



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**RECONOCIMIENTO DE EXPRESIONES FACIALES  
DE TRISTEZA UTILIZANDO APRENDIZAJE  
PROFUNDO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor:**

**Bach. Coronel Caján Erick Arturo**

**ORCID: <https://orcid.org/0000-0001-9112-8680>**

**Asesor:**

**Mg. Bravo Ruiz Jaime Arturo**

**ORCID: <https://orcid.org/0000-0003-1929-3969>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú 2021**

**APROBACIÓN DEL JURADO**

**RECONOCIMIENTO DE EXPRESIONES FACIALES DE TRISTEZA  
UTILIZANDO APRENDIZAJE PROFUNDO**

---

**Bach. Coronel Caján Erick Arturo**  
**Autor**

---

**Mg. Bravo Ruiz Jaime Arturo**  
**Asesor**

---

**Dr. Ramos Moscol Mario Fernando**  
**Presidente de Jurado**

---

**Ing. Sialer Rivera Noelia**  
**Secretario de Jurado**

---

**Mg. Bravo Ruiz Jaime Arturo**  
**Vocal de Jurado**

## **Dedicatorias.**

Lleno de regocijo, amor y esperanza, dedico este proyecto a cada uno de mis seres queridos, quienes han sido y son, piezas fundamentales en mi camino hacia la superación. A mi madre Nancy Esther Caján Alcántara, porque ella es la motivación más grande que tengo, quiero que sienta orgullo y sepa que su esfuerzo, esmero y dedicación valió la pena, es por ello que es una inmensa satisfacción haber culminado esta investigación. A mis hermanos Carlos y Aldo, por confiar siempre en mí y por contagiarme su firmeza. Finalmente, en homenaje a toda mi familia que demostró su cariño y apoyo, en sus diferentes acciones: abuelos, tíos y primos.

## **Agradecimientos.**

A Dios, porque sin Él no somos nada en esta vida. Uno puede planear hacer muchas cosas, pero Él siempre tendrá la última palabra. Agradezco, porque durante todo este tiempo transcurrido me brindó fuerza, vida, salud y una familia que me apoya en todo momento. Asimismo, quiero nombrar a los diferentes docentes que me motivaron a seguir investigando y que sus recomendaciones siempre las tomé en cuenta para ser mejor, y así lograr presentar un trabajo de calidad.

## Resumen.

En la actualidad, el presente tema de investigación, “El reconocimiento de expresiones faciales”, está copando muchos ámbitos de la ciencia. Puesto que, se está aplicando para mejorar muchos tipos de campos. En mención a ello, tenemos ejemplos como: la predicción de actividades emocionales del ser humano a partir de los comportamientos que expresa el mismo, predicción de relaciones interpersonales, en medicina con los tratamientos médicos, en juegos online con la realidad virtual y aumentada. Si ahondamos un poco más, el vocablo “expresiones-emociones humanas”, significa desde una vista general el modo universal para interactuar. Por ello, se precisa que, la identificación automática de expresiones faciales humanas tiene sus propias ventajas. El meollo puntualmente, en este tema tan interesante, radica en el rostro, ya que, por su naturaleza conlleva una estructura compleja en términos de formas, variación de redondez, género, edad, tipo de raza o etnia, y claro, las simetrías. Empero, con la llegada de la nueva tecnología y los avances del Deep Learning, es posible el reconocimiento facial y la clasificación de imágenes. En razón a ello, se tiene que delimitar que, existen algoritmos y métodos para detectar y reconocer rostros.

Ante el problema suscitado, se plantea reutilizar una Red Neuronal Convolutiva configurando ciertos parámetros, para que sea capaz de identificar y detectar expresiones de tristeza en un rostro. Por tal motivo, se usará el dataset FER2013 de la plataforma Kaggle. Luego, se empleará la librería Face Recognition para el reconocimiento facial. En alusión a lo anterior, indicar que, el sistema estará basado en el lenguaje de programación Python, lo cual nos será útil para trabajar de la mano con la herramienta OpenCV, que sería algo clave en este proceso. El diseño es, reconocer las expresiones faciales de tristeza de manera eficiente, y optimizando el tiempo de entrenamiento.

**Palabras Clave:** Reconocimiento facial, Expresiones faciales, Aprendizaje profundo.

## **Abstract.**

At present, the current research topic, "Recognition of facial expressions", is taking over many areas of science. Since, it is being applied to improve many types of fields. In reference to this, we have examples such as: the prediction of emotional activities of the human being from the behaviors expressed by the same, prediction of interpersonal relationships, in medicine with medical treatments, in online games with virtual and augmented reality. If we delve a little more, the word "human expressions-emotions" means from a general point of view the universal way to interact. Therefore, it is specified that the automatic identification of human facial expressions has its own advantages. Specifically, the crux of this interesting topic lies in the face, since, by its nature, it involves a complex structure in terms of shapes, variation in roundness, gender, age, type of race or ethnicity, and of course, symmetries. However, with the arrival of new technology and advances in Deep Learning, facial recognition and image classification is possible. Due to this, it must be defined that there are algorithms and methods to detect and recognize faces.

Given the problem raised, it is proposed to reuse a Convolutional Neural Network configuring certain parameters, so that it is able to identify and detect expressions of sadness on a face. For this reason, the FER2013 dataset from the Kaggle platform will be used. Then, the Face Recognition library will be used for facial recognition. In allusion to the above, indicate that the system will be based on the Python programming language, which will be useful to work hand in hand with the OpenCV tool, which would be key in this process. The design is to recognize the facial expressions of sadness efficiently, and optimizing the training time.

**Keywords:** Facial recognition, Facial expressions, Deep learning.

## ÍNDICE

<b>I. INTRODUCCIÓN</b> .....	<b>9</b>
1.1. Realidad Problemática. ....	9
1.2. Trabajos previos. ....	10
1.3. Teorías relacionadas al tema.....	24
1.4. Formulación del Problema. ....	34
1.5. Justificación e importancia del estudio.....	34
1.6. Hipótesis. ....	34
1.7. Objetivos. ....	34
1.7.1. Objetivo general.....	34
1.7.2. Objetivos específicos.....	35
<b>II. MATERIAL Y MÉTODO</b> .....	<b>35</b>
2.1. Tipo y Diseño de Investigación.....	35
2.2. Población y muestra. ....	35
2.3. Variables, Operacionalización.....	37
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad. ....	38
2.5. Procedimiento de análisis de datos. ....	38
2.6. Criterios éticos. ....	41
2.7. Criterios de Rigor Científico. ....	42
<b>III. RESULTADOS.</b> .....	<b>43</b>
3.1. Resultados en Tablas y Figuras .....	43
3.2. Discusión de resultados. ....	53
3.3. Aporte práctico.....	54
<b>IV. CONCLUSIONES Y RECOMENDACIONES.</b> .....	<b>69</b>
4.1. Conclusiones. ....	69
4.2. Recomendaciones. ....	70

<b>ANEXOS. ....</b>	<b>75</b>
<b>ANEXO 01: Resolución de Aprobación de Proyecto de Investigación.....</b>	<b>75</b>
<b>ANEXO 02: Formatos de Instrumentos de recolección de datos del Proyecto de Investigación.....</b>	<b>79</b>
<b>ANEXO 03: Codificación del sistema de detección facial utilizando red CNN. ....</b>	<b>80</b>



## I. INTRODUCCIÓN

### 1.1. Realidad Problemática.

(Andrade, 2019) explica que, en el último siglo, el beneficio en perfeccionar todos los semblantes de la interacción entre individuos y computadoras viene en incremento. Muchos estudios para distinguir impresiones faciales han favorecido la creación de medios para esta interacción. Principalmente en el sitio del reconocimiento de emociones humanas al prestar atención en expresiones faciales.

Es por ello que, el rol del reconocimiento maquinal de emociones vive progresando de manera progresiva, debido a la calidad que posee la fuerza de la computadora frente al beneficiario en la interacción persona-computadora, por la manera de interactuar de manera original, parecida a como se interactúa con otros individuos. (González, 2014)

(Ekman & L. Rosenberg, 2012) argumenta que, el rostro es uno de los más ricos canales de expresión en los seres humanos, el cual comunica emociones y señales sociales. Refiere que, viven 6 términos faciales prototípicos primordiales registrados generalmente. Estas son: furia, asco, duda, satisfacción, angustia y asombro. El reconocimiento atomizado de estos seis etapas fundamentales es el primer camino para realizar una salida, que, a oposición de otras, es cosmopolita, ya que es frecuente a todos los individuos.

A pesar que, en la actualidad los sistemas de reconocimiento facial tienen altos porcentajes de reconocimiento, no se ha logrado obtener un 100% de efectividad. (Moreano, 2021), indica que, esto se debe a las múltiples variables y la complejidad del rostro humano. Algunas de estas variables complejas de manejar son: pose, iluminación, y expresión facial. Estas variables son características de un sistema de reconocimiento facial que trabaje en ambientes no cooperativos.

A pesar de los trabajos realizados en estas áreas, no se ha logrado concluir satisfactoriamente acerca del proceso de extracción de características faciales. Sin embargo, algunos estudios han mostrado nuevos caminos acerca de cómo es el proceso de extracción de características. (Alonso Sierra & Castaño Saavedra , 2019)

En base a lo mencionado, se puede aseverar que, uno de los problemas al momento de reconocer expresiones faciales es la elección de un algoritmo de reconocimiento debido que las características de una imagen pueden variar, en iluminación y calidad.

## **1.2. Trabajos previos.**

Yu & Yin, (2018), en su estudio, “The Study of Facial Gesture Based on Facial Features”, en China. Nos indica que, en la actualidad, la totalidad de las indagaciones sobre gestos se agrupan en los gestos con el bolígrafo y con la mano, pero son menos frecuentes los gestos faciales. Basándose en ello, el problema radica en que el reconocimiento facial actual se utiliza principalmente para la identificación, pero interviene menos en la interacción. Por esta razón, los autores diseñaron un sistema con enfoque en los gestos faciales, los cuales a su vez se basan en los rasgos faciales para interactuar con el ordenador, liberando así la carga de ambas manos. Al mismo tiempo, es de gran importancia para que las personas discapacitadas interactúen con el ordenador. A continuación, se expone a dato el método que se llevó a cabo: La aplicación de gestos faciales basada en rasgos faciales que se propone en esta investigación se basa en la Arquitectura C/S. El sistema cuenta con 3 secciones: a) El módulo de adquisición de información facial, en el cual se utilizó la cámara ANC Core Duo (Versión 1080 HD). b) El análisis de la detección facial, el cual se realizó en el lado del servidor Face++. Aquí la información devuelta por el servidor se basó en reglas estipuladas y algoritmos de definición de gestos faciales. c) El módulo de interacción, donde se generó el feedback final. Para realizar el primer punto, se configuró las variables de entorno para java y JMF para que la cámara pueda comunicarse con el entorno JAVA sin dificultad.

