UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL TUCUMÁN

ESPECIALIDAD EN SISTEMA DE INFORMACION

# Sistema de Domótica para regular el estado emocional

*Author:*
Franco Agustin VILLAGRA

*Supervisor:*
Dr. Jorge Sardiñas GOTAY

*Una tesis presentada en cumplimiento de los requisitos para el posgrado de Especialista en Sistema de Información*

.
.

28 de diciembre de 2021

# Declaración de autoría

Yo, Franco Agustin VILLAGRA, declaro que esta tesis titulada, «Sistema de Domótica para regular el estado emocional» y el trabajo presentado en ella es de mi autoría. Confirmo que:

- Este trabajo se realizó total o principalmente mientras postulaba para un título de posgrado en esta Universidad.
- He dado crédito a cualquier parte de esta tesis que haya sido previamente sometida para obtener un grado académico de posgrado maestría u otro tipo de titulación en esta o cualquier otra universidad.
- He dado crédito a cualquier trabajo previamente publicado que se haya consultado en esta tesis
- He citado el trabajo consultado de otros autores , y la fuente de donde los obtuve.
- He dado crédito a las contribuciones de mis coautores, cuando los resultados corresponden a un trabajo colaborativo.
- Esta tesis esta basada en un trabajo hecho por mi, con excepción de las citas indicadas.

Firma:

Fecha:

UNIVERSIDAD TECNOLÓGICA NACIONAL

# *Resumen*

Facultad Regional Tucumán

Especialista en Sistema de Información

**Sistema de Domótica para regular el estado emocional**

by Franco Agustin VILLAGRA

La evolución de herramientas tecnológicas promete mejorar la calidad de vida y junto a estas una de las ramas tecnológicas más importante es la Domótica, la cual tiene la capacidad de automatizar una vivienda o edificación cualquiera incluyendo la integración de servicios de gestión energética, seguridad, comunicación y bienestar por medio de redes de comunicación cableados o inalámbricos. Este trabajo se enmarca en un proyecto más general cuyo propósito es predecir el estado emocional de una persona que ingresa al hogar para ambientar la casa de acuerdo con el estado emocional detectado, sin embargo, el alcance de este trabajo presupone únicamente la predicción de cuatro estados emocionales (alegre, enojado, neutral y sorprendido) a partir de la imagen del rostro de la persona que ingresa al hogar mediante el uso de técnicas del Deep Learning. En el trabajo los modos de ambientación de la casa son de carácter demostrativo.

Se utilizaron dos métodos para realizar la clasificación de los estados emocionales: EL primero, está basado en una red neuronal profunda del tipo convolucional o CNN por sus siglas en inglés (Convolutional Neural Network) que analiza la imagen completa del rostro de la persona y se encarga de extraer de manera automática las características fundamentales del rostro para establecer la clasificación. El segundo método, utiliza un vector de características que contiene las marcas biométricas presentes en el rostro que son propias de cada persona. Este vector de características es suministrado a un clasificador del tipo Máquina de Soporte Vectorial o (SVM) por sus siglas en inglés (Support Vector Machine).

Para ambos tipos de métodos se utilizó el lenguaje de programación Python con la biblioteca de código abierto TensorFlow. En el caso de la red neuronal de convolución el código de TensorFlow fue corrido mediante la librería de redes neuronales también de código abierto Keras y para el segundo método, se utilizó la librería dlib.

Los resultados de los modelos realizados muestran que el método basado en el conjunto de puntos claves del rostro brinda mayor porcentaje de acierto en la clasificación de los estados emocionales que cuando se utiliza el método basado en la imagen completa basado en el uso de la red neuronal de convolución.

# *Agradecimientos*

En estas líneas quiero agradecer a todas las personas que hicieron posible esta investigación y a las que de alguna manera estuvieron conmigo en los momentos difíciles, alegres, y tristes. Estas palabras son para ustedes. A mis padres y hermanos por todo su amor, comprensión y apoyo pero sobre todo gracias infinitas por la paciencia que me han tenido. No tengo palabras para agradecerles las incontables veces que me brindaron su apoyo en todas las decisiones que he tomado a lo largo de mi vida.

A mis amigos. Con todos los que compartí dentro y fuera de las aulas. Aquellos amigos del colegio y universidad, que se convierten en amigos de vida y aquellos que serán mis colegas, gracias por todo su apoyo y acompañamiento.

De manera especial a mi tutor de tesis Dr. Jorge Gotay , por haberme guiado en la elaboración de este trabajo de posgrado y haberme dado recomendaciones esenciales para que este trabajo sea realizado de la mejor manera , y a todos los profesores que hicieron posible mi formación , haciendo especial mención al Mg. Mario Figueroa quien me motivo con esta temática , indicándome que seria un tema interesante para la especialidad ...

# Índice general

**Declaración de autoría** III

**Resumen** V

**Agradecimientos** VII

**1. Introducción** 1
1.1. La domotica y las expresiones faciales . . . 1
1.2. Motivacion . . . 1
1.3. Objetivos . . . 2
1.3.1. Objetivo General . . . 2
1.3.2. Objetivo Especifico . . . 2

**2. Marco Teórico** 3
2.1. Las emociones y teoría de la mente . . . 3
2.1.1. Emociones . . . 3
2.1.2. Teoria de la Mente . . . 3
2.2. Patrones de reconocimiento . . . 4
2.2.1. Modelos Discretos . . . 4
2.2.2. Modelos Continuos . . . 5
2.3. Aprendizaje supervisado Vs aprendizaje no supervisado . . . 6
2.3.1. Algoritmos para la clasificación - con aprendizaje supervisado. 7
2.4. Problemas de Regresión Vs Problemas de clasificación . . . 8
2.5. Redes neuronales Artificiales . . . 9
2.5.1. ¿Qué son las redes neuronales artificiales? . . . 9
2.5.2. El perceptron multicapa . . . 11
2.5.3. Validación cruzada y Entrenamiento . . . 12
2.5.4. Tamaño y arquitectura de la red . . . 13
2.5.5. Funciones activación . . . 13
Funcion Sigmoide . . . 13
Función tangente hiperbolica . . . 14
Unidad lineal Rectificada (ReLU) . . . 15
Leaky/paramétrica ReLU) . . . 16
ReLU randomizada . . . 16
Softmax . . . 16
2.6. Redes neuronales convolucionales . . . 16
2.6.1. Historia y Fundamentos Biológicos de las Redes Neuronales Convolucionales . . . 17
2.6.2. Arquitectura de las Redes Neuronales Convolucionales . . . 17
2.6.3. ¿Qué es una convolución? . . . 18
2.6.4. Tipos de Capas y Neuronas en Redes Convolucionales . . . 19
2.7. Herramientas para la implementación de los clasificadores . . . 20
2.7.1. Python, Tensor Flow y Keras . . . 20

Python . . . 20
Tensorflow . . . 21
Keras . . . 22
2.7.2. OpenCV . . . 22
2.7.3. Interfaz grafica . . . 22
Diseño de la interfaz grafica . . . 24
Partes de la interfaz grafica . . . 24
2.8. Detección de rostro humano implementado en OpenCV . . . 25
2.8.1. Haar-like features . . . 25
2.8.2. Imagen Integral . . . 26
2.8.3. Proceso de aprendizaje . . . 27
2.8.4. Cascada atencional . . . 29
2.8.5. Proceso de detección . . . 30
2.9. Detección de Landmarks (puntos faciales) y DLIB . . . 31

**3. Configuracion experimental** **33**
3.1. Preparación de la base de datos (entrenamiento, validación y testeo) para los diversos modelos . . . 33
3.2. Modelos basados en redes neuronales de convolución . . . 34
3.3. Modelo basado en los puntos característicos del rostro y el clasificador SVM. . . . 34
3.3.1. Detección de puntos característicos (Landmarks) . . . 34
3.3.2. Implementación del clasificador SVM . . . 34
3.4. La interfaz gráfica en Mod2 y Mod3 . . . 34
3.5. El sistema de reconocimiento de emociones en un caso real . . . 35
3.6. Mejorando la predicción del sistema de reconocimiento de emociones para caso real . . . 35

**4. Analisis de los resultados** **37**
4.1. Resultados utilizando Mod1 y Mod2 . . . 37
4.2. Resultados utilizando Mod3 . . . 39
4.2.1. Resultados de la detección de puntos característicos (Landmarks) . . . 39
4.2.2. Resultados de la implementación de SVM . . . 39
4.3. Comparación de los modelos Mod2 y Mod3 . . . 39
4.4. Verificación de la interfaz . . . 40
4.4.1. Resultado de la pantalla de captura de imágenes . . . 40
4.5. Resultados del sistema en un caso real. . . . 41
4.6. Resultados del sistema mejorado para caso real . . . 43
4.6.1. Resultados del análisis con la Red Neuronal Convolucional en los diferentes escenarios . . . 43
Aciertos de las tomas en los 4 escenarios con Redes Neuronales Conv.(RNC) . . . 44
4.6.2. Resultados del análisis con SVM en los diferentes escenarios . . . 45
Resultados del análisis con SVM, en la primera toma en los cuatro escenarios . . . 45
Aciertos de las 5 tomas en todos los escenarios con SVM . . . 46
4.7. Comparación y conclusiones de los modelos e implementaciones realizadas . . . 47

**5.** **49**
5.1. Conclusión y recomendaciones . . . . . 49

**6.** **51**
6.1. Trabajos a futuro . . . . . 51

**Bibliografía** **53**

# Índice de figuras

2.1. Modelo Tridimensional Continuo de las Emociones . . . . . 6
2.2. Comparación entre la neurona biológica (A) y la neurona artificial (B). 10
2.3. Capacidad de decisión de las redes neuronales artificiales . . . . . 12
2.4. Criterio de parada del proceso de entrenamiento. . . . . 13
2.5. Gráfico de la función sigmoide y su derivada . . . . . 14
2.6. Gráfico de la función tangente hiperbólica y su derivada. . . . . 15
2.7. Gráfico de la función ReLU. . . . . 15
2.8. ReLU estándar,Leaky/paramétrica ReLU y ReLU randomizada . . . . 16
2.9. Estructura clásica de una red convolucional para clasificación . . . . . 18
2.10. Ejemplo de operación de convolución . . . . . 19
2.11. Ejemplo de Max Pooling. . . . . 20
2.12. Logo de Python . . . . . 21
2.13. Ejemplos de features , (Paul Viola and Michael J. Jones, 2001) . . . . . 26
2.14. Ejemplo del cálculo de la suma de píxeles en un rectángulo . . . . . 27
2.15. Representación visual del proceso de AdaBoost . . . . . 29
2.16. Descripción esquemática de una cascada atencional. . . . . 30
2.17. Distribución de los puntos faciales del modelo de 68 Landmarks . . . . 31
2.18. Puntos encontrados en la cara con la librería dlib. . . . . 32

3.1. Cantidad de imágenes para entrenamiento, validación y test . . . . . 33

4.1. Grafica de exactitud del entrenamiento y validación con 256 neuronas 37
4.2. Grafica de exactitud del entrenamiento y validación con 512 neuronas 38
4.3. Resultado del programa con la librería DLIB encontrando los puntos faciales del rostro. . . . . 39
4.4. Interface Grafica antes de capturar fotos del rostro con la cámara Web. 40
4.5. Interfaz gráfica, capturando imágenes con la cámara web . . . . . 41
4.6. Interfaz gráfica, con las fotos capturadas mediante la cámara web. . . . 42
4.7. Resultados del análisis de las imágenes con red neuronal convolucional 42
4.8. Resultados del análisis de las imágenes con SVM . . . . . 42
4.9. Resultados del análisis con red neuronal convolucional, para los 4 escenarios . . . . . 43
4.10. Resultados del análisis con SVM, para los 4 escenarios . . . . . 45

# Índice de cuadros

2.1. Conjuntos de emociones básicas propuestas por diferentes autores . . 5
2.2. Clasificación de las redes neuronales artificiales según tipo de aprendizaje y arquitectura. . . . . 11

3.1. Configuración de Mod1 y Mod2. . . . . 34

4.1. Exactitud de Mod1 y Mod2 . . . . . 37
4.2. Exactitud de Mod2 y Mod3. . . . . 39
4.3. Aciertos y fallas en el análisis con red neuronal convolucional, y SVM . 43
4.4. Resultado del analisis con RNC en Escenario Nř 1 . . . . . 44
4.5. Resultado del analisis con RNC en Escenario Nř 2 . . . . . 44
4.6. Resultado del analisis con RNC en Escenario Nř 3 . . . . . 44
4.7. Resultado del analisis con RNC en Escenario Nř 4 . . . . . 44
4.8. Resultado del analisis con SVM en Escenario Nř 1 . . . . . 46
4.9. Resultado del analisis con SVM en Escenario Nř 2 . . . . . 46
4.10. Resultado del analisis con SVM en Escenario Nř 3 . . . . . 46
4.11. Resultado del analisis con SVM en Escenario Nř 4 . . . . . 46

# Lista de abreviaciones

| | |
|---|---|
| **RNC** | **R**ed **N**euronal **C**onvolucional |
| **SVM** | **S**upport **V**ector **M**achine |
| **Mod1** | **M**odelo **1** |
| **Mod2** | **M**odelo **2** |
| **Mod3** | **M**odelo **3** |

***Esta tesis esta dedicada a:***

*Mis padres quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueno mas, gracias por inculcar en mi el ejemplo de esfuerzo y valentía.*

*Mis hermanos Noelia y Pablo por su cariñóo y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias.*

*A toda mi familia porque con sus oraciones, consejos y palabras de aliento hicieron de mi una mejor persona y de una u otra forma me acompanan en todos mis suenos y metas.*

*A todos los profesores y personas que hicieron posible que llegue a cumplir este objetivo, de tener una especialidad en sistema de informacion.*

.

# Capítulo 1

# Introducción

## 1.1. La domotica y las expresiones faciales

El avance de la Tecnologías de la información y la incidencia de esta en prácticamente todos los ámbitos de la vida ha producido que en estos últimos tiempos se haya hablado con mucha frecuencia de edificios inteligentes, viviendas domóticas, ciudades automatizadas. El hombre ha venido desarrollando, inventando y adaptando avances tecnológicos para su hogar. Los motivos han ido cambiando con el tiempo, aumentar la seguridad y hacer de la vivienda un lugar más confortable probablemente fueron los primeros objetivos de estas tecnológicas aplicadas a las viviendas. Posteriormente el ahorro energético, la mejora de la salud e higiene han sido otras metas importantes. En la actualidad ya se dispone de viviendas con estas características, y podemos encontrar casas equipadas con importantes sistemas los cuales disponen de diversos periféricos de entrada, cámaras web, luces, computadoras, etc. y periféricos de salida los cuales pueden estimular a las personas del hogar proporcionándole un mejor confort ((Cristian Alejandro Rojas T., 2019)) .

Para definir parte de lo anteriormente explicado surge el termino Domótica, cuyo fin es cubrir las necesidades de los habitantes del hogar, que se pueden derivar en numerosos aspectos: facilitar el control integral de la casa; aumentar la seguridad; incrementar el confort; mejorar las telecomunicaciones; ahorrar recursos naturales, dinero y tiempo; facilitar la oferta de nuevos servicios, etc. ((Stefan Junestrand, 2005)).

Mientras se dieron avances en el estudio de la tecnología, también los hubo en el estudio del comportamiento humano los cuales sirvieron para poder predecir ciertos hábitos en las personas; estos estudios en los últimos tiempos han evolucionado satisfactoriamente. Estudios realizados en el área de la Psicología muestran que existe una alta correlación entre las expresiones faciales y el estado de emoción presente en las personas ((Arias., 2006)) . Tanto es así que las expresiones faciales también ayudan a regular las interacciones entre personas, como ser conversaciones. Por lo que se puede postular que las expresiones del rostro juegan un rol importante en la interacción humana y en la comunicación no verbal((Saket S Kulkarni, 2009)). La clasificación de expresiones faciales adaptada a un computador podría ser utilizada como una excelente herramienta para que el hogar interactúe con una persona.

## 1.2. Motivacion

En la actualidad el estudio de la visión artificial está marcando un paradigma en el procesamiento de imágenes digitales, con la finalidad de extraer información del mundo real partiendo de imágenes o videos por medio del computador. El procesamiento digital de imágenes de rostros es una de las funcionalidades con más enfoque

centrándose en el reconocimiento facial para campos como la biometría, seguridad y reconocimiento de patrones. Una de las razones más importantes que ha llevado a este crecimiento, son la necesidad cada vez mayor de aplicaciones de seguridad y vigilancia utilizadas en diferentes ámbitos ((R. Gimeno Hernández, 2010)). Estas técnicas utilizadas para reconocimiento facial de personas pueden ser implementadas para encontrar puntos característicos para poder clasificar las diferentes emociones.

El principal inconveniente al implementar estos procesos de uno o varios rostros reside en la gran variación entre cada una de las imágenes y expresiones que puede llegar a tener una misma persona, estas discrepancias son producto, de diferentes tipos de luz artificial, luz natural o también conocida como luz fluorescente dependiendo de la variación del ambiente en el que fue captada, también varía por diferentes peinados, maquillaje e inclusive uso de accesorios, por lo que puede llegar a causar que la misma persona sea percibida de diferentes formas entre varias imágenes ((Edison Rene, 2017)). Para los humanos esta tarea es natural y es realizada constantemente sin preocuparse por el cómo, el trasladar esta tarea a una maquina conlleva a uno de los principales objetivos propuestos en este estudio.

Actualmente existen varios sistemas de aprendizaje automático y visión artificial o visión por computadora , orientados a detectar emociones ((C. Gonzalez Restrepo, 2014; David Costa Da Silva, 2018; Santiago., 2016; Ramon Zatarain-Cabada, 2016; Roca., 2017)), los cuales utilizan diferentes técnicas y algoritmos especializados , de los cuales para este trabajo se destacan 2 algoritmos especialmente el openCV para detectar las caras de las personas , y el DLIB para encontrar la ubicación de los principales hitos faciales ojos, nariz y boca para con estos poder predecir el estado emocional de la persona .

