INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“RECONOCIMIENTO DE EMOCIONES MEDIANTE IMÁGENES DE EXPRESIONES FACIALES BASADO EN ARQUITECTURA DE RED SIAMESA”

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA ELECTRÓNICA

PRESENTA:

LUIS XAVIER NEVÁREZ OCHOA

DIRECTOR DE LA TESIS:
DR. MARIO IGNACIO CHACÓN MURGUÍA

CHIHUAHUA, CHIH., OCTUBRE 2021

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO

Chihuahua, Chih. 11 de octubre de 2021

ROGELIO ENRIQUE BARAY ARANA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE

En cumplimiento con los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica, le notificamos que el documento de tesis del alumno C. LUIS XAVIER NEVÁREZ OCHOA, titulado *"Reconocimiento de emociones imágenes de expresiones faciales basado en arquitectura de red siamesa"* dirigido por el Dr. Mario Ignacio Chacón Murguía ha sido aprobado y aceptado para su impresión.

Por lo anterior, proponemos le sea concedida la autorización de impresión correspondiente.

Agradeciendo la atención a la presente, quedamos de usted:

ATENTAMENTE
Excelencia en Educación Tecnológica®
"La Técnica por el Engrandecimiento de México"

EDUCACIÓN | TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

DR. MARIO IGNACIO CHACÓN MURGUÍA
DIRECTOR DE TESIS

DR. JUAN ALBERTO RAMÍREZ QUINTANA
MIEMBRO DEL COMITÉ TUTORIAL

MTRA. ALMA DELIA CORRAL SÁENZ
MIEMBRO DEL COMITÉ TUTORIAL

MTRO. LUIS ENRIQUE GUERRA FERNÁNDEZ
MIEMBRO DEL COMITÉ TUTORIAL

Chihuahua, Chih. 18 de octubre de 2021

PROGRAMACIÓN DE EXAMEN DE GRADO

En cumplimiento con los requerimientos para otorgar el grado de Maestro en Ciencias en Ingeniería Electrónica, la División de Estudios de Posgrado e Investigación solicita la programación del examen de grado de **C. LUIS XAVIER NEVÁREZ OCHOA** con número de control **G19061418** donde sustentará la tesis titulada ***"Reconocimiento de emociones imágenes de expresiones faciales basado en arquitectura de red siamesa"***, dirigida por el Dr. Mario Ignacio Chacón Murguía.

La fecha de realización del examen es el 22 de octubre de 2021 a las 11:00 h en Sala virtual de Teams y el Jurado de Examen estará constituido de la siguiente manera:

Presidente: DR. MARIO IGNACIO CHACÓN MURGUÍA. Cédula grado: 8031020
Secretario: DR. JUAN ALBERTO RAMÍREZ QUINTANA. Cédula grado: 9769407
Vocal: MTRA. ALMA DELIA CORRAL SÁENZ. Cédula grado: 6960577
Suplente: MTRO. LUIS ENRIQUE GUERRA FERNÁNDEZ Cédula grado: 10213883

El dictamen de otorgamiento del Grado será responsabilidad del Jurado y debe estar de acuerdo con todos y cada uno de los requisitos establecidos por el reglamento e instructivos en vigor.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"La Técnica por el Engrandecimiento de México"

EDUCACIÓN
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

M.C. ROGELIO ENRIQUE BARAY ARANA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

CARTA CESIÓN DE DERECHOS

En la ciudad de Chihuahua el día 18 de octubre de 2021, el que suscribe C. LUIS XAVIER NEVÁREZ OCHOA con número de control G19061418, de la Maestría en Ciencias en Ingeniería Electrónica, adscrita a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del el Dr. Mario Ignacio Chacón Murguía y cede los derechos del trabajo titulado "Reconocimiento de emociones imágenes de expresiones faciales basado en arquitectura de red siamesa", al Tecnológico Nacional de México y/o Instituto Tecnológico de Chihuahua para su difusión, divulgación, transmisión, reproducción, así como su digitalización con fines académicos y de investigación.

C. LUIS XAVIER NEVÁREZ OCHOA

DECLARACIÓN DE ORIGINALIDAD

En la ciudad de Chihuahua el día 18 de octubre de 2021, el que suscribe C. LUIS XAVIER NEVÁREZ OCHOA de la Maestría en Ciencias en Ingeniería Electrónica, con número de control G19061418, adscrito a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual de la tesis titulada "Reconocimiento de emociones imágenes de expresiones faciales basado en arquitectura de red siamesa" bajo la dirección el Dr. Mario Ignacio Chacón Murguía; que el contenido es original y que las fuentes de información consultadas para su fundamentación están debidamente citadas y referenciadas.

C. LUIS XAVIER NEVÁREZ OCHOA

Agradecimientos

Esta tesis la quiero dedicar a mi novia Ana Laura Quiroz Rivera, quien gracias a su amor y apoyo incondicional he llegado hasta donde estoy ahora, pues con sus consejos y mente fría he podido solventar y evitar muchos obstáculos.

A mi familia, en especial a mis padres Edgar y Nora, mis abuelos Salvador†, Alba, Alma y Javier y mis tías Yolanda, Selene, Adriana y Sandra, pues han sido personas presentes durante toda mi vida, alimentando mi curiosidad con libros, platicas y experiencias de sus vidas. Asimismo, el apoyo que me han dado, alentándome siempre a investigar y aprender cosas nuevas, siempre con un enfoque de mejorar la calidad de vida de las personas, aun desde las cosas más pequeñas.

A mis mejores amigos Mariana e Isaac, a mis compañeros de maestría Eliacim, Vicente y Miguel, mis compañeros de laboratorio Abimael, Luis, Javier, David y Alonso y a mis amigos de la carrera Félix, Iván, Uriel y Fidel. Todos ellos me inspiraron a seguir adelante con mis proyectos de vida, dándome aliento y apoyo cuando lo requería, forjando así una amistad que perdurará a lo largo del tiempo.

A mis docentes de maestría e ingeniería y mis miembros de comité: Dr. Mario Chacón, Dr. Juan Ramírez, M.C. Alma Corral, M.C. Jose Robles, M.C. Enrique Hernández, M.C. Edgar Trujillo y M.C. Luis Guerra, quienes desde que exprese mi interés en unirme al programa de Maestría me brindaron sus conocimientos y me explicaron el panorama completo, para así poder tomar una elección. Además, por ser excelentes docentes quienes reivindicaron mi amor por la ciencia e investigación, siendo sus materias las que mas disfrute durante mi estadía en el Instituto Tecnológico de Chihuahua. Además, aquellos que forman parte de mi comité de maestría me han brindado sus correcciones de manera que pudiera mejorar, no solo mi proyecto de tesis, si no como investigador y desarrollador.

A mi mascota Pepe, un acompañante felino que me ha dado muchas alegrías y ha fungido como esponja de estrés, absorbiendo todos mis males con su inocencia y ternura.

Finalmente quiero agradecer al Tecnológico Nacional de México y al Consejo Nacional de Ciencia y Tecnología, mejor conocido como CONACYT, por brindarme instalaciones y equipo donde pudiera realizar mis investigaciones, estudios y tareas, darme apoyo económico suficiente para poder dedicarme en exclusividad a mis responsabilidades de maestría y el desarrollo de mi proyecto de tesis bajo el nombre de “Reconocimiento de Emociones Mediante Imágenes de Expresiones Faciales Basado en Arquitectura de Red Siamesa”.

RESUMEN

“RECONOCIMIENTO DE EMOCIONES MEDIANTE IMÁGENES DE EXPRESIONES FACIALES BASADO EN ARQUITECTURA DE RED SIAMESA”

Ing. Luis Xavier Nevárez Ochoa
Maestría en Ciencias en Ingeniería Electrónica
División de Estudios de Posgrado e Investigación
Tecnológico Nacional de México / I. T. Chihuahua
Chihuahua, Chih., 2020
Director de tesis: Dr. Mario Ignacio Chacón Murguía

Las técnicas de reconocimiento de emociones han evolucionado en los últimos años, pasando desde análisis realizados por personal especializado, hasta modelos computacionales que puedan diferenciar las emociones por medio de señales cerebrales, texto, voz y fotografías o videos. Entre estas, en tesis se hace uso de imágenes de expresiones faciales para la diferenciación y clasificación de emociones.

En el estado del arte se encuentran distintos métodos y tecnologías para discernir qué emoción presenta una persona, así como la cantidad de emociones utilizadas y cuáles son las más comunes de utilizar. Por ello, en este trabajo se tiene como objetivo la creación de un modelo de red profunda con arquitectura siamesa para el reconocimiento de emociones por medio de imágenes y que sea capaz de funcionar sobre un sistema embebido de manera rápida, portable y sin complicaciones de incompatibilidad de paquetes o una pobre optimización. Para desarrollar el modelo final de red siamesa se realizó una etapa previa llevando a cabo entrenamientos con pares de emociones que siguieran estas características: que fueran similares entre sí, que fueran medianamente similares, y que fueran distintas. El número de emociones se fue incrementando hasta que el modelo mantuviera una precisión general por arriba del 80%. Este límite se puede alcanzar hasta con 6 emociones, pero el modelo final resultó ser un modelo para 5 emociones con cerca de 90% de precisión.

En conclusión, se desarrollaron 3 modelos de arquitectura siamesa capaces de funcionar en sistemas de carácter embebido por medio de una migración de la red neuronal desde Matlab

hacia Python y se dan a conocer los resultados de precisión y rendimiento ante diferentes dispositivos. Los 3 modelos presentan rendimiento entre el 80% y 85% de precisión.

CONTENIDO

1. INTRODUCCIÓN 1
2. ANTECEDENTES DE LA TESIS 3
2.1. Bases teóricas 3
2.1.1. Modelos emocionales 3
2.1.2. Métodos y algoritmos para la de detección de emociones 5
2.1.3. Internet de las cosas orientado a emociones 7
2.2. Bases prácticas 9
2.2.1. Bases de datos para entrenamiento 10
2.2.2. Extracción de características 10
2.2.3. Clasificador de características 15
2.2.4. Sistemas embebidos en la detección de emociones 16
2.2.5. Aplicaciones de IoT 17
3. METODOLOGÍAS PARA LA DETECCIÓN DE EMOCIONES 19
3.1. Bases de datos 19
3.1.1. Extended Cohn-Kanade Dataset 19
3.1.2. FER-2013 21
3.1.3. KDEF 21
3.2. Preprocesamiento 23
3.2.1. Color a escala de grises 24
3.2.2. Escalamiento de resolución por medio de interpolación bicúbica 24
3.3. Red siamesa 25
3.4. Extracción de características 30

3.4.1. HOG 31
3.4.2. LBP 32
3.4.3. Red convolucional 33
3.5. Modelos de arquitectura siamesa 34
3.5.1. Método 1. (HOG + LBP) + MLP 34
3.5.2. Método 2. CNN + MLP 38
4. RESULTADOS DEL ENTRENAMIENTO DE LA RED 44
4.1. Pasos previos al entrenamiento de la red 44
4.2. Entrenamiento de la red siamesa. 47
4.3. Resultados preliminares. 47
4.3.1. Entrenamiento con 2 emociones. 48
4.3.2. Entrenamiento con 3 emociones. 49
4.3.3. Entrenamiento con 4 emociones. 51
4.3.4. Entrenamiento con 5 emociones. 52
4.3.5. Entrenamiento con 6 emociones. 53
4.4. Rendimiento de la red en sistemas embebidos 55
4.5. ONNX 56
4.6. Resultados de la red en Python 57
4.7. Análisis de la red 58
5. CONCLUSIONES Y TRABAJO A FUTURO. 64
6. REFERENCIAS 66

LISTA DE FIGURAS

Figura 2.1. Modelo Circunflejo de Russell.....4

Figura 2.2. Círculo de los adjetivos de Hevner. 5

Figura 2.3. Extracción de imágenes presentes en la base de datos de Kaggle [15].....10

Figura 2.4 Modelos con mejor clasificación por emoción en [18].....13

Figura 3.1 Muestra de la base de datos CK+42.....20

Figura 3.2 Muestra de imágenes de la base de datos FER-2013 22

Figura 3.3. Expresiones faciales de un sujeto de la base de datos KDEF para 5 emociones. ..23

Figura 3.5. Esquema general del preprocesamiento de las imágenes.....24

Figura 3.6. Arquitectura general de una red siamesa.....26

Figura 3.7. Funcionamiento de Contrastive Loss. 28

Figura 3.8. Funcionamiento de Triplet Loss.....29

Figura 3.9. Funcionamiento de Quadruplet Loss. 30

Figura 3.10. Imagen y sus características HOG. 32

Figura 3.11. Vecindario del pixel central y los 8 vecinos que se consideran para LBP.....32

Figura 3.12. Arquitectura común de una red CNN..... **¡Error! Marcador no definido.**

Figura 3.13. Esquema general del método HOG + LBP + MLP.....35

Figura 3.14. Matriz de confusión del Método 2 para el set de Pruebas.....38

Figura 3.15. Matriz de confusión del Método 2 para el set de validación.....39

Figura 3.16. Arquitectura de la subred CNN + MLP. 40

Figura 3.17. Esquema general del método CNN + MLP.....41

Figura 3.18. Matriz de confusión del Método 2 para el set de Pruebas.....42

Figura 3.19. Matriz de confusión del Método 2 para el set de Pruebas.....43

Figura 4.1. Tres factores de riesgo encontrados en la base de datos KDEF. a) Iluminación diferente al resto. b) Imagen vacía (negras). c) Aberración cromática.....45

Figura 4.2. Matrices de confusión para el set de pruebas en 2 emociones. Clasificación.....49

Figura 4.3. Matrices de confusión para el set de pruebas en 3 emociones. Clasificación.....52

Figura 4.4. Matrices de confusión para el set de pruebas en 4 emociones. Clasificación.....53

Figura 4.5. Matrices de confusión para el set de pruebas en 5 emociones. Clasificación.....55

Figura 4.6. Matrices de confusión para el set de pruebas en 6 emociones. Clasificación.....56

Figura 4.7. Mapa de activación de la primera capa convolucional para la clase Alegría.....60

Figura 4.8. Mapa de activación de la primera capa convolucional para la emoción Sorpresa. 60

Figura 4.9. Mapa de activación de la segunda capa convolucional para la emoción Alegría. .61

Figura 4.10. Mapa de activación de la segunda capa convolucional para la emoción Sorpresa. 62

Figura 4.11. Mapa de activación de la segunda capa convolucional para la emoción Alegría.62

Figura 4.12. Mapa de activación de la segunda capa convolucional para la emoción Sorpresa. 63

LISTA DE TABLAS

Tabla 3.1. Estructura de la subred CNN. 34

Tabla 3.2. Parámetros de entrenamiento del método 2.....36

Tabla 3.3. Tabla de desempeño por clase en el set de pruebas para el método 1. 37

Tabla 3.4. Tabla de desempeño por clase en el set de validación para el método 1. 37

Tabla 3.5. Estructura de la subred CNN. 39

Tabla 3.6. Parámetros de entrenamiento del método 2..... **¡Error! Marcador no definido.**

Tabla 3.7. Tabla de desempeño por clase en el set de pruebas para el método 2. 41

Tabla 3.8. Tabla de desempeño por clase en el set de validación para el método 2. 42

Tabla 4.1. Resultados de la red en el set de pruebas para 2 emociones. Nivel de similaridad. 48

Tabla 4.2. Resultados de la red en el set de pruebas para 3 emociones. Nivel de similitud.....50

Tabla 4.3. Resultados de la red en el set de pruebas para 4 emociones. Nivel de similitud.....52

Tabla 4.4. Resultados de la red en el set de pruebas para 5 emociones. Nivel de similitud.....54

Tabla 4.5. Resultados de la red en el set de pruebas para 5 emociones. Nivel de similitud.....55

Tabla 4.6. Especificaciones principales de los equipos. 57

Tabla 4.7. Tiempo de procesado de las redes para el set de entrenamiento en distintos equipos. 58

Tabla 4.8. Tiempo de procesado de las redes para el set de entrenamiento en distintos equipos. 59

CAPÍTULO 1.

1. INTRODUCCIÓN

El campo del reconocimiento de emociones humanas ha sido de interés para distintas áreas de investigación como por ejemplo el de medicina o psicología, ya que entrega información sobre el estado de las personas y cómo podrían reaccionar ante distintos escenarios. Debido al avance de tecnología de los últimos años, se han logrado diseñar y crear metodologías que permitan reconocer de manera efectiva, las emociones que presenta un individuo en un instante de tiempo. Entre los modelos de reconocimiento de emociones más destacados se puede encontrar aquellos que hacen uso de imágenes para reconocer las expresiones faciales, las que hacen uso de interfaces cerebro computadora (BCI), el análisis de texto y el reconocimiento por medio de la voz del individuo. Cada uno de los métodos descritos anteriormente tienen sus fortalezas y debilidades: BCI se considera como un molesto por el usuario, ya que requiere la colocación de electrodos en la cabeza de este, sin embargo, su desempeño es difícil de vulnerar y tiene buen rendimiento; el análisis de texto puede fallar mucho si la persona no es consistente en su escritura o contiene faltas ortográficas, pero tiene mucho potencial al usarse en redes sociales; el análisis de la voz puede contener ruido de fondo y depende de la duración de la grabación, pero es un método no intrusivo y no requiere dispositivos especializados para recoger la información; finalmente, el reconocimiento por medio de expresiones faciales depende de la resolución de las imágenes y es sensible a la iluminación, con la ventaja de que no es intrusivo y es el método más cercano al utilizado por los humanos, puesto que se depende en gran medida de la información que obtenida por medio de los ojos. Es este último el que se utilizará en el presente proyecto de tesis.

Así, el objetivo de esta tesis es la elaboración de una red de tipo siamesa que sea capaz de diferenciar entre distintas emociones. Además, se migrará el modelo entrenado hacia un sistema embebido, Nvidia Jetson Xavier, para obtener información sobre su capacidad de realizar

inferencias con redes neuronales previamente entrenadas. De las pruebas realizadas se obtuvieron tres modelos, dos que funcionan sobre cinco emociones y uno que lo hace sobre seis. Con estas redes se llevan a cabo dos tareas, una donde se conoce el nivel de similitud entre un par de imágenes y la otra lleva a cabo la clasificación con base en los niveles de similitud de una imagen con todas las demás.

La obtención del nivel de similitud entre dos imágenes es un proceso común para las redes de tipo siamesa, ya que es la tarea en la que basan su funcionamiento. Sin embargo, para que esta arquitectura de red pueda llevar a cabo una clasificación, se hacen necesarias varias consideraciones. En este caso, se realiza un proceso similar al de la primera tarea, pero esta vez sobre todo el conjunto de imágenes y después se hace uso de las clases con menor distancia a la que se desea clasificar.

Los resultados obtenidos son buenos para cinco emociones, ya que el promedio de precisión fue superior al 90%, pero al aumentar la cantidad, el modelo comienza a presentar fallas y problemas de entrenamiento, lográndose obtener únicamente un modelo de seis emociones con resultados adecuados.

El tiempo que el modelo requiere para la clasificación de emociones es relativamente lento cuando debe obtener los vectores de características de todas las imágenes del conjunto de datos, ya que esto le toma aproximadamente 8 segundos al modelo cuando se procesan 5 emociones. Sin embargo, es posible disminuir el tiempo a menos de un segundo si previamente se obtienen los vectores de características con la red y se almacenan. Esta información será utilizada cuando se quiera clasificar una nueva imagen o encontrar su nivel de similitud ante una clase en específico, permitiendo disminuir el procesamiento a únicamente una imagen. La contribución de este trabajo es la elaboración de redes siamesas para la detección de emociones que sean capaces de funcionar ante sistemas de carácter embebido, los cuales carecen de los recursos con los que sí cuentan los equipos computacionales modernos. Esto abre las puertas para el desarrollo de aplicaciones o dispositivos pequeños que puedan llevar a cabo el reconocimiento de emociones, siendo una opción barata y fiable.

CAPÍTULO 2.

2. ANTECEDENTES DE LA TESIS

Para la realización del proyecto de tesis se llevó a cabo un análisis de los trabajos realizados en los últimos años, así como lectura de los fundamentos teóricos que permitan ampliar el conocimiento sobre el trabajo a desarrollar. Dicha información se encuentra contenida en esta sección.

2.1. Bases teóricas

Para poder realizar un reconocimiento de emociones más acertado, es necesaria la creación de modelos que categoricen el aparente estado de la persona de manera eficaz, siguiendo ciertos parámetros. Los modelos utilizados en la literatura se suelen dividir en dos categorías, uno centrado en una aproximación dimensional y el otro en una categórica [1].

2.1.1. Modelos emocionales

En las aproximaciones dimensionales, las emociones humanas se conceptualizan al mapear las posiciones de las emociones en dos o tres dimensiones. Muchos de los modelos emocionales utilizan la excitación o excitación-relajación, la valencia que indica qué tan placentero o no es y/o la intensidad presente para realizar una categorización de la emoción presente. Uno de los clasificadores más conocidos es el llamado Modelo Circunflejo de Russell, el cual consiste en un arreglo bidimensional de la valencia y excitación para indicar, por medio de un arreglo circular, cuál es la emoción experimentada. Un ejemplo de tal modelo puede verse en la Figura 2.1, donde el eje vertical se corresponde a la excitación y el horizontal, a la valencia. A pesar de que el Modelo Circunflejo de Rusell sea de los más conocidos en cuanto a los modelos dimensionales, existen otros como el modelo de vectores, el PANA, Plutchik, el cubo de emociones de Lovheim, entre otros.