Luego de ello, se escribió el código del núcleo del gesto facial. Se estableció que el valor mínimo de la apertura de ojos sea 30 y el valor mínimo del cierre de ojos sea 10. El estado de la boca cerrada es 0 y el de la boca abierta es igual a 0. Por último, se ejecutó un análisis y tratamiento de los factores de error, ya que las personas suelen parpadear y abrir la boca de forma inconsciente. Se añadió un parámetro "count", que cuenta el número de veces que el sistema detecta cada gesto fácil e hizo uso del mismo para controlar si el comando se ejecuta o no tras el reconocimiento del gesto facial. Los efectos que se lograron en la actual pesquisa, van relacionados a las pruebas de la tasa de éxito que se realizaron para diferentes tipos de gestos faciales. Se diseñaron 20 casos de prueba, con una tasa de éxito media del 90%. Según la Ficha Técnica del Experimento de Expresión Facial, basados en los siguientes ítems: Gesto Facial, Frecuencia de las Pruebas, Tiempos de Éxito, Tiempos de Fallos y Tasa de Éxito respectivamente, los resultados son: 1) Ojo izquierdo abierto, ojo derecho cerrado y boca, 20, 19, 1, 95%. 2) Ojo izquierdo cerrado cerrado, ojo derecho abierto y boca, 20, 16, 4, 80%. 3) Ojo derecho cerrado, ojo izquierdo abierto, boca abierta, 2, 17, 3, 85%. 4) Ojo izquierdo abierto, ojo derecho abierto y boca, 20, 20, 0, 100%. 5) Vista izquierdo cerrado, vista derecho cerrado y boca, 20, 19, 1, 95%. 6) Vista izquierdo cerrado vista derecho cerrado y boca, 20, 18, 2, 90%. 7) Ojo izquierdo cerrado, ojo derecho abierto y boca, 20, 17, 3, 85%. En base a todo lo revisado, se concluye que, después de analizar la prueba funcional del sistema y el índice de éxito de los gestos faciales, se puede ver en los datos que el sistema de interacción de reconocimiento de gestos faciales cumple básicamente las funciones para las que fue diseñado, satisfaciendo así las necesidades de la interacción natural entre humanos y máquinas. La tasa de éxito de los gestos faciales puede alcanzar el 90%. Sin embargo, según los resultados de la experiencia, se puede saber que la tasa de reconocimiento de gestos faciales tiene una cierta influencia en la distancia entre la luz y la cámara, y necesita más mejoras. Al mismo tiempo, se enriquece la interacción gestual tridimensional.

Tümen, Faruk Söylemez & Ergen, (2017), realizaron la investigación, "Facial emotion recognition on a dataset using convolutional neural network", en

Turquía. En este estudio, el principal problema, radica en el conjunto de datos, puesto que hay 7 tipos de imágenes que constan de diferentes expresiones faciales, así como información de clase de todos los datos, y el proceso de prueba se realizó de acuerdo con estos valores. Lo que requiere de un computador potente para dar soporte a la ejecución de las 35.887 imágenes. En base a lo mencionado, fue que ellos desarrollaron un método de reconocimiento de expresiones faciales establecido en Puntos neuronales convolucionales (CNN) para clasificar automáticamente las expresiones faciales en el conjunto de datos FER2013. Por consiguiente, se explica el paso a paso del método de solución empleado: a) Primero, se seleccionó el conjunto de datos con el cual se va trabajar, en esta ocasión dicho dataset se denomina FER2013. b) Segundo, se realizó una función de extracción de características haciendo uso de una red CNN. La cual incluye la multiplicación de características y capas hash. Además, se utilizó una capa softmax. Con este método, se puede obtener un valor de activación de función diferente en cada posición de la imagen mediante la conversión con imágenes más grandes. c) Posterior a ello, se clasificó las propiedades obtenidas luego de utilizar la capa de convolución. Lo cual arrojó que según dimensión y muestra, crea un vector de características. d) Así pues, se escogió una capa de función de activación, ya que con la función ReLU se obtuvo un resultado en menor tiempo. Mencionar que, para no tener inconvenientes en el experimento, los investigadores ajustaron una tasa de aprendizaje adecuada al caso. e) Finalmente, se utilizó el método de regresión Softmax, el cual sirvió para la clasificación de datos altamente restringidos. En esta pesquisa, hay datos etiquetados para 7 clases en total. Se utilizó dicho clasificador para la última capa. Según el número de imágenes y tasa de éxito utilizados para el conjunto de datos de prueba (10%), se obtuvieron los siguientes resultados, basados en: Expresión Facial, Número de imágenes, Proporción de datos propios, y Tasa de Éxito. 1) Ira, 453, 0.091, 48.1%. 2) Disgusto, 28, 0.051, 82.1%. 3) Miedo, 423, 0.082, 47.5%. 4) Felicidad, 951, 0.105, 71.1%. 5) Tristeza, 701, 0.115, 46.2%. 6) Sorpresa, 367, 0.091, 80.7%. 7) Neutral, 666, 0.107, 46.7%. En la última etapa, CNN logró un 58,5% en los datos de verificación y un 57,1% en los datos de prueba. Teniendo en cuenta la tasa de éxito en humanos de  $65 \pm 5\%$  en este conjunto de datos, se ha obtenido

un éxito de clasificación adecuado. En conclusión, se observó que el método CNN desarrollado en el estudio arrojó buenos resultados en la detección de expresiones faciales sin ningún pretratamiento. Se ha logrado un éxito particularmente alto en las expresiones faciales de felicidad, inteligencia y disgusto. Cuando se examinaron todos estos logros, se obtuvo una tasa de éxito del 57,1% probando datos del 10%.

Liu, Hsu, Wang & Chiang, (2019), realizaron la investigación, “Real-Time Facial Expression Recognition Based on CNN”, en Vietnam. En esta indagación, el problema existente, fue que, cuando la cámara capturó imágenes a alta velocidad, se produjeron cambios en las características de la imagen en determinados momentos debido a la influencia de la luz y otros factores. Tales cambios podrían resultar en un reconocimiento incorrecto de la expresión facial humana. El método propuesto no utilizó la salida inmediata como referencia, sino que se refirió a la imagen anterior para promediar y facilitar el reconocimiento. De este modo, fueron capaces de reducir la interferencia por las características de las imágenes. El detalle del método, a continuación. Se dividió en tres pasos: a) Obtener e ingresar las imágenes a la arquitectura CNN para su reconocimiento. b) Obtener los resultados del reconocimiento de expresiones faciales. c) Promediar el resultado del reconocimiento con la salida anterior y volver a producirla. De esta manera, se puede lograr el reconocimiento en tiempo real mediante la integración de la arquitectura CNN y la cámara web. Se trabajó con el dataset FER2013 y se usaron los datos de 7 expresiones de dicho conjunto de datos. Para la detección de rostros se utilizó una cámara web Logitech C920, con el programa de captura de rostros Open CV y los parámetros del conjunto de entrenamiento facial haarcascade\_frontalface\_alt.xml. Se programó en Python en la plataforma Anaconda para lograr una tasa de detección de aproximadamente 10 imágenes por segundo. La arquitectura se divide en una capa de entrada, tres capas de convolución y agrupación, y capas completamente conectadas para obtener resultados de predicción de 7 expresiones. En este estudio, el resultado final se deriva del promedio del resultado del reconocimiento actual y el resultado del reconocimiento anterior porque el sistema considera tanto la arquitectura CNN

como la influencia del entorno de la imagen capturada. Finalmente, descrito el procedimiento desarrollado, agregar que, el empleo de una cámara web ayudó a lograr un método de reconocimiento de expresiones faciales en tiempo real eficaz. Los resultados experimentales muestran que después de tomar el promedio del resultado del reconocimiento actual y el anterior resultado del reconocimiento, las salidas demuestran resultados de reconocimiento más robustos y precisos gracias al método propuesto. Sin utilizar el método propuesto, las expresiones faciales durante un período de tiempo pueden generar resultados diferentes. Los histogramas acumulativos de expresiones para Neutral serían: Sin M.P (Enojado 3, Disgustado 0, Asustado 0, Feliz 1, Triste 1, Sorpresa 1, Neutro 88). Con M.P (Enojado 0, Disgustado 0, Asustado 0, Feliz 0, Triste 0, Sorprendido 0, Neutro 94). Realizando un resumen de todo lo tratado en este artículo, se propuso un método de ponderación promedio para evitar errores potenciales en el reconocimiento de expresiones faciales en tiempo real basado en la red neuronal convolucional tradicional. Con una cámara con altas velocidades de fotogramas, se redujo la influencia del ruido del entorno. Además, debido al método de ponderación promedio, los resultados del reconocimiento de expresiones faciales se volvieron más robustos de un cuadro a otro. Se mejoró la precisión del reconocimiento de expresiones faciales. Los efectos empíricos han confirmado que el sistema de reconocimiento de expresiones faciales propuesto es más confiable que el enfoque tradicional de CNN.

Yang, Zhang, Chen & Khan, (2019), realizaron la investigación, "Facial Expression Recognition Based on Facial Action Unit", en EE. UU. Aunque las unidades de acción pueden expresar con precisión las expresiones faciales, rara vez son utilizadas en el reconocimiento de expresiones faciales debido a la dificultad de posicionamiento preciso. El problema radica, en que la auto oclusión de la postura facial afectará seriamente la precisión del reconocimiento de expresiones faciales. Se propuso un método de reconocimiento de expresiones faciales de alto rendimiento basado en la unidad de acción facial. Se dividió principalmente en dos partes. En la primera parte, se obtuvieron 68 puntos de referencia faciales e histogramas de imágenes de gradientes

orientados (HOG), y los valores de las características de las unidades de acción se calculan en consecuencia. La segunda parte utilizó tres métodos de clasificación para realizar el mapeo de AU a FER. A continuación, se explica a detalle cómo se trabajó: Primero se realizó la extracción características AU. Utilizando 2 tipos principales de características: características de apariencia y características geométricas. Para las características de apariencia, se obtuvieron de los histogramas de gradientes. Mientras que las geométricas, se basaron en los resultados de las coordenadas de detección y seguimiento de puntos de referencia faciales. Se utilizó el modelo CE-CLM para marcar 68 puntos de referencia en la cara, el cual refleja los cambios de expresión de varias partes de la cara (ojos, cejas, boca, nariz, etc) Al extraer histogramas de gradientes orientados (HOG), se manejó el método de análisis de dispositivos importantes para disminuir la extensión del vector de características HOG, porque se necesitó reducir la dimensión para solucionar el inconveniente del estudio de expresión. Luego de ello, se usó la regresión de vectores de soporte para estimar la intensidad de las UA. Entre ellos, usamos un kernel lineal para mejorar la velocidad de entrenamiento, porque estaban interesados en métodos de detección en tiempo real. Como penúltimo paso, se efectuó un proceso de clasificación con los valores de AU. Se llevó a cabo la normalización mínima-máxima, para evitar que el valor característico sea demasiado grande o demasiado pequeño. Por consiguiente, se cuadró los resultados para hacer más sobresalientes las características de la expresión correspondiente, reducir influencias de otras características irrelevantes y otorgar una clasificación más precisa. Finalmente, se usaron 3 métodos de clasificación para lograr el mapeo de AUs a ocho expresiones. Los cuales son: SVM, Xg-Boost y DNN. Agregar que se ejecutó una validación cruzada 10 veces para método, buscando así la comparación de resultados para verificar el efecto. Para SVM, se usó un kernel lineal para la función kernel, porque puede mejorar efectivamente la eficiencia computacional y tiene poco impacto en los resultados de la clasificación. Se eligió el modelo gbtree como entrada de parámetro de refuerzo, que utiliza el modelo basado en árbol para el cálculo de refuerzo en Xgboost. Se utilizó la red neuronal profunda para construir un clasificador simple para la clasificación, y se diseñó una red neuronal de tres capas, cuyas capas de entrada son

características AU, la capa de salida son ocho expresiones básicas. Para la evaluación experimental, se comparó los resultados de tres métodos de clasificación diferentes en dos conjuntos de datos. Para el conjunto de prueba, se usó alrededor de 40 imágenes separadas para cada expresión (sin entrenamiento), totalizando 337 imágenes de prueba en el conjunto de datos CK + y un total de 240 imágenes de prueba en el conjunto de datos Oulu CASIA. Puede verse claramente que para el conjunto de datos CK + los métodos desarrollados lograron mejores resultados, especialmente el clasificador Xgboost con mayor precisión. Y para el conjunto de datos Oulu CASIA, la precisión promedio de los tres métodos es relativamente baja. En conclusión, se realizó el mapeo de la unidad de acción facial al reconocimiento de expresiones faciales. En primer lugar, la detección de hitos faciales y los histogramas de gradientes orientados se utilizaron para adquirir unidades de acción facial de acuerdo con la apariencia facial y las características geométricas, y luego se utilizaron diferentes métodos de clasificación para entrenar y clasificar los resultados anteriores, correspondientes a ocho tipos de expresiones.