## 1.3. Objetivos

### 1.3.1. Objetivo General

Elaborar un sistema que sea capaz de detectar las emociones, y en función de estas sea capaz de modificar el ambiente del hogar. Para hacer de él un espacio más confortable y adecuado al dueño de la vivienda

### 1.3.2. Objetivo Especifico

- Implementar el algoritmo de Viola Jones que utiliza la librería OpenCV para que el programa sea capaz de detectar un rostro mediante una cámara web, realizar una captura y un recorte del mismo.
- Diseñar una red neuronal de convolución que tenga la capacidad de clasificar el estado emocional a partir de imágenes obtenidas por cámara web .
- Describir el algoritmo de DLIB el cual permite detectar los diferentes puntos significativos de un rostro (boca, nariz, ojos, cejas).
- Diseñar un clasificador de tipo SVM que permita clasificar las emociones basadas en la información ofrecida por un sistema que detecta los diferentes puntos significativos de un rostro (boca, nariz, ojos, cejas).
- Implementar un sistema que produzca un tipo de audio que se corresponde con algún estado emocional.

# Capítulo 2

# Marco Teórico

## 2.1. Las emociones y teoría de la mente

### 2.1.1. Emociones

El diccionario de la Real Academia Española define emoción como una ńalteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somáticaż.

LeDoux ((J., 1999)) sostiene que las emociones son respuestas físicas controladas por el cerebro que permitieron a organismos antiguos sobrevivir en entornos hostiles y procrear.

Scherer ((K., 2000)) ofrece una definición operativa del término: las emociones son episodios de cambio coordinado en distintos componentes (activación neurofisiológica, expresión motora, experiencia subjetiva, etc.) en respuesta a eventos internos o externos de importancia para el organismo.

Para Damasio ((A., 2005)) las emociones son acciones que se expresan en el rostro, la voz o en conductas específicas, tendientes a mantener la homeostasis del organismo

Las emociones se dividen en: ((A., 2005; J., 1999))

1. Básicas o primarias: estados emocionales determinados biológicamente cuya expresión es universal e innata. De comienzo rápido y duración limitada, se hallan ligadas a conductas fundamentales para la supervivencia. Dado su valor adaptativo, este repertorio emocional estaría presente en otras especies. Alegría, tristeza, enojo, y miedo son las que parecen haber recibido mayor acuerdo.

2. Emociones complejas o secundarias: se trata de un amplio abanico de estados emocionales que surgen de la combinación de emociones primarias. Por ejemplo, el resentimiento surgiría de la combinación de tristeza y rabia. Damasio ((A., 1994)) entiende las emociones secundarias como la toma de conciencia del estado emocional y sus cambios somáticos al vincularlos con la experiencia previa. Para Baron-Cohen ((S., 2001)) , a diferencia de las emociones primarias, el reconocimiento de emociones secundarias requiere de la atribución, al interlocutor, de creencias, intenciones o algún estado mental, por lo tanto el reconocimiento de estos estados emocionales se logra a través de la Teoría de la mente (TdM).

### 2.1.2. Teoria de la Mente

La TdM fue definida por Premack ((D., 1978)) como la habilidad de conceptualizar estados mentales de otras personas (meta representaciones) para poder explicar y predecir gran parte de su comportamiento. Forma parte de la cognición social (CS), conjunto de habilidades cognitivas que nos permite dar sentido al mundo social e interactuar de forma efectiva con los demás. Se entiende por TdM la actividad

de representarse los estados mentales de los demás, por ejemplo, sus percepciones, objetivos, creencias, expectativas, etc. El intercambio social se ve regulado entonces en función de la creencia de que quienes nos rodean poseen una mente distinta de la nuestra, con intenciones, creencias, deseos y estados emocionales que podemos inferir e interpretar.

Dentro de las capacidades que requieren TdM encontramos el reconocimiento de estados emocionales secundarios a partir de la cara completa, la mirada, o la voz, la comprensión de las ńmeteduras de pataż sociales, la detección de la ironía, el juicio moral y la empatía. ((Maria Eugenia Tabernero, 2013))

## 2.2. Patrones de reconocimiento

Hay varios modelos para representar las emociones los cuales son usados para su categorización y organización. Estas categorías difieren dependiendo de las diferentes tareas y aplicaciones. La categorización es principalmente hecha sobre bases subjetivas porque los investigadores no coinciden en determinar un conjunto de etiquetas emocionales. Estos modelos de clasificación difieren principalmente en que algunos usan valores continuos y otros valores discretos para la abstracción de categorías Ambos modelos están relacionados ya que las categorías emocionales discretas pueden ser mapeadas en modelos continuos.((Humberto Perez Espinosa, 2010))
.

### 2.2.1. Modelos Discretos

Estos modelos se basan en el concepto de emociones básicas, que son la forma más intensa de las emociones, a partir de las cuales se generan todas las demás mediante variaciones o combinaciones de estas. Suponen la existencia de emociones universales, al menos en esencia, que pueden ser distinguidas claramente una de otra por la mayoría de la gente, y asociadas con funciones cerebrales que evolucionaron para lidiar con diferentes situaciones. Las emociones básicas son experimentadas por los mamíferos sociales y tienen manifestaciones particulares asociadas con ellas tales como expresiones faciales, patrones fisiológicos y tendencias de comportamiento. La dominación de esta teoría en el reconocimiento automático de emociones puede explicarse por el hecho de que la diferenciación entre emociones es relativamente clara, aun cuando dentro de estos conjuntos de emociones la necesidad de definiciones más detalladas ha sido abordada, por ejemplo, distinguiendo entre ira y cólera. Otra explicación puede ser que las representaciones estereotípicas de estas expresiones son asimiladas más fácilmente con el fin de generarlas y reconocerlas, y por lo tanto son más útiles para una construcción rápida de bases de datos y como un punto de partida para investigación emergente en este campo de investigación.

Algunos científicos definen una lista de emociones básicas desde su punto de vista, ver cuadro 2.1.

Como se puede observar en dicho cuadro, no existe un criterio único para definir qué emociones forman este conjunto. Los modelos discretos permiten una representación más particularizada de las emociones en las aplicaciones donde solamente se requiere reconocer un conjunto predefinido de emociones. Este enfoque ignora la mayor parte del espectro de expresiones emocionales humanas. Si un conjunto reducido de emociones básicas es usado como un punto de partida para el reconocimiento de emociones, surge la pregunta de si las mismas características y patrones de comportamiento son válidas tanto para emociones extremas como para emociones

CUADRO 2.1: Conjuntos de emociones básicas propuestas por diferentes autores

| Autor | Emociones básicas |
|---|---|
| Plutchik | aceptación, enojo, anticipación, asco, alegría, miedo, tristeza, sorpresa |
| Ekman, Friesem, Ellsworth2 | Ira, asco, miedo, alegría, tristeza, sorpresa. |
| Gray | Ira y terror, ansiedad, alegría. |
| Izard | Ira, desprecio, asco, angustia, miedo, culpa, interés, alegría, vergüenza, sorpresa. |
| James | Miedo, pena, amor, rabia. |
| Mowner, Friesem, Ellsworth2 | dolor, placer |
| Oatley y Johnson-Laird | Ira, asco, ansiedad, felicidad, tristeza. |
| Paksepp | Expectativa, miedo, alcance, pánico. |
| Tomkins | Ira, asco, angustia, miedo, alegría, vergüenza, sorpresa. |
| Watson | Miedo, amor, rabia |
| Weiner y Graham | felicidad, tristeza |

más sutiles .En el cuadro 2.1, tomada de ((A., 1990)), se muestran varios conjuntos de emociones básicas propuestos por distintos autores. Otro de los problemas de estos modelos es la investigación intercultural de emociones y la traducción correcta de términos emocionales o afectivos usados y muchos de estos términos tienen significados connotativos y denotativos diferentes en diferentes idiomas, no hay una solución satisfactoria a este problema. Algunos autores han llegado a la conclusión que la representación del espectro emocional mediante emociones básicas es demasiado compleja para su utilización en aplicaciones prácticas .((Humberto Perez Espinosa, 2010)).

### 2.2.2. Modelos Continuos

En estos modelos los estados emocionales son representados usando un espacio multidimensional continuo. Las emociones son representadas por regiones en un espacio n-dimensional. Los ejes no están relacionados con estados emocionales, sino con primitivas emocionales que son propiedades subjetivas que presentan todas las emociones. Un ejemplo muy conocido es el modelo Arousal-Valence. Este modelo describe las emociones usando un espacio bidimensional. Las emociones son descritas en términos de Valencia y Activación ((S., 2009)) .La Valencia, también llamada placer describe qué tan negativa o positiva es una emoción específica. Activación, también llamada intensidad, describe la excitación interna de un individuo y va desde estar muy tranquila hasta estar muy activa. Otro modelo similar al previo es el modelo tridimensional. Las dos primeras dimensiones son Valencia y Activación mientras la tercera es la energía o Dominación que describe el grado de control del individuo sobre la situación, o, en otras palabras, qué tan fuerte o débil se muestra el individuo. Ver Figura 2.1

El modelo tridimensional surge por la necesidad de distinguir entre emociones que se encuentran traslapadas en un espacio bidimensional. Añadir la tercera dimensión ayuda a distinguir entre emociones como miedo y enojo ya que ambas

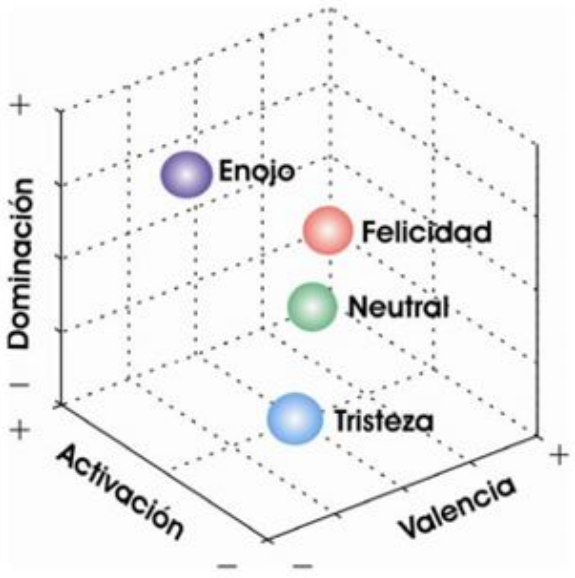


FIGURA 2.1: Modelo Tridimensional Continuo de las Emociones. Valencia-Activación-Dominación.

tienen Valencia y Activación similar. Los modelos continuos permiten mayor flexibilidad en las aplicaciones ya que no se limitan a un conjunto de emociones, sino que pueden representar cualquier estado emocional en el espacio multidimensional y trasladarlo a un conjunto de emociones básicas si así se requiere .Este tipo de modelos tiene la capacidad de representar de mejor manera la forma en que suceden las emociones en el mundo real, ya que muchas veces las emociones no se generan de forma prototípica, sino que pueden manifestarse como una mezcla de emociones o como ligeras expresiones emocionales difíciles de detectar. Al etiquetar bases de datos emocionales los modelos discretos son más adecuados para asignar estados preseleccionados a patrones psicológicos, mientras el enfoque continuo es más adecuado para evaluar la carga emocional ((Humberto Perez Espinosa, 2010))

## 2.3. Aprendizaje supervisado Vs aprendizaje no supervisado

La mayoría de los problemas de aprendizaje estadístico se dividen en una de dos categorías: supervisados o no supervisados. En el dominio de aprendizaje supervisado para cada observación de las mediciones del predictor $x_i, i = 1, \ldots, n$ hay una medida de respuesta asociada $y_i$. Se desea ajustar un modelo que relacione la respuesta con los predictores, con el objetivo de predecir con precisión la respuesta para observaciones futuras (predicción) o comprender mejor la relación entre la respuesta y los predictores (inferencia). Se utilizan muchos métodos de aprendizaje estadístico clásico como los de regresión lineal y la regresión logística, así como los enfoques más modernos como máquinas de vectores de soporte, que también operan en el dominio de aprendizaje supervisado.

Por el contrario, el aprendizaje no supervisado describe la situación algo más desafiante en la que para cada observación $i = 1, \ldots, n$, observamos un vector de medidas $x_i$ pero ninguna respuesta asociada $y_i$. No es posible ajustar un modelo

de regresión lineal, ya que no hay una variable de respuesta para predecir. En este contexto, en cierto sentido se trabaja ciego; La situación se conoce como no supervisada porque se carece de una variable de respuesta que pueda supervisar el análisis. ¿Qué tipo de análisis estadístico es posible? Se puede tratar de entender las relaciones entre las variables o entre las observaciones. Una herramienta de aprendizaje estadístico que se puede usar en este entorno es el análisis de clúster o clustering. El objetivo del análisis de clúster es determinar, sobre la base de $x_1, \ldots, x_n$, si las observaciones se dividen en grupos relativamente distintos. Por ejemplo, en un estudio de segmentación de mercado podríamos observar múltiples características (variables) para clientes potenciales, como código postal, ingresos familiares y hábitos de compra. Podríamos creer que los clientes se dividen en diferentes grupos, como los que gastan mucho y los que gastan poco. Si la información sobre los patrones de gasto de cada cliente estuviera disponible, entonces sería posible un análisis supervisado. Sin embargo, esta información no está disponible, es decir, no sabemos si cada cliente potencial gasta mucho o no. En esta configuración, podemos tratar de agrupar a los clientes sobre la base de las variables medidas, para identificar grupos distintos de clientes potenciales. Identificar dichos grupos puede ser de interés porque podría ser que los grupos difieran con respecto a alguna propiedad de interés, como los hábitos de gasto.

Muchos problemas caen naturalmente en los paradigmas de aprendizaje supervisados o no supervisados. Sin embargo, a veces la cuestión de si un análisis debe considerarse supervisado o no es menos claro. Por ejemplo, supongamos que tenemos un conjunto de n observaciones. Para m de las observaciones, donde m <n, tenemos mediciones predictivas y una medición de respuesta. Para las observaciones n - m restantes, tenemos mediciones predictoras, pero no medidas de respuesta. Tal escenario puede surgir si los predictores pueden medirse de manera relativamente fácil, pero las respuestas correspondientes son mucho más difíciles de recopilar. Se hace referencia a esta configuración como un problema de aprendizaje semi-supervisado. En este contexto, se desea utilizar un método de aprendizaje estadístico que pueda incorporar las observaciones m para las cuales las mediciones de respuesta están disponibles, así como las observaciones n m para las que no están disponibles. ((Gareth James, 2015))

### 2.3.1. Algoritmos para la clasificación - con aprendizaje supervisado.

Existen muchos algoritmos (clasificadores) para la clasificación con aprendizaje supervisado, entre ellos:

**Nearest Neighbors (KNN)** : el funcionamiento del algoritmo KNN radica en asignar la clase a un nuevo ejemplo basado en las observaciones de las clases de sus vecinos más cercanos. Es necesario contar con una representación de los datos en un espacio métrico y con una definición de distancia que aproxime de cierta forma la topología del espacio original

**Naive Bayes** : consiste en un clasificador bayesiano simple que asume independencia entre las características. La fase de entrenamiento consiste en computar para cada característica la cantidad de veces que es observada en cada clase, y de esta forma aproximar la probabilidad de que dicha característica indique la clase correspondiente

**Arboles de decisión** : son clasificadores que aproximan una función a partir de la ejecución de un conjunto de pruebas sobre los valores asociados a atributos predefinidos. Se utiliza el algoritmo de aprendizaje conocido como ID3 que significa ïnducción mediante árboles de decisión"que fue desarrollado por J. Ross Quinlan. Este

algoritmo genera de forma iterativa un árbol, eligiendo en cada nodo el atributo que maximiza la cantidad de información obtenida sobre el conjunto de entrenamiento.((Suilan Estevez-Velarde Yudivi, 2015))

**Regresión Logística** : La Regresión Logística es una técnica estadística multivariante que nos permite estimar la relación existente entre una variable dependiente no métrica, en particular dicotómica y un conjunto de variables independientes métricas o no métricas.

El Análisis de Regresión Logística tiene la misma estrategia que el Análisis de Regresión Lineal Múltiple, el cual se diferencia esencialmente del Análisis de Regresión Logística por que la variable dependiente es métrica; en la práctica el uso de ambas técnicas tiene mucha semejanza, aunque sus enfoques matemáticos son diferentes.

La variable dependiente o respuesta no es continua, sino discreta (generalmente toma valores 1,0). Las variables explicativas pueden ser cuantitativas o cualitativas; y la ecuación del modelo no es una función lineal de partida, sino exponencial; si bien, por sencilla transformación logarítmica, puede finalmente presentarse como una función lineal.

Así pues, el modelo será útil en frecuentes situaciones prácticas de investigación en que la respuesta puede tomar únicamente dos valores: 1, presencia (con probabilidad p); y 0, ausencia (con probabilidad 1- p).

El modelo será de utilidad puesto que, muchas veces, el perfil de variables puede estar formado por caracteres cuantitativos y cualitativos; y se pretende hacer participar a todos ellos en una única ecuación conjunta.

El modelo puede acercarse más a la realidad ya que muchos fenómenos, como los del campo epidemiológico, se asemejan más a una curva que a una recta. ((Salcedo Poma, 2017))

**Máquinas de Vectores de Soporte (SVM)** : las máquinas de soporte vectorial construyen una hipótesis mediante el cálculo de un hiperplano que separe a los elementos de cada clase. El problema de optimización asociado consiste en encontrar el hiperplano separador que maximiza la mínima de las distancias a cada uno de los elementos. En espacios muy esparcidos es bastante probable encontrar un hiperplano separador, o al menos minimice de forma considerable los elementos en la clase incorrecta ((Suilan Estevez-Velarde Yudivi, 2015))

## 2.4. Problemas de Regresión Vs Problemas de clasificación

Las variables pueden caracterizarse como cuantitativas o cualitativas (también conocidas como categóricas). Las variables cuantitativas toman valores numéricos. Por ejemplo: la edad, altura o ingresos de una persona, el valor de una casa y el precio de una acción. En contraste, las variables cualitativas toman valores en una de K diferentes clases o categorías. Los ejemplos de variables cualitativas incluyen el género de una persona (hombre o mujer), la marca del producto comprado (marca A, B o C), si una persona no paga una deuda (sí o no) o un diagnóstico de cáncer (leucemia mielógena aguda), Leucemia linfoblástica aguda o sin Leucemia). Tendemos a referirnos a los problemas con una respuesta cuantitativa como problemas de regresión, mientras que aquellos que involucran una respuesta cualitativa a menudo se denominan problemas de clasificación.