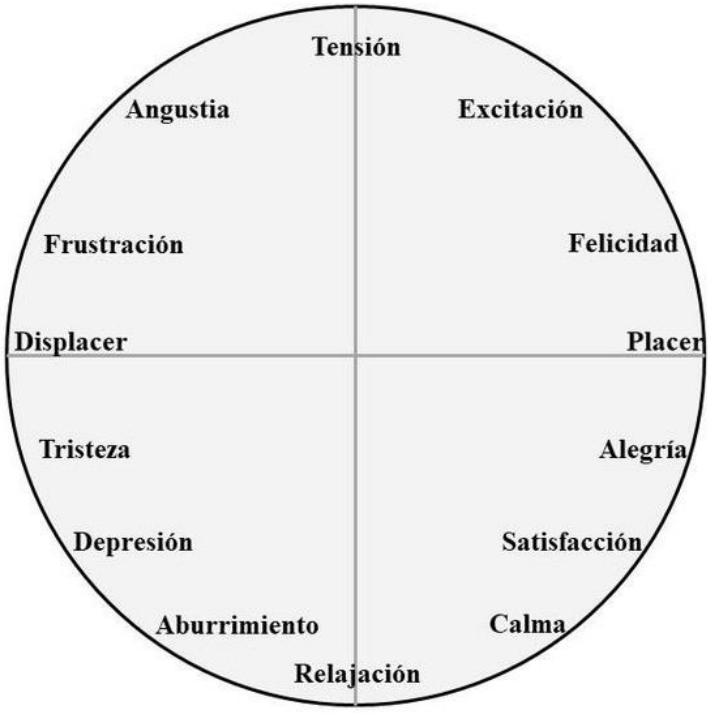

Figura 2.1. Modelo Circunflejo de Russell.

El otro tipo de modelos, los categóricos, están basados en la observación de las emociones humanas y, debido a que la percepción sobre éstas varían de individuo en individuo, se agrupan en categorías de emociones similares. Uno de los modelos más conocidos es la teoría afectiva de Ekman, el cual define la existencia de únicamente 6 emociones básicas: ira, alegría, sorpresa, asco, tristeza y miedo. En cambio, la investigadora Kate Hevner desarrolló su propio clasificador llamado Círculo de los adjetivos, en donde separa ocho emociones principales [1]. En la Figura 2.2 se puede observar el modelo de Hevner y su separación de ocho adjetivos principales, los cuales son los siguientes:

1. Respetable, digno, distinguido, sagrado, serio, sobrio, solemne y espiritual.
2. Oscuro, depresivo, doloroso, frustrante, sombrío, pesado, melancólico, fúnebre, patético, triste y trágico.
3. Soñador, anhelante, lastimero, suplicante, sentimental, afectuoso y ansioso.
4. Calmado, cómodo, lírico, silencioso, que satisface, sereno, suave y tranquilo.

5. Delicado, agradable, gracioso, humorístico, liviano, juguetón, original, vivaz y caprichoso.
6. Brillante, alegre, risueño, feliz, gozoso y festivo.
7. Agitado, dramático, excitante, estimulante, impetuoso, apasionado, incansable, sensacional, enaltecedor y triunfal.
8. Enfático, exultante, majestuoso, marcial, poderoso, robusto y vigoroso.

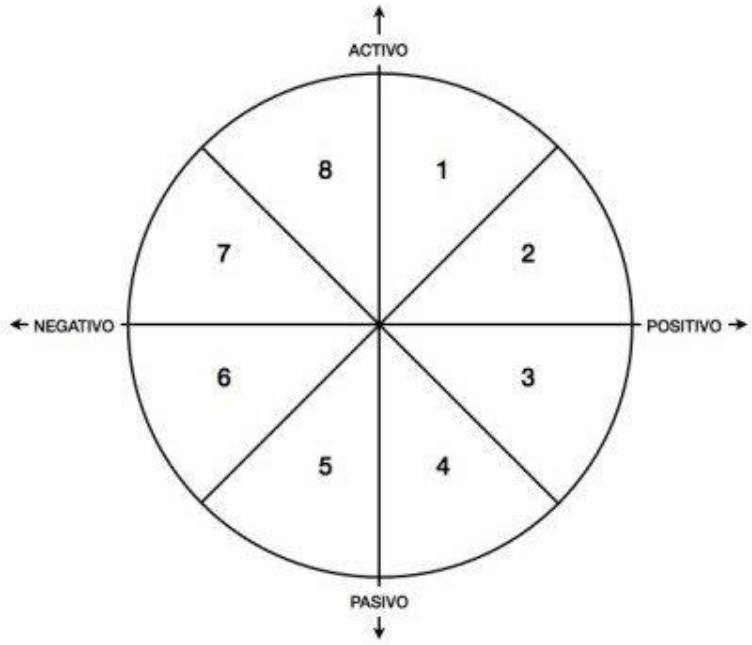

Figura 2.2. Círculo de los adjetivos de Hevner.

2.1.2. Métodos y algoritmos para la de detección de emociones

Entre los artículos encontrados, se puede encontrar diversos algoritmos para la detección de emociones, por ejemplo, en [1] se utiliza una detección multimodal utilizando la transformada de Hilbert Huang para características visuales y de audio, uno utiliza Haar Cascade como método principal para la extracción de características [2]. Otros, en cambio, utilizan técnicas de aprendizaje profundo para llevar a cabo tal tarea como se observa en [3], [4], [5], donde se utiliza una red neuronal convolucional, CNN por sus siglas en inglés. Otras técnicas encontradas en la literatura que optan por modelos que no incluyen redes neuronales son los

expuestos en [6], [7], [8], [9], los cuales logran buenos resultados en el análisis de distintas emociones en imágenes o secuencias de video.

Una característica que comparten los estudios anteriores es que realizan un análisis visual, sin embargo, existen otros artículos donde se utilizan diferentes métodos para llevar a cabo el reconocimiento, tal como se observa en [10], donde se analizan las expresiones faciales, tono de voz, electrocardiograma, pulso sanguíneo, electroencefalograma, respiración, temperatura corporal, electromiografía y actividad electrodérmica. En cambio, en [11], se utiliza un aprendizaje no supervisado con el uso de diccionarios adaptativos, o por sus siglas en inglés, UDADL.

En otra de las fuentes, se encontró un algoritmo basado en características gráficas laplacianas (GLFs) para analizar imágenes con rostros en tres dimensiones de diferentes bases de datos. En unos casos se utiliza únicamente un tipo de red o algoritmo, pero como se vio en la literatura, existen métodos diferentes para la extracción de características, como por ejemplo la utilización de modelos híbridos que combinan una o más redes para mejorar la precisión del reconocimiento. Tal tarea es realizada en [8], que combina un clasificador tipo K vecinos más cercanos (k-NN) con una red neuronal tipo perceptrón multicapa (MLP) para el reconocimiento en imágenes 3D, en [3], donde se utiliza una CNN en conjunto con una BRNN; la imagen de entrada se introduce en la primera red para la extracción de características de alto nivel, luego, en la BRNN, se aprenden los cambios de cada una de las imágenes de entrada. Otro de los artículos propone una arquitectura 3D CNN en conjunto con celdas de memoria de corto y largo plazo (LSTM) para llevar a cabo la tarea del reconocimiento de emociones, así como la combinación CNN y redes neuronales recurrentes (RNN) [4]. En [6] la identificación se lleva a cabo utilizando múltiples fuentes tales como expresiones faciales, poses, movimiento del cuerpo y voz, estas características espacio temporales se introducen en una 3D CNN y redes tipo DBN, donde se modela la información presente en audio y video para realizar un reconocimiento óptimo. La última combinación mencionada, CNN-RNN, es también utilizada en [5].

En el campo del aprendizaje profundo, existen diversas metodologías y algoritmos que sirven para llevar a cabo la tarea de clasificación de datos y su posterior detección. En los últimos tiempos, algoritmos como k-NN o las redes CNN han tenido un gran impulso debido al avance en el campo de la computación, brindando a los equipos de cómputo un mayor poder de procesamiento para la realización de tareas complejas. Es por esta razón que las redes neuronales han visto un incremento en su implementación en varios campos tecnológicos, pudiendo utilizarse en un gran margen de campos diferentes.

La utilización de redes neuronales para la identificación de emociones es ampliamente utilizada en conjunto con otros métodos, por ejemplo, en [2] se apoya de un detector tipo Haar Cascade y un filtro Sobel o en [8], donde se utiliza un modelo MLP en conjunto con un clasificador k-NN para reconocimiento facial en imágenes de tres dimensiones. Siguiendo un patrón de utilización en el estado del arte, los métodos con más aparición fueron los siguientes:

Las CNN, o por sus siglas en inglés *Convolutional Neural Network,* han ganado gran popularidad para aplicaciones de la vida real debido a su rendimiento y robustez superiores a otros métodos. Las CNN aprenden, de manera automática, características de alto nivel de sus imágenes de entrada, con la limitante de demandar una gran base de datos para el entrenamiento y muchos recursos computacionales [12].

El clasificador k-NN o k vecinos más próximos, es uno de los más simples existentes en el campo del *Machine Learning* y los algoritmos de clasificación de imágenes. Su premisa, a diferencia de otros algoritmos, no es la de aprender, sino que depende de las distancias entre los vectores de características. Una manera de describir su funcionamiento es bajo la premisa de: “Dime quiénes son tus vecinos y te diré quién eres.”, ya que el algoritmo k-NN clasifica datos desconocidos basándose en las clases que lo rodean, la que tenga mayor índice de aparición será la que determine su grado de pertenencia [13].

2.1.3. Internet de las cosas orientado a emociones

El paradigma de Internet de las cosas, o IoT por sus siglas en inglés, se ha visto en alza con la introducción de los sistemas embebidos en la vida cotidiana debido a su reducido tamaño y

la posibilidad de conectarse a Internet. Es con esto último que se crea una brecha entre los dispositivos embebidos, puesto que la capacidad de conectarse a la red les confiere una serie de ventajas que de otra manera no tendrían. Dependiendo de su necesidad de utilizar la red será el paradigma al cual pertenece. Si el dispositivo únicamente puede recabar datos y realizar procesamientos débiles o sencillos, obedecerá a la computación en el borde o *Edge Computing*. De tener capacidad de hacer procesamientos pesados y que la dependencia hacia la red no sea tan indispensable, obedecerá al Internet de las cosas.

El que se separen los dispositivos que puedan conectarse a Internet no implica una separación total, puesto que ambos sistemas pueden trabajar en conjunto para ofrecer una mayor cantidad de servicios y comodidades al usuario. Temas como la industria 4.0 y las casas inteligentes son ejemplos de cómo se utilizan en la actualidad los dispositivos IoT y *Edge Devices.*

En la actualidad, las redes sociales tienen un fuerte impacto en cuanto emociones se refiere, compañías como Facebook, Twitter, Google, entre otras, buscan entregar emociones placenteras al usuario para mantenerlos el mayor tiempo posible dentro de sus dominios. Otro uso que le dan es la entrega de publicidad a la cual el usuario pueda ser más susceptible a recibir de manera positiva. Es importante tomar en cuenta que las redes sociales han incrementado su número de usuarios con el tiempo, así como la manera en que atentan contra la privacidad del usuario. Sin embargo, es por eso último que se han convertido en una herramienta valiosa para el análisis de hábitos, tendencias y pensamientos de los distintos sectores sociales. Con la creciente aparición de inteligencias artificiales y algoritmos de *machine learning*, el análisis de todos esos datos se convierte en un trabajo automatizado y simple, pudiendo analizar una gran cantidad de información en cuestión de minutos [14]. Actualmente el objetivo es el de identificar la mayor cantidad de información necesaria de la manera más eficiente posible. Gracias a esto es que la detección automática de emociones utilizando redes sociales ha ganado tanta atención por parte de diversas compañías tecnológicas.

Como se vio en el párrafo anterior, las redes sociales forman uno de los pilares en la investigación para la identificación de emociones y el tener una cantidad de millones de

usuarios activos, les brindan una gran cantidad de información que puede ser utilizada para tales fines. Es importante reconocer el papel que han jugado los dispositivos en el borde, los cuales forman parte del paradigma de Internet de las cosas, puesto que fueron este tipo de aparatos los que facilitaron una conexión entre el usuario y la aplicación desde cualquier lugar. Asimismo, el dar opciones para compartir texto, imágenes, fotos personales, entre otras, no hace más que aumentar las bases de datos de las compañías e incrementar la importancia que se le da a los sistemas embebidos.

Con base en lo recabado en la recolección de la revisión del estado del arte, se encontró un campo de oportunidades en el área de la detección de emociones en los sistemas embebidos. Existen muchos enfoques para solventar el problema que conlleva la identificación, sin embargo, la gran mayoría se realiza en computadoras convencionales.

La utilización de los sistemas embebidos puede ser la respuesta para obtener mejores mediciones ya que estos dispositivos cuentan con gran movilidad y adaptabilidad. Además, la integración entre varios dispositivos de este tipo que se dediquen a monitorear distintos parámetros, podría reforzar los algoritmos actuales al añadir más clases que definan las distintas emociones que el humano experimenta a lo largo de su vida.

2.2. Bases prácticas

Para llevar a cabo una buena detección de emociones es necesaria la utilización de una base de datos que sirva como modelo base para el entrenamiento y pruebas de un algoritmo. Un punto importante para tener a cuenta es que, como humanos, nuestras emociones se encuentran en un estado de cambio constante y fluido, nunca se está 100% felices o 100% tristes. En su lugar, las emociones se encuentran mezcladas. Por ejemplo, al sentir sorpresa se puede experimentar alegría, como en una fiesta sorpresa, o miedo si viene de una fuente poco confiable o agradable. Es importante no encasillarse en una sola clase, como suele hacerse en otro tipo de clasificación. En su lugar, es más ventajoso observar la probabilidad de aparición de cada emoción y caracterizar su distribución [15].

2.2.1. Bases de datos para entrenamiento

Una de las bases de datos más utilizadas para el reconocimiento de emociones es la desarrollada para el reto de reconocimiento facial de Kaggle, llamado *FER13* [7] [15], el cual es una base de datos y entrenamiento de 28,709 imágenes, cada una compuesta por imágenes a

escala de grises de 48x48 pixeles. Cada uno de los rostros presentes ha sido ajustado para que tenga un tamaño similar al de los demás. Un ejemplo de las imágenes presentes en la base de datos es la mostrada en la Figura 2.3.

Figura 2.3. Extracción de imágenes presentes en la base de datos de Kaggle [15].

Otra de las bases de datos más comúnmente utilizadas es la desarrollada por la universidad Carnegie Mellon [18], conocida como *CK database*. Presenta 2,106 imágenes digitalizadas de 182 adultos de los cuales el 65% son mujeres. Además, entre los sujetos el 15% son afroamericanos y un 3% de latinos y asiáticos. Existe una versión ampliada, conocida como *CK+*, presenta un incremento del 22% en las imágenes y un 27% más sujetos.

2.2.2. Extracción de características

Muchos investigadores han buscado mejorar las formas en que los dispositivos detectan y tratan de entender los diferentes estados emocionales, aún más, teniendo presente una brecha de comunicación entre las personas y los dispositivos electrónicos. La dificultad de que un aparato sea capaz de reconocer, expresar y sentir emociones, pone un tope muy grande en su

habilidad interactiva con el humano. Sin embargo, es muy complicado que un dispositivo sea capaz de comprender las emociones de una persona limitándose a ciertos canales, por ejemplo, únicamente con la voz o el texto, puesto que carece de información que puede ser de ayuda para llevar a cabo una detección más acertada. Es por eso que actualmente se tienen dos métodos distintos para la extracción de las características. Mientras unos investigadores se centran únicamente en una fuente a analizar, por ejemplo, imágenes, videos, señales EEG [1], [2], [5]-[8], [11], [16], otros trabajos optan por características multimodales, las cuales se reciben por distintos medios tales como: audio, señales EEG, pulsos por minuto, etc. para obtener una medición más precisa [3], [6], [9], [10], [14], [17]. Las maneras más comunes para el análisis de datos con el fin de detectar emociones son listadas en [14][19]. Las más relevantes encontradas son las siguientes:

- **Emoción en texto.** Consiste en la identificación de la intensidad de sentimientos y opiniones en fragmentos de texto. Busca determinar si una oración o un documento expresa sentimientos positivos, negativos o neutrales dirigidos hacia un objeto o sujeto.
- **Análisis de texto.** Es un estudio computacional sobre cómo las opiniones, actitudes, perspectivas y emociones son expresados en el lenguaje. Es una tarea compleja puesto que el contexto puede cambiar el significado de ciertas palabras, como es el caso del sarcasmo o una mala estructuración de las oraciones. Esto acarrea muchas dificultades tanto para personas como para computadoras al determinar los sentimientos detrás de lo escrito.
- **Emoción en el sonido.** Dependiendo del entorno, esta tarea se vuelve más complicada gracias a la cantidad de ruido de fondo que podría aparecer, opacando la señal deseada. Otra cosa para tener en cuenta es que los algoritmos encargados de procesar y determinar las emociones en el audio están orientados hacia las conversaciones o palabras habladas. Es por esto último que ruidos no verbales, tales como risa, sollozos, suspiros, lamentos, no son útiles en la mayoría de los casos, aumentando el nivel de complejidad de los datos a analizar debido a que se interpretan como ruido. A pesar de lo anterior, la detección de emociones con base en el habla ha tenido un avance

significativo con la inclusión de dispositivos que reciben órdenes por comandos de voz, tales como los asistentes virtuales de Google Home o Amazon Alexa.

- **Emoción en imágenes y videos.** A pesar de que en este tipo de análisis se carezca de datos relativos a la comunicación, tal como el habla, es posible determinar una emoción por la observación de las expresiones faciales y los gestos. Resultado de esos análisis es la creación de modelos emocionales, descritos en la sección 2.1.1, que permiten clasificar y determinar emociones con base en imágenes o secuencias de video.

En [18] se llevó a cabo una investigación para comparar distintos descriptores de características y cuál obtiene una mejor caracterización de las expresiones faciales para realizar una clasificación de emociones utilizando *machine learning*. Estos eran: *Key Facial Landmark Detection* (KFL), *Saliency Mapping* (SAL), *Local Binary Pattern* (LBP) e *Histogram of Oriented Gradients* (HOG). Se realizaron distintas combinaciones entre estos métodos y en algunos casos se aplicó PCA para disminuir la redundancia en los datos. Por medio de las concatenaciones se obtuvieron 12 descriptores de características, los cuales fueron usados para entrenar 6 algoritmos de *machine learning* distintos. En la Figura 2.4 se muestran las mejores combinaciones entre descriptores de características y clasificadores para las distintas emociones a detectar, cortesía de [18].

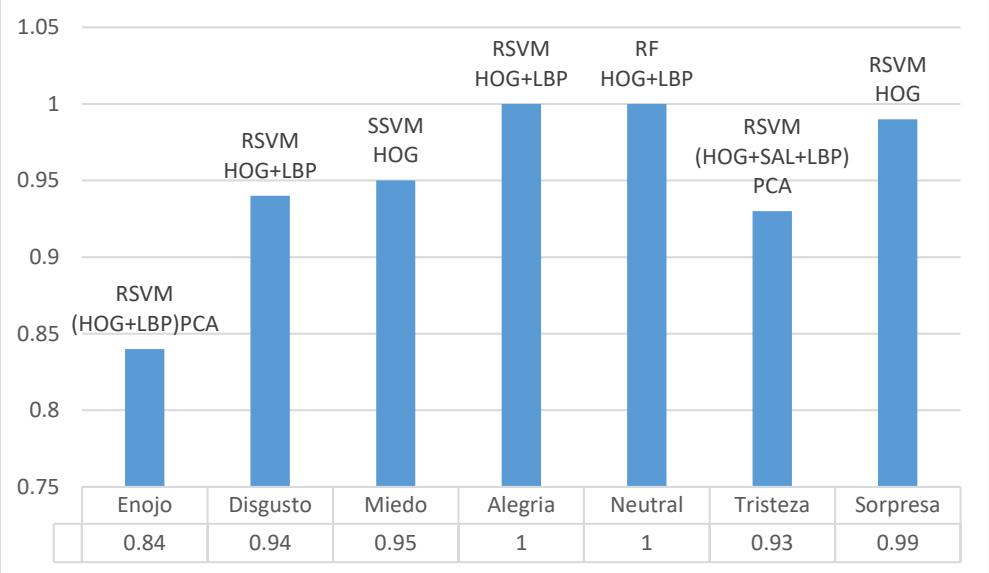

Figura 2.4 Modelos con mejor clasificación por emoción en [18].

