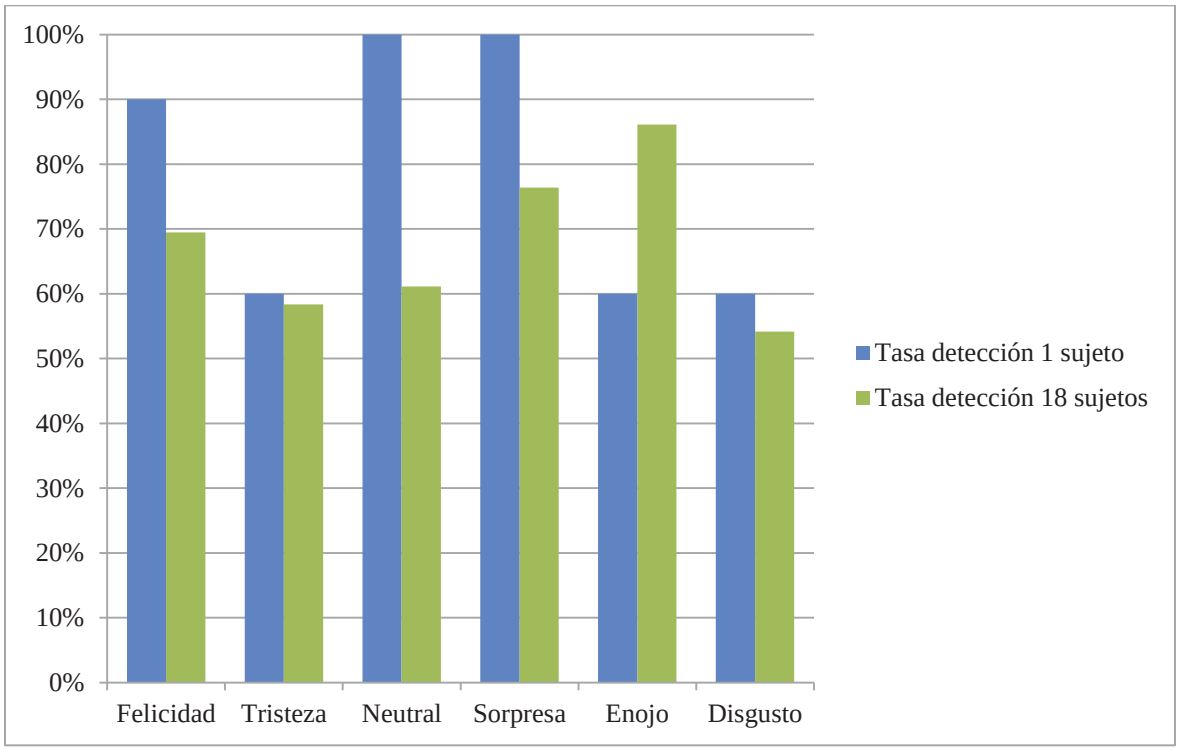


Figura 7.2: Tasa de detección para cada emoción utilizando 1 y 18 sujetos

En la figura 7.2, se compara la tasa de detección para cada emoción utilizando 1 y 18 sujetos. Se observa claramente que la tasa de detección disminuye al aumentar la cantidad de sujetos, con la excepción de la emoción Enojo, donde la detección se incrementa notablemente. Este hecho puede ser explicado debido a la naturaleza misma de los datos, sin embargo es posible concluir que a mayor número de sujetos de prueba, menor es la tasa de detección del clasificador.