Sin embargo, la distinción no siempre es tan nítida. La regresión lineal de mínimos cuadrados se usa con una respuesta cuantitativa, mientras que la regresión logística se usa típicamente con una respuesta cualitativa (dos clases o binaria). Como

tal, a menudo se usa como método de clasificación. Pero como estima las probabilidades de clase, puede considerarse como una regresión.((Gareth James, 2015))

## 2.5. Redes neuronales Artificiales

### 2.5.1. ¿Qué son las redes neuronales artificiales?

Una Red Neuronal es un algoritmo de cálculo que se basa en una analogía del sistema nervioso. La idea general consiste en emular la capacidad de aprendizaje del sistema nervioso, de manera que la Red Neuronal aprenda a identificar un patrón de asociación entre los valores de un conjunto de variables predictoras (entradas) y los estados que se consideran dependientes de dichos valores (salidas). Desde un punto de vista técnico, la Red Neuronal consiste en un grupo de unidades de proceso (nodos) que se asemejan a las neuronas al estar interconectadas por medio de un entramado de relaciones (pesos) análogas al concepto de conexiones sinápticas en el sistema nervioso. A partir de los nodos de entrada, la señal progresa a través de la red hasta proporcionar una respuesta en forma de nivel de activación de los nodos de salida. Los valores de salida proporcionan una predicción del resultado en función de las variables de entrada. Desde el punto de vista de implementación práctica, los nodos son elementos computacionales simples que emulan la respuesta de una neurona a un determinado estímulo. Estos elementos, como las neuronas en el sistema nervioso, funcionan como interruptores: cuando la suma de señales de entrada es suficientemente alta (en el caso de una neurona se dice que se acumula suficiente neurotransmisor), la neurona manda una señal a las neuronas con las que mantiene contacto (se genera un potencial de acción).

Esta situación se modela matemáticamente como una suma de pesos de todas las señales de llegada al nodo que se compara con un umbral característico. Si el umbral se supera, entonces el nodo se dispara, mandando una señal a otros nodos, que a su vez procesarán esa información juntamente con la que reciben de nodos adyacentes (Figura 2.2).

Evidentemente, la respuesta de cada nodo dependerá del valor de las interacciones con los nodos precedentes dentro de la estructura de la red. Como en el caso del sistema nervioso, el poder computacional de una Red Neuronal deriva no de la complejidad de cada unidad de proceso sino de la densidad y complejidad de sus interconexiones. La primera implementación práctica de estas ideas se describe en los trabajos de McCulloch y Pitts en 1946. A partir de este punto, algunos de los hitos principales en la investigación de este tipo de técnicas fueron: el diseño por Widrow y Hoff (1961) de la red conocida como Adalina (capaz de resolver problemas de regresión lineal), el desarrollo de la red con estructura de perceptrón simple en 1959 (con equivalencia al análisis discriminante y regresión logística) y las redes multicapa por Rosenblatt en 1986 (que permiten la resolución de situaciones no lineales). Por otra parte, los trabajos teóricos de Bishop y la aportación sobre redes autoorganizadas de Kohonen dotaron de fundamentos formales a este tipo de técnica. A partir de los trabajos pioneros, el interés sobre esta metodología se ha difundido a casi todos los ámbitos de la ciencia. Los distintos aspectos técnicos y las implicaciones de su utilización han sido investigados desde muchos puntos de vista, interesando, entre otros, a matemáticos, físicos, neurólogos, ingenieros, programadores y filósofos.

Desde un punto de vista práctico, existen muchos tipos de RN. En el cuadro 2.2 se recogen las más características. Para clasificarlas, podemos considerar dos criterios básicos: el modo de aprendizaje y el flujo de información. En una red, el modo de aprendizaje puede ser supervisado, es decir, la red recibe los patrones de entrada

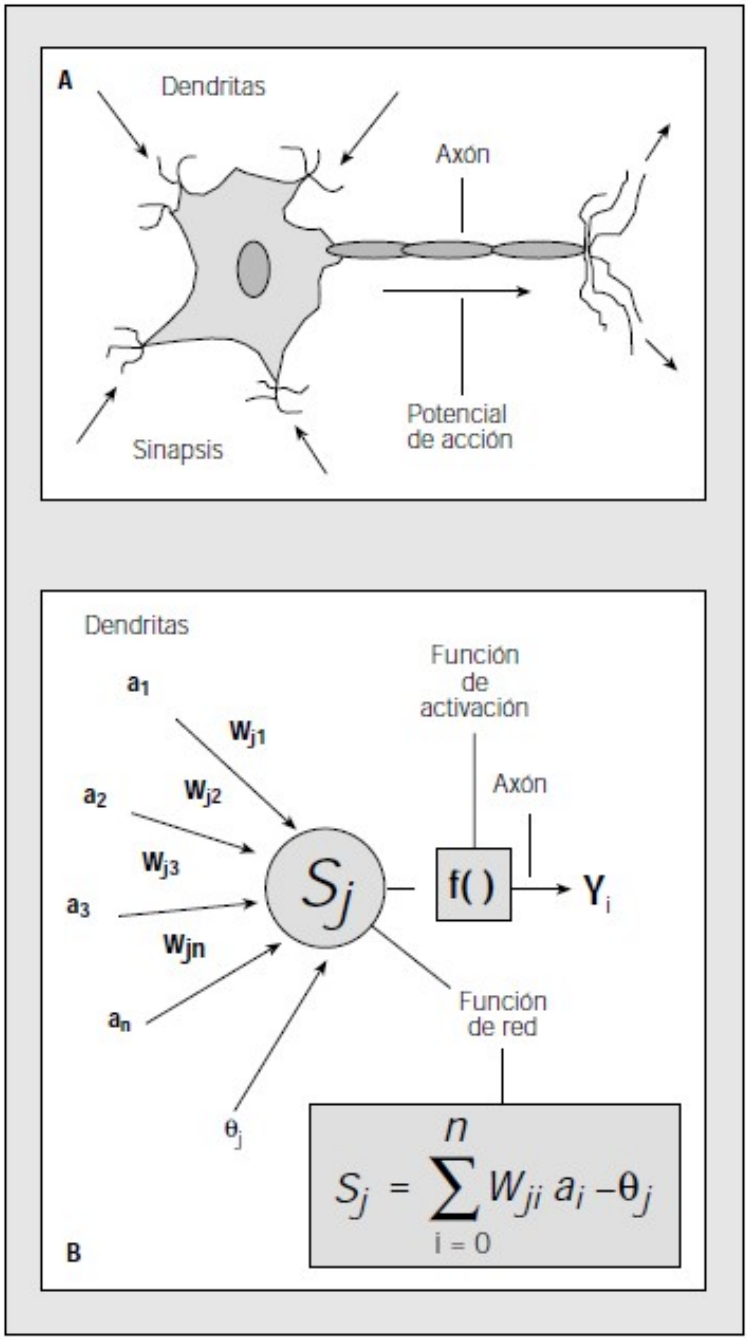


FIGURA 2.2: Comparación entre la neurona biológica (A) y la neurona artificial (B).

y la respuesta observada que debe aprender; o no supervisado si la red reconoce automáticamente en los datos el patrón que debe aprender. Por otra parte, el flujo de información que manejan puede ser unidireccional, cuando la información sigue una dirección única desde los nodos de entrada a los de salida; o realimentado, donde el flujo de información no es único al incorporar circuitos de realimentación entre capas de la red.

CUADRO 2.2: Clasificación de las redes neuronales artificiales según tipo de aprendizaje y arquitectura.

| Aprendizaje | Arquitectura Unidireccional | Arquitectura Realimentada |
|---|---|---|
| Supervisado | Perceptron | BSB |
| | Adalina | BSB |
| | Madalina | Fuzzy Cog. Map |
| | Perceptron Multicapa | |
| | GRNN | |
| | LVQ | |
| | Maquina de Boltzmann | |
| No supervisado | LAM | ART |
| | OLAM | Hopfield |
| | Mapas de Kohonen | BAM |
| | Neocognitrón | |
| Hibridos | Funcion de base radial(RBF) | |
| | Contrapropagación | |
| Reforzados | Aprendizaje reforzado | |

### 2.5.2. El perceptron multicapa

Dentro de las redes supervisadas unidireccionales, la estructura más utilizada es el llamado perceptrón multicapa (MLP, multilayered perceptrón). La arquitectura típica de este tipo de red está constituida por varias capas de nodos con interconexión completa entre ellos. El caso más sencillo en este tipo de red consiste en sólo 2 capas de neuronas, las de entrada y las de salida. De esta manera, podemos obtener un modelo adecuado para problemas lineales del tipo de la regresión lineal múltiple.

Si queremos analizar problemas no-lineales, es necesario incorporar otras capas de neuronas intermedias u ocultas (hidden units) (Figura 2.3)

En este tipo de red, una neurona recibe distintas entradas y activa una función de red (o regla de propagación) con unos pesos de entrada asociados (Figura 2.2). La computación de estos pesos se sigue de la aplicación de la función de activación que determina el nivel de activación de salida de la neurona. La entrada de las neuronas de la primera capa (entrada) son los valores de las variables predictoras y los niveles de activación de las neuronas de la última capa (salida) son los resultados de la red. Dentro de los parámetros que definen una red, la función de red más utilizada es de tipo lineal, y como función de activación más empleada está la función sigmoidea.

Las aplicaciones de estas Redes neuronales se realizan en diferentes áreas: Psicología, medicina, ingeniera, economía, Astronomía, etc.

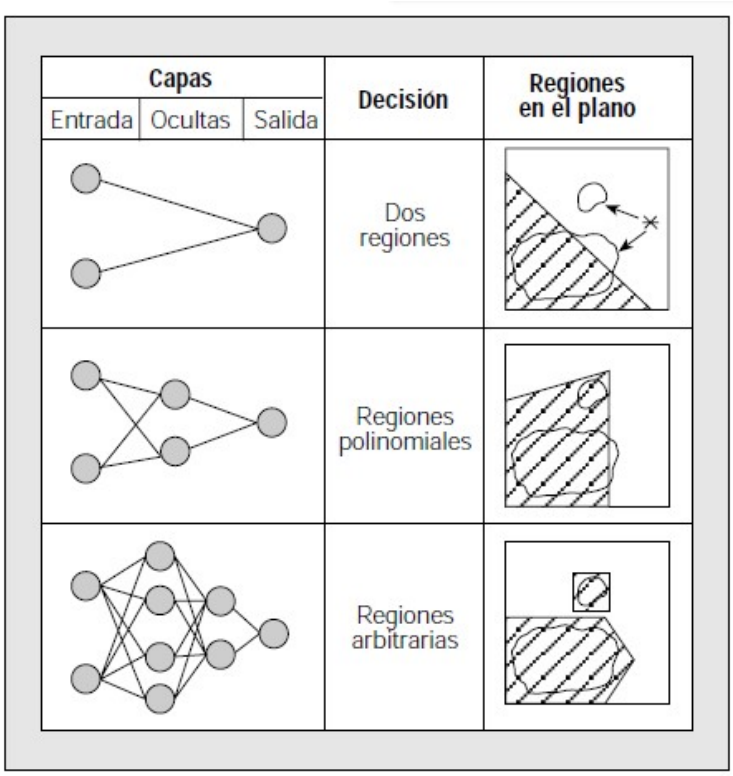


FIGURA 2.3: Capacidad de decisión de las redes neuronales artificiales (perceptrón multicapa). Con 2 variables de entrada, ante un problema de clasificación complejo en el plano

### 2.5.3. Validación cruzada y Entrenamiento

Para diseñar una red que sea eficaz es conveniente dividir los datos en 3 conjuntos, atendiendo a que cada uno de ellos mantenga la representatividad de la población origen: a) el conjunto de entrenamiento (training set); b) el conjunto de validación, y c) un conjunto de testeo. El conjunto de entrenamiento se usa para ajustar los pesos durante la fase de entrenamiento, mientras que el conjunto de validación se utiliza para decidir cuándo parar el proceso de entrenamiento. El entrenamiento debe parar cuando el error en el conjunto de validación comienza a crecer. De esta manera, nos aseguramos que la red es capaz de predecir correctamente los resultados de un conjunto de datos que no forman parte de los ejemplos de entrenamiento. Esta técnica se denomina validación cruzada (crossvalidation). Si continuamos el entrenamiento más allá de este punto, la red empieza a aprender de memoria los datos del conjunto de entrenamiento, pero pierde capacidad de generalización (Figura 2.4).

La búsqueda de una generalización óptima, que es la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento, requiere que se cumplan tres condiciones: a) que la información recogida en las variables sea suficiente es decir, una selección apropiada de las variables y una buena calidad en la recogida de datos; b) que la función que aprenda la red sea suave pequeños cambios en las variables de entrada produzcan pequeños cambios en las variables de salida, y c) que el tamaño de la base de datos sea suficiente. De esta manera aseguramos que el conjunto de entrenamiento sea representativo de la población a estudio. Excepto la segunda condición, el resto de los requisitos

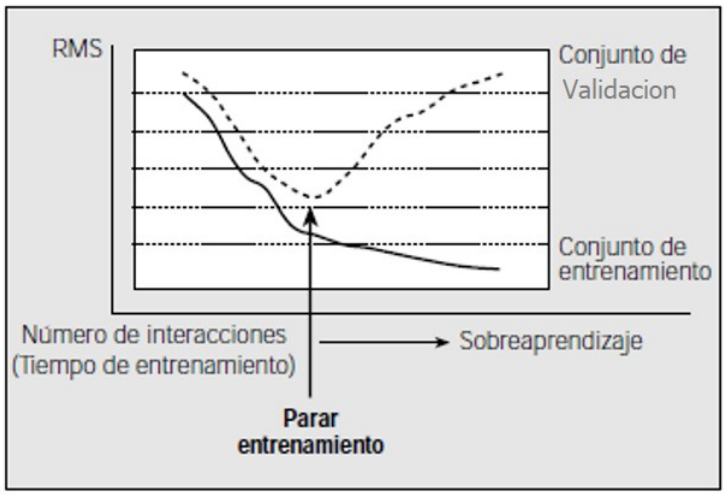


FIGURA 2.4: Criterio de parada del proceso de entrenamiento.

son comunes a cualquier técnica multivariante que se emplee. Una vez finalizado el entrenamiento, la red (entrenada) evalúa el conjunto de testeo y produce las correspondientes predicciones con datos que no se han utilizado en el entrenamiento ni en la validación cruzada. Esta prueba final nos aporta un resultado independiente acerca de la capacidad de generalización de la red.

### 2.5.4. Tamaño y arquitectura de la red

La arquitectura de una red viene determinada por el número de capas y nodos que la forman. La complejidad de la red viene determinada por el número de interconexiones que contiene. En general, no es inmediato establecer de forma exacta cuál será la arquitectura ideal para cada aplicación. Así, problemas de discriminante lineal o de regresión logística pueden solucionarse con redes simples. Los problemas surgen al enfrentarse a modelos más complicados (Figura 2.3). En diversas aplicaciones, un MLP con una única capa oculta puede ser adecuado en muchos casos. Existen algoritmos evolutivos que determinan, de forma automática, esta arquitectura óptima al aumentar o retirar nodos o capas del modelo. En cualquier caso, la arquitectura óptima debe alcanzarse, en la práctica, mediante un proceso iterativo, validando la capacidad predictiva de las distintas arquitecturas consideradas.((Javier Trujillano, 2004))

Finalmente se explica la función de activación (ů), que es una función no lineal que funciona como umbral para realizar redes neuronales no lineales.

### 2.5.5. Funciones activación

#### Funcion Sigmoide

Esta función es un caso especial de la función logística 2.1. Dónde L es el máximo de la función y K es lo abrupta que es la curva. Definiendo esta L = 1 y K = 1 obtendríamos la función sigmoide 2.2.

$$\sigma(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{2.1}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

Los números grandes negativos se convierten en 0 y los números grandes positivos en 1. Además, la función sigmoide tiene una buena interpretación como la velocidad de activación de una neurona. Pasa desde no activarse en absoluto (0), hasta el activarse completamente (1). Sin embargo, tiene dos grandes inconvenientes. En primer lugar, la sigmoide satura y acaba con los gradientes. Cuando la neurona satura en 0 o 1, gradiente en estas zonas es prácticamente cero. En el proceso de Backpropagation, proceso en el cual el error de una capa de neuronas se propaga a la anterior capa para realizar una corrección de los pesos de esta misma. Entonces si este gradiente local es muy pequeño, acabará con el gradiente general y no habrá señal a través de la neurona y sus pesos. Entonces es de vital importancia inicializar los pesos de las neuronas sigmoideas para evitar la saturación. Pero esos pesos no deben ser demasiado grandes, sino la mayoría de neuronas se saturarán y la red no aprenderá. El segundo problema se trata de que las salidas de las neuronas sigmoideas no están centradas a cero y eso no es lo deseable ya que las neuronas en las capas posteriores deberían recibir estas entradas centradas a cero. Esto puede resultar en un movimiento zigzag del gradiente para los pesos.

**Función tangente hiperbolica**

La función tangente hiperbólica es una alternativa a la función sigmoide, su fórmula (1.3) se define de la siguiente manera.

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.3}$$

Como se puede apreciar la gráficas comparación 2.2 y 2.3 función tangente hiperbólica es muy parecida, pero con una diferencia que arregla uno de los problemas anteriores. Las salidas de las neuronas están centradas a cero. Con lo cual es preferible el uso de este tipo de neuronas en la práctica.

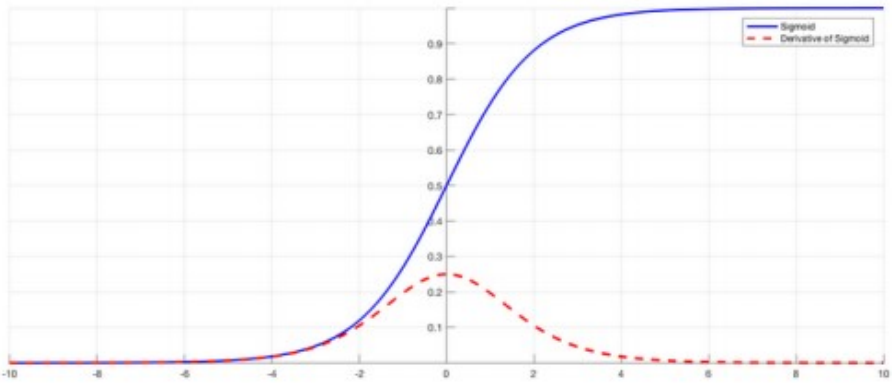


FIGURA 2.5: Gráfico de la función sigmoide y su derivada

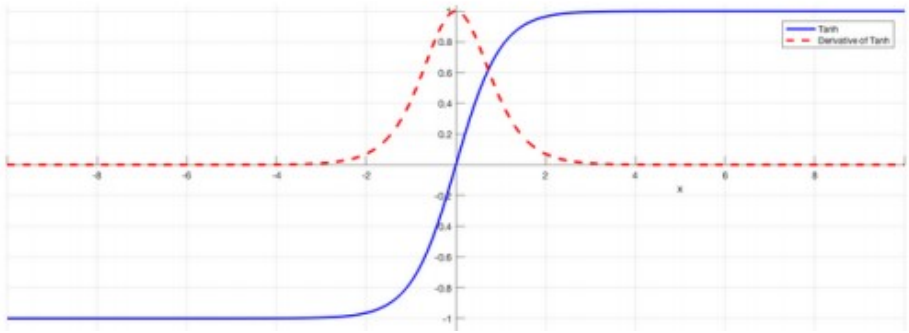


FIGURA 2.6: Gráfico de la función tangente hiperbólica y su derivada.