Como puede observarse, la combinación de HOG y LBP con un clasificador con kernel de función base radial, *RBF Kernel* por sus siglas en inglés, combinado con máquinas de vectores de soporte (RSVM) obtuvo los mejores resultados de manera general, logrando la mejor clasificación en dos de las 7 emociones a clasificar. Además, la caracterización lograda con HOG y LBP se encuentra como la mejor en 4 de las 7 emociones medidas, siendo en un caso necesaria la aplicación de PCA.

2.2.2.1 Histograma de gradientes orientados

Se enfoca en la estructura o la figura de los objetos en una imagen. A diferencia de los métodos para detección de bordes, HOG es capaz de indicar la dirección del borde. Esto lo realiza extrayendo el gradiente y la orientación de la imagen. Estos se calculan de manera localizada, es decir, la imagen se analizará en porciones o regiones. Con este algoritmo se obtiene el histograma de cada región, el cual se crea con relación al gradiente y orientación de los valores de los pixeles, de ahí su nombre. Generalmente se realiza un preprocesamiento de las imágenes y se escalan a un tamaño de 64x128 pixeles.

Para obtener los descriptores de HOG es necesario calcular los gradientes verticales y horizontales. Para esto se puede utilizar el kernel de Sobel, descrito en la ecuación 2.1a y luego de eso, se procede a calcular la magnitud y dirección del gradiente con las fórmulas (2.1) y (2.2).

$$Mag = \sqrt{g_x^2 + g_y^2} \tag{2.1}$$

$$\theta = \arctan\left(\frac{g_y}{g_x}\right) \tag{2.2}$$

Donde g_x y g_y son los gradientes horizontales y verticales de la imagen I, en este caso se utiliza un operador Sobel de kernel 3x3 para calcularlos.

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I$$
$$g_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} * I \qquad (2.1a)$$

Aplicando dichas ecuaciones a la imagen se logra obtener información de los gradientes presentes en la imagen.

Obtenidos los valores de los gradientes se divide la imagen original en celdas de un cierto tamaño, generalmente 8x8, y se calcula el histograma de cada grupo. Como los gradientes de la imagen suelen ser sensibles a los cambios de iluminación, es necesaria una normalización que vuelva robusto al algoritmo ante estas situaciones. Hecho esto bastará con concatenar el vector resultante de la normalización y así crear un vector más grande. Este vector contendrá las características de la imagen y su tamaño dependerá del tamaño en el que se dividirán la imagen y la ventana de normalización.

2.2.2.2 Patrones binarios locales

Es una técnica descriptiva para la clasificación de objetos en imágenes. Este método etiqueta los pixeles de la imagen al umbralizar el vecindario de un píxel y considerando el resultado como un numero binario. Debido a su simpleza computacional, se ha convertido en un método utilizado para la extracción de características ya que permite un análisis en tiempo real. El nombre deriva de su actuación sobre el vecindario de pixeles y a que el algoritmo arrojará el mismo resultado aun cuando se hayan realizado rotaciones sobre la imagen. [20]

Lo que hace este método es codificar la relación existente entre el pixel central con sus vecinos, todo en términos de sus intensidades de color. Además, el algoritmo es robusto ante variaciones de iluminación. El algoritmo generalmente trabaja sobre imágenes a escala de grises, pero es posible aplicarlo cuando hay color al trabajar sobre cada canal.

Los patrones locales binarios se obtienen moviendo un kernel de 3x3 sobre la imagen, el pixel central del bloque se utiliza como un valor de umbral contra el que se compararán los pixeles del vecindario. Posteriormente se realiza una sumatoria donde se codifican los valores resultantes. La expresión matemática que describe el comportamiento de *LBP* es el siguiente:

$$LBP = \sum_{p=0}^{P-1} s\left(i_p - i_c\right) * 2^p \tag{2.3}$$

$$s(i) = \begin{cases} 1 & i \geq 0 \\ 0 & i < 0 \end{cases} \tag{2.4}$$

Donde P representa el número de vecinos, i_p es el valor de intensidad del pixel vecino, i_c es la intensidad del pixel central, 2^p es el valor codificado de los pixeles que pasen el umbral dado por *s(i)*, donde *i* representa el valor resultante de la resta entre el pixel central y su vecino *p*.

2.2.3. Clasificador de características

La etapa de extracción de características permite la obtención de datos contenidos en las imágenes que se procesarán, logrando obtener información que permita distinguir entre las distintas clases a clasificar. Para completar este proceso se hace necesaria la utilización de un clasificador que será el encargado de analizar estos datos y agruparlos en sus respectivas categorías con base a lo aprendido en una etapa de entrenamiento previo.

Existen diversos métodos que permiten la creación de un clasificador, pero los derivados del *machine learning* destacan sobre los demás. Dentro de esta categoría, los basados en redes neuronales y aprendizaje profundo presentan los mejores resultados. Sin embargo, un buen desempeño viene con la complicación de que se hacen necesarios grandes conjuntos de entrenamiento para calibrar el sistema y a que el tiempo de ejecución se ve fuertemente afectado por el tipo de CPU/GPU utilizado [21], además, el tamaño en memoria que este tipo de sistemas requieren incrementa con el número de neuronas presentes en la red completa, razón por la cual no suelen utilizarse en sistemas embebidos, los cuales se encuentran limitados en este aspecto [22], [23], [24]. A pesar de esto, los avances tecnológicos de los últimos años han permitido

incrementar las capacidades de estos dispositivos, que junto a la creación de redes neuronales de bajo consumo permitirían la creación y adaptación de tecnologías que antes no podían llevarse a cabo en los sistemas embebidos.

2.2.4. Sistemas embebidos en la detección de emociones

Como se comentó en la sección anterior, existen diversos sensores para realizar una clasificación de las emociones. Muchos optan por la combinación de diferentes sensores para realizar dicha tarea, con el fundamento de que el humano requiere recibir varios estímulos para detectar una emoción. Entre las expresiones comunes para las personas se presentan las audiovisuales: expresiones faciales, tono de voz, la postura y los gestos; y la psicológica: respiración, rubor, temperatura de la piel, etc. Las computadoras pueden utilizarse para tomar lectura de alguno o varios de los parámetros anteriormente descritos. Estos serán usados para obtener ciertas características que, al juntarse, pueden utilizarse para inferir, con buena precisión, un estado emocional. Así, va en aumento la cantidad de sensores que pueden obtener señales humanas para luego procesarse y poder estimar cuál emoción se presenta. Entre estos sensores utilizables para la obtención de información destacan los siguientes: grabaciones de expresiones faciales, cambios tonales de voz, EEG, sensado de tensión muscular, ECG, temperatura corporal, entre otros [10].

A pesar de existir computadoras con capacidades para obtener todos esos conjuntos de información y procesarlos paralelamente, su tamaño, consumo eléctrico y peso, dificultan su transporte y se mantienen como dispositivos fijos, por lo que se requiere de dispositivos más modestos, pero con capacidad de realizar la tarea de la detección de emociones en tiempo real. Es aquí donde los sistemas embebidos encuentran campo de acción, ya que compensan las limitaciones de una computadora convencional, con la desventaja de que sacrifican la potencia computacional, por lo que se ven limitados en cuanto a la implementación de algoritmos muy complejos.

Los sistemas embebidos han visto un auge muy grande en el campo de la detección de emociones, aproximaciones a un campo práctico es en la conducción inteligente. Aprovechando el pequeño trabajo de los sistemas embebidos, es posible su colocación a bordo de vehículos

para obtener mediciones del usuario y detectar emociones tales como la confusión o fatiga, incrementando la seguridad en el camino. Un proyecto creado en el Instituto Tecnológico de Massachusetts, llamado *AutoEmotive*, lleva a cabo tal tarea, siendo éste un prototipo equipado con sensores y una cámara. De esta manera, el vehículo puede medir el nivel de estrés y fatiga del conductor. Cuando llegue a cierto nivel, se reproducirá música relajante, cambiará la temperatura y las luces interiores del automóvil o sugerirá al conductor rutas que evoquen menos estrés [14].

Otros campos que pueden verse afectados de manera positiva por la implementación de sistemas embebidos capaces de detectar emociones es el de la psicología, donde puede ayudar a la identificación de las emociones de pacientes con dificultades para expresar sus sentimientos, entre ellos personas con autismo y personas con síndrome de enclaustramiento. Otro de los campos que se pueden beneficiar de la inclusión de estos dispositivos es el sector de la salud, ya que se ha vuelto cada vez más dependiente de dispositivos y aplicaciones médicas. Una posible aplicación de la detección de emociones en pacientes es la interacción con un aparato que pueda leer y moldearse de acuerdo con el humor del usuario, para de este modo, encaminarlo a un estado de motivación. En caso de rehabilitación, esto es crucial puesto que podría ayudar a una recuperación más rápida y mejorar la calidad de vida del paciente [17].

2.2.5. Aplicaciones de IoT

Con el impulso que ha tenido el paradigma del Internet de las cosas y los dispositivos en el borde, ha sido posible la implementación de diversos algoritmos que permitan una ejecución en la nube, ahorrando costos computacionales e incrementando su despliegue en distintos campos. Es por eso que se han desarrollado diversas herramientas orientadas hacia la detección de emociones utilizando diversas entradas, ya sea imágenes, videos, audios, etc. En [14], se listan algunos de los softwares, APIs y herramientas de entrenamiento ya existentes que llevan a cabo esta tarea, a continuación, se mostrarán algunos.

***IntraFace*.** Es un paquete de software orientado hacia la publicidad. Ofrece tareas como seguimiento automatizado de rostros, estimación de la pose de la cabeza, reconocimiento de atributos faciales, análisis de las expresiones faciales en secuencias de videos, entre otras. Tiene

capacidad de medir la reacción de personas ante una charla o la sincronía de emociones entre diversas personas.

The Emotion API. Desarrollada por Microsoft, es una herramienta dedicada al reconocimiento de emociones en imágenes y videos. Como resultado del análisis de la entrada con la cual se alimente, imagen o video, regresará la probabilidad de presencia de ocho emociones distintas: enojo, disgusto, miedo, alegría, neutral, tristeza, sorpresa o desprecio. Tiene la capacidad de seguir la respuesta de una persona o un grupo de individuos que reaccionan ante un contenido presentado. Puede utilizarse en conjunto con los lenguajes de programación C#, Java, JavaScript, PHP, Python y Ruby.

ASC-Inclusion. Es un proyecto que apunta hacia la creación de una plataforma cibernética para la asistencia de niños con la condición del espectro autista para mejorar sus habilidades de comunicación socioemocionales. Se basa en el entendimiento y expresiones emocionales con base en las expresiones faciales, vocales y gestos corporales de este grupo de personas.

En este capítulo se llevó a cabo una investigación sobre el estado del arte actual del reconocimiento de emociones utilizando técnicas de aprendizaje profundo. A lo largo del capítulo se vieron temas teóricos y prácticos. En la primera sección se encontraron como se dividían las emociones desde puntos de vista psicológicos al introducir modelos categóricos que buscan conceptualizar las emociones en dos o más dimensiones, como el modelo circunflejo de Russel o el de Ekman. Otro de los conocimientos adquiridos fue el tipo de clasificación en las cuales ciertos modelos etiquetan las emociones, ya que en muchos trabajos se encontró que se suele trabajar con las emociones “base” o principales, como alegría, enojo, tristeza, miedo, sorpresa y desagrado. Modelos más robustos pueden incluir subcategorías más específicas como festivo, vivaz, tranquilo, etc.

En cuanto a las bases prácticas se encontraron aplicaciones, librerías, frameworks y APIs enfocados al reconocimiento de emociones en distintas plataformas y con distintos enfoques, por ejemplo, Emotion API de Microsoft o ASC-Inclusion.

CAPÍTULO 3.

3. METODOLOGÍAS PARA LA DETECCIÓN DE EMOCIONES

En esta sección se muestran algunas metodologías encontradas en la literatura referentes al reconocimiento de emociones en imágenes de expresiones faciales, además de un nuevo método que hace uso de una red de arquitectura siamesa, la cual contiene como clasificador una red neuronal convolucional. Además, se mostrarán diversos elementos que comparten las distintas técnicas aquí descritas, las bases de datos que fueron utilizadas y los resultados obtenidos.

3.1. Bases de datos

Para el entrenamiento y evaluación de la metodología empleada, así como en varias de las encontradas en el estado del arte, se utilizó la base de datos *Extended Cohn-Kanade Dataset* (CK+) [25], [26], *FER-2013* [27] y KDEF [28].

3.1.1. Extended Cohn-Kanade Dataset

La base de datos de *Cohn-Kanade* (CK) fue introducida en el año 2000 con el propósito de promover la investigación en el campo de la detección de expresiones faciales en secuencias de video. Desde su creación, esta base de datos se ha convertido en un referente para el desarrollo y evaluación de algoritmos enfocados al análisis de imágenes y videos que presenten expresiones faciales.

Una actualización de la base CK fue liberada en el año 2010 por Patrick Lucey [26] con el fin de incrementar la cantidad de muestras y el número de sujetos de prueba, así como una revisión y validación de la información contenida en la base de datos. Dicha actualización fue nombrada *Extended Cohn-Kanade Dataset* (CK+). Para el trabajo de tesis se utilizó una versión reducida bautizada como CK+, limitada a 7 emociones. Esta reducción se consiguió al extraer los últimos 3 cuadros de cada secuencia de video de la base de datos original.

Las características principales de la base de datos son las siguientes:

- 123 sujetos de prueba de edades comprendidas entre 18 a 50 años, de los cuales:
    - 69% son mujeres.
    - 81% Euroamericanos.
    - 13% Afroamericanos.
    - 6% De diversas etnias.
- 981 imágenes de expresiones faciales distribuidas de la siguiente manera:
    - Enojo: 135 imágenes.
    - Sorpresa: 249 imágenes.
    - Tristeza: 84 imágenes.
    - Desagrado: 54 imágenes.
    - Alegría: 207 imágenes.
    - Miedo: 75 imágenes.
    - Disgusto: 177 imágenes.
- Todas las imágenes son a escala de grises y con dimensiones de 48x48 píxeles.

Los participantes fueron guiados por un instructor para llevar a cabo una serie de 23 expresiones faciales. En todos los experimentos se iniciaba y terminaba en una expresión neutra. Una extracción de imágenes de la base de datos se muestra en la Figura 3.1.

Figura 3.1 Muestra de la base de datos CK+42.

3.1.2. FER-2013

La base de datos *Facial Emotion Recognition Challenge*, comúnmente abreviado como *FER-2013,* fue creada por Pierre-Luc Carrier y Aaron Courville como parte de un proyecto de investigación [27]. En este trabajo de tesis se utiliza una versión preparada por Goodfellow et al. para que concordara con 7 emociones: enojo, disgusto, miedo, alegría, tristeza, sorpresa y neutral. Esta base de datos acotada fue añadida al sitio web https://www.Kaggle.com en el 2013 como un concurso para la elaboración de un método que lleve a cabo un reconocimiento de emociones utilizando expresiones faciales. Los resultados fueron publicados en [27].

- Fue creada utilizando el API de Google *image Search* para encontrar las imágenes de rostros que concuerden con 184 palabras relacionadas con emociones.
- Cada imagen fue verificada y editada, en caso de ser necesario, por examinadores humanos
- 35,887 imágenes de expresiones faciales distribuidas de la siguiente manera:
    - Enojo: 4953 imágenes.
    - Sorpresa: 4002 imágenes.
    - Tristeza: 6077 imágenes.
    - Alegría: 8989 imágenes.
    - Miedo: 5121 imágenes.
    - Disgusto: 547 imágenes.
    - Neutral: 6198 imágenes.
- Todas las imágenes son a escala de grises y con dimensiones de 48x48 pixeles.
- Toda la base de datos se concentra en un archivo con formato *Comma Separated Values,* o por sus siglas en inglés, CSV.

Una extracción de imágenes de la base de datos se muestra en la Figura 3.2.

3.1.3. KDEF

La base de datos KDEF, del inglés *Karolinska Directed Emotional Faces*, es un conjunto de 4,900 imágenes de expresiones faciales de 70 distintos individuos. Fue creada por el

departamento de Neurociencia clínica, sección de Psicología del Instituto Karolinska, en Suecia [28].

Para la creación de esta base de datos, se tomaron fotos de 70 individuos, 35 hombres y 35 mujeres, expresando 7 emociones diferentes vistas desde 5 ángulos diferentes. Los sujetos de prueba seleccionados comprendían edades entre 20 y 30 años, sin accesorios en el rostro, barba o bigote. Todos los individuos vestían una camisa gris y se situaron a 3 metros de la cámara.

Figura 3.2 Muestra de imágenes de la base de datos FER-2013

Las imágenes se etiquetaron debido a las instrucciones dadas a los sujetos de prueba, puesto que se les solicitaba expresar cierta emoción al momento de tomar la foto, siendo quien tomaba la foto quien comprobara que no hubiera error de interpretación al verificar que la emoción que se estaba gesticulando correspondiera con la que se solicitaba. Un ejemplo de las imágenes que contiene la base de datos se encuentra en la Figura 3.3

Las emociones presentes en la base de datos son las siguientes:

- Miedo *(afraid*)
- Enojo *(angry)*
- Disgusto *(disgusted)*
- Alegre *(happy)*

- Neutral *(neutral)*
- Triste *(sad)*
- Sorprendido *(surprised)*

Figura 3.3. Expresiones faciales de un sujeto de la base de datos KDEF para 5 emociones.

3.2. Preprocesamiento

Como se vio en la sección 3.1, las bases de datos contienen imágenes con resoluciones variables, algunas de 48x48 pixeles a escala de grises, como lo son CK+ y FER-2013 o de 562x762 píxeles a color. Debido a que las redes neuronales requieren que los datos de entrada

compartan la misma estructura, se realiza una normalización de las imágenes, asegurándonos que el espacio de color sea a escala de grises y contengan la misma resolución. Esto último se logra al redimensionar las imágenes utilizando técnicas como la interpolación bicúbica.

El proceso utilizado para normalizar las imágenes previo su entrada a la red se muestra en la Figura 3.5. Una explicación sobre cada etapa se encuentra en las siguientes secciones.

Figura 3.4. Esquema general del preprocesamiento de las imágenes.

3.2.1. Color a escala de grises

Se define a I como una imagen en el espacio RGB de $(m,n \in \mathbb{R})$ dimensiones la cual se convertirá a escala de grises al realizar una suma ponderada de los canales *R*, *G* y *B*. De manera matricial, la operación se define como:

$$I_{RGB}^{mxn} \rightarrow I_{Gray}^{mxn} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \cdot \begin{bmatrix} 0.2989 & 0.5870 & 0.1140 \end{bmatrix} \tag{3.1}$$

Donde I_{gray}^{mxn} es la imagen en el espacio de escala de grises y es multiplicada por un vector de pesos para las capas RGB, respectivamente.

3.2.2. Escalamiento de resolución por medio de interpolación bicúbica

Sea I_{gray}^{mxn} Una imagen a escala de grises de dimensiones mxn la cual se desea modificarse a I^{48x48} se lleva a cabo una interpolación bicúbica como función de mapeo.

La interpolación bicúbica calcula el valor promedio ponderado de los pixeles que se encuentren en el vecindario 4x4 más cercano, es decir, los 16 pixeles vecinos más cercanos del

punto central. La manera en la que se calcula el valor de intensidad del pixel en la ubicación *(x, y)* se utiliza la ecuación (3.2):

$$v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j \tag{3.2}$$

Donde *v* es el valor de intensidad del pixel *(x,y)*, x^i y y^j corresponden a los 16 vecinos más cercanos a *v(x,y)* y a_{ij} son los coeficientes de multiplicación del pixel.

3.3. Red siamesa

La red siamesa (SiNN), introducida originalmente en [29] y más formalmente en [30], es una arquitectura de red neuronal comúnmente utilizada con algoritmos de *few-shot* y *one-shot learning*, es decir, cuando la base de datos carece de elementos suficientes para realizar un entrenamiento óptimo.

Una red siamesa contiene dos o más instancias de una subred, es decir, los pesos y parámetros son los mismos en todas ellas. Este tipo de arreglo permite al sistema medir la distancia entre clases y utilizar esta información para aprender si las entradas dadas a cada subred son similares o diferentes [31], es decir, su objetivo es encontrar una función de similitud entre clases, contrario a las redes tradicionales cuyo aprendizaje se basa en predecir entre clases. En el caso que las entradas sean imágenes, es posible utilizar CNNs para la etapa de extracción de características. Una manera gráfica de representar la arquitectura de una red siamesa se muestra en la Figura 3.3. En los siguientes párrafos se dará una explicación más detallada sobre la arquitectura de las redes siamesas.

En [32] y en [33] se explica que la arquitectura de red siamesa realiza un mapeo no lineal ϕ de su dominio de entrada *X*, dado por una imagen *I*, a un espacio euclídeo $\mathbb{R}$, es decir, extrae características del espacio *X* y lo convierte a otro espacio real, de manera matemática:

$$\phi : X \rightarrow \mathbb{R} \tag{3.3}$$

Este mapeo se logra gracias a la etapa de extracción de características realizada por las subredes y el aprendizaje de parámetros discriminativos. Dichas subredes serán definidas como

N_k donde k es el número de instancias de red existentes y todas compartirán los parámetros de los pesos W. A cada subred se le asigna una entrada X_k y producirá un vector de características $f_w(X_k)$. En este trabajo de tesis la incrustación no lineal ϕ varía con respecto el método de extracción de características. Más adelante se detallarán los métodos utilizados.