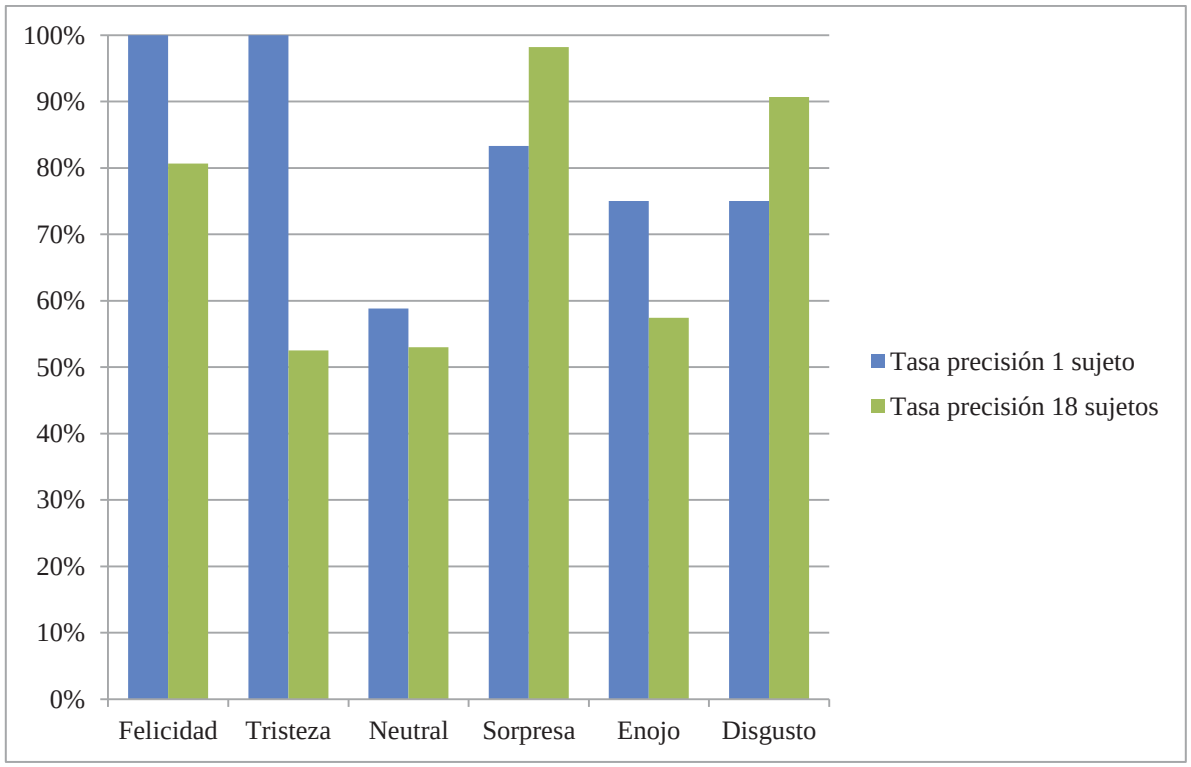


Figura 7.3: Precisión utilizando 1 y 18 sujetos de prueba

Por otro lado, al comparar la precisión del clasificador, en la figura 7.3, al utilizar 1 sujeto y 18 sujetos de prueba y entrenamiento, se observa una situación similar a la anterior. Esta vez, el clasificador que usa 18 sujetos nuevamente supera a la versión que sólo utiliza uno en la clase Sorpresa. Se añade a esto la clase Disgusto, que tuvo tasas de detección similar pero que en precisión supera a la versión de 1 sujeto.

Al comparar la exactitud del clasificador, se ve claramente que el clasificador que utiliza 18 sujetos es más exacto que su contraparte que utiliza sólo 1 sujeto, sin embargo la diferencia entre ambos no es tanta como si lo es la precisión y tasa de detección. La mayor variación de exactitud observada, se ve en la clase Sorpresa y Disgusto.

A continuación se comparó la diferencia de precisión, exactitud y tasa de detección para el clasificador utilizando 5 clases o emociones versus 6, ambas con 1 sujeto de prueba y entrenamiento. Se puede observar que la cantidad de clases o emociones afecta al clasificador. El clasificador que utiliza 6 emociones obtiene resultados peores que su contraparte que sólo utiliza 5 emociones. Sin embargo, esta diferencia no es excesiva. Sólo en el parámetro de exactitud se ve una diferencia mayor bastante significativa.