**Unidad lineal Rectificada (ReLU)**

La función de activación ReLU está definida por la siguiente expresión :

$$f(x) = max.(0, x) \tag{2.4}$$

Donde X es la entrada de la neurona. Es decir, la activación es llevada a cabo cuando pasa de cero. Se puede apreciar esta afirmación en el siguiente gráfico (Figura 2.7)

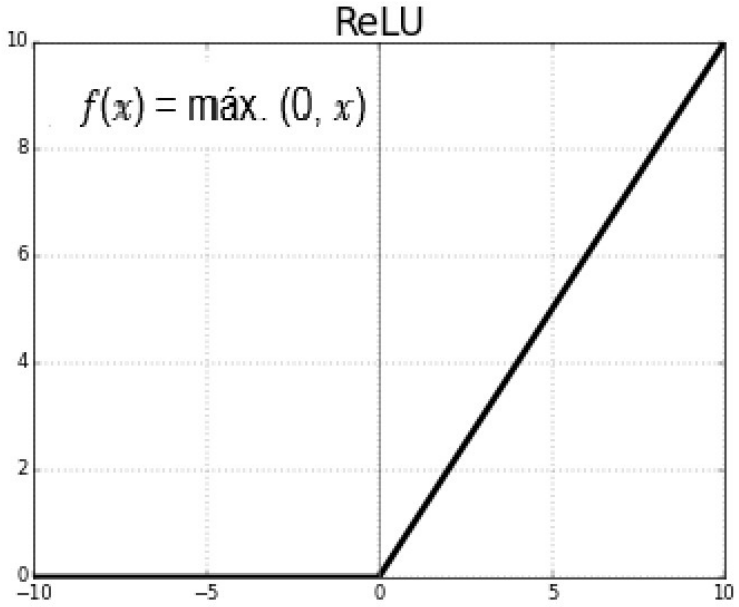


FIGURA 2.7: Gráfico de la función ReLU.

La función ReLU es más efectiva que la sigmoide y la alternativa más práctica, la tangente hiperbólica, ya que reduce eficazmente el costo de cálculo. Sin embargo, las neuronas con función ReLU presentan varios inconvenientes. En primer lugar, las neuronas durante el entrenamiento pueden quedar inutilizadas debido a que los pesos puedan ser actualizados de tal manera que la neurona no vuelva a ser activada nuevamente. Para evitar este problema, hay que elegir una tasa de aprendizaje lo suficientemente pequeña.

**Leaky/paramétrica ReLU)**

La Leaky ReLU es propuesta a la solución propuesta a la función de activación clásica ReLU. Su diferencia reside en que, mientras ReLU es cero para todo número más pequeño a cero (X<0), Leaky ReLU añade una pendiente para estos números por debajo de cero 2.5. Este parámetro es una constante más pequeña a 1. El caso concreto que está pendiente es configurado para cada neurona, la función es llamada paramétrica

$$\begin{cases} f(x) = \alpha x, (x < 0) \\ f(x) = x, (x \geq 0) \end{cases} \tag{2.5}$$

**ReLU randomizada**

La ReLU randomizada es una vertiente de la Leaky ReLU. En este caso se elige una pendiente aleatoria dentro de un rango, determinado dentro de una función de distribución uniforme, dentro del proceso entrenamiento. En el proceso de testing, está pendiente es fijada. A continuación, se muestran las gráficas de las distintas ReLU.

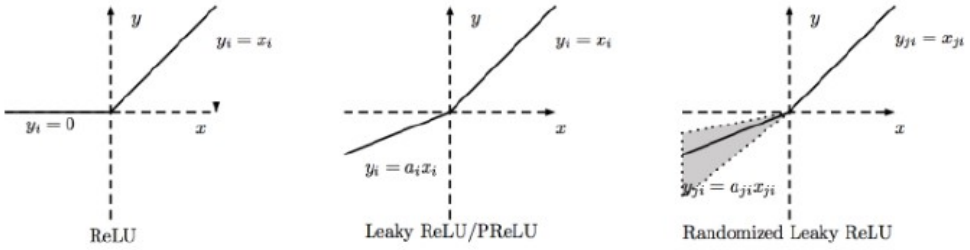


FIGURA 2.8: En la izquierda, la ReLU estándar. En el centro, Leaky/paramétrica ReLU con su definida. A la derecha, la ReLU randomizada, la zona gris define el rango de valores aleatorios que puede tomar en el proceso de entrenamiento.

**Softmax**

La función softmax, o función exponencial normalizada, es una generalización de la Función logística. Se emplea para çomprimirün vector K-dimensional, z, de valores reales arbitrarios en un vector K-dimensional(z) de valores reales en el rango [0, 1] ((Solsona., 2018)) . La función está dada por:

$$\begin{gathered} \sigma : \mathbb{R}^K \to [0,1]^K \\ \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad para \;\; j = 1, \ldots, K. \end{gathered} \tag{2.6}$$

## 2.6. Redes neuronales convolucionales

Las redes neuronales convolucionales son especialmente utilizadas en el campo de la visión artificial, puesto que sus neuronas se corresponden a campos receptivos de forma similar a las neuronas de la corteza visual primaria de un cerebro biológico.

Es importante mencionar que la arquitectura de red neuronal es una variación de la vista en el anterior apartado (perceptrón multicapa), sin embargo, gracias a que su aplicación es realizada en matrices bidimensionales, son especialmente efectivas en clasificación y segmentación de imágenes, así como en cualquier tipo de datos donde los mismos estén distribuidos de forma continua a lo largo de la entrada.

### 2.6.1. Historia y Fundamentos Biológicos de las Redes Neuronales Convolucionales

El origen de este tipo de red se encuentra en el Neocognitron , introducido por Kunihiko Fukushima en 1980. Dicho modelo fue mejorado por Yann Lecun en 1998, pues introdujo el aprendizaje basado en backpropagation. En el año 2012 este tipo de redes fueron refinadas por Dan Ciresan y fueron implementadas en GPU, consiguiendo un rendimiento computacional mejor a los obtenidos hasta entonces. Además, la calidad de los resultados resultó mucho mayor a la de otras técnicas previas en clasificación de imágenes como máquinas de soporte vectorial o cascadas de Haar.

Como se dijo anteriormente, esta arquitectura tiene una clara inspiración en la corteza visual del cerebro. Esta inspiración se debe en gran medida al trabajo realizado por Hubel y Wiesel en 1959, gracias al cual se comprendió en gran medida el funcionamiento de la corteza visual, sobre todo de las células responsables de la selectividad de orientación y detección de bordes en los estímulos visuales.

Hubel y Wiesel descubrieron dos tipos de neuronas: simples y complejas, las cuales, si bien son distintas, comparten la característica de excitarse si el estímulo visual que reciben está alineado con los patrones que estas células tienen. Ésta será una característica que se encuentran también en las células de este tipo de arquitectura((Martin., 2018))

### 2.6.2. Arquitectura de las Redes Neuronales Convolucionales

Las redes neuronales convolucionales consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Tras cada capa se suele añadir una función para realizar un mapeo causal no-lineal.

En su versión más común, dedicada generalmente a la clasificación, se encuentra al principio una fase de extracción de características. Esta fase está compuesta de neuronas convolucionales y de reducción de muestreo. Por el contrario, al final de la red se ven neuronas perceptrón sencillas para realizar la clasificación final sobre las características extraídas.

En la primera de las fases, y a medida que progresan los datos a través de la misma, se disminuye la dimensionalidad, lo cual hace que las neuronas de las capas lejanas sean mucho menos sensibles a las perturbaciones en los datos de entrada, sin embargo, estas neuronas son activadas por características cada vez más complejas.

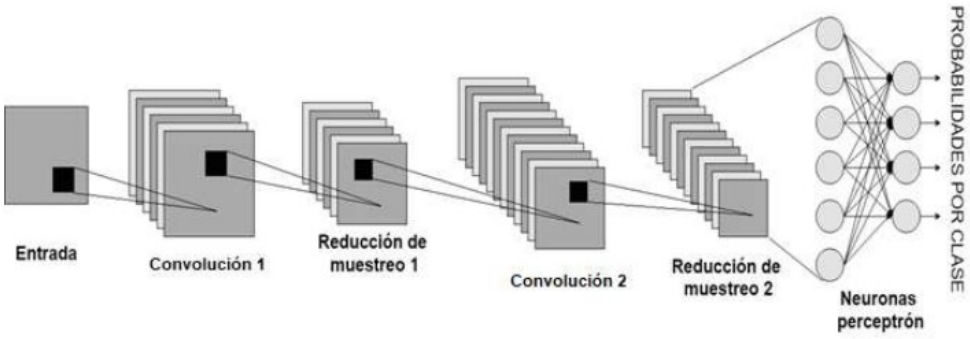


FIGURA 2.9: Estructura clásica de una red convolucional para clasificación

### 2.6.3. ¿Qué es una convolución?

El principal objetivo de la operación de convolución aplicada a las redes neuronales convolucionales es el de extraer características de la imagen de entrada.

La convolución, a grandes rasgos, es el proceso de añadir cada elemento de la imagen ponderado por un núcleo a sus vecinos locales.

El núcleo -kernel- previamente nombrado es una matriz cuadrada de tamaño impar e igual o menor a la imagen de entrada. Cada kernel es útil para determinadas tareas, como detección de bordes horizontales, verticales, etc.

Para realizar la convolución se procede de la siguiente manera, nótese que se realiza la operación sobre un fragmento de la imagen original del mismo tamaño al del núcleo:

1. Se voltea el núcleo tanto horizontal como verticalmente. Si el núcleo es simétrico, de forma obvia este paso no tiene ninguna consecuencia.

2. Se multiplica cada elemento de la matriz núcleo por su similar local de la imagen.

3. Se suman todos los resultados de las multiplicaciones del paso anterior.

4. El elemento central del fragmento tomado de la imagen obtendrá el valor que se obtuvo en el paso 3. Este proceso se repite, colocando el centro del núcleo sobre cada elemento de la imagen.

Sin embargo, se pueden encontrar problemas en los bordes de la imagen, pues se necesitaría multiplicar elementos del kernel con elementos que no existen en la imagen, pues están más allá de los bordes. Para solventar este inconveniente existen diversas alternativas:

- Extender los bordes de la imagen tanto como sea necesario
- Tomar los píxeles necesarios del otro extremo de la imagen
- Ignorar los cálculos más allá de los bordes.

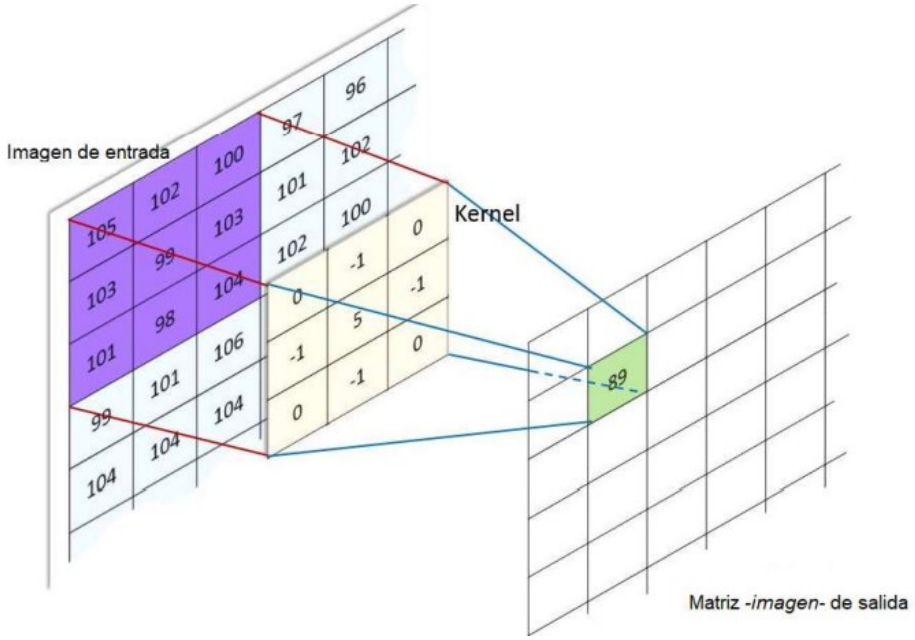


FIGURA 2.10: Ejemplo de operación de convolución

### 2.6.4. Tipos de Capas y Neuronas en Redes Convolucionales

Encontramos en este tipo de redes 3 capas distintas: de convolución, de reducción o muestreo y de clasificación. A continuación, describiremos cada una de ellas:

**De convolución** : En estas capas se encuentran las neuronas de convolución, cada una de estas neuronas tienen un kernel -generalmente distinto-, y se encuentra un gran número de estas neuronas por capa. Adicionalmente, cada neurona aplica la operación descrita en el apartado 2.6.3, produciendo una nueva imagen como resultado.

Es muy común que en este tipo de neuronas se utilice una función de activación llamada ReLU por las siglas en inglés de unidad lineal rectificada (descripta anteriormente).

Sin embargo, también existe una versión más suave, llamada softplus y definida como:

$$f(x) = \ln(1 + e^x) \tag{2.7}$$

**De reducción de muestreo**: La razón de la existencia de estas capas es la creencia de que la localización exacta de una característica es menos importante que su localización aproximada con respecto a otras características. De esta forma, la capa de reducción de muestreo permite reducir paulatinamente el consumo espacial de la representación de la imagen, el número de parámetros y el costo computacional de la red. Además, permite reducir el sobreajuste y aumentar la invariancia frente a la traslación.

Hay diversas funciones de reducción de muestreo no lineales entre las que elegir a la hora de implementar este tipo de capas en redes convolucionales, destacando por ser la más utilizada la técnica de max pooling, la cual se basa en elegir, de entre la región seleccionada, la activación máxima. Otras formas, también usadas son average pooling y L2-norm pooling.

La forma más común de este tipo de capas trabaja con filtros 2x2 con un salto de 2 elementos, consiguiendo por tanto reducir el muestreo en un 75 % de las activaciones al realizar cada operación sobre 4 elementos.

Cabe mencionar a su vez que la operación de reducción de muestreo -o pooling- actúa de forma independiente en cada capa de la entrada.

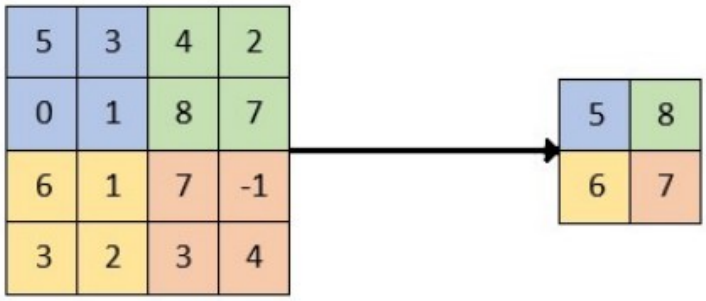


FIGURA 2.11: Ejemplo de Max Pooling.

**De clasificación**: Este tipo de capas cuentan con neuronas como las vistas en el anterior apartado, suelen recibir las características de las capas convolucionales y de pooling para decidir qué objetos puede haber en una imagen -en el caso de la clasificación de objetos en imágenes-.

Es importante mencionar la tendencia a eliminar este tipo de capas en algunas redes neuronales convolucionales, en este tipo de redes -llamadas FCN (Fully Convolutional Network)- el proceso de aprendizaje se hace en su totalidad mediante filtros, incluso en las capas de clasificación finales. Además, las redes totalmente convolucionales pueden manejar entradas de tamaño variable, mientras que aquellas redes con neuronas perceptrón no son capaces de hacerlo ((Martin., 2018)).

## 2.7. Herramientas para la implementación de los clasificadores

### 2.7.1. Python, Tensor Flow y Keras

#### Python

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. En los últimos años su utilización ha ido constantemente creciendo y en la actualidad es uno de los lenguajes de programación más empleados para el desarrollo de software.

Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar los más populares, como Windows, Mac OS X y Linux.

Pero, además, Python también puede funcionar en smartphones, Nokia desarrolló un intérprete de este lenguaje para su sistema operativo Symbian.

¿Tiene Python un ámbito específico? Algunos lenguajes de programación sí que lo tienen. Por ejemplo, PHP fue ideado para desarrollar aplicaciones web. Sin embargo, este no es el caso de Python. Con este lenguaje podemos desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica de usuario (GUI), para crear juegos, para smartphones y por supuesto, para aplicaciones web.


FIGURA 2.12: Logo de Python

Empresas y organizaciones del calibre de Industrial Light and Magic, Walt Disney, la NASA, Google, Yahoo, Red Hat y Nokia hacen uso intensivo de este lenguaje para desarrollar sus productos y servicios. Esto demuestra que Python puede ser utilizado en diversos tipos de sectores, con independencia de su actividad empresarial.

Entre las principales razones para elegir Python, son muchos los que argumentan que sus principales características lo convierten en un lenguaje muy productivo. Se trata de un lenguaje potente, flexible y con una sintaxis clara y concisa. Además, no requiere dedicar tiempo a su compilación debido a que es interpretado.

Python es open source, cualquiera puede contribuir a su desarrollo y divulgación.

Además, no es necesario pagar ninguna licencia para distribuir software desarrollado con este lenguaje. Hasta su intérprete se distribuye de forma gratuita para diferentes plataformas. ((Montoro., 2012))

**Tensorflow**

TensorFlow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos. Actualmente es utilizado tanto en la investigación como en los productos de Google, frecuentemente remplazando el rol de su predecesor de código cerrado, DistBelief. TensorFlow fue originalmente desarrollado por el equipo de Google Brain para uso interno en Google antes de ser publicado bajo la licencia de código abierto Apache 2.0 el 9 de noviembre de 2015.