De & Saha, (2015), realizaron la investigación, “A comparative study on different approaches of real time human emotion recognition based on facial expression detection”, en India. Aunque, el reconocimiento de emociones tiene una amplia literatura, todavía se considera un problema complicado para la investigación por las siguientes razones, como que la naturaleza y el entorno de las personas no son los mismos en todo el planeta. Las personas, a veces tienden a expresar un sentimiento que está en su mente pero que no siempre se pueden expresar a través de expresiones físicas de la misma manera que lo hacen los demás, lo que da lugar a estados emocionales mixtos y complejos que dificultan el reconocimiento de un sistema de reconocimiento de expresiones. Al ser su naturaleza, un estudio comparativo. Lo que se hizo fue evaluar distintos métodos de solución, y realizar una revisión de literatura sobre los distintos enfoques del reconocimiento de las expresiones humanas en tiempo real. Aquí no existe un único método sino varios, puesto que los autores son distintos en varios casos. *A continuación, según autor se explicará su método:*



(Srivastava, 2012), propuso un método de reconocimiento de expresiones faciales en periodo existente a partir de una entrada de video en vivo por una cámara web que enfrenta desafíos como: el diseño independiente de la persona, la posesión de las propiedades Light Invariant y Pose Invariant. Los pasos fueron los siguientes: a) Efectuar una extracción de marcos, principalmente para secuenciar el video de entrada de la cámara web en cuadros separados. b) Detectar los rostros, mediante los clasificadores Haar y Haar intercalado para procesar cada cuadro. Luego, se utilizó el algoritmo de detección de piel para detectar la parte de la piel de la cara para su posterior procesamiento. c) Ejecutar la extracción de puntos característicos, los cuales refieren a los puntos de la cara. d) Finalmente, clasificar las emociones haciendo uso del algoritmo SVM. Para clasificar emociones en clases como: Neutral, Felicidad, Ira y Emoción.

(Peng, Zhu, & Zhou, 2010), propusieron un método de reconocimiento de expresiones faciales en tiempo real basado en la detección de bordes adaptativa del operador canny. Los pasos fueron los siguientes: a) Realizar una captura de video, la cual es necesaria para secuenciar el video de entrada en secuencias de imágenes individuales. b) Efectuar un preprocesamiento de imágenes. c) Detectar el rostro para la prueba. d) Ejecutar la extracción de rasgos faciales. e) Por consiguiente, ejecutar también la extracción de características de expresión. f) Finalmente, clasificar y reconocer la expresión, siendo capaz de suprimir el ruido.

(M. Chawan, Manan, & Jinal, 2013), propusieron la implementación de una aplicación de reconocimiento de expresiones faciales y extracción de rasgos faciales en tiempo real en la que la cara frontal del sujeto se detecta automáticamente a partir de la imagen de entrada y se clasifica con respecto a siete emociones en tiempo real que son neutrales, ira, asco, miedo, alegría, tristeza y sorpresa. Los pasos fueron los siguientes: a) Transmisión en vivo. b) Obtención del marco de la imagen de entrada. c) Segmentación del color de piel del marco de la imagen de entrada. d) Detección del rostro. e) Detección de

ojos. f) Detección de labios. g) Usar el algoritmo de curva de Bezier en la región de ojos y labios detectada. h) Obtención del patrón binario cuando se seleccione una parte de la cara. i) Comparación y entrenamiento para reconocer la emoción especificada. j) Mostrar pantalla de salida según la emoción.

*Littlewort et al. (2011)*, plantearon diseñar una herramienta de software para el reconocimiento de expresiones faciales en tiempo real totalmente automático. Puede codificar automáticamente la intensidad de 19 acciones faciales y 6 expresiones faciales prototípicas diferentes. Los pasos fueron los siguientes: a) Detección de rostro. b) Detección de rasgos faciales. c) Registro facial. d) Extracción de características de Gabor. e) Clasificación SVM para estimar la Unidad de Acción. f) Reconocimiento de expresiones.

(He Li, Buenaposada, & Baumela, 2009), expusieron la idea de modelar un procedimiento de reconocimiento de expresiones faciales independiente del usuario en tiempo real que reconoce varias expresiones faciales a 30 fps. Los pasos fueron los siguientes: a) Alineación facial, se rastrea el rostro. b) Modelado facial, se aplican cambios en iluminación. c) Reconocimiento de expresiones faciales, el cual consta de subpasos: Reducción de dimensionalidad y Clasificación secuencial.

Yolcu et al. (2017), realizaron la investigación, “Deep learning-based facial expression recognition for monitoring neurological disorders”, en EE.UU. Aquí los autores encontraron que los problemas de reconocimiento de patrones cada vez son más complejos, por lo que se requieren Arquitecturas CNN más profundas para manejar imágenes de mayor tamaño. Ellos propusieron crear una nueva arquitectura, la cual primero segmenta los componentes faciales que son importantes para el reconocimiento de expresiones faciales y forma una imagen iconizada; posterior a ello, realiza la clasificación de la expresión facial utilizando la imagen de componentes faciales iconizados obtenida combinada con las imágenes faciales sin procesar. Este enfoque integra características locales basadas en partes con información facial holística para un reconocimiento robusto de expresiones faciales. Desde un aspecto granular, a

continuación, se argumenta el método propuesto: Para ejecutar la solución planteada se efectuaron 2 estructuras. La primera es una CNN en cascada, la cual sirvió para la segmentación guiada de reconocimiento de expresiones faciales. Para ello se valieron de 3 componentes faciales importantes: boca, cejas y ojos. La segmentación de los componentes faciales se formula como un problema de clasificación binaria, donde los bloques de  $16 \times 16$  de la imagen se clasifican como componente facial versus fondo. Las imágenes originales sin procesar y las máscaras de entrenamiento correspondientes se dividen en bloques no superpuestos de  $16 \times 16$ . A los bloques que tienen el 80% o más de su área cubierta por el componente facial o por clases de fondo se les asigna la clase mayoritaria y se mantienen para entrenamiento. El proceso general da como resultado una versión binaria iconizada de la cara de entrada. Esta primera estructura de CNN consta de siete capas, cuatro capas convolucionales (una capa con 16  $5 \times 5 \times 3$ , una capa con 16  $5 \times 5$ , una capa con 32  $5 \times 5$  y una capa con 32 filtros  $4 \times 4$ ), dos capas agrupadas y una capa completamente conectada. La segunda estructura de CNN en la cascada está entrenada para reconocer la expresión facial (feliz sorprendido, etc.), dada la imagen facial en bruto y la imagen iconizada correspondiente obtenida de la primera CNN. La segunda estructura de CNN incluye diez capas, cinco capas convolucionales (una capa con 64  $5 \times 5 \times 3$ , dos capas con 32  $5 \times 5$ , una capa con 64  $5 \times 5$  y una capa con 64 filtros  $4 \times 4$ ), cuatro capas agrupadas y una capa completamente conectada. ReLU se ha utilizado como una función de activación para ambos. Los pasos se usaron fueron 3. 1) Análisis de imágenes guiado: la cascada de CNN permite el entrenamiento guiado y desacoplado forzando las capas de la arquitectura para aprender a detectar y localizar los componentes faciales que se sabe que son importantes para el reconocimiento de expresiones faciales. 2) Fusión de información holística y basada en partes: la cascada de CNN permite la fusión de información basada en partes (primera CNN) y holística (segunda CNN) para mejorar el reconocimiento de expresiones faciales. 3) Desidentificación y privacidad del paciente: cuando las expresiones faciales u otras características morfodinámicas de la cara se utilizan con fines de diagnóstico médico o para monitorear los resultados del tratamiento o la progresión de la enfermedad, es fundamental des identificar la información de salud protegida. Los resultados

fueron, que usa solo la imagen de referencia binaria iconizada (CNN en cascada) supera al reconocimiento que usa la imagen de rostro sin procesar de 3 canales (CNN simple) en aproximadamente un 3% (89,99% frente a 86,86%). La combinación de una imagen facial sin procesar con una imagen de referencia iconizada (cascada de CNN) supera al reconocimiento mediante la imagen de rostro sin procesar de 3 canales (CNN único) en un 6,57% (93,43% frente a 86,86%). Se ve una matriz de desorden detallada para el reconocimiento de expresiones faciales utilizando una entrada de 4 canales utilizando la arquitectura CNN en cascada propuesta. El sistema propuesto detecta enfado con 92,93%, disgusto con 98,99%, miedo con 87,88%, feliz con 92,93%, tristeza con 88,89% y sorprende con una precisión del 98,99%. En virtud a lo expuesto, se concluyó que, las aplicaciones médicas requieren un análisis de imágenes de alta precisión. El sistema propuesto combina el conocimiento experto en el reconocimiento de expresiones faciales (importancia de los componentes faciales) con características complejas aprendidas del aprendizaje profundo para mejorar el rendimiento. Estos resultados experimentales muestran que la clasificación guiada por componentes faciales a través de la cascada propuesta de CNN logra mejorar la precisión del reconocimiento de expresiones faciales. La identificación automatizada de expresiones faciales puede revelar características de expresión facial clínicamente relevantes que pueden distinguir entre las condiciones de la enfermedad y servir como biomarcadores específicos de la enfermedad para ayudar en el diagnóstico clínico y para monitorear la respuesta terapéutica en pacientes con condiciones neurológicas.