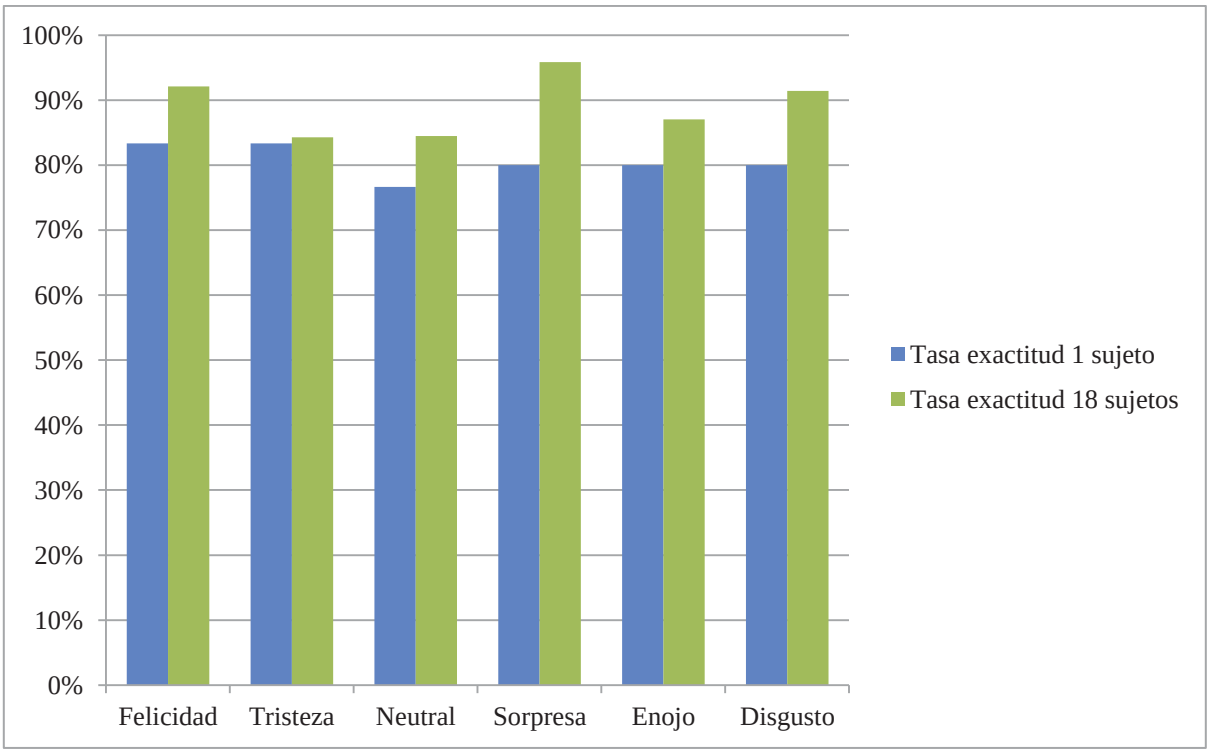


Figura 7.4: Exactitud utilizando 1 y 18 sujetos

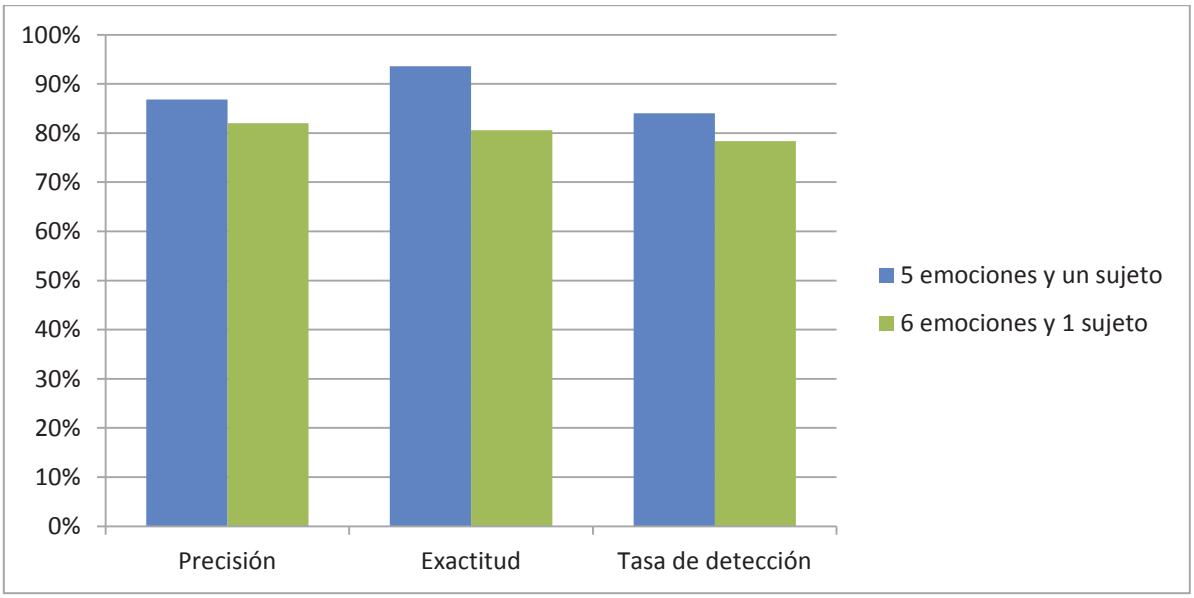


Figura 7.5: Precisión, exactitud y tasa de detección para clasificador con 5 emociones y con 6 emociones

Finalmente, al comparar la precisión, exactitud y tasa de detección del clasificador utilizando el número de sujetos de prueba como parámetro variable, se observa que la versión que utiliza sólo 1 sujeto de prueba logra una mayor precisión