TensorFlow es el sistema de aprendizaje automático de segunda generación de Google Brain, liberado como software de código abierto el 9 de noviembre de 2015. Mientras la implementación de referencia se ejecuta en dispositivos aislados, TensorFlow puede correr en múltiple CPUs y GPUs (con extensiones opcionales de CUDA para informática de propósito general en unidades de procesamiento gráfico).

TensorFlow está disponible en Linux de 64 bits, macOS, y plataformas móviles que incluyen Android e iOS. Los cómputos de TensorFlow están expresados como stateful dataflow graphs. El nombre TensorFlow deriva de las operaciones que tales redes neuronales realizan sobre arrays multidimensionales de datos. Estos arrays multidimensionales son referidos como "tensores". En junio de 2016, Jeff Dean de Google declaró que 1,500 repositorios en GitHub mencionaron TensorFlow, de los cuales solo 5 eran de Google.((Ortiz., 2018))

En el TensorFlow Dev Summit del 6 de marzo de 2019 se anunció la versión alfa de TensorFlow 2.010. TensorFlow 2.011 se centra en la simplicidad y la facilidad de uso, con actualizaciones importantes como el modelo de ejecución (modo eager), consolidar el uso de una API intuitivas de alto nivel (basada en Keras) y el despliegue flexible de modelos en cualquier plataforma. ((Martin Abadi, 2015))

#### Keras

Keras es una librería de redes neuronales escrita en Python y capaz de ejecutarse tanto sobre Tensorflow como sobre Theano. Sus principales características son ((Cortes., 2017)):

- Fácil y rápido prototipado gracias a su modularidad, minimalismo y extensibilidad.
- Soporta tanto redes neuronales convolucionales como recurrentes (así como la combinación de ambas)
- Soporta esquemas de conectividad arbitrarios (incluyendo entrenamiento multi-entrada y multi-salida)
- Corre en CPU y GPU
- Es compatible con Python 2.7-3.5

### 2.7.2. OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X, Windows y Android. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión etérea y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C, C++ optimizados y Python, aprovechando además las capacidades que proveen los procesadores multinúcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel ((Wikipedia, 2012))

### 2.7.3. Interfaz grafica

Hasta hace algunos años atrás, la GUI (Interfaz Gráfica de usuario) era considerada parte secundaria al desarrollar una aplicación. Sólo se ponía énfasis en lograr que la aplicación contara con todas las funcionalidades requeridas.

La GUI simplemente mostraba las acciones que se podían realizar sin dar importancia a cómo las veía el usuario.

Con el paso del tiempo, las aplicaciones comenzaron a formar parte de la vida cotidiana. Cada vez más usuarios, con o sin conocimientos, necesitaban interactuar con Interfaces de Usuario. Justamente, pensando en los usuarios inexpertos se comenzó a desarrollar una Ingeniería de Interfaces.

La Interfaz de Usuario es la parte del software que las personas pueden ver, oír, tocar, hablar; es decir, donde se pueden entender. La Interfaz de Usuario tiene esencialmente dos componentes: la entrada y la salida. La entrada es cómo una persona le comunica sus necesidades o deseos a la computadora. Algunos componentes de entrada comunes son el teclado, el ratón, un dedo (para pantallas sensibles al tacto: touch screen), y la voz de uno (para las instrucciones habladas). La salida es la forma en que la computadora transmite los resultados a lo solicitado por el usuario. Hoy en día el mecanismo de salida de la computadora más común es la pantalla, seguido de mecanismos que aprovechan las capacidades auditivas de una persona: de voz y sonido.

En la actualidad la GUI es parte fundamental de cualquier aplicación, y por lo tanto tiene tanta importancia como el desarrollo de la aplicación en sí. Existen tres puntos de vista distintos en una GUI: el Modelo del Usuario, el Modelo del Diseñador y el Modelo del Programador.

- Modelo del Usuario: el usuario tiene su propia visión del sistema y espera que se comporte de determinada manera. El modelo de usuario se puede conocer estudiándolo a través de test, entrevistas, realimentación.

- Modelo del Diseñador: el diseñador es quién se encarga de unir las ideas, necesidades y deseos del usuario, con las herramientas que dispone el programador para desarrollar el software. Este modelo consta de tres partes: la Presentación que es lo primero que llama la atención del usuario; luego adquiere más importancia la Interacción que es donde el usuario constata si el producto satisface sus expectativas. La tercera y última parte es la de Relaciones entre objetos aquí es donde se define la relación entre el modelo mental del usuario y los objetos de la Interfaz.

- Modelo del Programador: es el modelo más fácil de visualizar porque se puede especificar formalmente. Este modelo consta de los objetos que manipula el programador, distintos a los que maneja el usuario (el programador maneja una base de datos, el usuario la llama agenda o contactos). El usuario no ve los objetos que maneja el programador. Si bien el programador conoce de plataforma de desarrollo, sistema operativo, lenguajes y herramientas de programación, especificaciones; no significa que tenga la habilidad de proporcionar al usuario modelos más adecuados.

Los distintos modelos nos permiten conocer cómo visualizan la GUI cada uno de los actores. Cada uno de éstos son protagonistas al momento del diseño. Conocer cada punto de vista permite comprender los principios y reglas. ((Usuario., 2014))

Para la implementación de la interfaz gráfica en este trabajo se utilizó la librería Tkinter la cual proporciona a las aplicaciones de Python una interfaz de usuario fácil de programar. Además, es un conjunto de herramientas GUI de Tcl/Tk (Tcl: Tool Command Language), proporcionando una amplia gama de usos, incluyendo aplicaciones web, de escritorio, redes, administración, pruebas y muchos más.

Este viene incluido en Python por lo que se puede decir que es casi un standard de él. Se distribuye junto con el propio interprete de Python, es multiplataforma y está muy bien documentado ((Corso., 2017))

**Diseño de la interfaz grafica**

Se diseña una interfaz gráfica con el propósito de poder capturar múltiples imágenes con una cámara web, para que el usuario pueda decidir cuál de ellas analizar (con redes entrenadas para clasificar el estado emocional de la imagen). Para capturar la foto mediante la cámara web, cuando una persona sea detectada, se utiliza la librería de OpenCV.

Para el desarrollo de esta interfaz gráfica se usa la librería de Tkinter en Python, la misma está formada por la ventana principal la cual tiene: 4 botones de comando, 4 imágenes ,3 filas de botones radiales (cada una con 4 opciones), 2 etiquetas (para indicar el contenido de los botones radiales) y un comando para finalizar el programa.

**Partes de la interfaz grafica**

**Ventana Principal**

Para la creación de la ventana principal se configuran las medidas de la misma y se establece la distancia que tiene con respecto al vértice superior izquierdo de la pantalla, las medidas son de 690x350 pixeles y el vértice superior izquierdo de la ventana esta desplazada 10 cm a la derecha y 10 cm para abajo del vértice del monitor.

La ventana lleva el nombre de Detector de emociones, y el logo que muestra en la esquina superior izquierda

**Botones**

- Botón Iniciar Cámara

  La finalidad de este botón es que inicialice la cámara web, y que posteriormente el sistema realice la detección del rostro y sus respectivas capturas guardando dichas imágenes.

- Botón Analizar con Convolución

  Se utiliza para comenzar el análisis de la red ya entrenada (con Red Neuronal de Convolución) de forma tal que se pueda predecir el estado anímico de la persona (en una nueva ventana)

- Botón Analizar con SVM

  Se utiliza para comenzar el análisis de la red ya entrenada con SVM (Support Vector Machine) de forma tal que se pueda predecir el estado anímico de la persona (en una nueva ventana)

- Botón Salir

  Sirve para finalizar el programa. Cierra todas las ventanas abiertas.

**Cuadro de imágenes**

Utiliza un módulo de la librería pil, (ImageTK) para visualizar las diferentes imágenes que se capturaron del rostro de la persona a analizar.

Al inicio de la ejecución del programa estas imágenes contienen los números del 1 al 4 y una vez que la cámara web capture las fotos, dichos números serán reemplazados por las fotos.

**Botones radiales**

Los botones radiales, se utilizan para ofrecerle al usuario la posibilidad de elegir una opción entre varias.

Botones radiales para selección del número de fotos

Contiene 4 botones radiales, los cuales tienen las etiquetas de 1 Foto, 2 Fotos, 3 Fotos y 4 Fotos, estos sirven para indicar la cantidad de fotos que puede tomar la cámara web.

- Botones radiales para seleccionar el intervalo entre fotos

  Posee 4 botones de radio, los cuales tienen las etiquetas de 1 seg, 2 seg., 3 seg. y 4 seg, estos sirven para indicar la cantidad de segundos que demora en capturar entre una y otra foto.

- Botones radiales para selecciona una de las 4 imágenes capturadas

  Arriba de cada imagen aparece un botón de radio con la etiqueta Numero 1, Numero 2, Numero 3, Numero 4. Se utiliza, para seleccionar una de las imágenes capturadas, que posteriormente será analizada al presionar el botón de análisis correspondiente.

  Para que esta opción se encuentre habilitada se deberá previamente haber capturado las fotos con Iniciar Cámara, se habilitan los botones de acuerdo a la cantidad de fotos capturadas.

## 2.8. Detección de rostro humano implementado en OpenCV

El rostro humano es un objeto dinámico que tiene un alto grado de variabilidad en su apariencia lo cual hace que su detección sea un problema difícil de tratar en visión por computador

Para este trabajo se utiliza una técnica de detección de rostros frontales como etapa inicial de un sistema automático para el reconocimiento de emociones a partir del análisis del movimiento y deformación del rostro.

La detección de objetos mediante clasificadores en cascada basados en funciones de Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones en su documento, Rapid Object Detection using a Boosted Cascade of Simple Features en 2001, Con el aporte de este algoritmo de Viola-Jones se pudo construir la librería OpenCV. ((Paul Viola, 2001))

El algoritmo de Viola-Jones se basa en una serie de clasificadores débiles denominados Haar-like features que se pueden calcular eficientemente a partir de una imagen integral. Estos clasificadores, que por sí mismos tienen una probabilidad de acertar solo ligeramente superior a la del azar, se agrupan en una cascada empleando un algoritmo de aprendizaje basado en AdaBoost para conseguir un alto rendimiento en la detección, así como una alta capacidad discriminativa en las primeras etapas.

### 2.8.1. Haar-like features

Las Haar-like features son los elementos básicos con los que se realiza la detección. Estas features son rasgos o características muy simples que se buscan en las imágenes y que consisten en la diferencia de intensidades luminosas entre regiones

rectangulares adyacentes. Las features por tanto quedan definidas por unos rectángulos y su posición relativa a la ventana de búsqueda y adquieren un valor numérico resultado de la comparación que evalúan.

En el trabajo presentado por Viola-Jones existen tres tipos de features representadas en la (Figura 2.13):

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

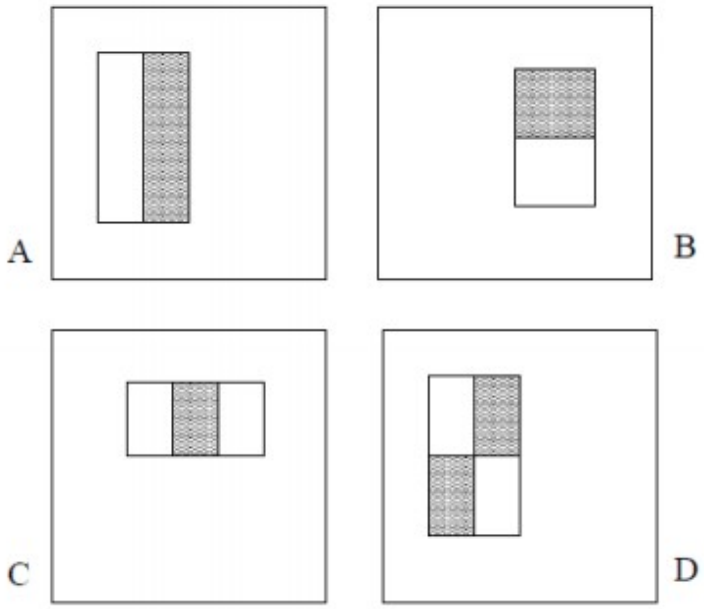


FIGURA 2.13: Ejemplos de features de dos, tres y cuatro rectángulos y su posición relativa a la ventana de búsqueda (Paul Viola and Michael J. Jones, 2001). La suma de los píxeles en las áreas grises se resta a la de las áreas blancas

En el trabajo de Viola-Jones, las features se definen sobre una ventana de búsqueda básica de 24x24 píxeles, lo que da lugar a un gran número de features posibles.

### 2.8.2. Imagen Integral

La suma de los píxeles de un rectángulo puede ser calculada de manera muy eficiente empleando una representación intermedia denominada imagen integral. La imagen integral en el punto (x, y) contiene la suma de todos los píxeles que están

arriba y hacia la izquierda de ese punto en la imagen original.

$$ii(x,y) = \sum_{x' \leq x,\ y' \leq y} i(x',y') \tag{2.8}$$

donde ii(x,y) es la imagen integral y i(x,y) es la imagen original.

La imagen integral se puede calcular en un solo barrido de la imagen empleando el siguiente par de sentencias recurrentes:

$$s(x,y) = s(x,y-1) + i(x,y) \tag{2.9}$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \tag{2.10}$$

donde s(x,y) es la suma acumulada de la fila x, con s(x,-1)=0 y ii(-1,y)=0.

Usando la imagen integral, cualquier suma rectangular se puede calcular con cuatro referencias a memoria como se muestra en la (Figura 2.14). Las features de dos rectángulos se pueden computar con 6 referencias a memoria puesto que comparten vértices. En el caso de features de tres rectángulos se pasa a 8, y a 9 para features de cuatro rectángulos.

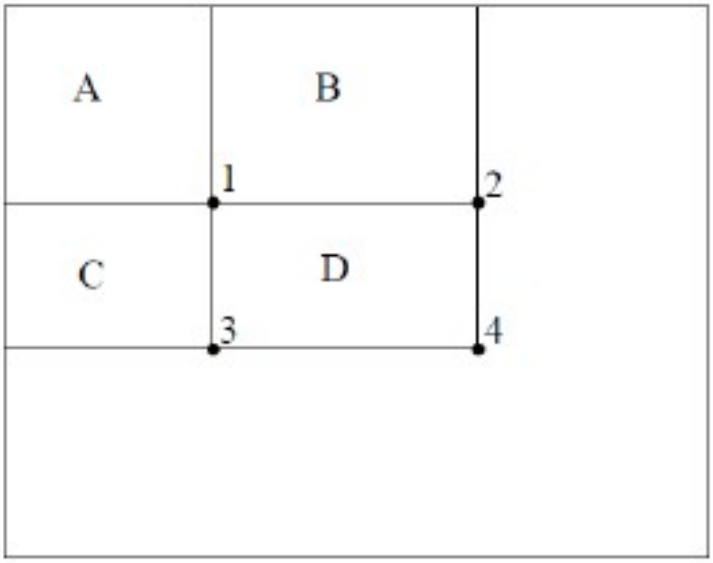


FIGURA 2.14: Ejemplo del cálculo de la suma de píxeles en un rectángulo . El valor de la suma de los píxeles en el rectángulo D es igual al valor de la imagen integral en 4 (A+B+C+D) menos el valor de la imagen integral en 2 y en 3 (A+B, A+C) y más el valor de la imagen integral en 2 y en 3 (A+B, A+C) y más el valor de la imagen integral en 1 (A).

### 2.8.3. Proceso de aprendizaje

Es necesario realizar un proceso de entrenamiento supervisado para crear la cascada de clasificadores. Este proceso se realiza mediante un algoritmo basado en AdaBoost, un meta algoritmo adaptativo de machine learning cuyo nombre es una abreviatura de adaptative boosting.

El boosting consiste en tomar una serie de clasificadores débiles y combinarlos para construir un clasificador fuerte con la precisión deseada. AdaBoost fue introducido por Freund y Schapire en 1995 resolviendo muchas de las dificultades prácticas asociadas al proceso de boosting ((Freund, 1999))

En el procedimiento de Viola Jones, AdaBoost se utiliza tanto para seleccionar un pequeño set de features de las 180000 posibles como para entrenar el clasificador.

Para seleccionar features, se entrenan clasificadores débiles limitados a usar una única feature. Para cada feature, el clasificador débil determina el valor umbral que minimiza los ejemplos mal clasificados. Un clasificador débil hj(x) por tanto consiste en una feature fj, un valor umbral j y un coeficiente pj indicando la dirección del signo de desigualdad.

$$h_j(x) = \begin{cases} 1 & \text{si } p_j f_j(x) < p_j \theta_j, \\ 0 & e.o.c, \end{cases} \tag{2.11}$$

A continuación, se describe el algoritmo de AdaBoost empleado y se incluye una representación visual del proceso (Figura 2.15). En cada ronda se selecciona un clasificador débil y por tanto una feature.

- Se parte de un conjunto de imágenes $(x_1, y_1), , (x_n, y_n)$ donde $y_i = 0, 1$ para ejemplos negativos y positivos respectivamente.
- Se inicializan los pesos $w_1, i = \frac{1}{2m}, \frac{1}{2l}$ para $y_i = 0, 1$ respectivamente donde m es el número de negativos y l el número de positivos.
- Para cada ronda, t = 1, , T:

a. Normalizar los pesos:

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \tag{2.12}$$

b. Para cada feature, j, entrenar un clasificador $h_j$ que solo use una feature. El error se evalúa teniendo en cuenta los pesos $w_t$,

$$\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right| \tag{2.13}$$

c. Se escoge el clasificador, $h_t$, con menor error $ct$

d. Se actualizan los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \tag{2.14}$$

Donde ei = 0 si el ejemplo $x_i$ se clasifica correctamente y 1 en caso contrario y

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \tag{2.15}$$

El clasificador fuerte final es:

$$h(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t, \\ 0 & \text{si } e.o.c, \end{cases} \tag{2.16}$$

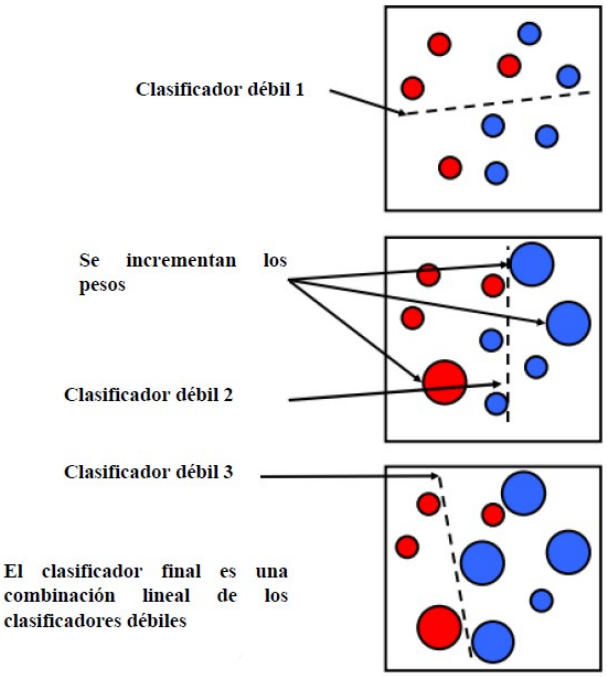


FIGURA 2.15: Representación visual del proceso de AdaBoost . En cada ronda se aumenta el peso de los ejemplos mal clasificados anteriormente y se busca un nuevo clasificador que minimice el error.