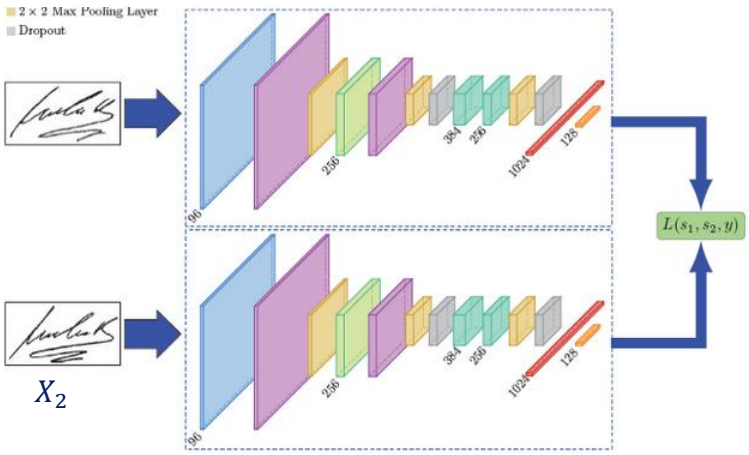

Figura 3.5. Ejemplo de una red siamesa para verificación de firmas [33].

Para obtener el nivel de similitud entre las clases, se utiliza la función de distancia D_W, es decir, las entradas X que pertenezcan a la misma clase tendrán un valor bajo, mientras que las que sean de clases diferentes tendrán una alta separación. La manera en la que se obtiene dicho valor de proximidad es utilizando la distancia Euclidiana como nuestra función de energía, definida por la ecuación (3.4):

$$D_W(X_1, X_2) = \| f_W(X_1) - f_W(X_2) \| \tag{3.4}$$

El resultado obtenido es luego utilizado por el método de clasificación seleccionado para predecir si las imágenes comparadas son similares o diferentes.

3.3.1 Funciones de pérdida

El entrenamiento de las redes siamesas es distinto al de arquitecturas de redes tradicionales debido a que se requieren los vectores de características de las subredes y la comparación entre ellas con la función de distancia D_W. Debido a esto, la función de pérdida se encuentra limitada y generalmente se utilizan tres modelos para llevar a cabo el entrenamiento: pérdida contrastiva,

por triplete y por cuatrillo. En este documento se detallarán estas funciones de pérdida en las siguientes subsecciones.

3.3.1.1 Pérdida contrastiva.

Se refiere al método que tradicionalmente se utiliza para entrenar las SiNN [34], y se le conoce comúnmente como *Contrastive Loss*. La idea principal consiste en la construcción de un espacio euclídeo donde los pares similares, llamados *pares positivos*, que corresponden a objetos o imágenes bajo la misma etiqueta, son más cercanos los unos de los otros, mientras que los pares distintos, llamados *pares negativos*, correspondientes a objetos de clases diferentes se encuentran alejados.

Este tipo de entrenamiento no requiere mucha supervisión debido a que únicamente se necesita la información de los pares, es decir, si son positivos o negativos. Otro detalle por comentar es que los modelos entrenados bajo la pérdida contrastiva son considerados como soluciones holísticas, es decir, se enfoca en distancias absolutas donde las distancias relativas tienen mayor importancia en ciertos casos [35].

La expresión que define la función de pérdida *Contrastive Loss* se muestra en la ecuación (3.5):

$$L(X_1, X_2, y) = (1-y)\frac{1}{2}D_W^2 + y\frac{1}{2}\max(0, m - D_W)^2 \tag{3.5}$$

Donde X_1 y X_2 son las entradas al sistema, y es una etiqueta binaria que indica si pertenecen a la misma clase o no, 0 y 1 respectivamente, m es un margen que anula la pérdida de dos clases distintas que presenten una distancia muy grande y D_W corresponde a las distancias euclidianas entre los vectores de características y se encuentra definido en la ecuación (3.4) [36].

De manera gráfica, la función de pérdida *Contrastive Loss* opera como se muestra en la Figura 3.7, donde se observa que clases similares son atraídas entre sí mientras que los vectores de distinta clase se alejan.

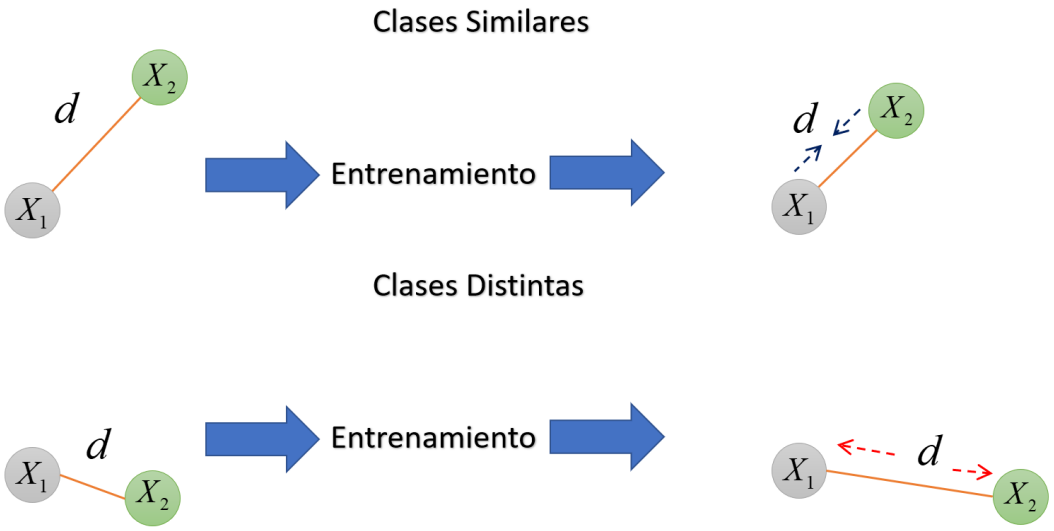

Figura 3.6. Funcionamiento de Contrastive Loss.

3.3.1.2Pérdida de triplete.

La función de pérdida por triplete, *Triplet Loss* por sus siglas en inglés, hace uso de las distancias relativas entre las clases y las métricas de información local. Este método utiliza ejemplos vecinos a una matriz de entrada ancla, X_a, llamado punto de anclaje, es decir, compara la matríz de entrada X_a con un ejemplo de la misma clase y otro de clase diferente [30].

La técnica *Triplet Loss* crea un triplete con la forma (X_a, X_+, X_-), donde X_+ es una matriz vecina con la misma etiqueta que la de X_a, mientras que X_- es otra matriz con una clase diferente a X_a y por tanto, a X_+ [30]. En pocas palabras, se tendrá una entrada base X_a, la cual será comparada con una entrada positiva (de la misma clase) X_+ y con una negativa (de diferente clase) X_-. La distancia euclídea entre la matriz base y la entrada positiva es minimizada mientras que las entradas negativas se maximizan. Lo anterior puede formularse por la ecuación (3.6) y (3.7):

$$L(X_a, X_+, X_-) = \max\left(\left\| f_W(X_a) - f_W(X_+)\right\|^2 - \left\| f_W(X_a) - f_W(X_-)\right\|^2 + \alpha, 0\right) \qquad (3.6)$$

O en su forma reducida:

$$L(X_a, X_+, X_-) = \max\left(d_1^2 - d_2^2 + \alpha, 0\right) \qquad (3.7)$$

Donde α es un margen utilizado para fijar una distancia mínima entre los vectores de diferentes clases, d_1 es la distancia entre el vector ancla X_a y el vector de su misma clase X_+. Finalmente, d_2 es la distancia que existe entre X_a y X_-. En la Figura 3.8 se observa de manera visual como una red va agrupando las clases al entrenarse con la función de pérdida por triplete.

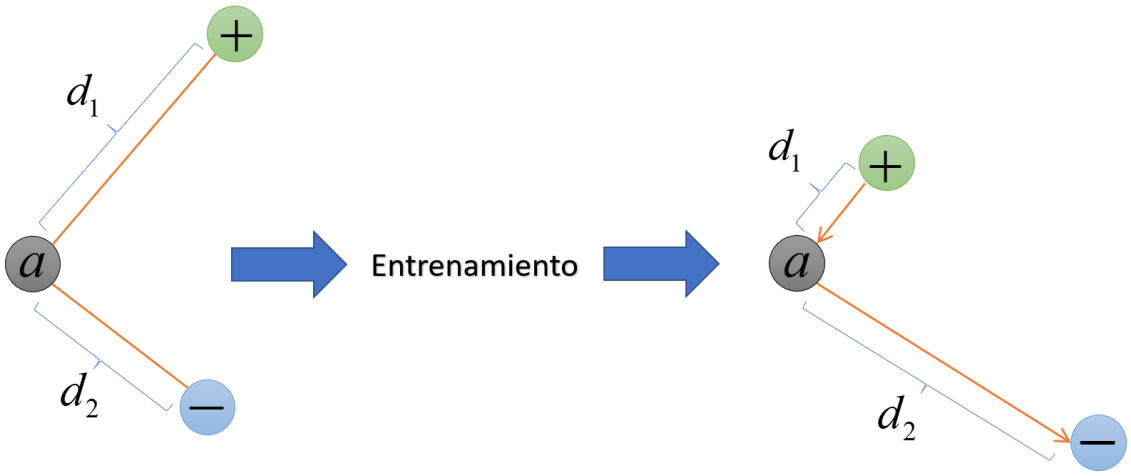

Figura 3.7. Funcionamiento de Triplet Loss.

3.3.1.3Pérdida de cuatrillo.

La función de pérdida por cuatrillo, o *Quadruplet Loss*, parte de un concepto similar al de *Triplet Loss*, con la diferencia de que se incluye otra muestra, la cual deberá ser de una clase distinta a todas las anteriores. El cuatrillo creado con la forma (X_a, X_+, X_-, X_b), donde X_a, X_+, y X_- conservan el mismo significado que en la perdida por triplete, mientras que el nuevo término X_b, representa a una nueva clase distinta a X_a, X_+ *y* X_- que se añade a la ecuación. En *Triplet Loss* el algoritmo busca que la distancia entre X_a y X_+ se minimice y que la distancia entre X_a y X_- se vea incrementada. Con la adición de X_b, ahora la imagen X_- aumentará la distancia no solo con X_a, si no que también de X_b. *Quadruplet Loss* muestra una mejora en cuanto al número de iteraciones necesarias para separar las clases con respecto a pérdida contrastiva y por triplete, ya que se incrementa el número de clases que se alejan entre sí.

La ecuación que describe el funcionamiento de la funcion de pérdida *Quadruplet Loss* se muestra en las ecuaciones (3.8) y (3.9).

$$L(X_a, X_+, X_-, X_b) = \max\left(\left\| f_W(X_a) - f_W(X_+)\right\|^2 - \left\| f_W(X_a) - f_W(X_-)\right\|^2 + \alpha_1, 0\right) + \max\left(\left\| f_W(X_a) - f_W(X_+)\right\|^2 - \left\| f_W(X_-) - f_W(X_b)\right\|^2 + \alpha_2, 0\right) \quad (3.8)$$

O en su forma reducida:

$$L(X_a, X_+, X_-, X_b) = \max\left(d_1^2 - d_2^2 + \alpha_1, 0\right) + \max\left(d_1^2 - d_3^2 + \alpha, 0\right) \quad (3.9)$$

Lo anterior se describe de manera gráfica en la Figura 3.9, donde se observa un ejemplo del funcionamiento de la función de perdida *Quadruplet Loss*.

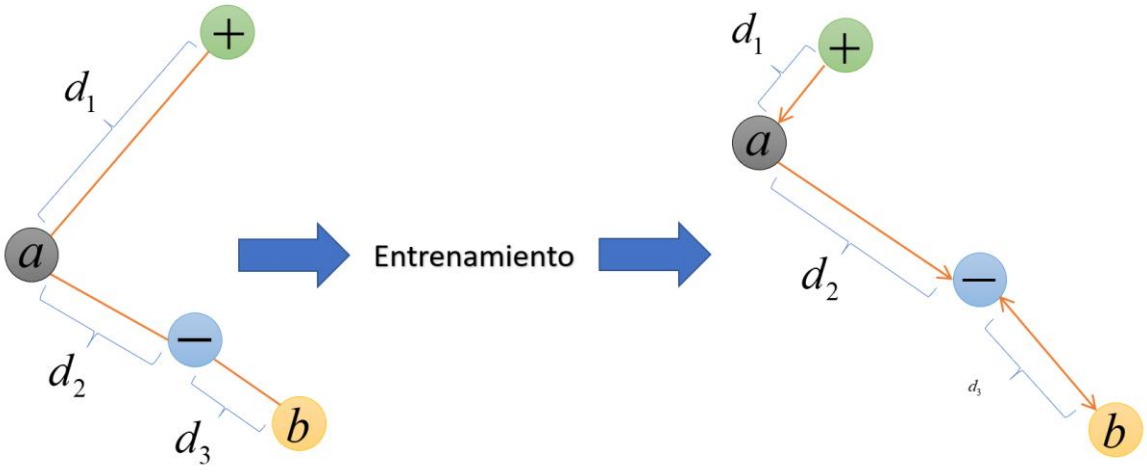

Figura 3.8. Funcionamiento de Quadruplet Loss.

3.4. Extracción de características

Como se ha dicho en secciones anteriores, las redes siamesas no realizan una tarea de clasificación, sino de encontrar si sus dos entradas son similares o no. Es esta misma particularidad permite a las redes siamesas el entrenarse con pocas muestras y aun así tener buen desempeño.

Una de las partes más importantes de las redes siamesas es la extracción de características, ya que esta se realiza sobre las *n* entradas al mismo tiempo en las instancias de la red, la cual debe realizarse de la misma manera para cada imagen de entrada. Existen diversos métodos que nos permiten realizar dicha tarea, pero en este trabajo se decidió por utilizar una combinación

de *HOG* y *LBP* y redes neuronales convolucionales. La razón es que de esta manera se puede realizar una comparación entre un método que realice una extracción de características manual (*HOG* y *LBP)* y uno que use aprendizaje profundo (redes convolucionales).

3.4.1. HOG

En esta subsección se describirá el proceso realizado para obtener los vectores de características HOG, los cuales son utilizados por el método 1, descrito en la sección 3.5.1.

Para obtener los descriptores de HOG es necesario obtener primero los gradientes verticales y horizontales de las imágenes de entrada, los cuales nos brindan información sobre la dirección en la que el valor de intensidad de los pixeles va cambiando con respecto a sus vecinos.

La forma de obtener los gradientes es por medio de una convolución con los kernel Sobel vertical S_v y horizontal S_H, definidos en la ecuación (3.10):

$$S_V = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad S_H = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{3.10}$$

De la operación anterior se obtienen los gradientes verticales y horizontales de la imagen por medio de una operación de convolución (3.11) y (3.11a):

$$g_V = conv(I, S_V), \quad \mathrm{g}_H = conv(I, S_H) \tag{3.11}$$

$$conv = \sum_{m=0}^{M} \sum_{n=0}^{N} I[m,n] \cdot S[M-m, N-n] \tag{3.11a}$$

Donde M y N corresponden al tamaño horizontal y vertical de la imagen.

Estos valores de gradiente horizontal y vertical se introducen en las ecuaciones 2.1 y 2.2 para obtener el valor de magnitud del gradiente y su dirección. Luego estos valores se contabilizan y se agrupan por número de apariciones, es decir, se obtiene el histograma de magnitud e inclinación de la figura, El último paso consiste en realizar una normalización de la información. Un ejemplo gráfico de la aplicación de HOG a una imagen es la que se muestra en la Figura 3.10.

Al procesar la imagen y obtenerse las características se consigue un vector de tamaño fijo.

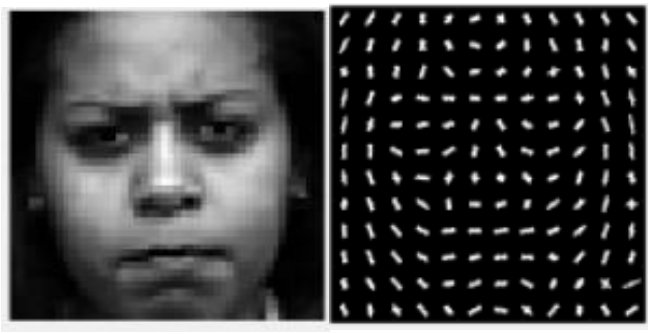

Figura 3.9. Imagen y sus características HOG.

3.4.2. LBP

Al igual que con HOG, los patrones locales binarios son utilizados por el método 1, por lo que en esta subsección se detallará el proceso que se lleva a cabo para obtener los vectores de características LBP.

Para extraer las características LBP de las imágenes de entrada se define un tamaño de celda que agrupará las características locales, donde a mayor el tamaño mayor pérdida de detalles, esta suele ser de 4 y 8 pixeles vecinos.

Para definir el número de pixeles vecinos se realizaron experimentos con la conectividad de vecinos 4 y 8, resultando en que los mejores desempeños se encontraban al utilizar los 8 vecinos más próximos, siguiendo un patrón radial como el que se muestra en la figura 3.10.

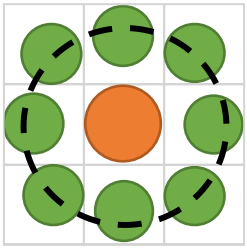

Figura 3.10. Vecindario del pixel central y los 8 vecinos que se consideran para LBP.

Al final del proceso se consigue un vector de características con un tamaño de 1x59 datos, el cual se concatena con las características HOG para obtener un nuevo arreglo de datos. Este número de define por la siguiente expresión:

$$características = (P*(P-1))+3 \tag{3.12}$$

Donde *P* es el número de pixeles vecinos a utilizar, en este caso de 8.

3.4.3. Red convolucional

Las Redes Neuronales Convolucionales (CNN) son, en la actualidad, uno de los algoritmos de *Deep Learning* más utilizados. Su uso típico es en tareas relacionadas con imágenes tal como el reconocimiento de objetos, segmentación, tracking, entre otras [29].

Una CNN consiste principalmente de las siguientes capas:

- **Capa convolucional.** Es núcleo de las CNN y su función principal es la de extraer las características importantes de la imagen por medio de la operación de la convolución.
- **Capa de *pooling*.** Luego de que la imagen fuera procesada por la capa convolucional, se obtienen mapas de características de la imagen. Dichos mapas son muy grandes y el procesarlos conllevaría una carga computacional muy alta. Es por eso que se lleva a cabo una operación para reducir las dimensiones de los mapas de características y mantener únicamente los datos relevantes.
- **Capa de completamente conectada.** Las capas pasadas procesan una imagen hasta obtener un mapa de características, por lo que, en conjunto, resultan ser simplemente un extractor de características. El siguiente paso es clasificar esa nueva información. Para pode llevar a cabo la clasificación se utiliza una red neuronal completamente conectada se toman las características procesadas por la convolución y el *pooling*, se aplanan y se utiliza como entrada para la red neuronal interconectada. Esta se encargará de procesar los datos y, finalmente, arroja los resultados de la clasificación.

La arquitectura de una red neuronal convolucional suele estar estructurada como se muestra en la Figura 3.12.

3.5. Modelos de arquitectura siamesa

Como se ha mencionado a lo largo del capítulo, los modelos de arquitectura siamesa contienen una etapa extractora de características que puede ser manual, por ejemplo, HOG y/o LBP, o integrada con la red. Esta sección es la que se instancia y le da una forma similar al de una pareja de siameses. y un clasificador por lo que es posible generar nuevas metodologías al modificar estas dos secciones. En las siguientes secciones se proponen dos arquitecturas de redes siamesas al utilizar distintas metodologías para la extracción de características y el reconocimiento de la emoción por medio de las expresiones faciales.

3.5.1. Método 1. (HOG + LBP) + MLP

El nombre se debe a la manera en la que las características son obtenidas de las imágenes de entrada, ya que primero se extraen características HOG y LBP, concatenándose antes de ingresar a la red completamente conectada.

El primer método desarrollado corresponde a un modelo donde la etapa de extracción de características se realiza de manera manual, obteniendo primero las características HOG y concatenándolas con las características obtenidas por LBP. Este nuevo vector de características se introduce a una red completamente conectada, la cual es entrenada utilizando la función de pérdida *contrastive Loss K* y un optimizador de tipo *ADAM* y procesará la información obtenida de HOG y LBP para obtener un vector de salida de 2048 características.

La estructura de la capa completamente conectada es la mostrada en la Tabla 3.1. El modelo de red siamesa que se diseñó para este método se muestra de manera gráfica en la figura 3.11. La entrada a la red se muestra en la ecuación 3.13 y tiene una salida de 2048 caracteristicas.

$$InputFullyConnected = \left[HOG \quad LBP \right] \tag{3.13}$$

Tabla 3.1. Estructura de la subred CNN.