Álvarez et al. (2018), realizaron el estudio, "Facial Emotion Recognition: A Comparison of Different Landmark-Based Classifiers", en El Salvador. Indicaron que el gran inconveniente relacionado al reconocimiento facial es que requiere un entrenamiento extenso y consume mucho tiempo. Como parte del método, se trajeron 2 métodos diferentes para estudiar los datos: (1) la información de los 68 puntos, usando la posición en "x" y "y". El segundo proceso fue, las distancias entre esos puntos, en "x" y "y", con retrato de la misma ente, pero con expresión neutra. En este apartado, a detalle se explicará dicho procedimiento: Primero, se seleccionó el datase Cohn-Kanade, el cual tiene imágenes de

rostros etiquetadas con 7 emociones: ira, disgusto, desprecio, miedo, alegría, tristeza y sorpresa. Segundo, se efectuó el procedimiento HOG y un algoritmo de descubrimiento de sitios de referencia. En tercer lugar, se trabajó en Dlib, y los datos resultantes funcionaron como ingresos para ejercitar y experimentar diferentes algoritmos de sistematización: Naive Bayes, SVM, Decision Tree, Random Forest, AdaBoost y Multilayer Perceptron, este último consiguió el mejor resultado. En cuarto lugar, la agrupación de la secuencia de imágenes fue al azar, de manera arbitraria. Luego se empleó el mismo conjunto de clasificadores a los 68 pares y se obtuvo mejores resultados en cada uno de ellos. Por último, se entrenó una red neuronal en Matlab y se consiguió el excelente rendimiento en una red con una sola capa oculta de 10 neuronas. Referido a los resultados obtenidos se efectuaron según Clasificador, Exactitud y BER de 58 puntos, Exactitud y BER de Distancias: a) Naive Bayes : (52.13, 54.79) , (82.05, 22.53), b) SVM: (58.97, 57.86) , (38.46, 75.50), c) Decision Tree: (52.99, 57.56), (76.92, 30.09), d) Random Forest: (59.83, 52.52), (82.05, 23.66), e) AdaBoost: (36.75, 76.42), (38.46, 73.72), y f) Multilayer Perceptron: (67.52, 46.74), (88.03, 17.99).

Como argumento final se puede decir que, el uso de puntos de referencia para clasificar expresiones faciales puede conducir a buenos resultados. Alimentar al clasificador con un incorporado de identificaciones de distancia entre los sitios de referencia de la cara neutra y otras expresiones puede mejorar considerablemente los resultados. Con base en los resultados, se observa que se utilizaron diferentes clasificadores y Perceptrón Multicapa mostró el mejor rendimiento.

Kim, Hwan An, & Kang, (2017), realizaron la investigación, "Facial expression recognition system using machine learning", en Corea del Sur. En su estudio ellos buscan solucionar la ralentización del movimiento debido que la fatiga está relacionada con los accidentes de tráfico, y la fatiga del conductor se puede predecir en función de las expresiones faciales de los conductores. Hay dos procesos principales en el sistema propuesto, que son la detección de rostros y el reconocimiento de expresiones faciales (FER). El proceso de detección de rostros utiliza características similares a las de Haar, y la región de interés se

restablece para reducir la variable de cambios de apariencia. El proceso FER extrae el histograma de características de gradientes orientados (HOG) de cada región facial, y luego, se realiza una máquina de vectores de soporte para clasificar la expresión facial final. Ahora, desde una perspectiva granular se argumenta el método: En los procesos de detección de rostros y FER, si no hay una región clasificada, la salida será la misma que la entrada. El conjunto de datos de cada persona está compuesto por imágenes de estados de enojo, felicidad y neutralidad. Primero, se efectúa la detección de rostros basándose en las características similares a Haar, dimensión 64 x 64. Después se inició un proceso de reordenación del ROI, el cual es redimensionado a 32 por 32 píxeles para extraer las características del HOG, sólo para el contorno de la cara. El segundo paso es, el reconocimiento de expresiones faciales. Aquí existen dos clasificadores: un clasificador de personas y un clasificador de expresiones faciales. Independientemente de la expresión facial, en el clasificador de personas se utiliza SVM etiquetado como características HOG de cada persona. Luego de haber realizado lo expuesto anteriormente se obtuvo que, en los resultados experimentales, el sistema reconoció exactamente la expresión facial de una determinada persona, y el sistema propuesto tenía la puntuación F1 de 0,8759. Se concluye que, el proceso de reordenamiento del ROI minimizó el factor de cambio ambiental y la estructura jerárquica de la persona, y la clasificación de las expresiones faciales mejoró la tasa de clasificación. Con base en este estudio, podría aplicarse a las interfaces de vehículos personalizadas para prevenir accidentes de tráfico combinando este sistema propuesto con el vehículo.

Laha, Fang, Chen, Wang & Zhang, (2019), realizaron la investigación, Lightweight Deep Learning Model For Facial Expression Recognition en China. Buscan reducir el costo computacional mientras mantienen una precisión relativamente alta en la segmentación de imágenes para reconocimiento de expresiones. El modelo utiliza una red neuronal convolucional denominada Inception para extraer características de las entradas y los cuellos de botella de la red troncal para comprimir el modelo y aprender la función de manera eficiente en materia de segmentación y selección de imágenes. Se obtuvo que

el modelo propuesto es altamente eficiente, y se puede implementar en dispositivos equipados con recursos computacionales limitados. Los resultados demuestran que el modelo ligero de aprendizaje profundo propuesto puede alcanzar niveles relativamente altos precisión para el reconocimiento de expresiones y reducir el cálculo de costo significativamente.

Patras & Pantic, (2016), realizaron la investigación, Dynamics of facial expression: recognition of facial actions and their temporal segments from sequences of facial profile images, en Holanda. La mayoría de los sistemas automatizados existentes para el análisis de expresiones faciales intentan reconocer algunas expresiones emocionales prototípicas, como la ira y la felicidad. Up Facial, dicho sistema consta de una serie completa de ejercicios sencillos que buscan mejorar el aspecto de tu rostro desde el interior, reduciendo el tamaño de la cara de manera casi inmediata, aliviando tensiones musculares y hasta rebajando la fatiga visual. El algoritmo realiza tanto la segmentación automática de un video de entrada en expresiones faciales representadas como el reconocimiento de segmentos temporales (es decir, inicio, vértice, desplazamiento) de 27 AU que ocurren solos o en combinación en el video de perfil de cara de entrada. Se logró una tasa de reconocimiento del 87%. offset) de 27 AU que se producen solas o en una combinación en el video de perfil de cara de entrada.

Bonilla Meza, (2016), Application of Neuronal Networks for recognition of faces in free environments, en México. El inconveniente de la indagación parte de la coexistencia de aspectos humanos en un retrato preciso; para trasladar a cabo la programación de muchos dibujos del rostro de sujetos objetivo y fundar que conciernen al mismo individuo libremente de las situaciones del retrato. Debido a ello, se ejecutó un método de tipo experimental que sirva para el proceso de clasificar las condiciones de la imagen. A partir de esta indagación, se plantea prolongar aumentando la plataforma de retratos con la que está en este momento. Concluyeron que, es viable transportar a cabo la clasificación de aspectos humanos realizando el procedimiento de redes neuronales en lugares

no vigilados, y empleando el clasificador neuronal PCNC y la plataforma de retratos LFW.

Navarro Briones, (2020), en su estudio, realizado en la Universidad Católica Santiago de Guayaquil, concluye que: El recopilamiento de data fundamentado en especialistas referidas al tema, así como las consideraciones de las personas que laboran en dicha zona, ratifican 8 expresiones faciales que pueden ser capaces de detectar el proceso de atención al cliente, y se indican a continuación: felicidad, ira, miedo, sorpresa, tristeza, desagrado, desprecio y neutral. Asimismo, manifiesta que, luego de los tests efectuados con el prototipo se puede inferir que la supervisión y control de las expresiones faciales se puede emplear como forma de enriquecimiento para la compilación de datos en la atención de un modo autómatas hacia un producto o servicio.

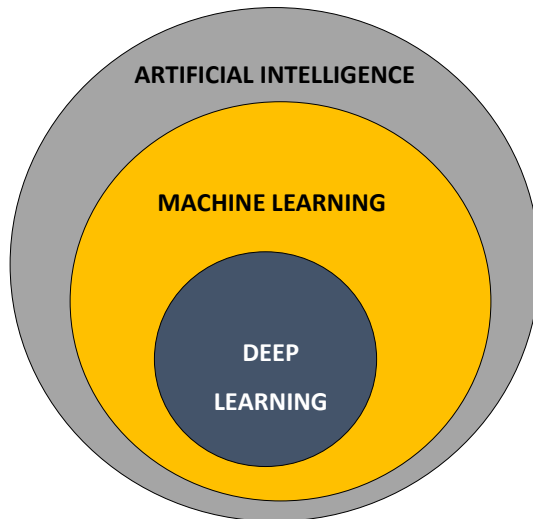
### **1.3. Teorías relacionadas al tema.**

#### **1.3.1 Aprendizaje profundo**

De acuerdo a Yann LeCun, Yoshua Bengio y Geoffrey Hinton en el 2015, el Deep Learning se ha convertido en una técnica que ha permitido mejorar significativamente el estado del arte de reconocimiento visual de objetos por lo que se plantea su estudio en la presente investigación. El Deep Learning permite descubrir complejas estructuras en grandes volúmenes de datos mediante la utilización del algoritmo “backpropagation” o propagación en reversa por su traducción literal en español, algoritmo que indica cómo un computador debería cambiar sus parámetros internos. Un aspecto importante de lo expuesto por LeCun Bengio y Hinton, es que las capas mencionadas no son diseñadas por expertos humanos, sino que estas capas son aprendidas desde los datos mediante procedimientos generales de aprendizaje. El Deep learning (DL) o por su traducción desde el inglés, es un subcampo del “Machine Learning” (ML/Aprendizaje de máquina) el cual es a su vez un subcampo de la Inteligencia artificial conocido como AI, por sus siglas del inglés (ver Figura 1). De la literatura examinada se puede argumentar que el campo de aplicación en el cual el DL se



ha enfocado logrando excepcionales resultados se encuentra alrededor de lenguaje hablado, lenguaje natural, visión incluso juegos por computador, mientras que el clásico aprendizaje de máquina sigue dominando campos como la regresión lineal o árboles de decisión.



*Figura 1.* Representación de la relación del Deep Learning dentro de la Inteligencia Artificial. Fuente: Towards Data Science (2019)

En este sentido es oportuno indicar que la Inteligencia Artificial (AI) tiene como meta proveer al usuario de un conjunto de algoritmos que puedan resolver cierta clase de problemas, mismos que los humanos pueden resolver de manera automática e intuitiva pero que para el computador constituye un reto tecnológico, cuyo enfoque está en las inferencias, planificación, heurística, etc. Un ejemplo de algoritmos de ML son las conocidas Artificial Neural Networks (ANNs) que aprenden de los datos y su función especial es el reconocimiento de patrones, basado en forma similar a como el cerebro humano trabaja. El Deep learning pertenece a este tipo de algoritmos y desde ese punto de vista, tiene ya más de 60 años de edad con respecto a técnicas que actualmente se pueden implementar.

Respecto a la comparativa del Aprendizaje Profundo con el Aprendizaje Automático, Sewak, Sahay, & Rathores (2018), en su estudio, indican que recientemente el Deep Learning ha mostrado resultados prometedores en varias aplicaciones de Inteligencia Artificial como reconocimiento de imágenes, procesamiento de lenguaje natural, modelado de lenguaje, traducción automática neuronal, etc. Aunque, en general, es más costoso

computacionalmente en comparación con las técnicas clásicas de aprendizaje automático, sus resultados se muestran más efectivos en algunos casos.

Por otro lado, Dong & Liang (2019), en su investigación argumentan que, el Deep learning es una subclase del Machine learning. En comparación con el aprendizaje automático tradicional, el aprendizaje profundo es más adecuado para el procesamiento de big data. El rendimiento del algoritmo aumenta con el aumento del volumen de datos. A diferencia del aprendizaje automático tradicional, el aprendizaje profundo no se basa en la determinación artificial de las características de la aplicación. En cambio, intenta obtener directamente funciones de nivel superior de los datos y lograr un modelo de aprendizaje automático de nivel profundo a través de múltiples transformaciones de funciones.