y tasa de detección, sin embargo, es superado por el clasificador que utiliza 18 sujetos en el parámetro de

exactitud, esto era un resultado esperable al observar los resultados de la figura 7.4, donde el clasificador que utilizó 18 sujetos supera en todas las clases la exactitud del clasificador que utiliza sólo 1 sujeto de prueba.

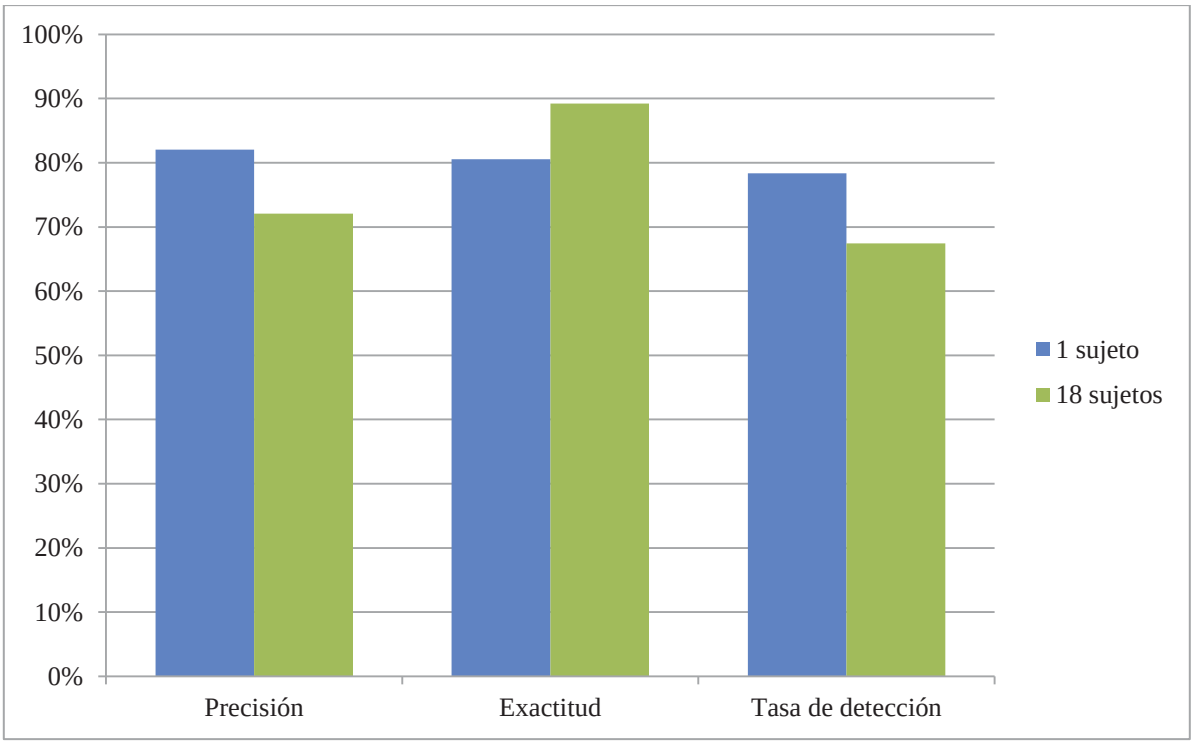


Figura 7.6: Precisión, exactitud y tasa de detección para clasificador con 1 y 18 sujetos de prueba