### 2.8.4. Cascada atencional

En vez de construir un único clasificador mediante el proceso descrito en el apartado anterior, se pueden construir clasificadores más pequeños y eficientes que rechacen muchas ventanas negativas (es decir, aquellas que no incluyan ninguna instancia del objeto buscado) manteniendo casi todas las positivas (es decir, las que contienen una instancia del objeto buscado). Estos clasificadores más simples se utilizan para rechazar la mayoría de las ventanas de búsqueda y solo en aquellas en las que hay mayores probabilidades de encontrar caras se llama a clasificadores más complejos que disminuyan el número de falsos positivos. Este proceso se representa en la 2.16.

Se obtiene así una cascada de clasificadores, cada uno de los cuales es entrenado con AdaBoost y después sus valores umbrales se ajustan para minimizar los falsos negativos.

La cascada entrenada por Viola-Jones tiene 38 etapas y más de 6000 features pero de media se evalúan únicamente 10 features por ventana de búsqueda ((Paul Viola, 2001))

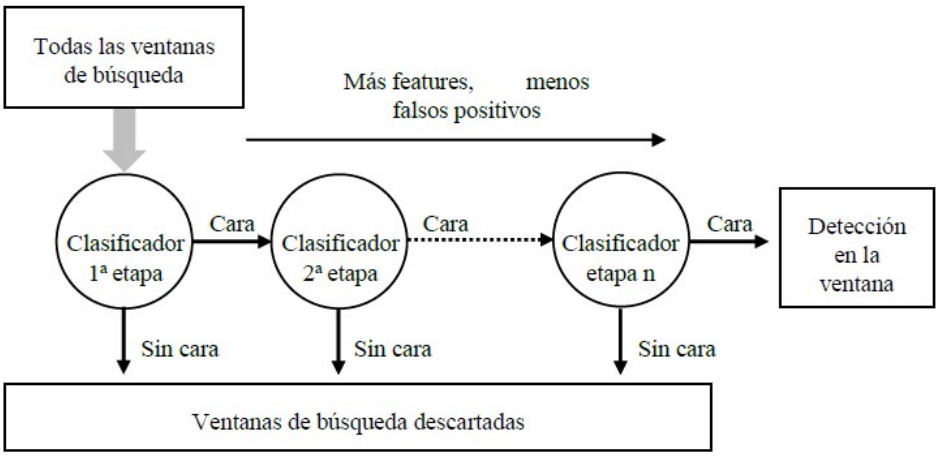


FIGURA 2.16: Descripción esquemática de una cascada atencional [10]. Una serie de clasificadores se aplican a cada ventana de búsqueda. Los clasificadores iniciales rechazan un gran número de ventanas rápidamente.

### 2.8.5. Proceso de detección

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

Para normalizar se emplea la varianza:

$$\sigma^2 = m^2 - \frac{1}{N}\sum x^2 \tag{2.17}$$

Donde m es la media del valor de los píxeles, que puede calcularse a partir de la imagen integral. La suma de los píxeles al cuadrado se puede obtener a partir de una imagen integral de la imagen al cuadrado.

La cascada de features se evalúa sobre una ventana de búsqueda cuadrada que barre la imagen con incrementos de unos pocos píxeles. La búsqueda se realiza a distintas escalas obtenidas al multiplicar la escala anterior por un factor de escala, normalmente entre 1.1 y 1.3.

Puesto que el detector es poco sensible a pequeñas translaciones y diferencias de escala, se suelen producir múltiples detecciones alrededor de cada cara. De hecho, se puede exigir que las detecciones tengan un determinado número mínimo de detecciones vecinas para disminuir el número de falsos positivos.

Para combinar las detecciones que se refieren al mismo objeto, se fusionan las detecciones cuyas áreas se solapen más de un determinado valor umbral y el recuadro con la detección final se calcula como la media de todos los recuadros que se han fusionado ((Barrero., 2015)) .

## 2.9. Detección de Landmarks (puntos faciales) y DLIB

Las marcas faciales dicen mucho de una persona. Ellas permiten transmitir sentimientos, también ayudan en la comunicación verbal y no verbal. Las marcas faciales de una cara son expresiones que hablan de sentimientos, acciones, pensamientos e indudablemente son datos que se pueden analizar.

Para facilitar el proceso de clasificación de expresiones faciales se buscan los puntos característicos del rostro o Facial Landmarks, que nos permitirá reconocer 67 puntos faciales agrupados en partes de un rostro como: Boca, Cejas, Ojos, Nariz, Mandíbula

Los métodos basados en landmarks utilizan las relaciones geométricas entre puntos característicos de la cara. Para ello, es necesario detectar la posición de estos landmarks de forma precisa, ya que de esta detección dependen resultados futuros.

DLIB es un conjunto de herramientas de machine learning implementado en Python que fue diseñado para resolver problemas en muchas áreas industriales, incluyendo robótica, dispositivos móviles o visión artificial. DLIB es una librería de código abierto y licencia libre. La librería Dlib incluye, además de una clase que detecta el rostro en la imagen, otra clase que permite obtener las coordenadas (x, y) de los puntos que definen la pose de un objeto (en el caso del rostro, se obtiene las coordenadas de los Landmarks que definen la cara). En concreto, se emplea un modelo de 68 puntos faciales (que se muestra en la (Figura 2.17) ).((Roca., 2017))

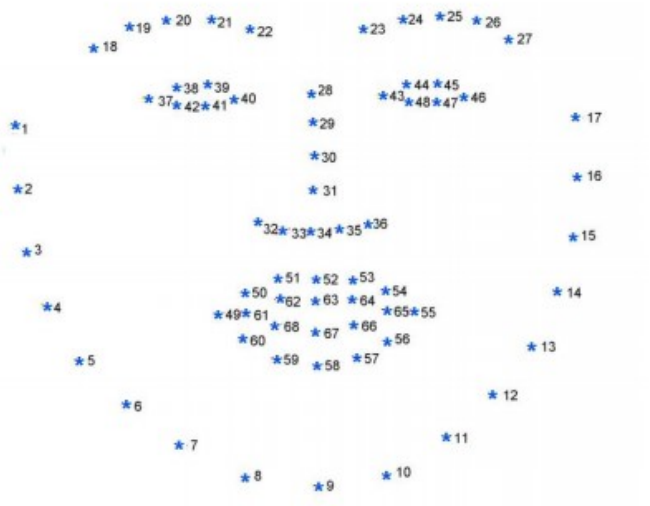


FIGURA 2.17: Distribución de los puntos faciales del modelo de 68 Landmarks

Los 68 puntos de referencia del rostro se distribuyen de la siguiente forma: 17 puntos que definen el contorno de la cara, 6 puntos que definen el contorno de cada ojo, 5 puntos que definen el contorno de cada ceja, 9 puntos para definir la nariz y 20 puntos para definir los contornos de los labios ((Scherhag, 2018))

En la figura a continuación se muestran algunos ejemplos de imágenes procedentes de la base de datos Cohn-Kanade, con la representación de los 68 Landmarks detectados en cada rostro. Se representan varias emociones diferentes para observar cómo se detectan los puntos en cada expresión facial.

FIGURA 2.18: Puntos encontrados en la cara con la librería dlib.

# Capítulo 3

# Configuracion experimental

## 3.1. Preparación de la base de datos (entrenamiento, validación y testeo) para los diversos modelos

Se hizo una selección de imágenes, las cuales provienen de la base de datos KDEF (Karolinska Directed Emotional Faces) que contiene un total de 4900 imágenes las cuales tienen un tamaño de 562x762 pixeles, con 32 bit de color, de las cuales se seleccionaron los rostros de frente de diversas personas con diferentes emociones quedando de este modo un total de 800 imágenes.

De estas imágenes se seleccionaron 200 por cada emoción las cuales son: Feliz, Enojado, Sorpresa y Neutral, permitiendo de este modo tener 4 clases bien diferenciadas

Las 800 imágenes seleccionadas fueron divididas en tres subconjuntos:

Conjunto de entrenamiento con 560 imágenes que representa el 70 %.

Conjunto de validación con 120 imágenes que representa 15 %.

Conjunto de testeo con 150 imágenes que representa el 15 %

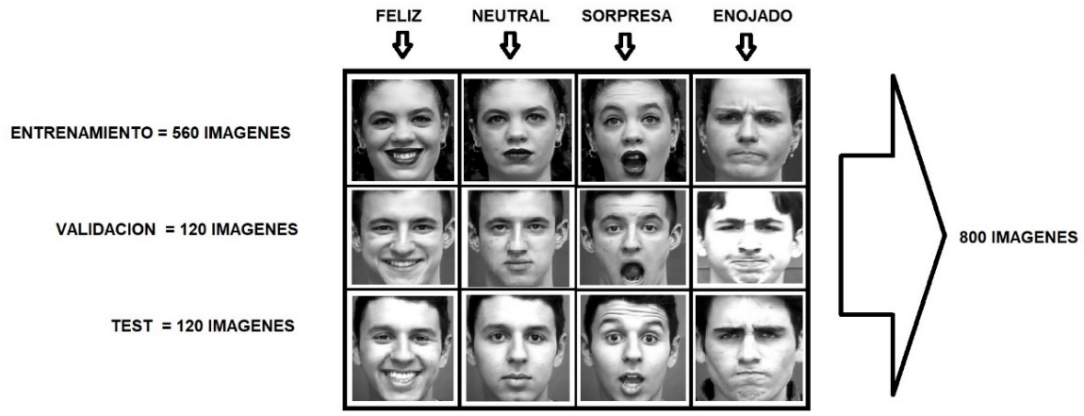


FIGURA 3.1: Cantidad de imágenes para entrenamiento, validación y test

El tamaño de las imágenes se modificó a 150 pixeles de altura y 150 pixeles de ancho, es decir de 150x150 pixeles.

Cada imagen fue convertida a escala de grises para su posterior uso.

## 3.2. Modelos basados en redes neuronales de convolución

Se proponen dos modelos de redes de convolución Mod1 y Mod2 cuyas arquitecturas se diferencian únicamente en la cantidad de filtros. La cantidad de capas, tamaños de los filtros y funciones de activación se describe en el cuadro 3.1

CUADRO 3.1: Configuración de Mod1 y Mod2.

| **Caracteristicas** | **Mod1** | **Mod2** |
|---|---|---|
| Funcion de activación: | Relu | Relu |
| Función de activación de las ultimas capas | Softmax | Softmax |
| Capas de convolución | 2 | 2 |
| Tamaño del primer filtro | 3,3 | 3,3 |
| Tamaño del segundo filtro | 2,2 | 2,2 |
| Capas de Max pooling | 2 | 2 |
| Tamaño del filtro | 2,2 | 2,2 |
| Épocas (número de veces que itera sobre el set de datos de entrenamiento) | 50 | 50 |
| Pasos (número de veces que se procesa la información en cada una de las épocas) | 100 | 100 |
| Clases | 4 | 4 |
| Numero de imágenes a procesar en cada paso | 32 | 32 |
| Learning rate | 0,0004 | 0,0004 |
| Cantidad de filtros (también llamadas neuronas) | 256 | 512 |

## 3.3. Modelo basado en los puntos característicos del rostro y el clasificador SVM.

Este modelo Mod3 se ejecuta en dos etapas o pasos .

### 3.3.1. Detección de puntos característicos (Landmarks)

Se detectan los puntos característicos del rostro (Landmarks) utilizando la librería dlib. Es decir, los puntos característicos son ubicados en las siguientes zonas: boca, ceja derecha, ceja izquierda, mandíbula y nariz. Luego, en ésta misma etapa se realiza el correspondiente etiquetado a dichos puntos característicos.

### 3.3.2. Implementación del clasificador SVM

Se realiza la clasificación de las emociones mediante un clasificador SVM que toma como entrada los puntos faciales característicos encontrados en la detección de puntos característicos.

## 3.4. La interfaz gráfica en Mod2 y Mod3

Se analiza el comportamiento de la interfaz gráfica (desarrollada por el autor) junto a la cámara web la cual tiene el propósito de formar una pequeña base de

datos con imágenes del rostro (de 1 a 4 fotos), se evaluara si la detección y captura de rostros del programa es adecuada, y si produce un recorte correcto de la cara de la persona, otro aspecto importante a verificar es si la cantidad de capturas que realiza es acorde a lo que el usuario ha prefijado en la interfaz gráfica, y si los tiempos que se establecen en el programa entre captura y captura de imágenes se cumplen.

## 3.5. El sistema de reconocimiento de emociones en un caso real

Se evaluará el sistema de reconocimiento de emociones en un caso real, es decir que se analizara la capacidad del programa para clasificar las emociones de las fotos capturadas mediante la cámara web con la red neuronal convolucional y SVM . También se escuchará como ejecuta un tema musical acorde a la emoción que encuentre.

## 3.6. Mejorando la predicción del sistema de reconocimiento de emociones para caso real

Para mejorar la exactitud en la clasificación de las redes entrenadas en mod2 y mod3 , se añadirán imágenes del usuario del sistema (el autor del trabajo) a la base de datos para el reentrenamiento con fines de ver si de este modo se puede obtener un mayor número de aciertos en la clasificación.

Para evaluar el comportamiento de las redes anteriormente entrenadas aplicadas en este sistema se incluirán 4 escenarios , con las diferentes emociones , estos son :

- Rostro sin variación
- Rostro con anteojos comunes
- Rostro con anteojos de sol
- Rostro con gorra

Cada escenario será analizado un total de 5 veces con cada emoción, de esta forma se determinará el éxito y el comportamiento con cada imagen, es decir que al ser 4 escenarios , tener 4 emociones cada uno y realizar 5 repeticiones , se producirán en total 80 imágenes para analizar.

# Capítulo 4

# Analisis de los resultados

## 4.1. Resultados utilizando Mod1 y Mod2

Al realizar el entrenamiento durante 50 épocas, del Mod1 y Mod2, se pudo observar los siguientes resultados:

CUADRO 4.1: Exactitud de Mod1 y Mod2

| | **Mod1** | **Mod2** |
|---|---|---|
| Exactitud de entrenamiento | 97,11 % | 99,47 % |
| Exactitud de validación | 78 % | 84 % |
| Exactitud de testeo | 77 % | 82 % |

En la siguiente grafica se muestran los resultados de la exactitud de entrenamiento y de validación en las diversas épocas de Mod1:

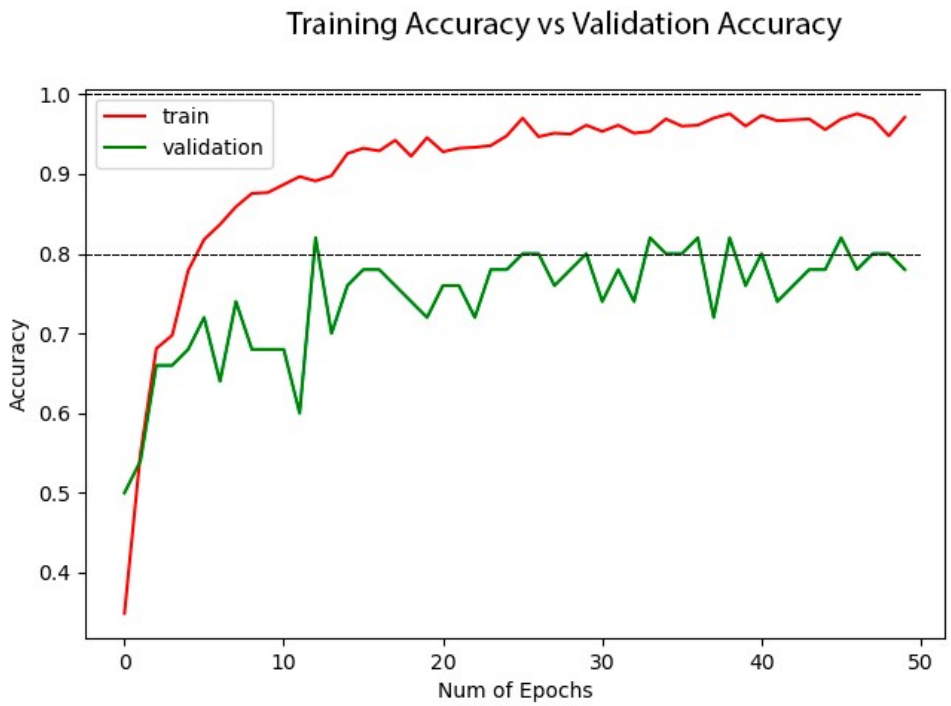


FIGURA 4.1: Grafica de exactitud del entrenamiento y validación con 256 neuronas

Con esta grafica puede apreciarse el proceso de entrenamiento el cual llega a tener una exactitud de prácticamente el 95 por ciento en las 50 épocas

La grafica también muestra los resultados de la validación los cuales desde la época 45 a 50 oscilan entre el 75 y 82 por ciento de aciertos. Siendo un valor relativamente alto.