Extractor CNN	
Capa	Características
FULLY CONNECTED	2048

Como se mencionó anteriormente, el entrenamiento se realizó utilizando la función de pérdida *contrastive Loss*, un optimizador tipo ADAM y se llevó a cabo durante 1000 iteraciones, realizando un respaldo de los pesos cada 100 iteraciones. El valor *margin* da el valor mínimo de separación entre clases diferentes y su selección se debió a diversas pruebas realizadas, los valores de gradientes y la tasa de aprendizaje se obtuvieron de una red siamesa con arquitectura similar, pero con un enfoque a la detección de caracteres, finalmente, el tamaño de lote seleccionado se debe al tamaño de memoria de la tarjeta gráfica utilizada en el entrenamiento, pues era el valor máximo permitido sin saturar su capacidad. Los parámetros completos del entrenamiento se muestran en la Tabla 3.2.

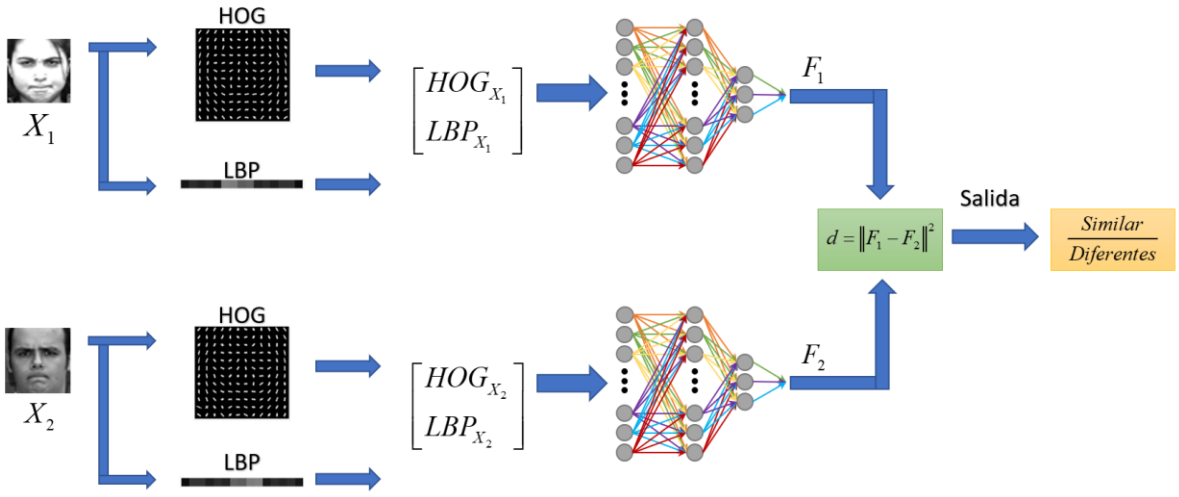

Figura 3.11. Esquema general del método HOG + LBP + MLP.

3.5.1.1 Resultados

Debido a la forma en la que funcionan las redes siamesas se obtuvieron dos matrices de confusión, la manera de obtención se indica en su correspondiente subsección, 3.5.1.2 para detección de similitud y 3.5.1.3 para clasificación. Para ambos casos, la imagen ancla se obtiene al extraer una imagen de la base de datos. Posteriormente, se obtienen los vectores de características de la emoción ancla y de todas las otras imágenes de la base de datos. Los siguientes resultados se llevaron a cabo con la base de datos CK+, cuyos detalles se encuentran en la sección 3.1.1

Tabla 3.2. Parámetros de entrenamiento del método 2.

Parámetros de entrenamiento	
Parámetro	Valor
Margen	0.3
Optimizador	ADAM
Gradiente descendente	0.9
Gradiente descendente cuadrado	0.99
Tasa de aprendizaje	0.0001
Tamaño de lote	180
Función de pérdida	*Contrastive Loss*
Iteraciones	*1000*

Como se ha manejado en capítulos anteriores, la manera en la que las redes siamesas miden el grado de similitud entre una clase u otra es por medio de la distancia euclídea entre sus vectores de características, por lo que se procede a obtener la diferencia entre la imagen ancla y todas las otras emociones del conjunto de datos y, finalmente, se ordenan de menor a mayor, siendo las distancias más pequeñas las que más posibilidades tienen de ser de la misma clase que la imagen ancla. Con este vector contenedor de las menores distancias, es posible catalogar que imágenes pueden ser consideradas similares por medio de una umbralización. Por el contrario, si se desea clasificar la imagen, se consideran las clases de las ***N*** imágenes más cercanas a las características de la imagen ancla. La clase que se asignará será aquella etiqueta con mayor número de apariciones.

3.5.1.2 Detección de similitud

Para este método, HOG+LBP, se catalogan como clases similares todas aquellas imágenes que presenten una distancia euclideana igual o menor a 0.1 de la imagen ancla. La tabla de desempeño para cada clase con el conjunto de pruebas se muestra en la Tabla 3.3, mientras que el conjunto de validación se encuentra en la Tabla 3.4. La primera fila indica el mejor promedio

de detección por emoción luego de procesar todo el conjunto de imágenes, la segunda fila indica la peor detección. La última fila indica el promedio general de detección por emoción.

3.5.1.3 Detección de clase

Para la detección de la clase se prescinde del uso del umbral y se realiza un conteo de las ***N*** clases más cercanas con ayuda de un histograma. Para el método 1, HOG+LBP, se consideraron las tres menores distancias y la etiqueta con mayor número de apariciones es la que se asigna como la clase de pertenencia.

Tabla 3.3. Tabla de desempeño por clase en el conjunto de pruebas para el método 1.

	Enojo	Desprecio	Disgusto	Miedo	Alegría	Tristeza	Sorpresa	Promedio General
Promedio	81%	75%	76%	77%	78%	80%	80%	79%
Mejor	91%	85%	93%	85%	88%	85%	91%	88%
Peor	68%	68%	62%	62%	68%	72%	70%	67%

Tabla 3.4. Tabla de desempeño por clase en el conjunto de validación para el método 1.

	Enojo	Desprecio	Disgusto	Miedo	Alegría	Tristeza	Sorpresa	Promedio General
Promedio	71%	69%	69%	74%	73%	67%	71%	71%
Mejor	93%	77%	88%	83%	96%	77%	93%	86%
Peor	55%	65%	34%	52%	55%	55%	34%	50%

Para la tarea de clasificación se obtuvo una matriz de confusión de la etapa de pruebas y validación, mismas que se encuentran en las Figuras 3.12 y 3.13. Cabe recordar que estas

matrices de confusión no representan los resultados de la Tabla 3.3 y 3.4, ya que las tablas representan el etiquetado con base en el nivel de similitud mientras que la Figura 3.12 y 3.13 al realizar una clasificación.

3.5.2. Método 2. CNN + MLP

En este método se lleva a cabo una extracción de características automática por medio de una red neuronal artificial profunda (RNAP) de tipo convolucional cuya salida es un vector de 2048 dimensiones, el cual será utilizado para obtener las distancias entre las imágenes. A diferencia del método anterior (HOG+LBP), aquí todas las características se obtienen de manera automática por la RNAP.

La estructura de la RNAP consta de 1 capa de entrada, 3 capas convolucionales con función de activación ReLu y cada una conectada a una capa de *Max Pooling*. Finalmente, las características se introducen a una capa completamente conectada cuya salida son 2048 características. La estructura de la RNAP se muestra en la Tabla 3.5 y de manera gráfica en la Figura 3.14. Finalmente, la estructura completa de la red siamesa del método 2 está contenida en la Figura 3.15.

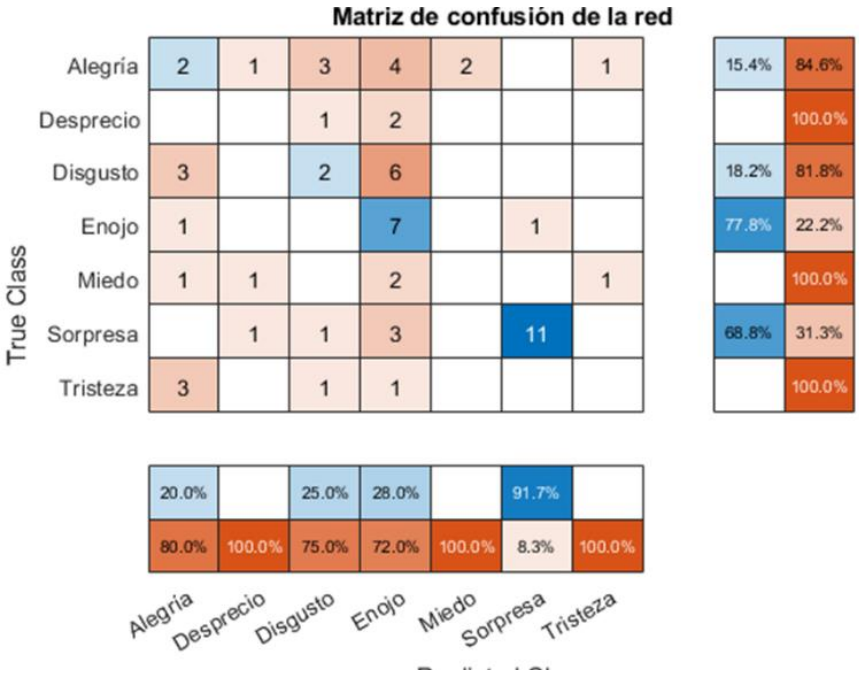

Figura 3.12. Matriz de confusión del Método 1 (HOG+LBP) para el conjunto de pruebas.

Tabla 3.5. Estructura de la subred RNAP.

RNAP	
Capa	Características
Entrada	48x48x1
CONV1	10x10 (128 filtros)
CONV2	7x7 (256 filtros)
CONV3	4x4 (256 filtros)
MAX POOLING	2x2 (stride = 2)
FULLY CONNECTED1	2048 (Narrow-normal)

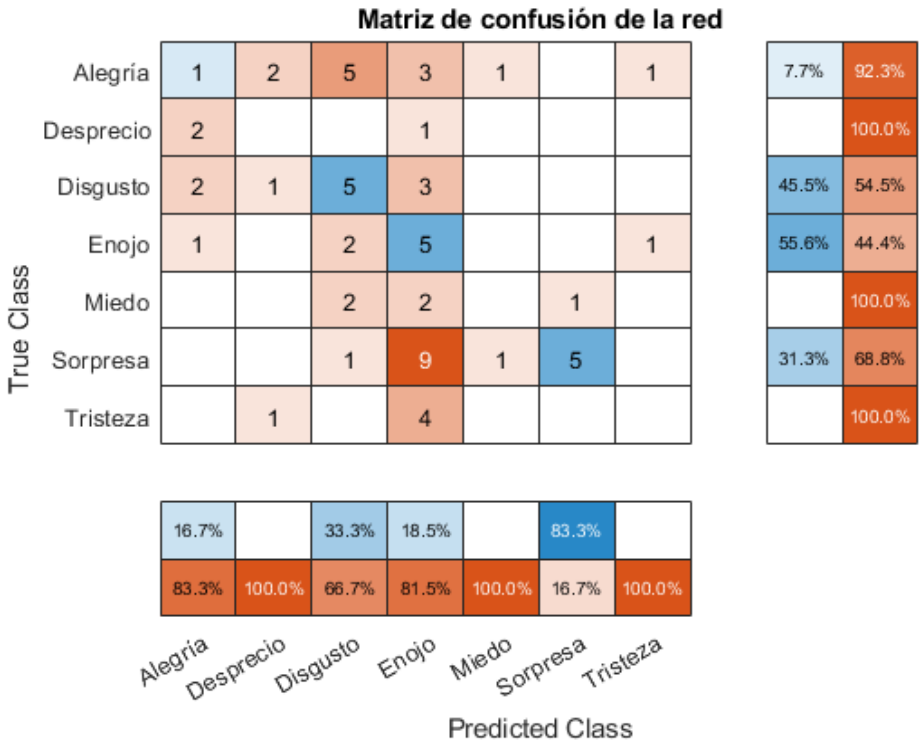

Figura 3.113. Matriz de confusión del Método 1 para el conjunto de validación.

Para el entrenamiento se utilizó la función de pérdida *Contrastive Loss*, con un margen de 0.3. Los demás parámetros que se utilizaron para llevar a cabo el entrenamiento se listan en la Tabla 3.6.

3.5.2.1 Resultados

El proceso de obtención de las distancias es el mismo que el realizado para el método 1, con la diferencia de que se utiliza las características obtenidas de manera automática por la red y no las calculadas con HOG y LBP. Sin embargo, se mantiene la diferencia entre la tarea de detección de similitud y la clasificación de imágenes.

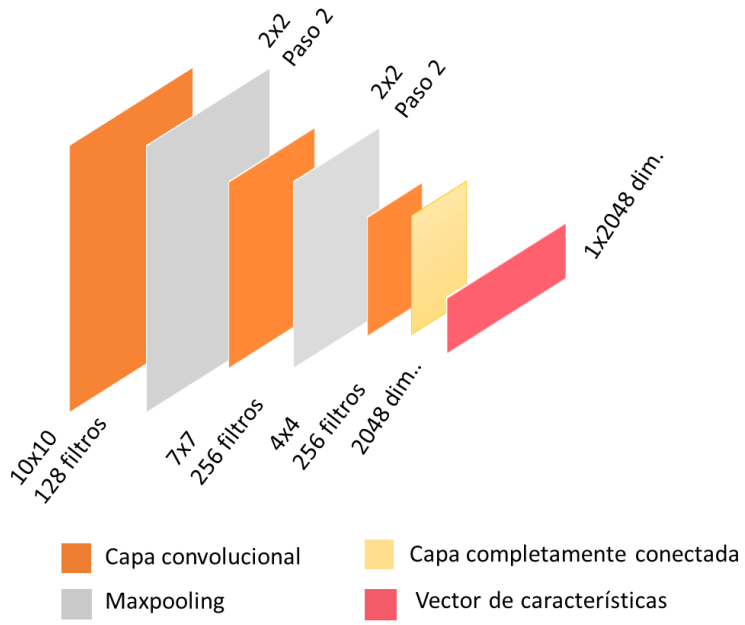

Figura 3.14. Arquitectura de la RNAP.

3.5.2.2 Detección de similitud

Para este método, RNAP, se catalogan como clases similares todas las imágenes que presenten una distancia igual o menor a 0.1 de la figura ancla. La tabla de desempeño para cada clase con el conjunto de pruebas se muestra en la Tabla 3.7, los resultados del conjunto de validación se muestran en la Tabla 3.8.

3.5.2.3 Detección de clase

Para la detección de la clase se prescinde del uso del umbral y se realiza un conteo de las ***N*** clases más cercanas con ayuda de un histograma. Para el método 2 se consideraron las tres

menores distancias y la etiqueta con mayor número de apariciones es la que se asigna como la clase de pertenencia.

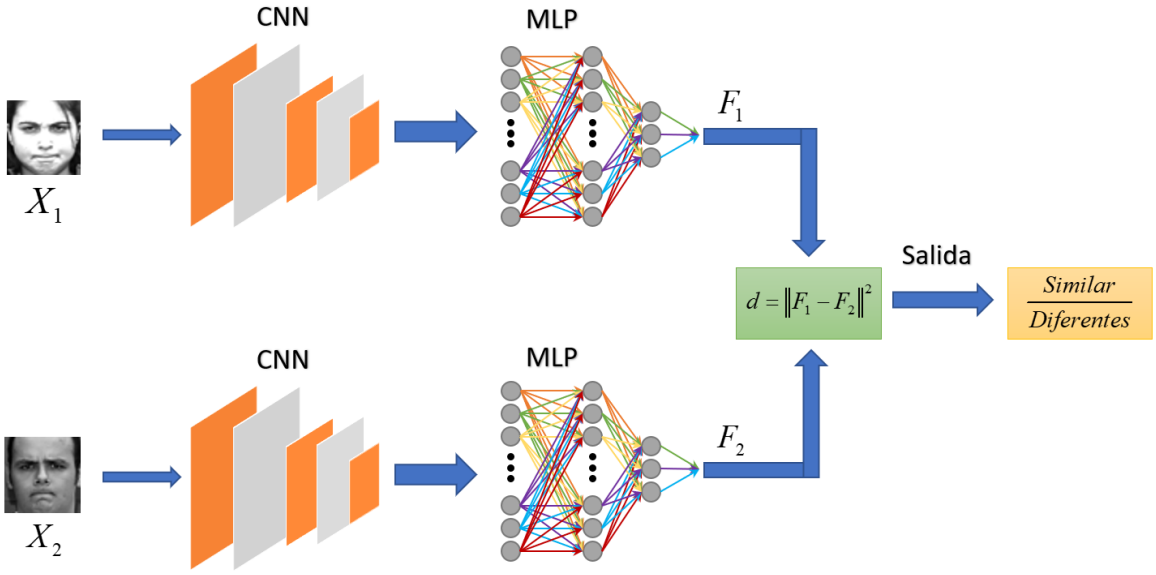

Figura 3.15. Esquema de la red siamesa del método RNAP.

Tabla 3.6. Tabla de desempeño por clase en el conjunto de pruebas para el método 2.

	Enojo	Desprecio	Disgusto	Miedo	Alegría	Tristeza	Sorpresa	Promedio General
Promedio	85%	95%	88%	84%	90%	83%	85%	**87%**
Mejor	92%	97%	97%	92%	95%	90%	92%	**93%**
Peor	72%	90%	90%	70%	80%	75%	67%	**77%**
Varianza	0.12	0.64	0.06	0.34	2.83	0.01	0.02	**----**
Desviación	0.35	0.80	0.25	0.58	1.68	0.11	0.16	**----**

Al igual que en los resultados para la detección de clase del método 1, se obtienen las matrices de confusión para el conjunto de pruebas y de validación, mismas que se encuentran en las Figuras 3.16 y 3.17.

Tabla 3.7. Tabla de desempeño por clase en el conjunto de validación para el método 2.

	Enojo	Desprecio	Disgusto	Miedo	Alegría	Tristeza	Sorpresa	Promedio General
Promedio	82%	84%	90%	87%	96%	90%	93%	**88%**
Mejor	87%	90%	97%	93%	98%	92%	97%	**93%**
Peor	79%	79%	82%	75%	80%	87%	67%	**78%**
Varianza	0.2	0.46	1.07	0.06	0.01	0.13	3.13	**----**
Desviación	0.44	0.68	1.03	0.25	0.12	0.366	1.77	**----**

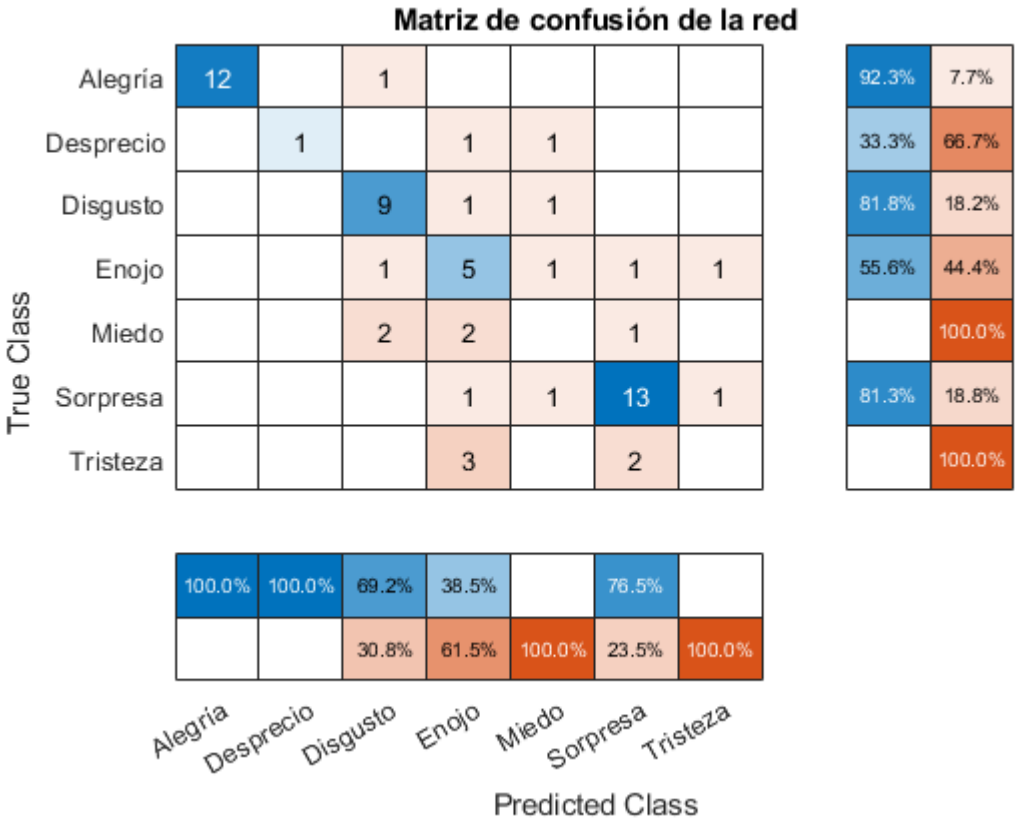

Figura 3.126. Matriz de confusión del Método 2 para el conjunto de Pruebas

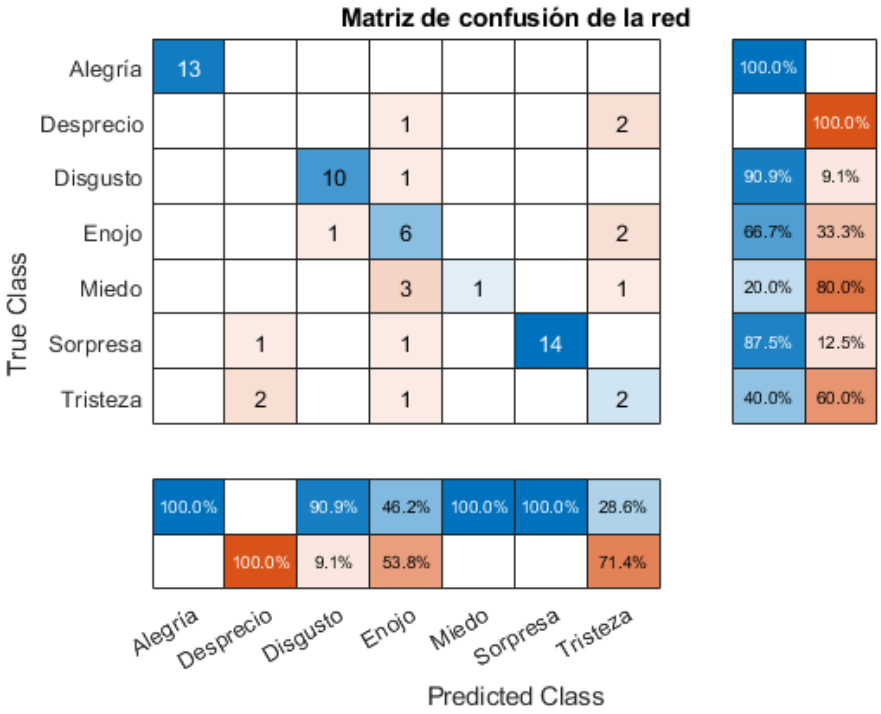

Figura 3.137. Matriz de confusión del Método 2 para el conjunto de pruebas

Los resultados obtenidos del método 1, HOG+LBP, comparados con RNAP, muestran como el funcionamiento obtenido con técnicas de *Machine Learning* supera ampliamente al que se logra al extraer las características por métodos con parámetros fijos. Además, estos experimentos nos sirven como un acercamiento a los resultados que pueden obtenerse al cambiar a una mejor base de datos, puesto que la utilizada para este capítulo cuenta con problemas de desbalanceo de clases y una baja resolución de entrada. Sin embargo, estas mismas carencias, junto con los resultados obtenidos por el método 2, demuestran las capacidades de las redes de arquitectura siamesas en el ámbito de *few-shot learning*, ya que se lograron obtener resultados con promedios generales por encima del 85% con una cantidad de imágenes baja. Además, el sistema es capaz de funcionar con clases nunca vistas en el entrenamiento, debido a que su entrenamiento parte de acotar distancias entre las características de una misma clase, mientras aleja las de clases diferentes.