Esto nos dice que al aumentar el número de sujetos de prueba, logramos una mayor exactitud al costo de una menor precisión. Esto es, después de todo, un comportamiento esperable. Al haber un único sujeto de prueba, el clasificador tenderá a ser más preciso ya que existe menos dispersión de los datos. En cuanto a una exactitud más alta a mayor número de sujetos de prueba, también es esperable, ya que al tener más datos de prueba como ejemplo, el clasificador debería ser capaz de encontrar uno más cercano y similar de manera más sencilla. El encontrar un objeto similar del set de entrenamiento tampoco implica que pertenezca a la misma clase debido a las particularidades propias de la emoción expresada por cada sujeto, y eso termina por explicar que aunque la exactitud aumenta, no lo hace la tasa de detección, la cual sigue el mismo comportamiento que sigue el parámetro de precisión, al menos en sus tasas promedio.

8. Conclusiones

El problema del reconocimiento de expresiones faciales no ha sido resuelto completamente, y los enfoques existentes tienen tanto ventajas como desventajas. La detección en tiempo real requiere una gran cantidad de capacidad de cálculo, sin embargo, el rápido avance de la tecnología podría hacer posible este tipo de tecnologías más común en nuestros computadores en un futuro.

En este proyecto se ha abordado la problemática de la detección de emociones a través de las expresiones faciales, logrando construir un prototipo con éxito relativo con una precisión promedio del 67,59% en un modo persona no dependiente, y reproduciendo los resultados de [26]. Tal como se muestra en la sección de resultados, el clasificador evidenció un número mayor de falsos positivos en las clases Neutral, Enojo y Tristeza, mientras que la clase Felicidad y Sorpresa mostraron mejores resultados. Es importante señalar que, tal como determinó Ekman [3], ni siquiera el ser humano es capaz de clasificar las emociones básicas a partir de expresiones faciales con un 100% de precisión, y dicha tarea parece ser más complicada de lo que parece.

Si bien los sistemas existentes revisados logran ser fiables cuando son persona dependiente, los resultados no necesariamente son igualmente de buenos con sistemas persona no dependiente, siendo este uno de los principales problemas de la detección de emociones. Este fenómeno se apreció durante la implementación del sistema. Se evidenció que la gran variabilidad de los datos entre dos rostros de diferentes personas hace dificultosa la clasificación de las emociones, obteniendo malos resultados. Esto llevó a construir un modelo distinto en el cual se calculó una matriz de proyección para cada sujeto en particular en lugar de una única proyección para todo el conjunto de datos. Este enfoque logró resolver en parte el problema en casos con gran cantidad de sujetos de prueba, obteniendo resultados mucho mejores. En el caso de un sistema persona-dependiente, PCA resultó ser una técnica tremendamente eficaz y eficiente, debido a que en el caso de una misma persona, la variabilidad de los datos se encuentra en las expresiones particulares correspondientes a cada emoción, y no a las características específicas del rostro de la persona.

Una posibilidad interesante es la integración con algún sistema de reconocimiento facial. Detectar primero el rostro más cercano, y luego realizar la clasificación de acuerdo a la matriz de proyección correspondiente a ese rostro en particular, podría ser un enfoque mucho más escalable y probablemente más eficiente.

Como trabajo a futuro queda la inquietud de experimentar con máscaras en 3D. Intentar clasificar un objeto 3D, como lo es el rostro humano, con imágenes en 2d, podría no ser la mejor solución, ya que una rotación en cualquiera de los ejes puede afectar bastante los resultados. Diversos trabajos han utilizado máscaras 3D para medir las expresiones faciales, por lo que esta podría ser una evolución natural del enfoque propuesto. Una máscara 3D tiene la ventaja que luego, se puede crear una proyección 2d a

partir de la primera, que se encuentre siempre alineada, resolviendo el problema del alineamiento del rostro y eliminando la restricción del sistema desarrollado de funcionar únicamente con rostros frontales.