#### **1.3.1.1 Historia de las redes neuronales y Deep learning**

Según la literatura relacionada con el tema, se puede descubrir que el Deep Learning tiene sus primeras apariciones en 1940 enfocada desde diversas perspectivas como Cibernética, Conexionismo y las ya mencionadas Artificial Neural Networks (ANN), estas últimas y pese a que las ANNs, se inspiran en la estructura del cerebro humano para emular su funcionamiento y como sus neuronas interactúan unas con otras, se debe especificar que no significa que se trate de un modelo real de un cerebro humano, por el contrario las ANNs inspiradas en la estructura cerebral real presentan un paralelismo entre la idea básica de cerebro y como este puede comportarse ante ideas de aprendizaje.

Sus primeras apariciones teóricas datan de 1943 en las publicaciones de McCulloch y Pitts quienes estudiaron las características neuronales de las actividades nerviosas y su relación entre ellas para la determinación de actividades lógicas, tanto así que su primera representación funcional trataba de un clasificador binario, que implicaba en una adición de las señales input, que a su vez se multiplican por valores de pesos correctamente seleccionados, debido a que los pesos que se utilizaron para identificar la etiqueta de la clase a

discriminar debían ser introducidos por un humano, no fue considerada como aprendizaje automático.

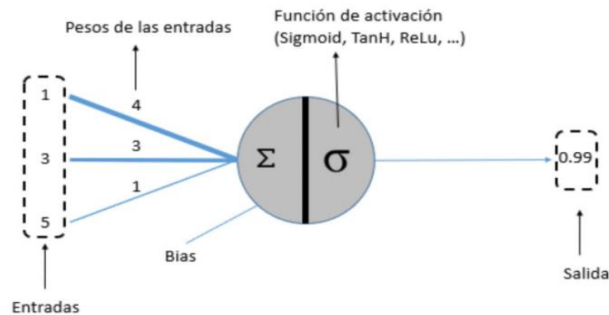


Figura 2: Representación de la RNA-Neural Artificial. Fuente: Ibixa (2019)

### 1.3.1.2 Principales algoritmos del Deep Learning

Según (Deng & Yu, 2014) el tipo de enseñanza, conforme a su edificación y a su propósito que pretenden, llevan a cabo una clasificación en 3 grupos:

- Redes profundas para enseñanza no inspeccionado.
- Redes profundas para enseñanza inspeccionado.
- Redes profundas híbridas.

### 1.3.2 Detección y reconocimiento de rostro

Las personas para el reconocimiento de otras se ayudan de las fisonomías, ya que con el apoyo de la vista se logra procesar retratos de individuos que deseen, para ello la tecnología permite duplicar este proceso con el apoyo de una serie de pasos o secuencias complejas teniendo como fin, el objeto de identificar a otra persona. Se tiene en cuenta que, la comparación de imagen del rostro se efectúa para corroborar y autenticar la identidad de la persona mediante el proceso de reconocimiento facial, teniendo como parte de ello, un dataset de imágenes alojadas y estructuradas en una base datos.

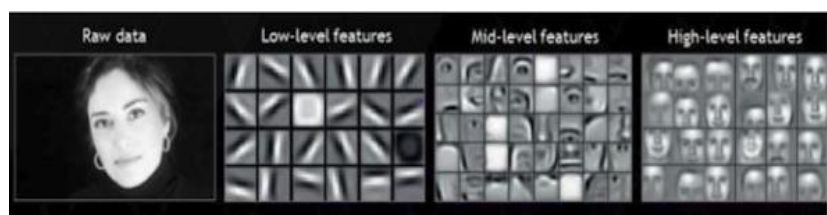


Figura 4: Identificación completa mediante partes concretas de la cara. Fuente: Recuero (2019)

### 1.3.2.1 La expresión facial.

Según (Science Learning Hub, 2007) en su pesquisa, ostenta que existe una nítida diferencia entre los músculos de la parte facial respecto a los del resto del cuerpo. Indica que, es debido a la unión de manera directa a la piel, mas no a los tendones ni huesos. Lo cual denota que, una mínima contracción en la parte mencionada, tirará de la piel, y ello advertirá un cambio en la expresión facial. En otros términos, si se refiere a la parte superior, la piel es aquella que efectúa el movimiento de los huesos, en consecuencia, la actividad está relacionada a dicho aspecto de las expresiones.

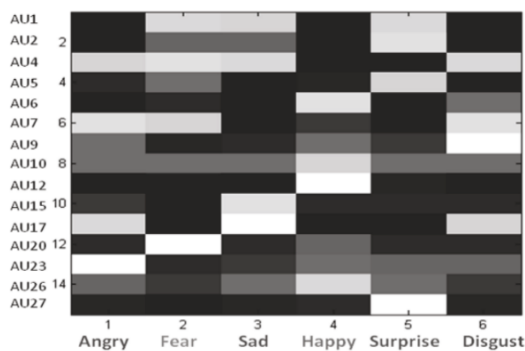


Figura 5: Relación de AU con la emoción básica identificada. Fuente: Pellejero, Grinblat & Uzal (2018)

### 1.3.3 Redes Neuronales Convolucionales.

Las redes neuronales convolucionales están infundidas en la organización del método visual humano. Según K. Fukushima (1980), el aprendizaje vive establecido en que cuando la escapatoria de neuronas parejos es laborosa o trasgrede en zonas concretos de la capa superior de neuronas. Hoy por hoy, los procedimientos de reconocimiento de patrones fundados en redes neuronales convolucionales son los que poseen excelente trabajo como el reconocimiento a mano alzada. Las redes convolucionales viven constituidas en 2 tipos de capas: las capas convolucionales y las capas con submuestra. Cada capa posee una ordenación topográfica, coligada con una perspectiva bidimensional en proporción a un campo propenso. En todo ámbito de cada capa, vive un número disparejo de nervio con cada nervio coherente con un agregado de obligaciones sinápticos de ingreso coligados con nervio apropiables a una línea de la capa anterior. El mismo agregado de pesos, pero con una nueva cintura de forma rectangular está coligado con neuronas comercializadas en otros terrenos. Un fan-in chico en estas neuronas beneficia al procedimiento del gradiente descendente a proceder y irradiarse mediante diferentes capas sin

que su trabajo se evapore tanto como para no proceder, La distribución de conectividad jerárquica dispone las medidas de la red neuronal en una zona providencial en la cual la optimización fundada en el gradiente descendente anda conforme. Bengio (2006) nos indica que, un punto transcendental es que este tipo de redes aun manejándose con pesos fortuitos en las iniciales capas sin ejercicio poseen un excelente ejercicio que una arquitectura profunda totalmente preparada manejando una enseñanza vigilada; así mismo, su trabajo avanza grandemente contrastada con una red convolucional totalmente preparada. Para concluir, Hinton & Osindero (2006), nos explican que, tenemos que recordar que estas redes formaron el excelente caso de preparación triunfante de una arquitectura difícil incluso antes de la venida de las Redes Neuronales Profundas.

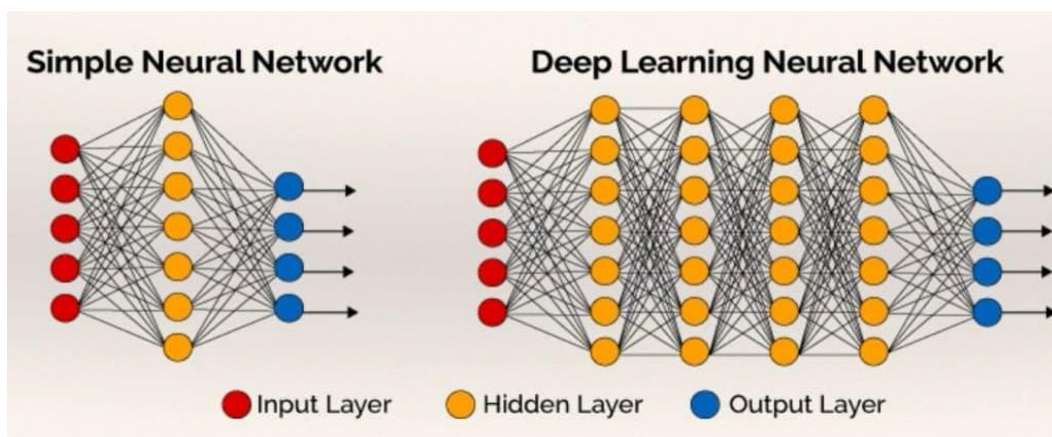


Figura 6. Red Neuronal simple y Red Neuronal profunda. Villanueva (2010)

Se observa que las derivaciones de salida de varias neuronas son las derivaciones de ingreso de otras, y así continuamente. Pero se visualiza un par de discrepancias entre las capas. Los nervios de tono rojo no poseen derivaciones de ingreso. Son indagación que se dará a los nervios. Asimismo se nota que las neuronas de tono azul tienen derivaciones de salida que no viven juntas a otros nervios. Son la investigación de salida de la red neuronal. Dependiendo del número de capas ocultas de tono amarillo se puede estimar o conversar de una red neuronal difícil.

### 1.3.4 Autoencoders.

Han estado manejados como dispositivos arquitectos de instrucción de arquitecturas difíciles, en el que cada nivel es coligado con un Autoencoder que prepara primeramente la capa libremente de las restantes. El mismo es preparado para recopilar el ingreso  $x$  en alguna forma correspondiente  $c(x)$  tal que el acceso logre ser reformada a partir de esta representación. La formulación manejada trasciende el juicio del desliz cuadrático proponiendo la minimización de la verosimilitud logarítmica de la restauración, en aquel momento dada la forma  $c(x)$  tenemos:

$$RE = -\log P(x|c(x))$$

Si la repartición de  $x|c(x)$  es de tipo Gaussiano rescataremos el juicio de la falta cuadrático medio. Los ingresos  $x_i$  son de dos elementos o estimadas como posibilidades de tipo binomial, la función de precio está entregada por la ecuación:

$$-\log P(x|c(x)) = -\sum_i x_i \log f_i(c(x)) + (1 - x_i) \log(1 - f_i(c(x)))$$

donde  $f()$  es convocado decodificador y  $f(c(x))$  es la restauración derivada por la red, en este tema un resultante de dígitos en el nivel (0,1) adquiridos, por ejemplo, manejando un cargo de tipo sigmoide.

Un Autoencoder con un ingreso n-dimensional y una dimensión de recopilado de al menos n logra contenidamente instruirse únicamente el cargo identidad dado que no está una situación para impedir este tipo de simbolización; sin embargo, este caso trata de evitarse. Algunos procesos para el entrenamiento son: 1. Entrenamiento con gradiente descendente estocástico (Regularización de tipo l2 de los parámetros) 2. Añadir sonido en la auto codificación 3. Instituir una situación de iliquidez en el código 4. Extender la probabilidad del modelo generativo.