La arquitectura de red correspondiente al método 2 será utilizado en el capítulo siguiente, donde se lleva a cabo un reentrenamiento con una mejor base de datos.

CAPÍTULO 4.

4. RESULTADOS DEL ENTRENAMIENTO DE LA RED

En el capítulo pasado se mostraron resultados experimentales para redes de tipo siamesa y su desempeño ante el conjunto de imágenes CK+. En este capítulo se cambió a la base de datos de KDEF, publicada por el Instituto Karolinska con motivo de crear una base de datos para la identificación de emociones, la cual cuenta con más imágenes, 4,900 dividido entre 70 individuos, y se añaden emociones, siendo en total las emociones de: miedo, enojo, disgusto, alegre, neutral, triste y sorprendido, con 700 fotos cada una. Además, la resolución de las imágenes también es diferente, aumentando de los 42x42 pixeles a 100x100. Para esto se realizó un reentrenamiento de la red, ajustando el tamaño de entrada a las nuevas dimensiones. La arquitectura de la red es la misma que la del método 2, RNAP. Debido a que la base de datos y el número de emociones fue diferente el reentrenamiento se realizó siguiendo el siguiente esquema:

- Se entrena la red con 2 emociones siguiendo el esquema: emociones similares, medianamente similares y diferentes.
- Se anotan los resultados y se analizan cuales emociones funcionan bien juntas y cuáles no.
- Se incrementa el número de emociones en 1 y se repite el proceso.

Lo anterior es repetido hasta que el desempeño de la red disminuya a menos del 70% de precisión para tres o más emociones. En este caso, el límite máximo para la red se encontró entre 5 y 6 emociones para el entrenamiento.

4.1. Pasos previos al entrenamiento de la red

Antes de comenzar con el entrenamiento se realizó un análisis exploratorio sobre la base de datos KDEF, esto con fines de asegurar una uniformidad de los datos y conocer su estructura

para eliminar información que pudiera introducir información no relevante al entrenamiento de la red.

Debido a este análisis fue posible encontrar 3 factores de riesgo:

- Imágenes con iluminación distinta al del resto.
- Imágenes vacías o completamente negras.
- Aberraciones cromáticas.

Para el primer punto se optó por conservar las muestras, ya que al momento de procesar las imágenes se realiza una binarización, la cual nos entregaba imágenes muy similares a las que no presentaban ese cambio de luminosidad. En el segundo caso, donde existían imágenes completamente negras, se decidió eliminar las muestras, puesto que el mantenerlas ponía en riesgo el correcto entrenamiento de la red. A pesar de haber eliminado dichas imágenes se encontró la solución de aplicar técnicas de *data augmentation* para recuperar la información perdida, ya que este fallo apareció únicamente en algunas muestras que correspondían a fotos de perfil, por lo que se podía utilizar el otro perfil del individuo y aplicar una transformación en la imagen invirtiéndola con respecto al eje y. Sin embargo, debido a que se decidió utilizar únicamente imágenes frontales para el entrenamiento, se decidió únicamente eliminarlas del conjunto general. Finalmente, las aberraciones cromáticas se solucionaban al procesar la información, transformando el espacio de color al de escala de grises. Un ejemplo de los 3 factores de riesgo se puede observar en la Figura 4.1, donde se muestran imágenes de los 3 casos.

Figura 4.1. Tres factores de riesgo encontrados en la base de datos KDEF. a) Iluminación diferente al resto. b) Imagen vacía (negras). c) Aberración cromática.

Una vez encontrados y solucionados los posibles errores en la base de datos, se procede a realizar un preprocesamiento de los datos.

4.1.1 Preprocesamiento de los datos.

En esta etapa se realiza una discriminación en el conjunto de datos, dejando fuera todos los elementos que no se consideraron necesarios para el entrenamiento o que pudiesen dificultar el proceso de aprendizaje de la red.

Como ya se mencionó en la sección 3.1.3, dedicada al conjunto de datos KDEF, cada imagen tiene un código único que brinda información sobre esta, tal como la sesión (A o B), el género del actor (**F**emenino o **M**asculino), su número identificatorio (del 1 al 30), la emoción gesticulada y el perfil facial (lateral izquierdo, lateral derecho, semiperfil izquierdo, semiperfil derecho y frontal). La información anterior es utilizada para llevar a cabo un filtrado de datos y conservar la información relevante para el posterior entrenamiento. Debido a pruebas y experimentos realizados, se decidió preservar las imágenes que presentaban el rostro frontal y semiperfiles, las imágenes que estaban de perfil completo fueron ignoradas.

Las imágenes que se obtuvieron al llevar a cabo el proceso anterior pasarán a una siguiente etapa, descrita en la siguiente subsección.

4.1.2 Procesamiento de los datos.

Antes de ingresar la información a la red, es necesario llevar a cabo un tratamiento de los datos, de manera que compartan la siguiente serie de características:

- Cambio al espacio de color a escala de grises.
- Normalización de los valores de pixeles (0 a 1).
- Escalamiento de resolución a 100x100 pixeles.

Para el cambio del espacio de color y el escalamiento de resolución se utilizarán las técnicas mencionadas en las secciones 3.2.1, Color a escala de grises, y 3.2.2, Escalamiento de resolución. La normalización se llevó a cabo por medio de la siguiente expresión matemática:

$$I_{ij} = \frac{I_{ij}}{\max(I)} \tag{4.1}$$

Donde I_{ij} corresponde al valor de intensidad del pixel (i, j) para una imagen I y $\max(I)$ el valor máximo de intensidad en la imagen.

4.2. Entrenamiento de la red siamesa.

A diferencia de redes neuronales convolucionales tradicionales, donde el entrenamiento se realiza ingresando una sola entrada para generar una inferencia, en las redes siamesas se necesitan 2 o más entradas, dependiendo de la función de pérdida. Para este trabajo de tesis se optó por utilizar la función pérdida contrastiva, detallada en la sección 3.3.1.1, por lo que fue necesaria la creación de duplas de imágenes y una etiqueta indicando si pertenecen a la misma clase, *Similar*, o son diferentes, *Distintas*.

Para la creación de las duplas se realizó una selección aleatoria, manteniendo un balance de clases al mantener una probabilidad de 50% de que sean clases diferentes o similares. Las muestras se almacenaban en lotes de 120 pares, los cuales serían utilizadas para realizar el entrenamiento de la red. Como se vio en la sección dedicada a la función de pérdida, la red aprendería a minimizar las distancias entre las características de emociones similares y a incrementar las de emociones diferentes. La red se entrenó durante 1,000 épocas cuando se introducían de 2 a 5 emociones diferentes. Cuando se incrementaban a 6 emociones, las épocas se aumentaban a 1,300.

Antes de entrenar el modelo final, se realizaron pruebas con diferente cantidad de emociones, variando estas para conocer que clases son más fáciles de distinguir y cuáles no. Varios de los resultados se mostrarán en la siguiente sección.

4.3. Resultados preliminares.

Para conocer los límites de la red se llevaron a cabo entrenamientos con 2, 3, 4, 5, 6 y 7 emociones, indicando también las emociones que la red mejor diferenciaba y aquellas con las que la red presentaba problemas. En las siguientes subsecciones se mostrarán resultados relevantes para distintas cantidades de emociones.

4.3.1. Entrenamiento con 2 emociones.

Como primer experimento se entrenó la red con emociones que en perspectiva humana son contrarias (alegría y enojo, alegría y tristeza). Al obtener resultados con precisiones mayores al 90%, se llevaron a cabo entrenamientos con las de emociones restantes. Se trató de mantener un esquema de selección en el que se elegían emociones similares, distintas y neutras.

Como se mencionó anteriormente, el entrenamiento con emociones que se perciben como diferentes, arrojó resultados prometedores para la detección de similitud y clasificación, logrando precisiones con valores mayores al 90% de precisión en el conjunto de entrenamiento y pruebas. Esto se ve reflejado en la Tabla 4.1 donde se muestra el resultado para la tarea etiquetar el nivel de similitud, y en la Figura 4.2, donde se clasificaban las emociones, ambas realizadas sobre el conjunto de pruebas. En el conjunto de entrenamiento se lograron resultados del 100% de precisión.

Con la información obtenida se pudo determinar que las emociones miedo y sorpresa se gesticulan con expresiones faciales muy similares, razón por la cual la red tiene problemas para determinar el nivel de similitud de las clases y para clasificar. Caso contrario ocurre con las demás emociones, donde se obtuvieron buenos resultados en ambas tareas, logrando resultados cercanos al 100% de precisión y en algunos casos, del 100%. Al igual que en el capítulo 3, las tablas y matrices no representan los mismos experimentos, puesto que las tablas corresponden a la tarea de nivel de similitud y las matrices al de clasificación. Esto es igual para todas las pruebas.

Tabla 4.1. Resultados de la red en el conjunto de pruebas para 2 emociones. Nivel de similitud.

Emociones	TP	TN	FP	FN	Precisión
Enojo	33	45	0	5	93.97
Alegría	39	43	1	0	98.79
Neutral	96	80	17	2	90.26
Sorpresa	96	97	1	1	98.97

Miedo	0	40	0	37	51.95
Sorpresa	0	45	32	0	58.44

4.3.2. Entrenamiento con 3 emociones.

Debido a que se obtuvo una alta precisión en la mayoría de los casos al entrenar con 2 emociones, menos en miedo y sorpresa, se decidió continuar y aumentar la complejidad al incluir otra emoción. Se siguió un esquema similar al anterior, donde se trató de agrupar emociones con diferencias marcadas y no tan marcadas y aquellas que son similares en expresiones.

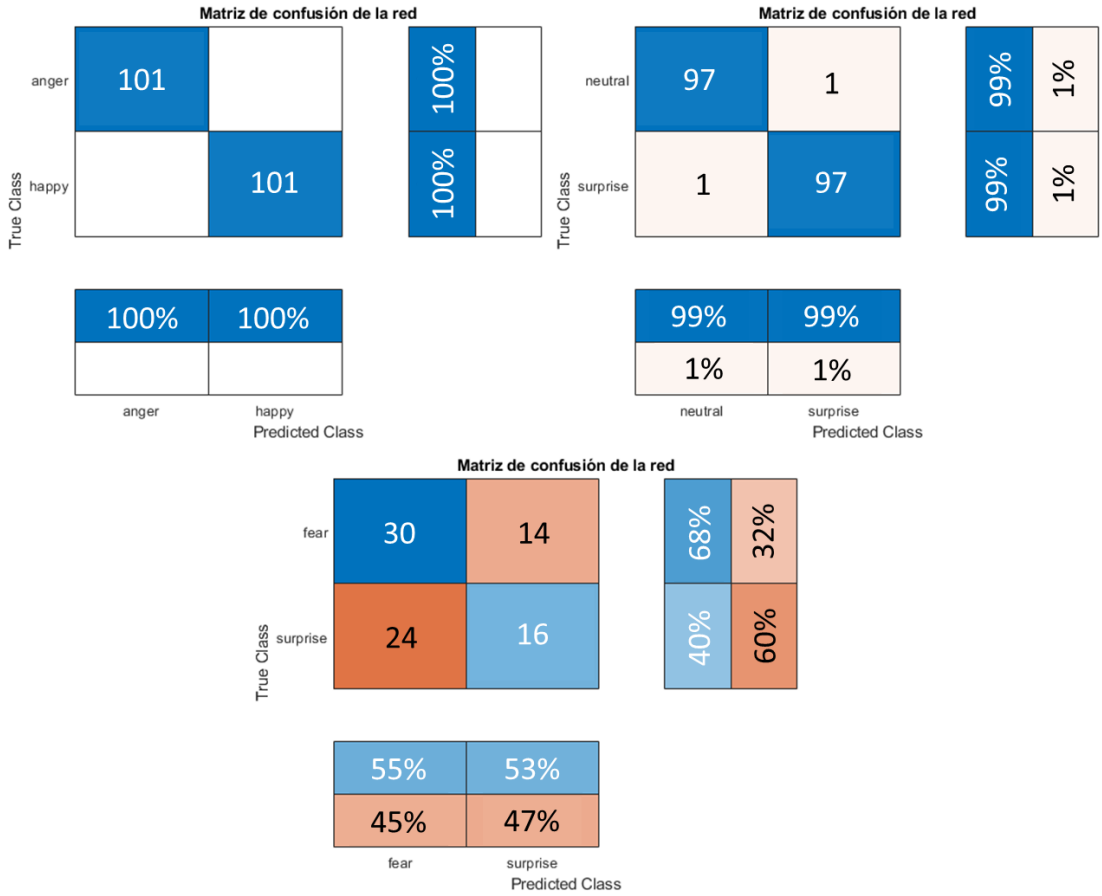

Figura 4.2. Matrices de confusión para el conjunto de pruebas en 2 emociones, de izquierda – derecha de arriba abajo son: alegría y enojo, neutral y sorpresa, miedo y sorpresa. Clasificación.

Algunos de los resultados sobre el conjunto de pruebas se muestran en la Tabla 4.2, donde se obtuvieron resultados para la tarea de nivel de similitud y en la Figura 4.3 en la cual se muestran las matrices de confusión al realizar una tarea de clasificación. Al igual que con dos emociones, los resultados sobre el conjunto de entrenamiento arrojaban una precisión del 100%, por lo que no fueron incluidos en el texto.

Los resultados generales de precisión fueron buenos, alcanzando resultados entre el 80% al 90% de precisión en la mayoría de los casos, tanto para la tarea de clasificación como en la de nivel de similitud. Un detalle que llamó la atención fue la mejora obtenida al combinar las emociones de sorpresa y miedo con alguna otra, pues como se vio en la sección anterior, al entrenar la red para diferenciar entre dichas clases se obtenían rendimientos malos, cosa que no pasaba al añadir una nueva emoción. Una de las razones por la cual esto sucede es por una posible compensación de la red, debido a que, al tener otra clase en el entrenamiento, se incrementan los datos y la información de la cual puede aprender.

Tabla 4.2. Resultados de la red en el conjunto de pruebas para 3 emociones. Nivel de similitud.

Emociones	TP	TN	FP	FN	Precisión
Alegría	61	191	37	4	86.01
Tristeza	68	169	25	31	80.89
Enojo	50	199	0	44	84.98
Disgusto	70	204	19	0	93.52
Sorpresa	65	194	0	34	88.4
Neutral	77	187	3	26	90.1
Disgusto	39	78	6	2	93.6
Sorpresa	19	89	10	7	86.4
Miedo	18	69	4	34	69.6

4.3.3. Entrenamiento con 4 emociones.

Debido a que los experimentos con 3 emociones mostraron rendimientos superiores al 80% en la mayoría de los casos, se decidió incrementar el número de emociones. Puesto que no se tienen muchas emociones en el conjunto, en específico son 7, los entrenamientos no pudieron seguir completamente el esquema de: emociones similares, diferentes y neutrales.

Como en las subsecciones anteriores, se mostrarán algunos de los resultados obtenidos en el conjunto de pruebas. En la tabla 4.3 se almacenan los datos obtenidos al entrenar con 4 emociones para la tarea de determinar niveles de similitud y en la Figura 4.4, diversas matrices de confusión para la tarea de clasificación de emociones. Al igual que en los casos anteriores, la precisión del conjunto de entrenamiento fue del 100%.

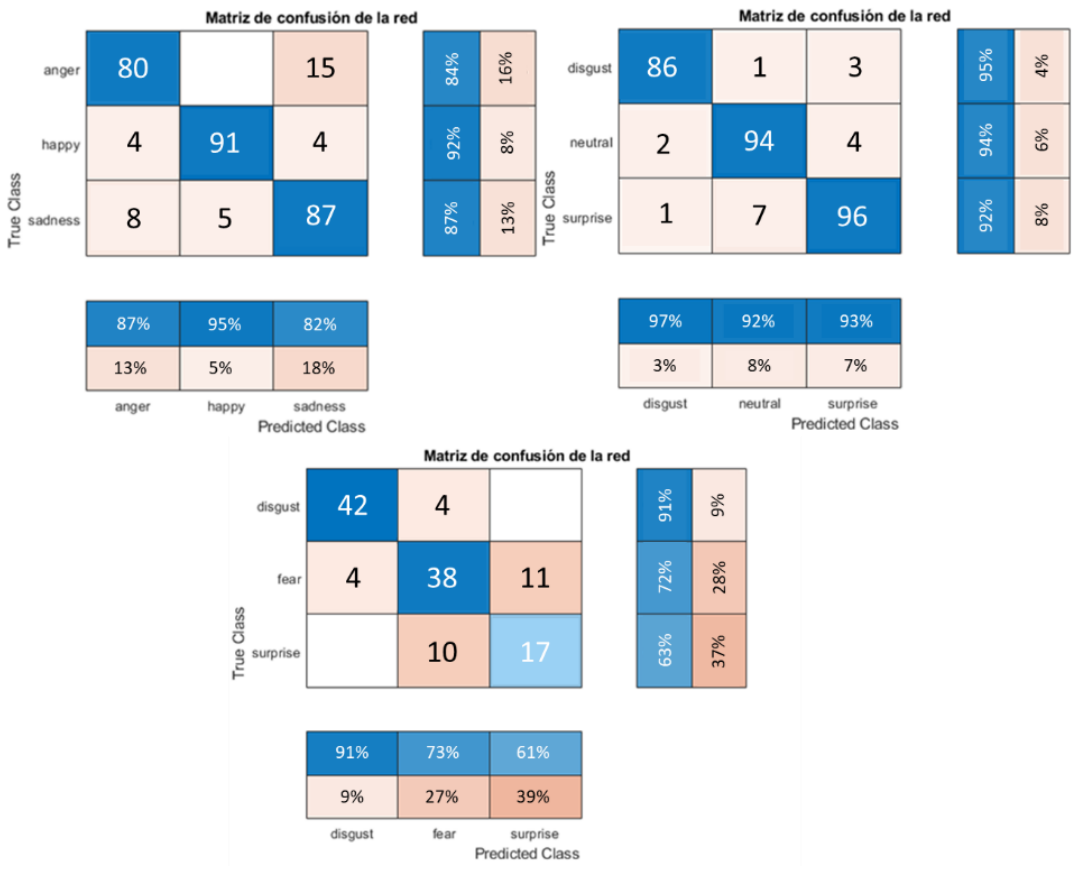

Figura 4.3. Matrices de confusión para el conjunto de pruebas en 3 emociones. Clasificación.