La iluminación es un factor importante también. Algoritmos como el de Viola/Jones pueden presentar problemas en la detección de caras en condiciones de poca iluminación, y la clasificación se puede volver menos precisa también en estas condiciones aunque se lleve a cabo el proceso de ecualización del histograma de la imagen. Diversos filtros de detección de bordes pueden servir para este propósito, tal como el filtro de Canny, sin embargo, estos filtros suelen requerir parámetros de umbrales, que pueden variar en sus valores óptimos dependiendo de la iluminación. Un umbral muy amplio genera muchos bordes inexistentes, mientras que un umbral muy pequeño puede terminar en pocos bordes detectados. Este filtro se implementó y se probó, sin embargo, no se apreciaron mejoras en la clasificación, sin embargo, este tipo de filtros se puede utilizar para encontrar puntos claves en el rostro, puntos que librerías como Stasm no logran detectar eficientemente en rostros no neutrales, y que fue uno de los motivos de abandonar su uso.

La utilización de lógica difusa podría ser una opción a tomar en cuenta. Está claro que las emociones no son estados binarios, y generalmente una expresión se genera por la mezcla de distintas expresiones. En este trabajo se logró calcular la intensidad de la emoción utilizando la cara neutral como referencia. Este método resultó ser sencillo y confiable, sin embargo, puede ser mejorado.

Respecto a la eficiencia, se concluye que un número grande de imágenes de entrenamiento termina por ralentizar el inicio del sistema. Este problema podría ser resuelto guardando las matrices de proyección, y el conjunto de entrenamiento ya proyectado, en un archivo. De esta forma, se puede cargar los datos ya procesados en lugar de procesarlos cada vez que se inicia el sistema.

La integración de sistemas capaces de reconocer las emociones con otros sistemas, tiene una potencialidad inimaginable. Cada vez toma más importancia el aspecto afectivo en la interacción persona-computador y persona-robot. Una de estas potencialidades se mostró al integrar el sistema desarrollado con un sistema de recomendaciones multimedia, con resultados muy interesantes.

El reconocimiento de expresiones faciales para el análisis de emociones del usuario no es la única forma indirecta de obtener dichas emociones, sino que hay otras técnicas, tales como el reconocimiento del habla, de gestos e incluso el uso de sensores. La unión de dichas técnicas en sistemas multimodales podría aumentar la capacidad de la máquina de entender las emociones del usuario y reaccionar con mayor precisión tanto a la emoción como al contexto, disminuyendo la frustración en la interacción persona-computador. Dotar a la máquina con capacidades humanas no debe ser el objetivo. Las máquinas deben ir más allá y brindarnos habilidades que no poseemos y que nos limitan. Máquinas capaces de “aprender”, “ver” y “pensar” cambiarán la forma en que interactuamos con ellas en un futuro, de maneras que actualmente no podemos imaginar.

9. Referencias

[1] N. Fragopanagos, J.G. Taylor, "Emotion recognition in human–computer interaction", Neural Networks 18, 2005 pp 389–405.

[2] R.W. Picard, E. Vyzas, J. Healey, "Toward machine emotional intelligence: analysis of affective physiological state". IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 2001 pp1175–1191.

[3] P. Ekman, W.V. Friesen, "Unmasking the face: a guide to recognizing emotions from facial clues". Englewood Cliffs, NJ: Prentice-Hall; 1975.

[4] C. Reynolds and R. Picard, "Designing for affective interactions". In Proceedings of 9th International Conference on Human-Computer Interaction, August 5–10, 2001, New Orleans, Louisiana, USA.

[5] R.W. Picard, J. Klein, "Computers that recognize and respond to user emotion: theoretical and practical implications". Interacting with Computers 14(2), 2002, pp141–169.

[6] B. Reeves and C.I. Nass, "The media equation: how people treat computers, television, and new media like real people and places". Cambridge University Press, 1996.

[7] C. Busso, Z. Deng, S. Yildirim, M. Bulut, M. C. Lee, A. Kazemzadeh, S. Lee, U. Neumann and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information". Sixth International Conference on Multimodal Interfaces, ICMI 2004, 205–211.