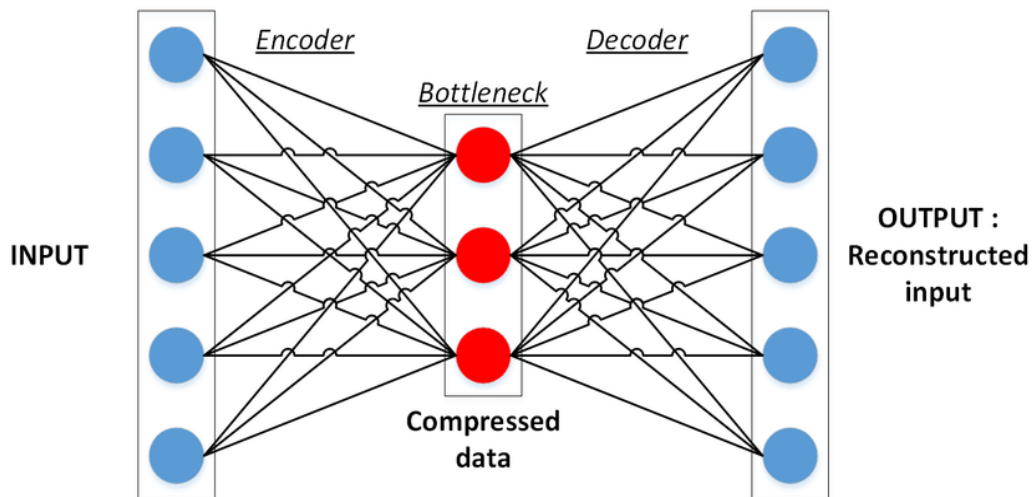


Figura 7. Automatic Post-Disaster Damage Mapping Using Deep-Learning Techniques for Change Detection: Case Study of the Tohoku Tsunami. Sublime y Kalinichieva (2019)

#### 1.3.4.1 Autoencoders convolucionales.

Un uso muy público de los codificadores automáticos es emplearlos a los retratos. El truco consiste en sustituir capas completamente conectadas por capas convolucionales. Estos, junto con las capas agrupadas, cambian el ingreso de ancho y delgado (digamos 100 x 100 px con 3 canales - RGB) a estrecho y grueso. Esto favorece a la red a extraer peculiaridades percibidas de los retratos y, por lo tanto, a lograr una forma del área latente mucho más precisa. El paso de restauración maneja upsampling y convoluciones. La red resultante se denomina codificador automático convolucional.

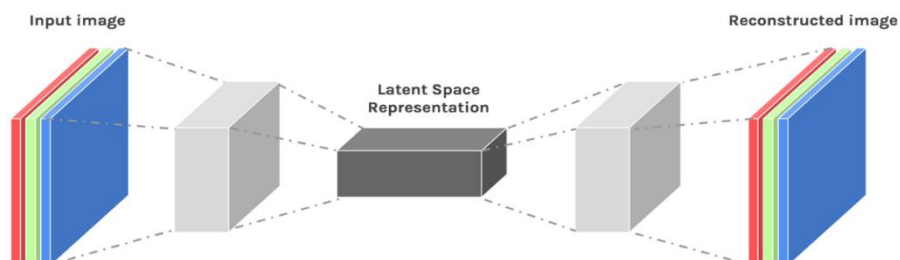


Figura 8. Arquitectura de codificador automático convolucional: asigna un espacio de entrada amplio y delgado a un espacio latente estrecho y grueso. Julien (2017)

#### 1.3.4.1.1 Calidad de reconstrucción

La reconstrucción de la imagen de entrada suele ser borrosa y de menor calidad. Esto es una consecuencia de la compresión durante la cual hemos perdido información.

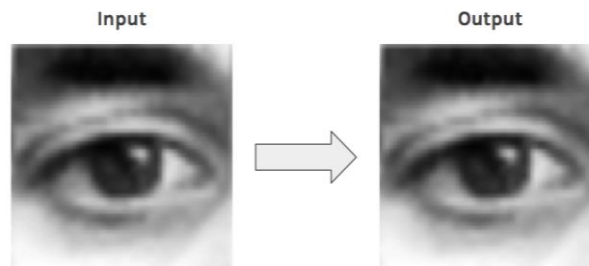


Figura 9. El CAE está capacitado para reconstruir su input. Julien (2017)

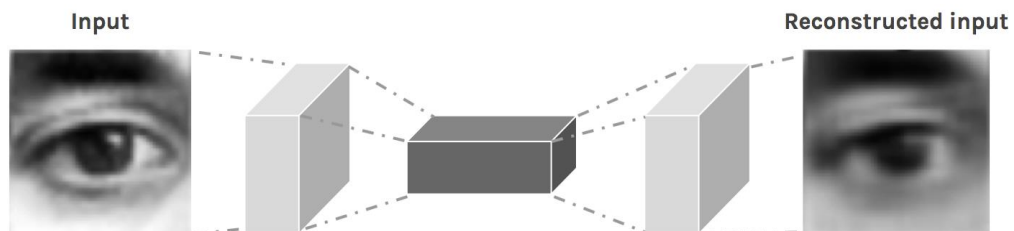


Figura 10. La imagen reconstruida está borrosa. Julien (2017)

#### 1.3.5 Máquina de Boltzman.

Las máquinas de Boltzmann son una gran clase de modelos de redes neuronales, pero la más utilizada en aplicaciones prácticas es la RBM. El modelo RBM en sí es muy simple, solo una red neuronal de dos capas, por lo que no puede considerarse como una categoría de aprendizaje profundo en sentido estricto. Sin embargo, la Deep Boltzmann Machine (DBM) puede considerarse como la promoción de la GBR. No es difícil estudiar DBM después de comprender la GBR, por lo que este artículo se centra principalmente en la GBR. Volviendo a la estructura de RBM, es una red neuronal de dos capas, como se modela en la siguiente figura:



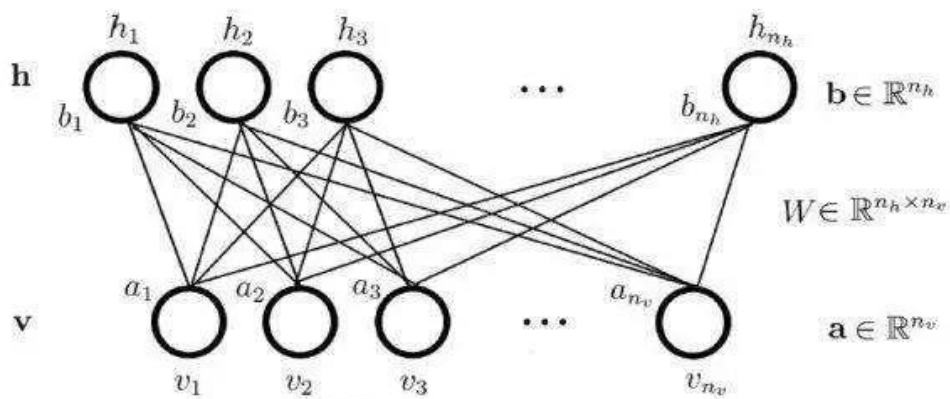


Figura 11. Estructura RBM, dos capas.

Las neuronas de la capa superior forman una capa oculta y el valor de las neuronas de la capa se esconde con el vector  $h$ . Las neuronas de la capa inferior forman la capa visible y el vector  $v$  se utiliza para representar el valor de las neuronas en la capa visible. La capa oculta y la capa visible están completamente conectadas, lo que es similar a DNN. Las neuronas en la capa oculta son independientes y las neuronas en la capa visible también son independientes. El peso de la conexión se puede representar mediante una matriz  $W$ . La diferencia con DNN es que RBM no distingue entre avance y retroceso. El estado de la capa visible puede actuar sobre la capa oculta, y el estado de la capa oculta también puede actuar sobre la capa visible. El coeficiente de sesgo de la capa oculta es el vector  $b$ , y el coeficiente de sesgo de la capa visible es el vector  $a$ . Los RBM de uso común son generalmente binarios, es decir, ya sea una capa oculta o una capa visible, el valor de sus neuronas es solo 0 o 1. El objetivo principal de la Máquina de Boltzmann es optimizar la solución de un problema. El trabajo de Boltzmann Machine es optimizar los pesos y la cantidad asociados con este problema en particular.

### 1.3.5.1 Algoritmo de entrenamiento

Como sabemos que las máquinas Boltzmann tienen pesos fijos, por lo que no habrá un algoritmo de entrenamiento porque no necesitamos actualizar los pesos en la red. Sin embargo, para probar la red, necesitamos establecer los pesos y para encontrar la función de consenso (CF).

La máquina Boltzmann tiene un conjunto de unidades  $U_i$  y  $U_j$  y tiene conexiones bidireccionales.

#### **1.4. Formulación del Problema.**

¿De qué manera se puede utilizar el aprendizaje profundo para el reconocimiento de expresiones faciales de tristeza?

#### **1.5. Justificación e importancia del estudio.**

El presente proyecto se justifica puesto que, en nuestros días, ha habido una gran demanda de sistemas capaces de identificar seres humanos basándose en diversos aspectos de su fisiología humana (biometría). Debido a la debilidad en seguridad que estamos viviendo en la actualidad, en nuestro país, en nuestra ciudad, e incluso en nuestro propio vecindario. Adicional a ello, se justifica porque, hoy en día las personas estamos expuestos a ser engañados, sobornados, amenazados. Y es aquí donde surge la necesidad inminente de cambios en los protocolos de seguridad, a sistemas parcialmente automatizados. Asimismo, se fundamenta porque si bien es cierto se ha optado por resolver el problema simplificándolo a operaciones matriciales y metodologías de reconocimiento de patrones generalizadas, pero no consideran otras áreas como la de la percepción facial. Dadas todas las condiciones mencionadas anteriormente, se hace necesaria la realización de la presente investigación en el área de reconocimiento facial.

#### **1.6. Hipótesis.**

Con el uso del algoritmo de aprendizaje profundo, Red Neuronal Convolutiva y la librería Face Recognition, se busca reconocer las expresiones faciales de tristeza de manera eficiente.

#### **1.7. Objetivos.**

##### **1.7.1. Objetivo general.**

Desarrollar un sistema de reconocimiento de expresiones faciales de tristeza utilizando aprendizaje profundo.

### **1.7.2. Objetivos específicos.**

- a) Seleccionar el dataset que cumpla con protocolos de selección.
- b) Seleccionar el algoritmo de aprendizaje profundo para el reconocimiento de expresiones faciales de tristeza.
- c) Aplicar el reconocimiento de expresión facial en un sistema.
- d) Realizar las pruebas y obtener resultados.

## **II. MATERIAL Y MÉTODO**

### **2.1. Tipo y Diseño de Investigación.**

#### **2.1.1 Tipo de Investigación.**

Tipo cuantitativa, puesto que, su característica es la aplicación de las ilustraciones logrados durante el plazo de la investigación. Se precisa mencionar, que se utilizarán datos de tipo numérico para su posterior análisis o transformación en caso sea necesario. Los datos en mención estarán relacionados con las variables descritas en el apartado de Operacionalización

#### **2.1.2 Diseño de Investigación.**

Diseño cuasi experimental. Dado que, el uso de la variable independiente “Algoritmos de aprendizaje profundo”, se utilizará para el reconocimiento de expresiones faciales de tristeza mejorando en términos de eficiencia y tiempos.

### **2.2. Población y muestra.**

#### **2.2.1 Población.**

La población que se estima, son 6 algoritmos de Deep Learning, los cuales son extraídos de artículos de corte científico, basándose en la relación de los mismos con la identificación o reconocimiento facial. Se detalla en la siguiente tabla lo mencionado:

Tabla 1.