En la gráfica se puede apreciar que a partir de las 30 épocas, ya se posee un resultado de entrenamiento y validación adecuados, se eligió una cantidad de 50 épocas porque a pesar de que no se observa una disminución de la exactitud en la validación con una cantidad mayor tampoco se obtiene una mejor ganancia del sistema , y con una cantidad menor el resultado en la exactitud de entrenamiento es inferior, sin embargo para esta implementación se considera que a partir de 30 épocas brinda un resultado aceptable. . En la siguiente grafica se muestran los resultados de la exactitud de entrenamiento y de validación en las diversas épocas de Mod2:

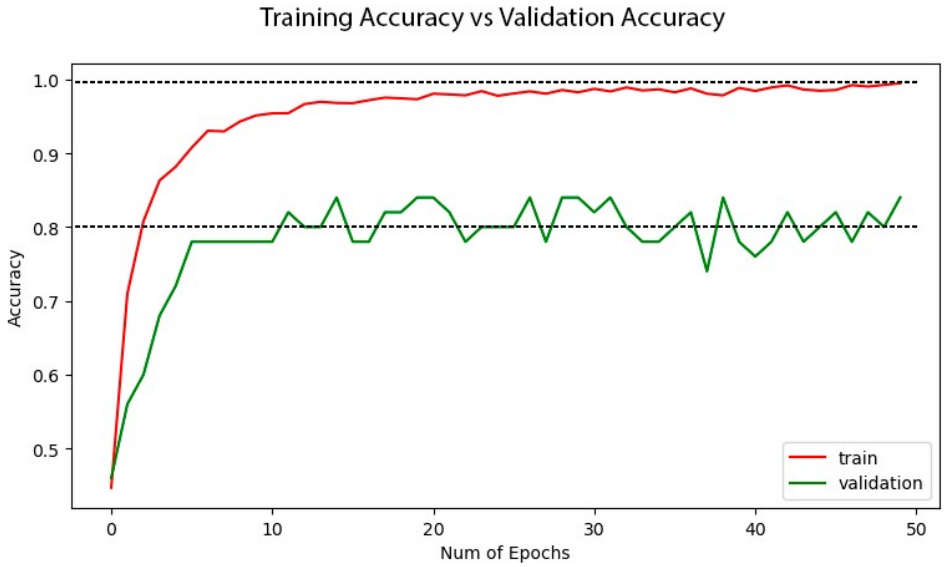


FIGURA 4.2: Grafica de exactitud del entrenamiento y validación con 512 neuronas

Con esta grafica puede apreciarse el proceso de entrenamiento el cual llega a tener una exactitud de prácticamente el 100 por ciento en las 50 épocas

La grafica también muestra los resultados de la validación los cuales desde la época 45 a 50 oscilan entre el 78 y 85 por ciento de aciertos. Siendo un valor relativamente alto.

En la gráfica se puede apreciar que a partir de las 20 épocas, ya se posee un resultado de entrenamiento y validación adecuados, se eligió una cantidad de 50 épocas porque a pesar de que no se observa una disminución de la exactitud en la validación con una cantidad mayor tampoco se obtiene una mejor ganancia del sistema , y con una cantidad menor el resultado en la exactitud de entrenamiento es inferior, sin embargo para esta implementación se considera que a partir de 20 épocas brinda un resultado aceptable.

Puede apreciarse claramente que con la Red Neuronal obtenida en el mod2, la exactitud de testeo es más altas, motivo por el cual, para armar la red neuronal de convolución, elegiremos una cantidad de 512 filtros.

## 4.2. Resultados utilizando Mod3

### 4.2.1. Resultados de la detección de puntos característicos (Landmarks)

Los resultados obtenidos con esta implementación, en el cual se utilizó la librería de dlib fueron totalmente exitosos dado que se pudo encontrar todos los puntos del rostro (landmarks), y posteriormente con estos realizar la clasificación de las diferentes partes de la cara. Lo descripto puede apreciarse en la imagen (perteneciente al conjunto de datos para el entrenamiento) mostrada a continuación (Figura 4.3).

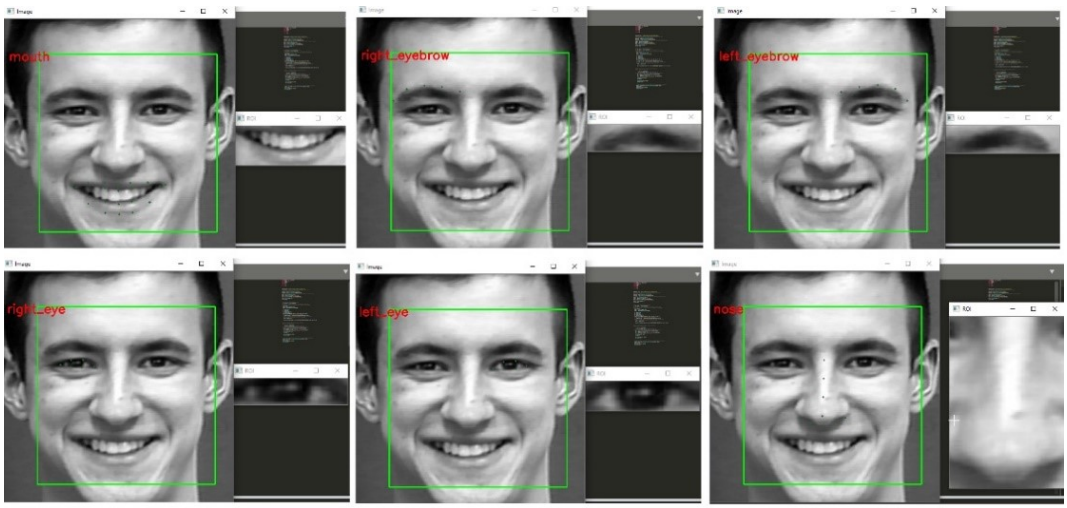


FIGURA 4.3: Resultado del programa con la librería DLIB encontrando los puntos faciales del rostro.

### 4.2.2. Resultados de la implementación de SVM

En un total de 120 ensayos los resultados exitosos con esta implementación son de 103 de ellos, lo cual representa un 86 % de asertividad. Los resultados no exitosos (14 %), en los cuales nos indica una emoción que no es la correspondiente para la imagen.

Del modelo Mod3 se pudo observar en los casos no exitosos, que la emoción que tenía el segundo valor de porcentaje normalmente era la emoción correcta, por lo que se plantea para un trabajo futuro la posibilidad de añadir un sistema de verificación con el Mod2, para de este modo poder reducir el error obtenido ,

## 4.3. Comparación de los modelos Mod2 y Mod3

En el testeo de ambos modelos, con imágenes diferentes con las que fueron entrenadas, se vieron los siguientes resultados:

CUADRO 4.2: Exactitud de Mod2 y Mod3.

| | **Mod2** | **Mod3** |
|---|---|---|
| Exactitud de testeo | 82 % | 86 % |

En Mod3 la exactitud de testeo es mayor, por lo cual se concluye que se logra un mejor resultado encontrando puntos específicos de la cara (landmarks) y posteriormente entrenando nuestra red neuronal con SVM, esto puede deberse a que estamos guiando a que el sistema estudie ciertos factores del rostro para hacer la clasificación.

## 4.4. Verificación de la interfaz

### 4.4.1. Resultado de la pantalla de captura de imágenes

Se utiliza la interfaz gráfica, y la misma es evaluada con las diferentes funcionalidades que debe cumplir el programa, con este fin se detalla y se somete a prueba el funcionamiento de cada parte, a continuación, se muestra la pantalla inicial.


FIGURA 4.4: Interface Grafica antes de capturar fotos del rostro con la cámara Web.

La interfaz gráfica está compuesta por 4 botones, uno que permite iniciar cámara, otro que aporta la posibilidad de cerrar la aplicación y 2 más que permiten analizar la imagen capturada. También cuenta con 4 imágenes que en su estado inicial muestran números y dependiendo de la opción que se elija en el botón radial 1 2 3 o 4 Fotos, será la cantidad de imágenes tomadas cuando la cámara detecte rostros.

Otra opción que tiene la interfaz es la de definir cada que intervalo de tiempo se realiza una toma (al detectarse un rostro), la cual al obtener varias fotos evita que sean idénticas por intervalos de tiempo muy reducido, esto se ajusta con el botón radial que tiene una etiqueta cuyo texto muestra: Intervalo entre fotos (s): 1 2 3 o 4 seg.

Al presionar el botón de iniciar cámara, se visualiza lo que la webcam puede captar, hasta que encuentre un rostro y automáticamente realizara una captura de la cara, este proceso terminara cuando se tome la cantidad de imágenes que se seleccionó en el botón radial

A continuación, se muestran los resultados obtenidos al capturar una serie de imágenes a partir de la cámara web (Figura 4.5):

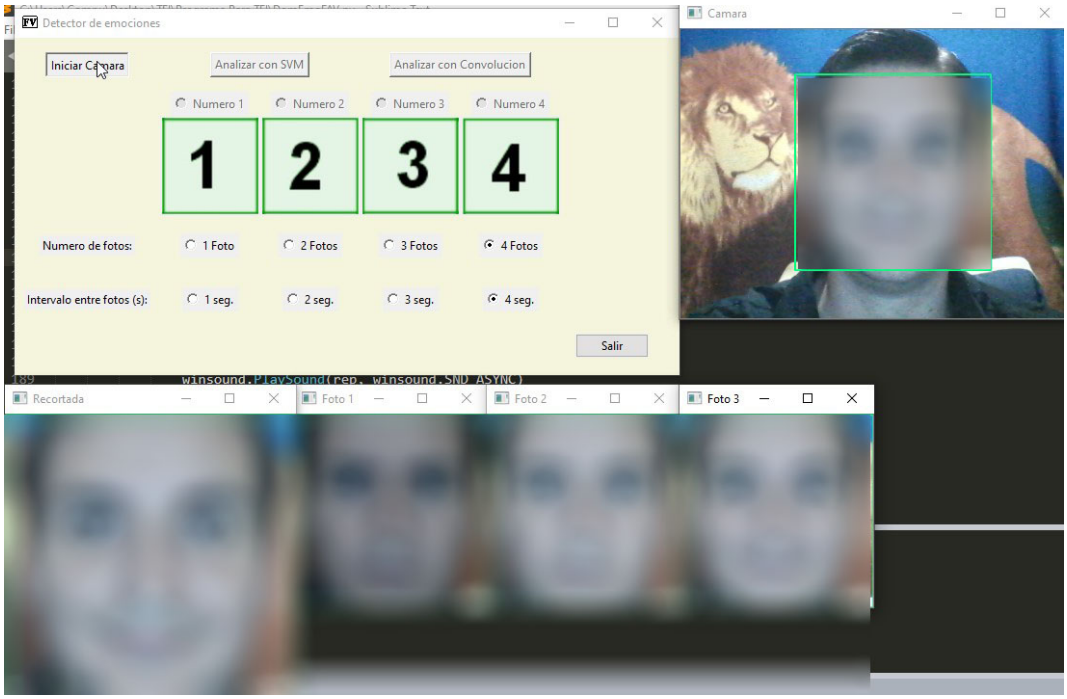


FIGURA 4.5: Interfaz gráfica, capturando imágenes con la cámara web

La toma de imágenes de la cámara web se muestra en 2 ventanas, una contiene la imagen que va filmando la cámara (en todo su campo visual, esta ventana tiene el título de Cámara), una segunda ventana muestra la filmación, pero únicamente del sector de la cara (ventana con nombre Recortada), este recort

También se muestran 4 ventanas (las cuales aparecen según el número de fotos que se elija), que son las imágenes del rostro capturadas por webcam.

El sistema de captura se produce cada determinado tiempo el cual es definido con el botón de radio, es decir si se elige como en la foto un intervalo de tiempo de 4 segundos, capturara una nueva imagen una vez que transcurre este tiempo.

## 4.5. Resultados del sistema en un caso real.

Con la cámara web se capturaron 4 fotos del rostro del autor de este documento, haciendo clic en el botón iniciar cámara, estas imágenes forman parte de una pequeña base de datos para estudiar el correcto funcionamiento de la interfaz gráfica.

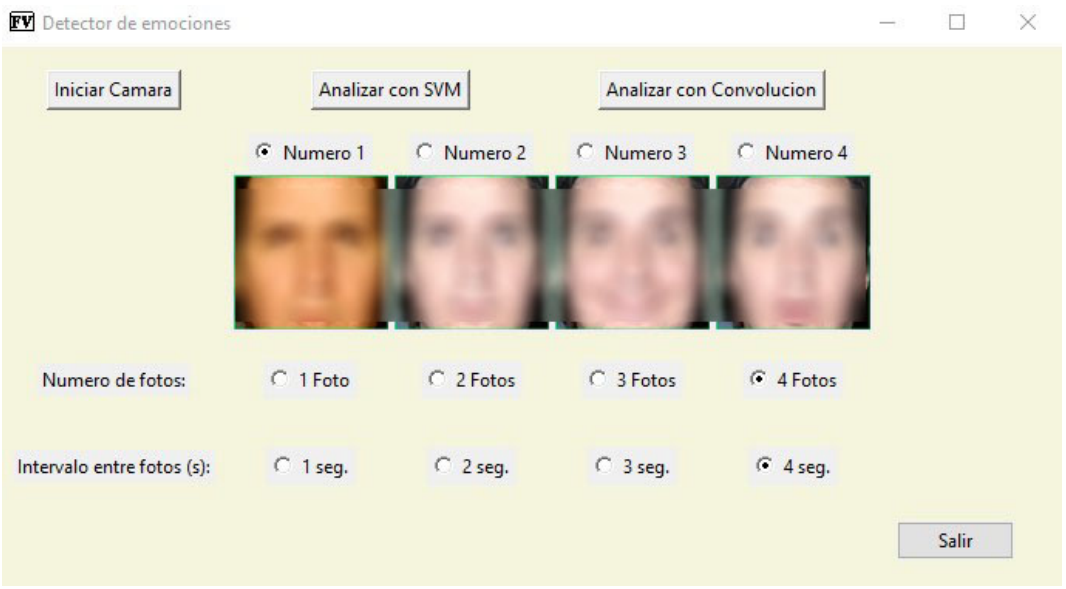


FIGURA 4.6: Interfaz gráfica, con las fotos capturadas mediante la cámara web.

Se escoge el número de imagen a analizar, y se hace clic sobre el botón analizar con convolución, con un procedimiento análogo se obtiene el análisis con SVM.

A continuación, se muestra 8 análisis de los cuales 4 fueron analizados con convolución y los otros 4 con SVM , los cuales nos muestran casos que fueron positivos.

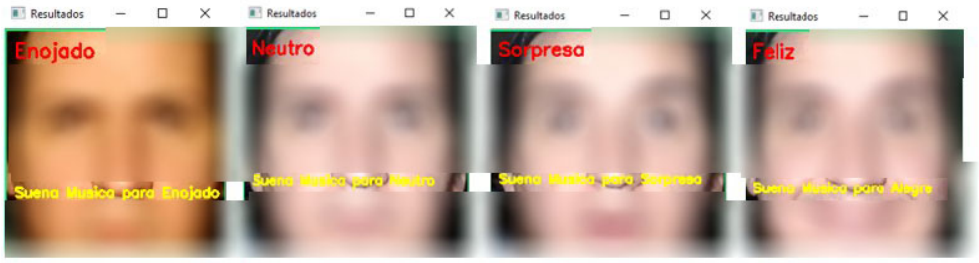


FIGURA 4.7: Resultados del análisis de las imágenes con red neuronal convolucional


FIGURA 4.8: Resultados del análisis de las imágenes con SVM

En la siguiente tabla puede apreciarse los aciertos y fallos con un total de 80 imágenes analizadas.

CUADRO 4.3: Aciertos y fallas en el análisis con red neuronal convolucional, y SVM

| | Aciertos | | Fallos | |
|---|---|---|---|---|
| | Cantidad | Porcentaje( %) | Cantidad | Porcentaje( %) |
| Convolución | 66 | 82,5 | 14 | 17,5 |
| SVM | 69 | 86,25 | 11 | 13,75 |

## 4.6. Resultados del sistema mejorado para caso real

Se muestran los resultados de las diferentes imágenes analizadas (capturadas mediante la cámara web), las cuales fueron tomadas con el autor del trabajo en 4 escenarios seleccionados , esto conforma un total de 80 imágenes para analizar con la Red neuronal de convolución y con SVM .

### 4.6.1. Resultados del análisis con la Red Neuronal Convolucional en los diferentes escenarios

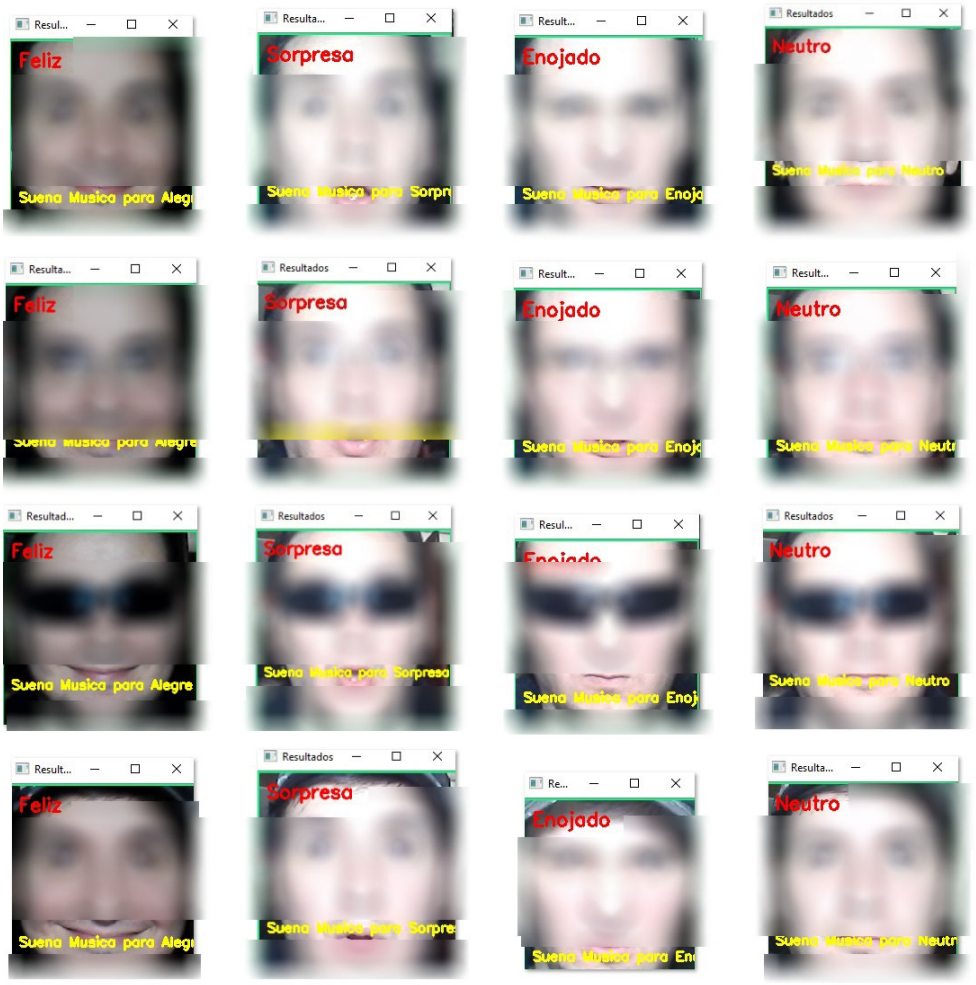


FIGURA 4.9: Resultados del análisis con red neuronal convolucional, para los 4 escenarios

**Aciertos de las tomas en los 4 escenarios con Redes Neuronales Conv.(RNC)**

CUADRO 4.4: Resultado del analisis con RNC en Escenario Nř 1

| Escenario Nř 1 - Sin variacion externa en el rostro | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Aserto | Fallo | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 5 | 5 | 4 | 5 |
| Total de aciertos | 19 de 20 (95 %) | | | |

CUADRO 4.5: Resultado del analisis con RNC en Escenario Nř 2

| Escenario Nř 2 Con anteojos comunes (transparentes) | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Fallo | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Fallo | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 4 | 4 | 5 | 5 |
| Total de aciertos | 18 de 20 (90 %) | | | |

CUADRO 4.6: Resultado del analisis con RNC en Escenario Nř 3

| Escenario Nř 3 Con anteojos de sol | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Fallo | Aserto | Fallo |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Fallo | Aserto | Aserto |
| 5 toma | Fallo | Fallo | Aserto | Aserto |
| Aciertos por emocion | 4 | 2 | 5 | 4 |
| Total de aciertos | 15 de 20 (75 %) | | | |

CUADRO 4.7: Resultado del analisis con RNC en Escenario Nř 4

| Escenario Nř 4 Con gorra | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Fallo | Aserto | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 4 | 5 | 5 | 5 |
| Total de aciertos | 19 de 20 (95 %) | | | |

En los 4 escenarios de las 80 capturas del rostro , 71 fueron exitosas , lo que nos da un 88,75 por ciento de clasificaciones correctas

### 4.6.2. Resultados del análisis con SVM en los diferentes escenarios

Para esta etapa se realiza un procedimiento análogo al realizado en el punto anterior, donde el programa analizara mediante la red con SVM las imágenes que fueron tomadas (80 imágenes en total).