[8] O. Çeliktutan, S. Ulukaya and B. Sankur, "A comparative study of face landmarking techniques", EURASIP Journal on Image and Video Processing 2013, 2013:13.

[9] S.V. Ioannou, A.T. Raouzaniou, V.A. Tzouvaras, T.P. Mailis, K.C. Karouzis, S.D. Kollias, "Emotion recognition through facial expression analysis based on a neurofuzzy network". Elsevier Neural Networks 18, 2005, pp 423–435.

[10] Z. Hammal, L. Couvreur, A. Caplier. & M. Rombaut, "Facial expression recognition based on the belief theory: Comparison with different classifiers". In: Roli, F. & Vitulano, S., (Eds.). Image Analysis and Processing. Heidelberg: Springer-Verlag, v. 3617 de Lecture Notes in Computer Science, 2005, pp 743–752.

[11] S.V. Ioannou, G. Caridakis, K.C. Karpouzis and S.D. Kollias, "Robust feature detection for facial expression recognition". Journal on Image and Video Processing, 2, 2007.

[12] G. Castellano, L. Kessous and G. Caridakis, “Emotion recognition through multiple modalities: Face, body gesture, speech”. In: C. Peter and R. Beale, (Eds.). Affect and Emotion in Human-Computer Interaction. Heidelberg: Springer-Verlag, v. 4868 de Lecture Notes in Computer Science, 2008, pp 92–103.

[13] T.F. Cootes and C.J. Taylor. “Active shape models–smart snakes”. In Proc. of British Machine Vision Conference, 1992, pp 266–275.

[14] Lanitis, C.J. Taylor, and T.F. Cootes. “An automatic face identification system using flexible appearance models”. Image and Vision Computing, 13(5), 1995, pp 393–401.

[15] S. Milborrow and F. Nicolls, “Active shape models with SIFT descriptors and MARS”, VISAPP 2014.

[16] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. “Active shape models - their training and application”. Computer Vision and Image Understanding 61(1), 1995, 38–59.

[17] S.Z. Li and A.K. Jain, “Handbook of face recognition”, Springer, New York, 2005.

[18] T.F. Cootes, G. Edwards, and C.J. Taylor. “Active appearance models”. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 2001, pp 681–685.

[19] Matthews and S. Baker. “Active appearance models revisited”. International Journal of Computer Vision, 60(2), 1992, pp 135–164.

[20] M.B. Stegmann, B.K. Ersboll, and R. Larsen. “FAME–a flexible appearance modeling environment”. IEEE Transactions on Medical Imaging 22(10), 2003, pp 1319–1331.

[21] C.H.. Hjortsjö, Man's face and mimic language, 1969.

[22] P. Ekman, and W.V. Friesen, “Facial action coding system (FACS): manual”. Consulting Psychologists Press, 1978.

[23] Y. Piparsaniyan, V.K. Sharma, K.K. Mahapatra, ”Robust facial expression recognition using Gabor feature and bayesian discriminating classifier”. IEEE International Conference on Communicatoin & Signal Processing ICCSP 2014, 3rd-5th April 2014, Adhiparashakti Engineering College, Melmaruvathur, Tamilnadu.

[24] Halder, A. Jati, G. Singh, A. Konar, A. Chakraborty, R. Janarthanan, “Facial action point based emotion recognition by principal component analysis”, SocProS (2) 2011: 721-733.

[25] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose", IEEE Trans. Pattern Anal. Mach. Intelligence, 23(6), pp. 643-660, 2001.

[26] A.P. Gosavi and S.R. Khot, "Facial expression recognition using principal component analysis", International Journal of Soft Computing and Engineering (IJSCE), 3(4), 2013.

[27] A.P.Gosavi and S.R. Khot, "Facial expression recognition using principal component analysis with singular value decomposition", International Journal of Advance Research in Computer Science and Management Studies 1(6), 2014.

[28] G. Chanchi, "Arquitectura basada en contexto para el soporte del servicio de VOD de IPTV móvil, apoyada en sistemas de recomendaciones y streaming adaptativo", Anteproyecto de Tesis de Doctorado, Colombia, 2014.