*Población de Algoritmos de Machine Learning para la Investigación.*

N°	Algoritmos de Deep Learning
1	Redes Neuronales Profundas
2	Codificadores Automáticos
3	Máquinas Boltzmann limitadas
4	Red Neuronal Recurrente de Larga Duración y Corta Memoria
5	Red Neuronal Convolutiva
6	Red de Perceptrón Multicapa

Fuente: Elaboración propia.

### **2.2.2 Muestra.**

Por conveniencia, para el desarrollo del sistema enfocado en el reconocimiento de expresiones faciales de tristeza se ha optado por el algoritmo de Red Neuronal Convolutiva, puesto que, durante la revisión del material bibliográfico se encontró mejores resultados cuando se hizo uso del mismo.

### 2.3. Variables, Operacionalización.

Tabla 2.

Matriz de variables e indicadores para la Investigación.

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
<b>VARIABLE</b>				
<b>INDEPENDIENTE</b>	Consumo	de Uso	de	
Algoritmo	de recursos	consumo	de	
aprendizaje		CPU		
profundo		Uso de memoria		
			$C_e = \sum_j^n \frac{C_{e_j}}{n}$	
			$C_m = \sum_j^n \frac{C_{m_j}}{n}$	Registro electrónico
<b>VARIABLE</b>				
<b>DEPENDIENTE</b>	Rendimiento	Exactitud		Ficha de observación
Reconocimiento			$E = \frac{VP + VN}{VP + VN + FP + FN}$	
de expresiones		Precisión	$P = \frac{VP}{VP + FP}$	
faciales de			$R = \frac{VP}{VP + FN}$	
tristeza		Recall		

Fuente: Elaboración propia.

## **2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.**

### **2.4.1 Técnica de recolección de datos.**

#### **2.4.1.1 Registro Electrónico.**

Será mediante esta técnica que se recopilará la información concerniente para el desarrollo de la investigación. En primer lugar, se identificó que el repositorio sea seguro y fiable. En segundo lugar, se evaluó si cuenta con una licencia de uso, para que tenga mayor credibilidad y veracidad. El dataset denominado FER2013, se encuentra en el repositorio de Kaggle, y se ha utilizado para varias investigaciones de corte científico a nivel mundial. En países como China, Japón, India, Vietnam.

### **2.4.2 Instrumento de Recolección de datos.**

#### **2.4.2.1 Ficha de Observación.**

Conforme al proceso de desarrollo de la propuesta se gestionará cada cambio. Con el motivo de salvaguardar y crear un asegurado verdadero de los intentos elaborados para detectar un rostro con la expresión facial de tristeza, según el algoritmo de red neuronal convolucional seleccionado.

## **2.5. Procedimiento de análisis de datos.**

Se plantea reutilizar una Red Neuronal Convolutiva para que sea capaz de identificar y detectar expresiones de tristeza en un rostro. En razón a ello, se utilizará el dataset FER2013 de la plataforma Kaggle. Luego, se empleará la librería Face Recognition para el reconocimiento facial. En mención a lo anterior, indicar que, el sistema estará basado en el lenguaje de programación Python, lo cual nos será útil para trabajar de la mano con la herramienta OpenCV, que sería algo clave en este proceso. La finalidad es, reconocer las expresiones faciales de tristeza de manera eficiente, y optimizando el tiempo de entrenamiento.

Las fórmulas empleadas se describen a continuación:

**Grado de consumo de CPU:**

$$C_e = \sum_j^n \frac{C_{e_j}}{n}$$

Donde:

$C_e$ : Es el grado de consumo de CPU

$C_{e_j}$ : Es el grado de consumo de CPU en la prueba  $j$

$n$ : Es el total de pruebas

**Grado de consumo de memoria:**

$$C_m = \sum_j^n \frac{C_{m_j}}{n}$$

Donde:

$C_m$ : Es el grado de consumo de memoria

$C_{m_j}$ : Es el grado de consumo de memoria en la prueba  $j$

$n$ : Es el total de pruebas

**Promedio de tiempo de respuesta:**

$$Tr = \sum_j^{n_f} \frac{t_{f_j} - t_{f_i}}{n}$$

Donde:

$Tr$ : Es el tiempo de respuesta

$t_{f_j}$ : Es el tiempo final de respuesta

$t_{f_i}$ : Es el tiempo inicial de respuesta

$n$ : Es el total de pruebas

### **Exactitud**

$$E = \frac{VP + VN}{VP + VN + FP + FN}$$

Donde:

E: Es la exactitud

VP: Es el verdadero positivo

VN: Es el verdadero negativo

FP: Es el falso positivo

FN: Es el falso negativo

### **Precisión**

$$P = \frac{VP}{VP + FP}$$

Donde:

P: Es la precisión

VP: Es el verdadero positivos

FP: Es el falso positivo

### **Recall**

$$R = \frac{VP}{VP + FN}$$

Donde:

R: Es la recall o exhaustividad

VP: Es el verdadero positivo

FN: Es el falso negativo

### **F – Score**

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

Donde:

F: Es el F -Score

P: Es la precisión

R: Es la recall



## AUC

$$AUC = \sum_{i \in (P+N)} \frac{(TPR_i + TPR_{i-1})(FPR_i + FPR_{i-1})}{2}$$

Para lograr el máximo rendimiento y eficiencia, es importante lograr el equilibrio adecuado entre la elección de la arquitectura y el modelo de computación. Los marcos de procesamiento de gráficos existentes apuntan principalmente a una arquitectura y adoptan uno de los modelos de computación anteriores (es decir, modelos centrados en vértices/bordes o SPMV). Sin embargo, no es transparente qué modelo de arquitectura y computación maximiza el rendimiento y la eficiencia del sistema en su conjunto.

Si bien es bueno indicar que cada algoritmo mostrará un comportamiento diferente, ya sea por la técnica utilizada para procesar y clasificar, como por la distribución y naturaleza de los datos suministrados. Así de cada algoritmo se puede extraer valores como la precisión que ha tenido con un conjunto de datos de entrenamiento o el consumo de recursos de la máquina en la que se ejecute. Entre los criterios que conllevan la dimensión rendimiento están: memoria usada para ajuste o entrenamiento, tiempo necesario para el ajuste y tiempo necesario para la predicción de las etiquetas.

El consumo de cpu y memoria, servirá para compararse con trabajos futuros que hagan uso de otros algoritmos pero que se enfoquen en la misma problemática y saber de esta manera, si el tamaño del conjunto de datos, la alternancia de parámetros, y nivel de complejidad causan algún efecto en el procesador.

## 2.6. Criterios éticos.

### 2.6.1 Conformabilidad.

Se producirán resultados en base al fundamento y desarrollo de la investigación, los cuales deberán ser correctamente aprobados y legitimados por expertos en

la materia. Dichas personas, como criterios tendrán la subjetividad y veracidad en las declaraciones dadas y expresadas.

### **2.6.2 Derechos de Autor.**

La compilación de investigaciones y artículos, en el presente trabajo están debidamente referenciados, indicando datos importantes como: autor del documento, año de publicación y título de la investigación.

## **2.7. Criterios de Rigor Científico.**

### **2.7.1 Originalidad.**

Se detallará de donde provienen las fuentes bibliográficas correctamente analizadas y consultadas, con el fin de ser ponderados en fundamento al porcentaje resultante del programa anti plagio.

### **2.7.2 Validez.**

Se evaluarán los datos alcanzados durante el transcurso de la investigación, tomando como fuentes de validez, el criterio de especialistas e ingenieros calificados. Todo ello, será para fundamentar su veracidad y autenticidad.

### **2.7.3 Veracidad.**

La documentación expuesta en el presente trabajo, se exhibirá sin manipulación alguna. Puesto que, será totalmente veraz y confiable.

### III. RESULTADOS.

#### 3.1. Resultados en Tablas y Figuras

A continuación, se muestran los efectos después de implementar el modelo secuencial de red neuronal convolucional. Se obtuvieron los valores de los indicadores descritos en la Tabla 3.

Tabla 3.

*Valores de los indicadores en las 50 épocas realizadas de la Red Neuronal Convolutacional.*

Época	Precisión	Recall	F1 Score
1	1.0000	1.0000	1.0000
2	1.0000	0.7550	0.8604
3	1.0000	0.5660	0.7229
4	0.5980	0.5880	0.5930
5	0.9770	1.0000	0.9884
6	0.8634	1.0000	0.9267
7	1.0000	0.5277	0.6908
8	0.6667	1.0000	0.8000
9	1.0000	0.7627	0.8654
10	1.0000	1.0000	1.0000
11	1.0000	0.9544	0.9766
12	1.0000	1.0000	1.0000
13	0.1818	0.9187	0.3036
14	1.0000	1.0000	1.0000
15	1.0000	1.0000	1.0000
16	1.0000	1.0000	1.0000
17	1.0000	1.0000	1.0000
18	1.0000	0.9536	0.9762
19	0.7822	0.5333	0.6342
20	1.0000	1.0000	1.0000
21	1.0000	0.7684	0.8690
22	1.0000	1.0000	1.0000
23	0.9000	1.0000	0.9474

24	1.0000	1.0000	1.0000
25	1.0000	0.9230	0.9600
26	1.0000	0.8737	0.9326
27	0.7777	1.0000	0.8750
28	1.0000	1.0000	1.0000
29	1.0000	0.8822	0.9374
30	1.0000	0.9630	0.9812
31	0.7660	1.0000	0.8675
32	0.6461	0.4930	0.5593
33	0.8555	0.7820	0.8171
34	1.0000	0.0836	0.1543
35	0.9824	0.3370	0.5018
36	1.0000	0.5867	0.7395
37	0.7880	0.5660	0.6588
38	0.5988	1.0000	0.7491
39	1.0000	0.7770	0.8745
40	0.8208	0.6026	0.6950
41	1.0000	0.7176	0.8356
42	1.0000	1.0000	1.0000
43	0.9204	1.0000	0.9586
44	1.0000	1.0000	1.0000
45	0.9645	0.8344	0.8947
46	1.0000	1.0000	1.0000
47	0.8426	1.0000	0.9146
48	0.5660	0.3330	0.4193
49	0.7530	1.0000	0.8591
50	1.0000	0.8330	0.9089

---

Nota: Los datos fueron calculados en base a pruebas realizadas en la Red Neuronal Convolutiva.

Fuente: Elaboración Propia.

En función de los efectos alcanzados de la Tabla 3, se pudo conseguir el valor del indicador F1-score, primero se tuvo que obtener el promedio de los valores de la precisión, su valor fue de 0.9050 y de la misma manera se obtuvo el promedio de los valores alcanzados de Recall de cada una de las épocas, su valor promedio fue de 0.8383, por lo tanto, utilizando la fórmula del F1-Score:

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

Se efectúa el reemplazo de los valores y se obtuvo:

$$F1 = 2 \frac{0.9050 \times 0.8383}{0.9050 + 0.8383}$$

$$F1 = 0.84$$

Los cálculos de los indicadores de precisión el cual considera verdaderos positivos (TP) y falsos efectivos y Recall utiliza falsos negativos fueron calculados por la misma red neuronal convolucional, estos valores corresponden a valores en pixeles de las imágenes.