**Resultados del análisis con SVM, en la primera toma en los cuatro escenarios .**

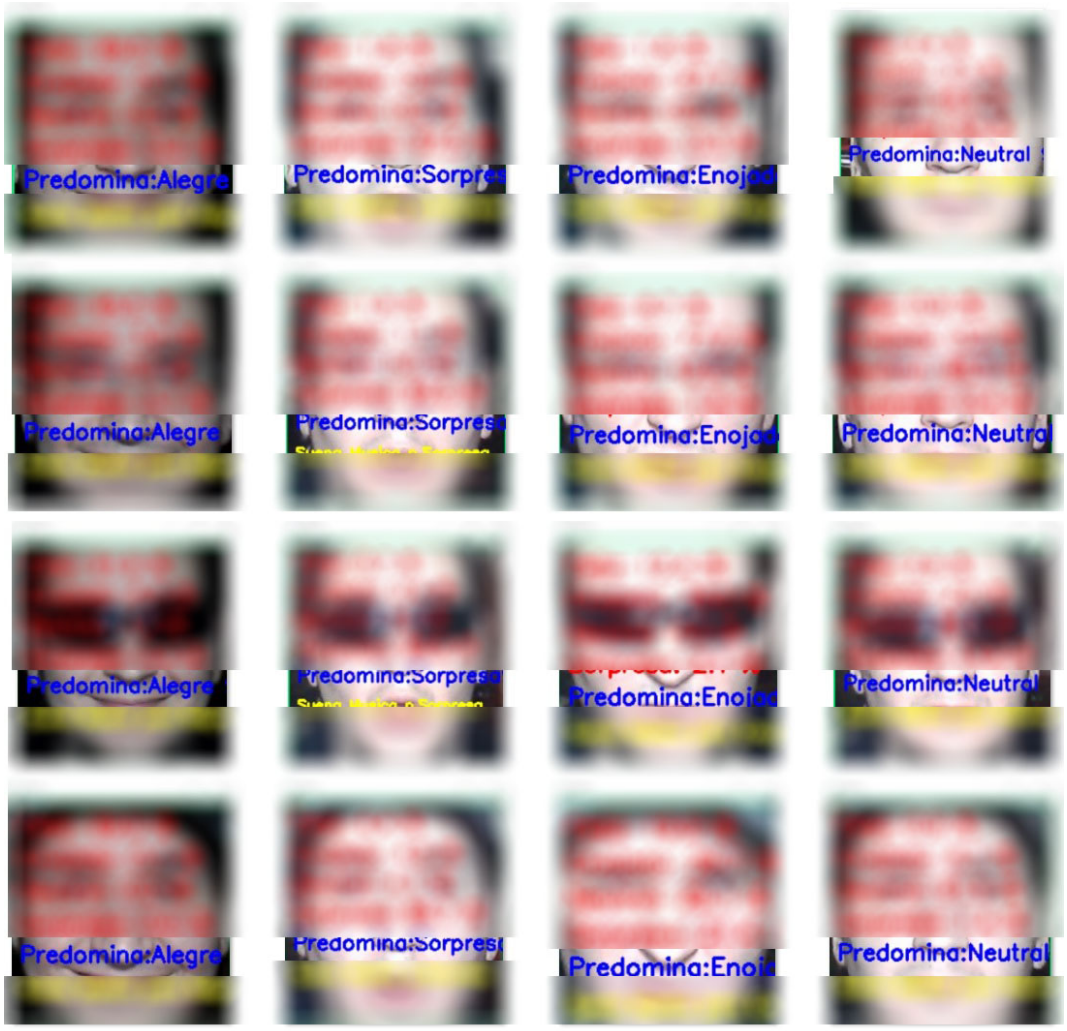


FIGURA 4.10: Resultados del análisis con SVM, para los 4 escenarios

**Aciertos de las 5 tomas en todos los escenarios con SVM**

CUADRO 4.8: Resultado del analisis con SVM en Escenario Nř 1

| Escenario Nř 1 - Sin variacion externa en el rostro | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Aserto | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 5 | 5 | 4 | 5 |
| Total de aciertos | 20 de 20 (100 %) | | | |

CUADRO 4.9: Resultado del analisis con SVM en Escenario Nř 2

| Escenario Nř 2 Con anteojos comunes (transparentes) | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Aserto | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 5 | 5 | 5 | 5 |
| Total de aciertos | 20 de 20 (100 %) | | | |

CUADRO 4.10: Resultado del analisis con SVM en Escenario Nř 3

| Escenario Nř 3 Con anteojos de sol | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Fallo | Aserto |
| 4 toma | Aserto | Aserto | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 5 | 5 | 4 | 5 |
| Total de aciertos | 19 de 20 (95 %) | | | |

CUADRO 4.11: Resultado del analisis con SVM en Escenario Nř 4

| Escenario Nř 4 Con gorra | | | | |
|---|---|---|---|---|
| | Feliz | Sorpresa | Enojado | Neutral |
| 1 toma | Aserto | Aserto | Aserto | Aserto |
| 2 toma | Aserto | Aserto | Aserto | Aserto |
| 3 toma | Aserto | Aserto | Aserto | Aserto |
| 4 toma | Aserto | Aserto | Aserto | Aserto |
| 5 toma | Aserto | Aserto | Aserto | Aserto |
| Aciertos por emocion | 5 | 5 | 5 | 5 |
| Total de aciertos | 20 de 20 (100 %) | | | |

En los 4 escenarios de las 80 capturas del rostro solo una fallo, lo que nos da un 98 por ciento de clasificaciones correctas

## 4.7. Comparación y conclusiones de los modelos e implementaciones realizadas

En el presente trabajo se abordaron dos métodos para clasificar el estado emocional de una persona, lo que permite afirmar que es posible realizar la clasificación de emociones por múltiples caminos, siendo unos más preciso e idóneos que otros.

El primer método en el que se orientan los modelos (mod1, mod2) está formado por una red neuronal profunda, la cual aprende a clasificar diferentes emociones según diferentes imágenes de entrada agrupadas en 4 clases diferentes. En mod1 , se obtiene un 78,5 % de clasificaciones correctas (con imágenes con la que no fueron entrenadas ), es decir que podría utilizarse en un entorno el cual acepte un error de 21,5 % aproximadamente , y en mod2 , con un total de 512 neuronas se obtiene un 82 % de clasificaciones correctas, con estos valores se llegó a la conclusión que si bien el desempeño de ambos modelos presentan resultados aceptables, es preferible para la detección de emociones utilizar la red del mod2 (por su mayor exactitud).

En mod3 se implementó el segundo método de detección de emociones, el cual gracias a una librería denominada dlib (que tiene la capacidad de encontrar los puntos significativos de la cara) y a la red armada con SVM, permitió clasificar diversas emociones estimando un porcentaje para cada una. Se observa en este modelo que se puede tener una exactitud del 86 por ciento con este método.

Se puede concluir del segundo método que, al generar un vector de salida con los puntos más importantes del rostro, su entrenamiento y clasificación es enfocada en los sectores específicos que señalan el estado emocional de una persona brindando de esta manera un resultado optimo gracias a su estratificación de puntos. Con lo expuesto anteriormente se llega a determinar que de los 2 métodos utilizados el segundo presenta un mejor resultado que el primero, gracias a que el mismo estratifica la búsqueda de determinados puntos faciales, por el contrario, en el método Nř 1 hay que tener especial cuidado en que la clasificación se realice por el estado emocional y no por otros factores ajenos a este, como ser la iluminación y otros aspectos. Esta conclusión puede verse fundamentada en los mod2 y mod3 (donde se observa un mejor resultado en la detección de emociones en el mod3).

Se evaluó el desempeño de la interfaz gráfica y se pudo ver la capacidad del programa para producir un recorte de la imagen en el rostro. El programa es capaz de capturar 4 fotos mediante la cámara web, y posteriormente analizar dichas imágenes, con los métodos antes descriptos (realizados en el mod2 y mod3). Con la integración descripta, se tuvo un programa 100 % funcional, el cual puede analizar diferentes emociones a partir de las imágenes capturadas por la webcam, y suministrar una música de fondo acorde a la emoción. Por otra parte, se logró mejorar la cantidad de aciertos , esto fue posible gracias a que se entrenó nuevamente las redes de mod2 y mod3 con imágenes de la persona que utiliza el sistema .

Se puede concluir que esta interfaz es de vital importancia para que este sistema pueda ser utilizado por cualquier usuario promedio (el cual no posea conocimientos avanzados en informática), dando de este modo la ventaja de que las personas puedan capturar sus propias imágenes, escogiendo que tipo de evaluación prefieren (por Convolución o SVM).

# Capítulo 5

## 5.1. Conclusión y recomendaciones

Se logro mediante el uso de la cámara web y de la librería OpenCV capturar el rostro de la persona a analizar, de esta forma se almacena dicha imagen en la memoria formando una pequeña base de datos la cual será utilizada para evaluar la emoción que expresa la misma.

La red neuronal convolucional (mod2) tuvo la capacidad de detectar las 4 emociones, con un resultado exitoso del 82 por ciento, y el clasificador SVM (mod3) tuvo mejores resultados (con las 4 emociones) con una cantidad de éxitos del 86 %, lo que nos permite concluir que entre estos 2 métodos el que se basa en muestras estratificada, como es el caso de SVM con puntos específicos de la cara, nos proporciona un mejor método para la clasificación.

Se pudo observar que la interfaz gráfica nos facilita la captura de imágenes del rostro con la cámara web , como así también los botones y herramientas para el estudio de dichas imágenes . Otro aporte importante que se observó es que se puede mejorar la exactitud de la clasificación de las redes reentrenándolas al añadirle fotos de la persona a analizar, teniendo una cantidad de éxitos del 88,75 % para las redes neuronales convolucionales , y del 98 % para las SVM. También se determinó que si la cara presenta variabilidad con anteojos u gorra no determina una dificultad para la red SVM, causando una leve dificultad en el caso que se utiliza una red de convolución al usar anteojos de Sol al tener una precisión del 75 % con estos anteojos.

La implementación de sonidos acorde a la emoción encontrada, a modo demostrativo, fue perfectamente implementada con Python.

Con lo antes mencionado se concluye que es posible realizar un sistema con una interfaz gráfica amigable al usuario, que tenga la capacidad de clasificar 4 emociones en las personas, las cuales son: Felicidad, enojo, Neutralidad, sorpresa, y acorde a esto modificar el ambiente de la casa, en este trabajo solo se optó por modificar el sonido del ambiente, sin embargo, también se podría modificar la iluminación color de luces y otros factores.

Con esto queda demostrado que la domótica puede seguir incursionando en la vida de las personas aportándoles la posibilidad de vivir mejor, e interactuando con ellos, aprendiendo de las emociones humanas y permitiendo de esa forma crear un ambiente confortable.

# Capítulo 6

## 6.1. Trabajos a futuro

Se plantea la posibilidad de trabajos a futuro en los cuales se combine los resultados aportados por el mod2 (de la red neuronal convolucional), con los resultados del mod3 (del sistema con SVM), de modo tal que ambos resultados nos permitan validar la correcta elección de la emoción para reducir los posibles errores al detectarla e inclusive agregar un tercer sistema de detección de emociones para poder hacer competir los resultados.

También se presenta como posibilidad de trabajo a futuro añadir más emociones para que el sistema pueda detectarlas, aumentando de esta forma el espectro de posibilidades.

Otro punto interesante para trabajo a futuro es el de acompañar el sonido con la iluminación del hogar, y agregar un sistema de chat para que interactúe con la persona (acorde a la emoción detectada).

# Bibliografía


A., Damasio (1994). *El error de Descartes.*

— (2005). *En busca de Spinoza. Neurobiología de la emoción y los sentimientos.*

A., Ortony (1990). *Whats basic about basic emotions?*

Arias., Mauricio Gonzalez (2006). *Aspectos Psicológicos y Neurales en el Aprendizaje del Reconocimiento de Emociones.*

Barrero., Eloy Parra (2015). *Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal.*

C. Gonzalez Restrepo S. Rincon Montoya, O. L. Quintero Montoya. (2014). *Metodologia para Deteccion de Caracteristicas Faciales con Fines de Reconocimiento de Emociones.*

Corso., Ing.Cynthia Lorena (2017). *Lenguaje Python.*

Cortes., Carlos Antona (2017). *Herramientas modernas en redes neuronales: La libreria keras.*

Cristian Alejandro Rojas T. Gabriel Elías Chanchí G., Katerine Márceles Villalba. (2019). *Propuesta de una Arquitectura IoT para el control domótico e inmótico de edificaciones.*

D., Premack (1978). *Does the chimpanzee have a theory of mind.*

David Costa Da Silva Belen Rios Sanchez, Julian Fierrez Aguilar. (2018). *Reconocimiento facial no invasivo en dispositivos moviles.*

Edison Rene, Caballero Barriga. (2017). *Aplicación practica de la visión artificial para el reconocimiento de rostros en una imagen, utilizando redes neuronales y algoritmos de reconocimiento de objetos de la biblioteca OpenCV.*

Freund Y., Schapire R. (1999). *A short Introduction to Boosting.Journal of Japanese Society for Artificial Intelligence,.*

Gareth James Daniela Witten, Trevir Hastie Robert Tibshirani. (2015). *An Introduction to Statistical Learning- with Applications in R.*

Humberto Perez Espinosa, Carlos Alberto Reyes Garcia. (2010). *Reconocimiento de Emociones a partir de voz basado en un Modelo Emocional Continuo.*

J., LeDoux (1999). *El cerebro emocional.*

Javier Trujillano, J. M. (2004). *Aproximacion metodologica al uso de redes neuronales artificiales para la prediccion de resultados en medicina.*

K., Scherer (2000). *Psychological Models of Emotion.*

Maria Eugenia Tabernero, Daniel Gustavo Politis. (2013). *Reconocimiento de emociones basicas y complejas en la variante conductual de la demencia frontotemporal.*

Martin., Pablo Pastor (2018). *Usando Redes Neuronales Convolucionales para convertir caracteristicas Visuales en estimulos Sonoros.*

Martin Abadi Ashish Agarwal, Paul Barham Eugene Brevdo (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.*

Montoro., Arturo Fernández (2012). *Python 3 al descubierto.*

Ortiz., Giovanni Barrero (2018). *Aplicación de deep learning usando tensorflow para analisis de la calidad de software desarrollado en ibm rpg.*

Paul Viola, Michael Jones. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features.*

R. Gimeno Hernández, J. R. Morros (2010). *Estudio de Tecnicas de reconocimiento Facial.*

Ramon Zatarain-Cabada Maria Lucia Barron Estrada, Gilberto Muñoz-Sandoval. (2016). *Plataforma de reconocimiento multimodal de emociones.*

Roca., Sheila Novoa (2017). *Analisis de operadores de textura para reconocimiento de expresiones faciales.*

S., Baron-Cohen (2001). *The Reading the Mind in the Eyes.*

S., Steidl (2009). *Automatic Classification of Emotion-Related User States in Spontaneous Childrens Speech.*

Saket S Kulkarni Narender P Reddy, SI Hariharan. (2009). *Facial expression (mood) recognition from facial images using committee neural networks.*

Salcedo Poma, Celia Mercedes. (2017). *Estimacion de la ocurrencia de incidencias en declaraciones de polizas de importacion.*

Santiago., Elsa Irene Herrera (2016). *Arquitectura para el reconocimiento de emociones basado en caracteristicas faciales.*

Scherhag Dhanesh Budhrani, U. (2018). *Detecting Morphed Face Images Using Facial Landmarks.*

Solsona., Albert Vilagran (2018). *Facial Expression Detection using Convolutional Neural Networks.*

Stefan Junestrand Xavier Passaret, Daniel Vazquez. (2005). *Domotica y hogar digital.*

Suilan Estevez-Velarde Yudivi, Almeida Cruz. (2015). *Evaluacion de algoritmos de clasificacion supervisada para el minado de opinion en twitter.*

Usuario., Diseño de Interfaz Grafica de (2014). *OpenCV.*

Wikipedia (2012). *OpenCV.*