Los resultados obtenidos al entrenar con 4 emociones fueron generalmente buenos, teniendo casos donde la precisión general del sistema era de 90%. Debido a la mejora de desempeño logrado al agrupar las emociones de sorpresa y miedo con otras, se decidió seguir incluyéndolas en los experimentos, logrando así resultados que arrojan detalles interesantes.

Al juntar sorpresa y miedo junto con enojo y neutral la precisión del sistema se mantenía entre el 80% y 90%. Sin embargo, al juntarla con tristeza y alegría o disgusto y tristeza, los resultados disminuían enormemente, obteniendo resultados inferiores al 50%, convirtiéndose en una red no viable. De este experimento se llegó a la conclusión de que las emociones de miedo y sorpresa sirven como catalizador, ya que la combinación con otras clases provoca un aumento o disminución de los resultados. En este caso, la etiqueta tristeza podría ser aquella que provoca un mal rendimiento del sistema al combinarse con miedo y sorpresa.

Tabla 4.3. Resultados de la red en el conjunto de pruebas para 4 emociones. Nivel de similitud.

Emociones	TP	TN	FP	FN	Precisión
Neutral	36	115	7	9	90.42
Enojo	22	127	0	18	89.22
Sorpresa	30	123	4	10	91.62
Miedo	18	116	21	12	80.24
Enojo	28	114	9	16	85.03
Alegría	35	128	0	4	97.6
Sorpresa	35	125	0	7	95.81
Neutral	34	120	8	5	92.22
Alegría	45	0	122	0	26.95
Tristeza	40	0	127	0	23.95
Sorpresa	40	0	127	0	23.95
Miedo	39	0	128	0	23.35

4.3.4. Entrenamiento con 5 emociones.

El entrenamiento con 5 emociones arrojó resultados positivos, con precisiones arriba del 70% en la mayoría de los casos. Algunos de los resultados obtenidos se muestran en la Tabla

4.4, donde se utiliza el conjunto de pruebas para la tarea de encontrar la similitud entre imágenes, y la Figura 4.5, en el cual se realiza una clasificación entre las 5 emociones. Los resultados de precisión obtenidos del conjunto de entrenamiento variaban entre el 92 y el 99%.

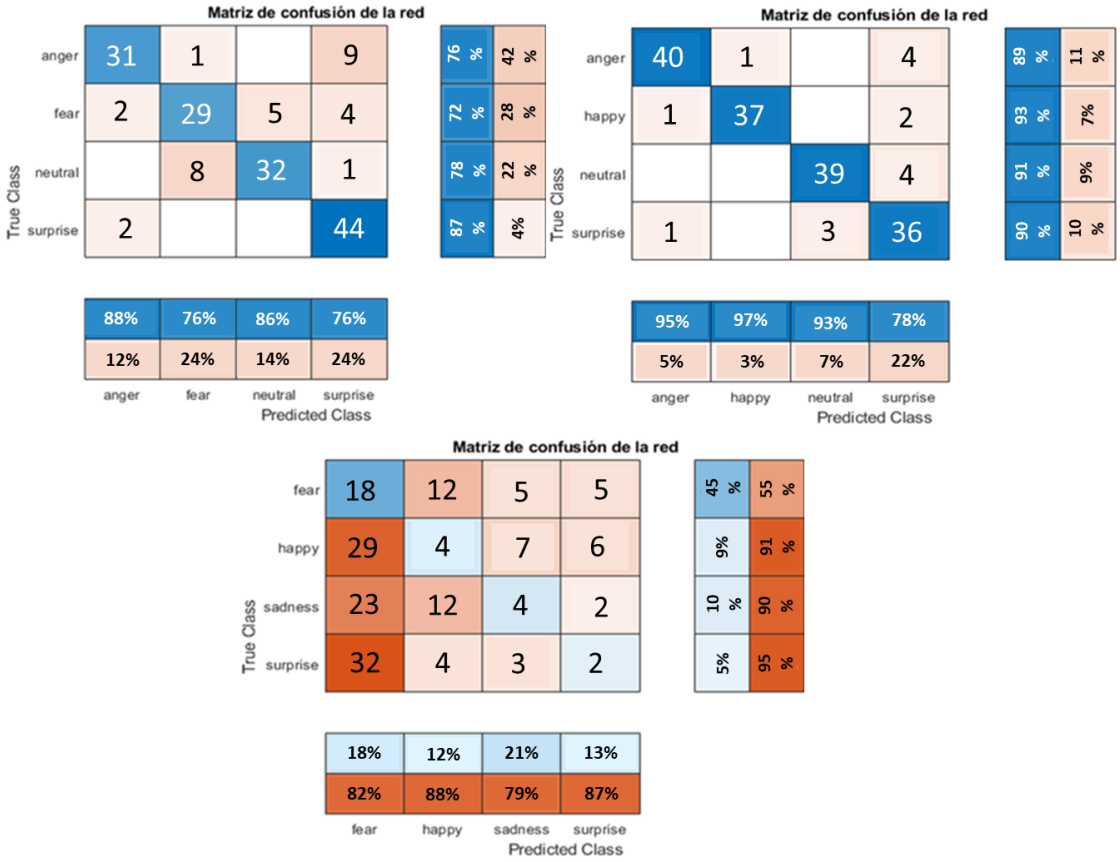

Figura 4.4. Matrices de confusión para el conjunto de pruebas en 4 emociones: enojo, miedo, neutral y sorpresa, enojo, alegría, neutral y sorpresa, miedo, alegría, tristeza y sorpresa. Clasificación.

4.3.5. Entrenamiento con 6 emociones.

A diferencia de los entrenamientos con 2 a 5 emociones, donde se llevaron a cabo 1,000 épocas, el entrenamiento con 6 emociones necesitó 1,300 iteraciones para lograr resultados óptimos. Si se incrementaban o disminuían la cantidad de épocas, la red mostraba una incapacidad para diferenciar correctamente las emociones con las que se entrenaba.

En esta ocasión únicamente dos redes alcanzaron niveles de precisión aceptables. Para las emociones alegría, enojo, sorpresa, miedo, tristeza y disgusto, se tuvo un promedio general del 83%, siendo la clase tristeza aquella con el peor desempeño, de 63.75% y alegría la más alta, con una precisión del 90-84%. En cambio, la red entrenada con las emociones enojo, alegría, miedo, neutral, tristeza y disgusto, obtuvo un rendimiento general del 59%, siendo la peor clase disgusto, con una precisión del 34.6% y la mejor alegría, con un resultado del 81.27%.

En la Tabla 4.5 se condensan los resultados de la red con el conjunto de pruebas en la tarea de distinguir el nivel de similitud. En cambio, en la Figura 4.6 se muestran las matrices de confusión para las dos redes finales, en este caso, los resultados son para la tarea de clasificación.

Se realizaron entrenamientos con todas las combinaciones posibles de emociones, de las cuales solo dos redes entregaron resultados superiores al 40% de precisión. Adicionalmente se entrenó un modelo con las 7 emociones, el cual tenía un pésimo desempeño, razón por la cual se decidió guardar los mejores modelos para 5 y 6 emociones

Tabla 4.4. Resultados de la red en el conjunto de pruebas para 5 emociones. Nivel de similitud.

Emociones	TP	TN	FP	FN	Precisión
Enojo	10	158	6	35	80.38
Alegría	36	169	0	4	98.09
Sorpresa	27	156	12	14	87.56
Neutral	24	165	5	15	90.43
Miedo	12	141	28	28	73.21
Enojo	18	118	46	27	65.07
Alegría	31	169	0	9	95.69
Sorpresa	35	167	1	6	96.65
Neutral	15	138	32	24	73.21
Tristeza	19	144	25	21	77.99

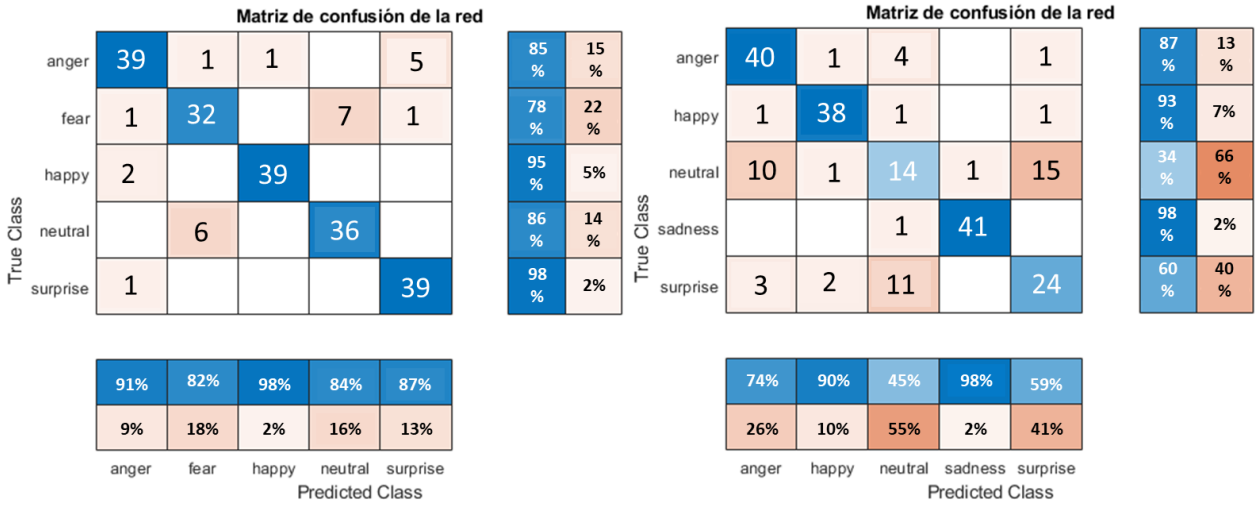

Figura 4.5. Matrices de confusión para el conjunto de pruebas en 5 emociones: enojo, miedo, alegría, neutral y sorpresa y enojo, alegría, neutral, tristeza y sorpresa. Clasificación.

Tabla 4.5. Resultados de la red en el conjunto de pruebas para 5 emociones. Nivel de similitud.

Emociones	TP	TN	FP	FN	Precisión
Enojo	25	110	105	11	53.78
Alegría	40	164	42	5	81.27
Miedo	19	88	121	23	42.63
Neutral	28	150	70	3	70.92
Tristeza	14	170	40	27	73.31
Disgusto	32	55	145	19	34.66
Enojo	27	194	24	6	88.05
Alegría	21	207	0	23	90.84
Sorpresa	40	183	26	2	88.84
Miedo	14	191	23	23	81.67
Tristeza	6	154	53	38	63.75
Disgusto	35	188	17	11	88.84

4.4. Rendimiento de la red en sistemas embebidos

De la etapa anterior se seleccionaron las mejores redes para 5 y 6 emociones y se guardaron y exportaron para comprobar su funcionalidad en distintos sistemas, especificados en la tabla 4.6, con el lenguaje de programación Python. En específico se seleccionaron: una computadora

personal y el sistema embebido Nvidia Jetson Xavier elegida por su amplio poder de procesamiento, compatibilidad con el sistema operativo Linux y una arquitectura de CPU tipo ARM, el cual es de menor potencia que x64, pero con mayor eficiencia energética. Los resultados obtenidos se analizaron y compararon para determinar la viabilidad de implementar este tipo de arquitectura de red en dispositivos con bajos recursos computacionales. Además, se intentó utilizar la red en el dispositivo Raspberry Pi 3B+. Sin embargo, no se pudo llevar a cabo debido a incompatibilidades con los paquetes usados para exportar la red, pues los modelos de red onnx no están adaptados para funcionar con modelos de Raspberry inferiores a la versión 4. Este tipo de formato para redes neuronales se detallará en la siguiente subsección.

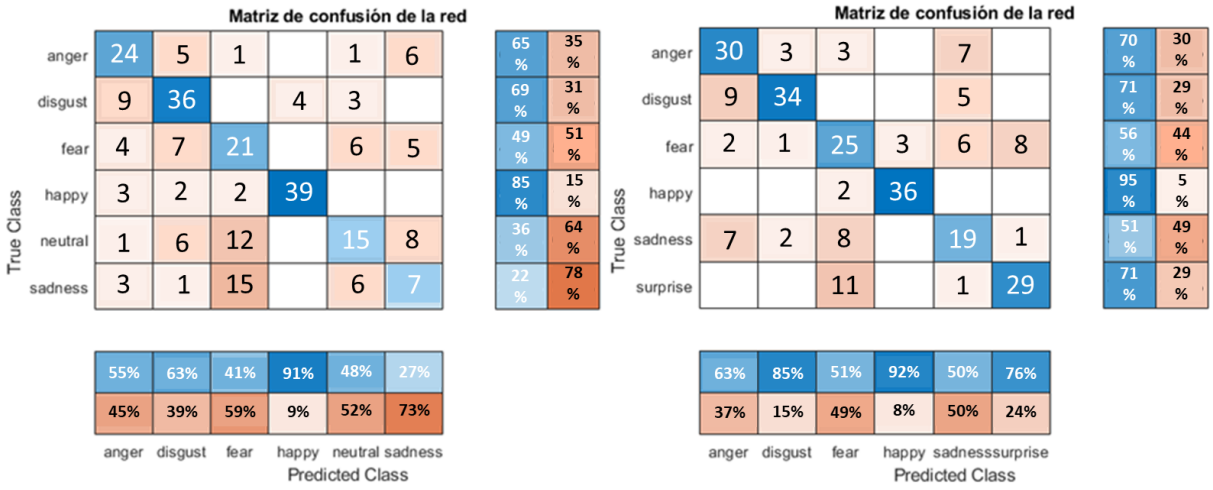

Figura 4.6. Matrices de confusión para el conjunto de pruebas en 6 emociones: enojo, disgusto, miedo, alegría, neutral y tristeza y enojo, disgusto, miedo, alegría, tristeza y sorpresa. Clasificación.

4.5. ONNX

Cada lenguaje de programación maneja tipos de archivos diferentes, lo mismo ocurre en los distintos frameworks utilizados para *Machine Learning* y la manera en la que los modelos entrenados se almacenan. Un ejemplo de esto es Matlab, cuyas arquitecturas de red se guardan bajo el formato ".mat", o Tensorflow, librería de Python, la cual utiliza el formato propio ".pb". Debido a esta problemática es que en septiembre del 2017 las compañías Facebook y Microsoft se unen para crear el formato ONNX, por sus siglas en ingles *Open Neural Network Exchange*, lo que en español se traduce como "Intercambio de Redes Neuronales Abierto".

Como se mencionó en el capítulo anterior, las redes seleccionadas con 5 y 6 emociones se entrenaron en el software de Matlab, por lo que no se podría utilizar en Python de manera normal, por lo que fue necesario hacer uso de la conversión hacia el formato ONNX. Matlab no incluye una función integrada, por lo que se debe instalar un complemento disponible en la página de Matlab. Caso similar en Python, donde se debe descargar el paquete con el gestor de paquetes "pip". Una vez hecha la migración de formatos, es posible utilizar las redes sin necesidad de llamar otras librerías o paquetes.

4.6. Resultados de la red en Python

Gracias a la portabilidad ofrecida en Python, la ejecución en distintas plataformas se realiza de manera rápida y sencilla, por lo que se decidió ejecutar la red en dos plataformas diferentes: una PC y el sistema embebido. Los resultados obtenidos de dichas ejecuciones se utilizaron para realizar comparaciones de desempeño y para comprobar que se comportara de la misma manera en Matlab y en Python. Dichos resultados se mostrarán más adelante.

Las especificaciones de los equipos se encuentran contenidos en la Tabla 4.6, donde se muestran los elementos que cada uno contiene.

Tabla 4.6. Especificaciones principales de los equipos.

	PC	Nvidia Jetson Xavier
CPU	Intel i5 10400f 6-cores	ARM v8.2 8-core
RAM	16 Gb LPDDR4	32Gb LPDDR4*
GPU	Nvidia RTX 3070	Nvidia Volta 512core
VRAM	8 Gb	32Gb LPDDR4*
Tensor Cores	184 Tensor Cores	64 Tensor Cores

- *Memoria compartida.

Los resultados de las redes de 5 y 6 emociones se mantuvieron idénticos para todos los casos, es decir, en Matlab y al utilizar las redes en Python, por lo que se tomó como exitosa la conversión y migración de la red. La métrica que se utilizó para comparar entre los dispositivos

fue el tiempo que requirió para procesar toda la base de datos y la clasificación de todas las imágenes contenidas. Se llevaron a cabo 6 pruebas diferentes, variando el conjunto de entrenamiento y pruebas, sobre tres redes siamesas basadas en la desarrollada en el método 2. Los resultados obtenidos se muestran en las siguientes tablas: Tabla 4.7, Tabla 4.8, la columna "Emociones" indica cuantas y cuales emociones fueron usadas para el entrenamiento de la red. La columna "total" muestra el tiempo que se necesita para realizar una clasificación la primera vez, las siguientes veces que se ejecute el programa, ya no se hará necesario volver a procesar todo el conjunto.

En un caso práctico solo una imagen entraría a la etapa de extracción de características mientras que la base de datos se almacenaría ya procesada, es decir, se procesan y se almacenan las características a la espera de que otra imagen entre al sistema y se compare para clasificar la emoción correspondiente. En este caso, tanto la computadora como la Jetson nano tardaron un tiempo de un segundo aproximadamente.

4.7. Análisis de la red

Para conocer de mejor manera el funcionamiento de la red se realizó un análisis sobre los mapas de activación de las distintas capas convolucionales. Esto nos proporcionó información sobre las características que la red aprendió de las distintas emociones, así como posibles futuras mejoras a la red.

Tabla 4.7. Tiempo de procesado de las redes para todo el conjunto de entrenamiento en distintos equipos.

Equipo	Emociones de entrenamiento	Tiempo para procesar todo el conjunto (s)	Tiempo de clasificación (s)	Total (s)
Computadora	5 -- An, Ha, Ne, Af, Su	5	2	7
	5 -- An, Ha, Ne, Sa, Su	5	2	7
	6 -- An, Ha, Di, Sa, Af, Su	5	3	8
Jetson Xavier	5 -- An, Ha, Ne, Af, Su	21	12	33
	5 -- An, Ha, Ne, Sa, Su	21	12	33
	6 -- An, Ha, Di, Sa, Af, Su	24	18	42

Como se mencionó en al principio de este capítulo, los modelos entrenados se basan en la arquitectura desarrollada en el método 2, la RNAP y esta cuenta con 3 capas convolucionales, cada una con diversa cantidad de filtros. Por motivos de simplificación se decidió agrupar los filtros en una sola imagen, de manera que se facilitara una vista general de los mapas de

activación, los cuales se muestra en la Figura 4.7, correspondientes a la primera capa convolucional ante una imagen de la clase Alegría. La Figura 4.8 es el resultado de introducir una imagen de la clase Sorpresa.

Los mapas de activación nos indican la activación de las neuronas por medio de una representación gráfica, donde mientras más blanco sea el pixel, más fuerte fue su reacción ante la imagen. Destacan la apariencia de filtros que detectan bordes, con un resultado similar al que se obtienen con las funciones Sobel vertical y/u horizontal. Además, se observan grupos de filtros que son los que mayor activación presentan, los cuales parecen identificar detalles como ojos, cejas, pelo, fosas nasales y los labios.

Tabla 4.8. Tiempo de procesado de las redes para el conjunto de pruebas en distintos equipos.

Equipo	Emociones	Tiempo de procesado de imágenes (s)	Tiempo de clasificación (s)	Total (s)
Computadora	An, Ha, Ne, Af, Su	2	1	3
	An, Ha, Ne, Sa, Su	2	1	3
	An, Ha, Di, Sa, Af, Su	3	1	4
Jetson Xavier	An, Ha, Ne, Af, Su	9	2	11
	An, Ha, Ne, Sa, Su	9	2	11
	An, Ha, Di, Sa, Af, Su	11	3	14

En la Figura 4.9 se encuentra el mapa de activación de la segunda capa convolucional para la emoción Alegría, mientras que en la Figura 4.10 se introdujo la emoción Sorpresa.

La cantidad de filtros con baja activación se incrementó con respecto a la primera capa convolucional, ya que se nota un gran grupo de imágenes con colores grises. A pesar de esto, se puede observar cómo diversos filtros van encontrando detalles más finos, como podría ser las pestañas o bordes de los ojos y caras. Es notable la predilección del modelo por enfocarse al área de los ojos y boca, áreas que destacan por ser aquellas que más cambian al gesticular una emoción.

Los resultados de la tercera capa convolucional se refieren a detalles más finos, sin embargo, no es posible distinguir figuras claras debido a la baja densidad de pixeles, producto de la aplicación de la convolución y *Max-Pooling*. Así, la Figura 4.11 muestra el mapa de activación

de los filtros de la tercera capa convolucional para la emoción Alegría, mientras que la Figura 4.12 muestra los resultados de la emoción Sorpresa.

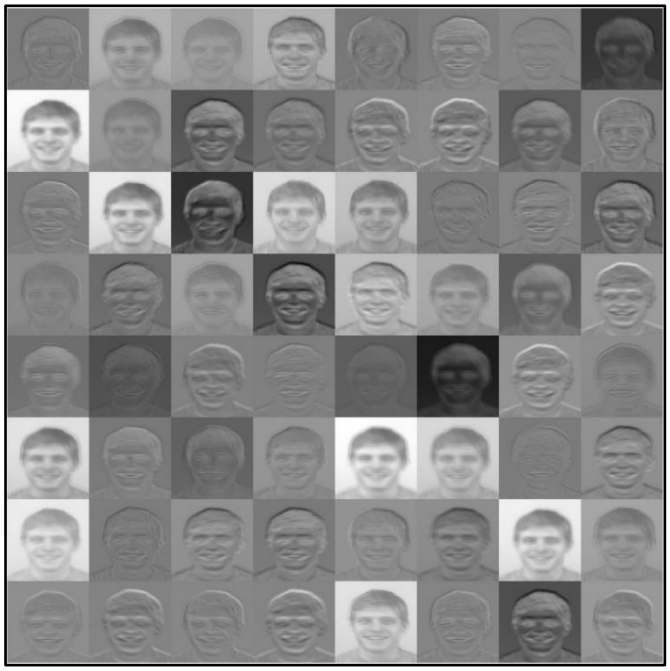

Figura 4.7. Mapa de activación de la primera capa convolucional para la clase Alegría.

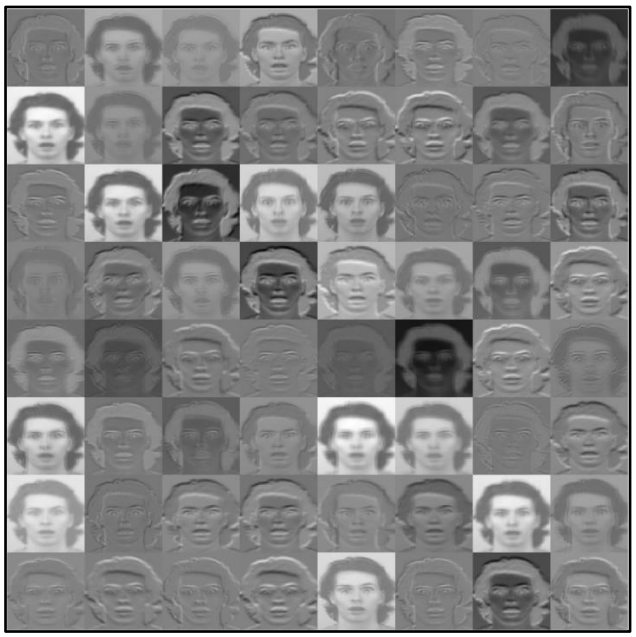

Figura 4.8. Mapa de activación de la primera capa convolucional para la emoción Sorpresa.

Esta última capa convolucional aprendió detalles más abstractos y detalles finos, razón por la cual se vuelve complicado entender que zonas faciales logra detonar las neuronas de esta última etapa de extracción de características. A pesar de esto, en algunos filtros se puede distinguir ligeramente el rostro del individuo, más precisamente los ojos y boca de este.

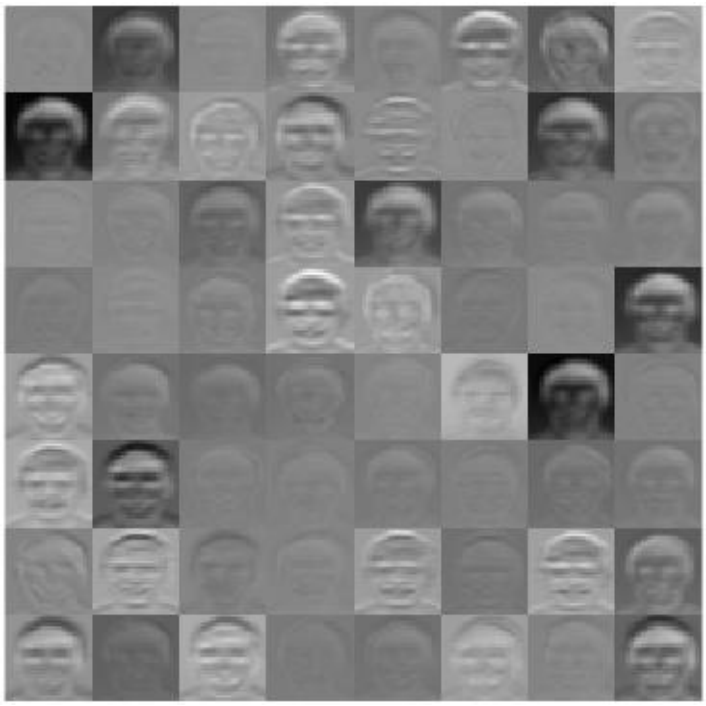

Figura 4.9. Mapa de activación de la segunda capa convolucional para la emoción Alegría.

Resumiendo, en este capítulo se presentaron los resultados de la red elaborada en el método 2 sobre una nueva base de datos, KDEF introducida en el capítulo 3.1.3. Para obtener los mejores resultados se realizaron pruebas con distintos números de emociones hasta obtener la máxima cantidad de clases posibles manteniendo un buen desempeño. De estas pruebas salieron 3 redes diferentes, dos que logran clasificar 5 emociones y una de 6, todas con la misma estructura.

Otro de los puntos a destacar es que se tuvo un mal desempeño al hacer una clasificación binaria de miedo y sorpresa. No obstante, el añadir una emoción para hacer una clasificación ternaria resultaba en un incremento en el desempeño de la red. No en todos los casos existió un

aumento de desempeño al añadir incluir otra emoción, ya que solo pocas redes entrenadas con 5 y 6 emociones lograron resultados superiores al 80% de precisión.

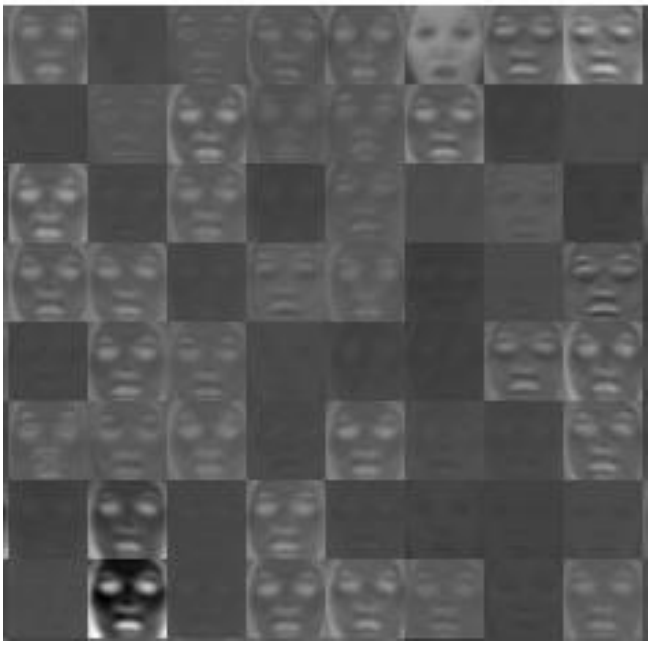

Figura 4.10. Mapa de activación de la segunda capa convolucional para la emoción Sorpresa.

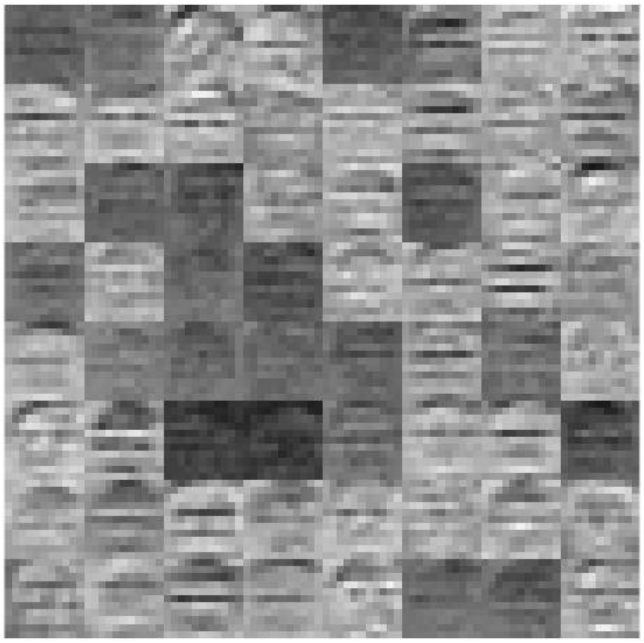

Figura 4.11. Mapa de activación de la tercera capa convolucional para la emoción Alegría.

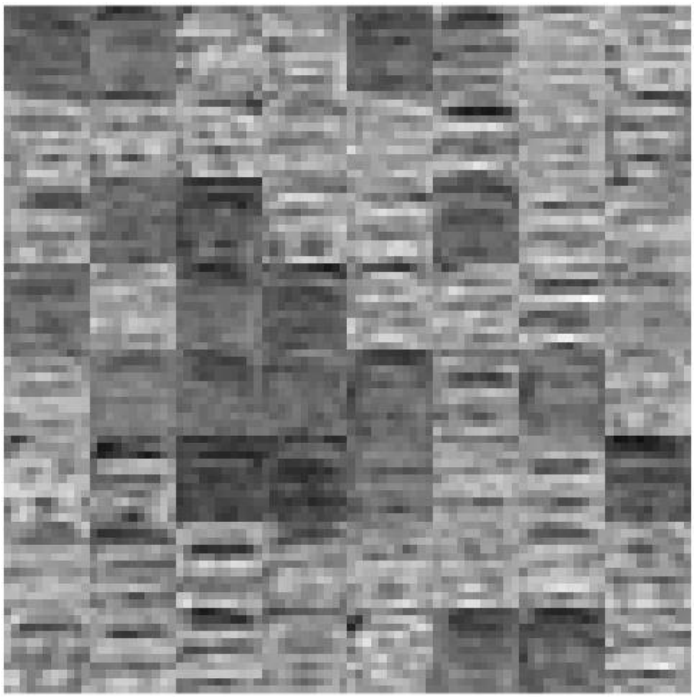

Figura 4.12. Mapa de activación de la tercera capa convolucional para la emoción Sorpresa.

En este capítulo también pudo conocerse el desempeño de la red al migrarla a otra plataforma, siendo el desarrollo original realizado en Matlab y de este exportado al lenguaje de programación Python, el cuál es libre y se puede utilizar en una gran variedad de plataformas, como la Nvidia Jetson Xavier. De esta manera, se pudo comprobar que la red funcionaba igual sin importar la plataforma, lo único que se vería afectado es la velocidad de procesamiento, ya que este se encuentra ligado al hardware del dispositivo.

Finalmente se realizó un análisis de los mapas de activación de la red, permitiéndonos conocer un poco sobre las zonas que activan los filtros de las capas convolucionales, los detalles generales y específicos que se captan y como cambia ante distintas emociones. De esta manera, se pudo observar que las regiones oculares y aquellas cercanas a la boca son las que más activan a las neuronas. También es de destacar la aparición de múltiples filtros cuyo funcionamiento aparente es el de detección de distintos bordes en las imágenes de expresiones faciales.

5. CONCLUSIONES Y TRABAJO A FUTURO.

En esta tesis se propuso el desarrollo de una red de arquitectura siamesa para la detección de emociones implementada en un sistema embebido. Inicialmente, se realizó un análisis sobre teoría emocional y modelos que definen características que diferencian las distintas emociones humanas, métodos usados en la actualidad para la detección de emociones, rendimiento y precisión, la selección en la cantidad de emociones, cuales son más utilizadas y bases de datos enfocadas a la clasificación de emociones. Gracias a esta etapa de análisis, fue posible la generación de un criterio de selección de la información, además de algunos problemas asociados al tratar de diferenciar entre emociones con gestos similares, como es el caso de enojo y desagrado. Se encontró un área de oportunidad al encontrar poca información sobre detección de emociones con redes siamesas, siendo este uno de los motivos por lo que se optó desarrollar el tema de tesis con esa arquitectura de red.

En principio se intentó realizar el entrenamiento de la red final en el lenguaje Python. Sin embargo, se terminó migrando el desarrollo a Matlab debido a problemas que surgieron por el tipo de red, como lo fue que la mayoría de *frameworks* y librerías no cuentan con métodos y funciones que permitan un entrenamiento y análisis profundo, cosa que no ocurría completamente en Matlab. Aun cuando no cuenta con herramientas propias para redes siamesas, es posible adaptar las funciones para obtener información de su funcionamiento.

Como fase preliminar se desarrollaron dos modelos cuya diferencia radicaba en la etapa de extracción de características, siendo el primero de ellos por medio de una obtención manual con los descriptores de HOG y LBP, y el segundo haciendo uso de técnicas de *Deep Learning* con redes convolucionales. Gracias a los resultados de los experimentos de la fase preliminar se encontró que la red profunda presentaba mejores resultados tanto en el conjunto de pruebas como en el de validación. El promedio general de la red convolucional, método 2, fue de 87% y 88%, mientras que el método 1 obtuvo 80% y 71% para pruebas y validación respectivamente, razón por la cual se optó por utilizar como red principal.

Con base en la red desarrollada con el método 2, RNAP, se realizaron tres redes: dos entrenadas con cinco emociones y una con seis. La cantidad se seleccionó al encontrarse que el modelo propuesto tenía como límite las seis emociones para el entrenamiento, ya que con este número se comenzaron a presentar precisiones muy bajas, existiendo únicamente un caso donde hubo resultados medianamente favorables.

Los métodos desarrollados se probaron en un sistema embebido para conocer la viabilidad de desarrollar e implementar futuras redes en este tipo de dispositivos. Los resultados obtenidos fueron iguales a los logrados en Matlab, mientras que el tiempo de procesado de toda la red y obtención de métricas aumentó de 8 a 42 segundos, en el caso de 6 emociones, y de 7 a 33 segundos para 5 emociones. Para estos resultados se debe considerar que se llevó a cabo el procesamiento total de la base de datos, cosa que en la práctica solo se realizará una vez, por lo que el tiempo de procesamiento que verá el usuario final será mucho menor.

Existe varias áreas de oportunidad para un posible trabajo a futuro, entre estos se destaca el desarrollo de una aplicación que funcione en un dispositivo embebido, como un teléfono inteligente, que realice inferencias con la red desarrollada y sea capaz de identificar las emociones del usuario. Otro de las posibles mejoras es una optimización sobre la arquitectura de la red, debido a que, en el análisis de los mapas de activación, se descubrió una cantidad de neuronas cuya excitación era muy baja para todas las emociones, dando a entender que su aporte era bajo. Lo mismo aplica para la salida de la red, donde se descubrió que existían dimensiones del vector de salida que nunca se activaron.

6. REFERENCIAS

[1] Mo, S., Niu, J., Su, Y. and Das, S. *A novel feature set for video emotion recognition*. Neurocomputing, vol. 291, pp.11-20, 2018.

[2] Yang, D., Alsadoon, A., Prasad, P., Singh, A. and Elchouemi, A. *An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment.* Procedia Computer Science, vol. 125, pp.2-10, 2018.

[3] Yan, J., Zheng, W., Cui, Z., Tang, C., Zhang, T. and Zong, Y. *Multi-cue fusion for emotion recognition in the wild.* Neurocomputing, vol. 309, pp.27-35, 2018.

[4] Yu, Z., Liu, G., Liu, Q. and Deng, J. *Spatio-temporal convolutional features with nested LSTM for facial expression recognition.* Neurocomputing, vol. 317, pp.50-57, 2018.

[5] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali and M. Zareapoor, *Hybrid deep neural networks for face emotion recognition*. Pattern Recognition Letters, vol. 115, pp. 101-106, 2018.

[6] R. Favaretto, P. Knob, S. Musse, F. Vilanova and Â. Costa. *Detecting personality and emotion traits in crowds from video sequences*. Machine Vision and Applications, vol. 30, no. 5, pp. 999-1012, 2018.

[7] N. Roopa. *Emotion Recognition from Facial Expression using Deep Learning.* International Journal of Engineering and Advanced Technology, vol. 8, no. 6, pp. 91-95, 2019.

[8] P. Tarnowski, M. Kołodziej, A. Majkowski and R. Rak. *Emotion recognition using facial expressions.* Procedia Computer Science, vol. 108, pp. 1175-1184, 2017.

[9] S. Kumar, M. Bhuyan, B. Lovell and Y. Iwahori. *Hierarchical uncorrelated multiview discriminant locality preserving projection for multiview facial expression recognition.* Journal of Visual Communication and Image Representation, vol. 54, pp. 171-181, 2018.

[10] M. Ali, A. Mosa, F. Machot and K. Kyamakya. *Emotion Recognition Involving Physiological and Speech Signals: A Comprehensive Review*. Studies in Systems, Decision and Control, vol. 109, pp. 287-302, 2017.

[11] K. Yan, W. Zheng, Z. Cui, Y. Zong, T. Zhang and C. Tang. *Unsupervised facial expression recognition using domain adaptation-based dictionary learning approach*, Neurocomputing, vol. 319, pp. 84-91, 2018.

[12] Y. Liu, X. Yuan, X. Gong, Z. Xie, F. Fang and Z. Luo. *Conditional convolution neural network enhanced random forest for facial expression recognition.* Pattern Recognition, vol. 84, pp. 251-261, 2018.

[13] A. Rosebrock. *Deep Learning for Computer Vision with Python Starter Bundle*. PyImageSearch, 1st ed., pp. 67-80, 2017.

[14] C. Marechal et al., *Survey on AI-Based Multimodal Methods for Emotion Detection*, Lecture Notes in Computer Science, vol. 11400, pp. 307-324, 2019.

[15] A. Rosebrock, *Deep Learning for Computer Vision with Python Practicioner Bundle*, PyImageSearch, 1st ed., pp. 141-146, 2017.

[16] D. Derkach and F. Sukno. *Automatic local shape spectrum analysis for 3D facial expression recognition.* Image and Vision Computing, vol. 79, pp. 86-98, 2018.

[17] M. Egger, M. Ley and S. Hanke. *Emotion Recognition from Physiological Signal Analysis: A Review*. Electronic Notes in Theoretical Computer Science, vol. 343, pp. 35-55, 2019.

[18] J. Kwong, F. Garcia, P. Abu and R. Reyes. *Emotion Recognition via Facial Expression: Utilization of Numerous Feature Descriptors in Different Machine Learning Algorithms.* TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 2045-2049, 2018.

[19] M. Ley, M. Egger and S. Hanke. *Evaluating Methods for Emotion Recognition based on Facial and Vocal Features.* The 2019 European Conference on Ambient Intelligence, vol. 2492, pp. 1-7, 2019.

[20] T. Ojala, M. Pietikainen and T. Maenpaa. *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, 2002.

[21] E. Silva Cruz, C. Jung and C. Esparza Franco. *Facial expression recognition using temporal POEM features,* Pattern Recognition Letters, vol. 114, pp. 13-21, 2018.

[22] K. Zidek, J. Pitel and A. Hosovsky. *Machine learning algorithms implementation into embedded systems with web application user interface*, 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES), pp. 77-82, 2017.

[23] G. Bonaccorso, *Machine Learning Algorithms*, 1st ed. Birmingham, UK: Packt Publishing, pp. 6-42, 2017.

[24] S. Raschka and V. Mirjalili. *Python Machine Learning*, 3rd ed. Birmingham: Packt Publishing Ltd., pp. 19-51, 2019.

[25] Z. Ambadar, J. Cohn, and L. Reed. *All smiles are not created equal: Morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous.* Journal of Nonverbal Behavior, vol. 33, pp. 17–34, 2009.

[26] P. Lucey, J. Cohn, T. Kanade, J. Saragih and Z. Ambadar. *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression.* CVPRW, pp. 94-101 2011.

[27] Goodfellow I.J. et al. *Challenges in Representation Learning: A Report on Three Machine Learning Contests. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, vol. 8228, pp. 117-124, 2013.

[28] E.Lundqvist, D., Flykt, A., & Öhman, A. *The Karolinska Directed Emotional Faces - KDEF*, Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, pp. 2-2, 1998.

[29] S. Berlemont, G. Lefebvre, S. Duffner, C. Garcia. *Class-Balanced SiameseNeural Networks*. Neurocomputing, Elsevier, vol. 273, pp. 47-56, 2017.

[30] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Llados, J., and Pal, U., *SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification*, arXiv e-prints, pp. 1-7, 2017.

[31] P. Baldi and Y. Chauvin, *Neural Networks for Fingerprint Recognition*, Neural Computation, vol. 5, pp. 402-418, 1993.

[32] J. Bromley, I. Guyon and Y. LeCun, *Signature verification using a "Siamese" time delay neural network*, NIPS, vol. 6, pp. 737–744, 1994.

[33] Ravichandiran, S., *Hands-On Deep Learning Algorithms with Python.* 1st ed. Birmingham: Packt, pp.437 – 456, 2019.

[34] Roy, S., Harandi, M., Nock, R. and Hartley, R., *Siamese Networks: The Tale of Two Manifolds*. IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3046-3055, 2019.