En el siguiente apartado, en la tabla 4 se pudo prestar atención los efectos adquiridos de los indicadores de la Dimensión Rendimiento, planteados en la matriz de variables y operacionalización, que son: precision, recall y f1-score:

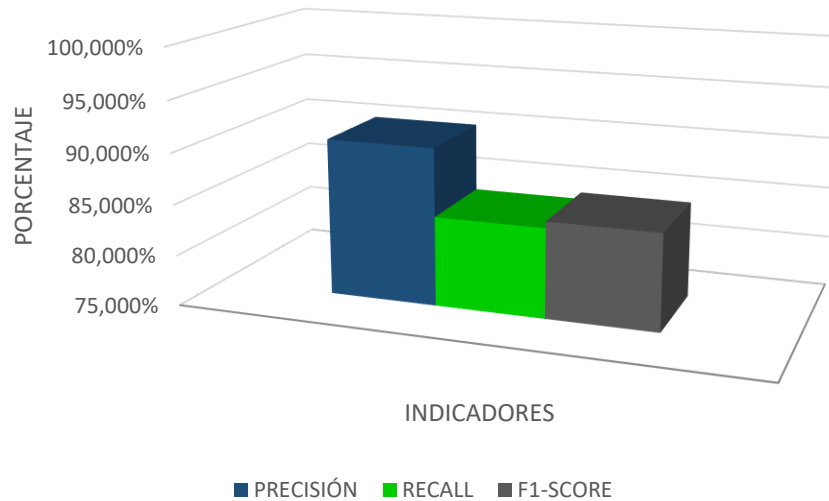
Tabla 4.

*Resultados de los indicadores de la dimensión rendimiento de la Red Neuronal Convolucional.*

Precision	Recall	F1 Score
0,9050	0,8383	0,8449

Fuente: Elaboración propia.

Mediante un gráfico estadístico, se visualizaron los valores reflejados en la Tabla 4, lo mencionado se representa en la Figura 11.



*Figura 11.* Gráfico con los valores de los indicadores de precisión, Recall y f1-score. Fuente: Elaboración propia.

Los datos muestran que la Red Neuronal Convolutiva en términos de precisión tuvo un (0,90), en términos de recall tuvo un (0,83), y en términos de f1 score obtuvo un (0,84).

Se muestran los efectos logrados de la etapa de entrenamiento utilizando una red neuronal convolutiva, lo cual se plasmó en la Tabla 5.

Tabla 5.

*Valores de Loss y Val\_Loss en las 50 épocas realizadas de la Red Neuronal Convolutiva (Train).*

Época	ETAs	Loss	Val_loss
1	714	1.7226	1.7135
2	696	1.3868	1.2829
3	698	1.2115	1.1567
4	742	1.1207	1.0682
5	707	1.0619	1.0331

6	693	1.0185	1.0558
7	708	0.9860	1.0435
8	739	0.9611	0.9363
9	737	0.9278	1.1109
10	772	0.9116	1.0122
11	744	0.8863	0.9465
12	701	0.8606	0.9927
13	687	0.8444	0.8821
14	695	0.8217	0.8939
15	703	0.8013	0.9490
16	755	0.7875	0.8606
17	729	0.7602	0.9982
18	728	0.7437	0.8943
19	703	0.7279	0.9628
20	713	0.7121	0.7386
21	713	0.6881	0.6742
22	709	0.6670	0.6296
23	739	0.6530	0.6170
24	711	0.6381	0.6555
25	743	0.6070	0.6401
26	731	0.5924	0.6607
27	715	0.5866	0.6093
28	690	0.5633	0.5690
29	707	0.5423	0.5691
30	770	0.5244	0.6455
31	749	0.5202	0.6464
32	732	0.4929	0.6012
33	700	0.4846	0.5995
34	935	0.4615	0.4920
35	855	0.4478	0.5453
36	682	0.4352	0.5472
37	678	0.4175	0.7029
38	677	0.4138	0.5402

39	676	0.3860	0.5534
40	676	0.3787	0.5958
41	676	0.3635	0.5446
42	675	0.3566	0.4479
43	675	0.3381	0.6027
44	675	0.3298	0.5086
45	685	0.3214	0.4685
46	698	0.3053	0.5885
47	677	0.3065	0.3660
48	674	0.2834	0.4965
49	675	0.2772	0.5098
50	675	0.2619	0.4460

Fuente: Elaboración Propia.

La función de pérdida como se observa en la Figura 12, tuvo en cuenta las probabilidades o la incertidumbre de una predicción (línea azul) en función de cuánto varía la predicción del valor real (línea amarilla). Esto no delimitó una visión más matizada de qué tan bien se desempeñó el modelo, lo ideal es conseguir una incertidumbre de 0.

## Pérdida

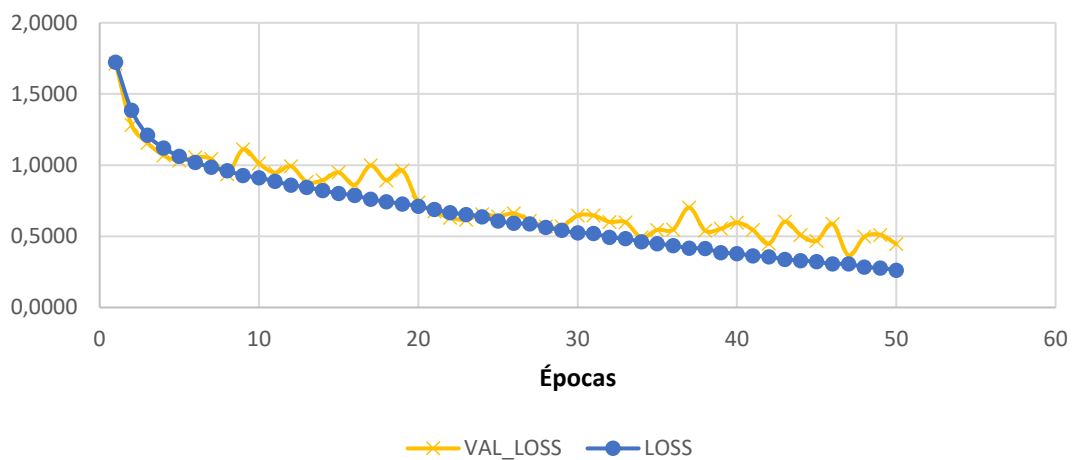


Figura 12. Diagrama de dispersión de la función de pérdida de la Red Neuronal Convolutiva. Fuente: Elaboración propia.



Se presentan los resultados obtenidos de Accuracy, etapa de entrenamiento utilizando una red neuronal convolucional, lo cual se plasmó en la Tabla 6.

Tabla 6.

*Valores de Accuracy y Val\_Accuracy en las 50 épocas realizadas de la Red Neuronal Convolutacional (Train).*

Época	ETAs	Accuracy	Val_Accuracy
1	714	0.3853	0.3992
2	696	0.5263	0.5551
3	698	0.5957	0.6033
4	742	0.6352	0.6576
5	707	0.6580	0.6624
6	693	0.6771	0.6585
7	708	0.6904	0.6569
8	739	0.6994	0.7085
9	737	0.7165	0.6494
10	772	0.7194	0.6831
11	744	0.7277	0.7126
12	701	0.7380	0.6951
13	687	0.7444	0.7305
14	695	0.7543	0.7264
15	703	0.7625	0.7129
16	755	0.7686	0.7480
17	729	0.7778	0.6674
18	728	0.7842	0.7328
19	703	0.7917	0.7243
20	713	0.7974	0.7581
21	713	0.8061	0.7306
22	709	0.8149	0.7440
23	739	0.8219	0.8323
24	711	0.8239	0.8362
25	743	0.8387	0.8471
26	731	0.8428	0.8464

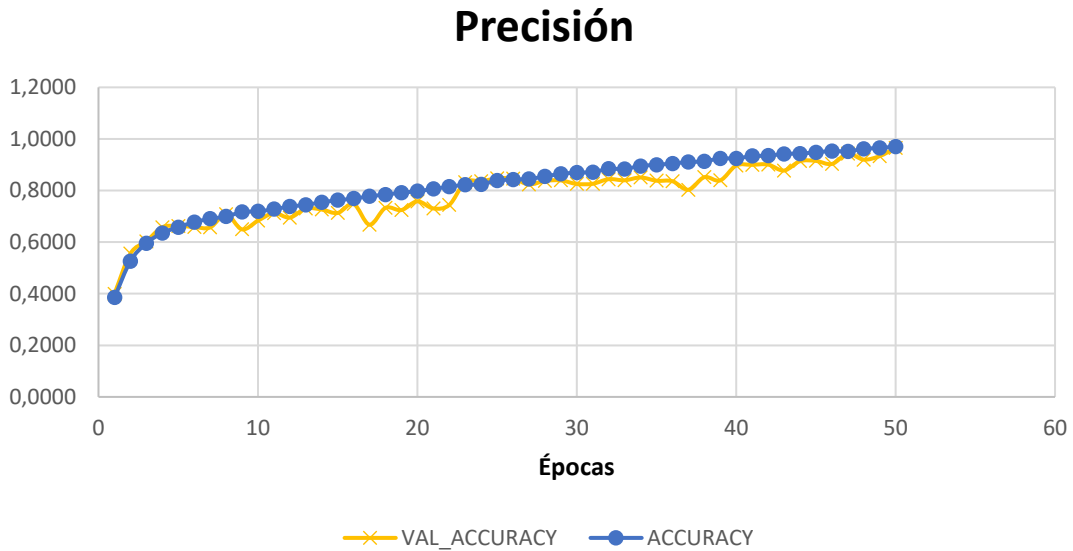
27	715	0.8452	0.8236
28	690	0.8546	0.8374
29	707	0.8645	0.8392
30	770	0.8693	0.8257
31	749	0.8714	0.8262
32	732	0.8839	0.8441
33	700	0.8838	0.8397
34	935	0.8950	0.8507
35	855	0.8997	0.8381
36	682	0.9047	0.8362
37	678	0.9105	0.8030
38	677	0.9133	0.8517
39	676	0.9244	0.8393
40	676	0.9243	0.8971
41	676	0.9336	0.8996
42	675	0.9349	0.9007
43	675	0.9419	0.8775
44	675	0.9434	0.9138
45	685	0.9481	0.9149
46	698	0.9529	0.9030
47	677	0.9518	0.9428
48	674	0.9614	0.9198
49	675	0.9650	0.9335
50	675	0.9703	0.9652

---

Fuente: Elaboración Propia.

La función de Precisión como se observa en la Figura 13, se usó para calcular el desempeño de un modelo de codificación, se midió normalmente en porcentaje. La precisión es el cálculo de presentimientos donde el precio predicho es parejo al valor existente. Se pudo notar que el indicador está aumentando en cada época, lo que nos delimitó que se está acercando a un valor de uno. Se pudo pensar que ambas funciones están inversamente proporcionales, pero no fue así. Cada función fue independiente y se evaluó ambas, para poder validar el

modelo. En esta ocasión, se examinó que ambas funciones dieron resultados de nivel óptimo, por lo que se infirió que el modelo planteado está siendo efectivo, dado que, tiene tendencias a cero en pérdida y a uno en ganancia mientras las épocas aumentan.



*Figura 13.* Diagrama de dispersión de la función de precisión de la Red CNN. Para finalizar la sección de los indicadores, se especificaron los valores de consumo de grado de CPU, consumo de memoria y el tiempo de ejecución, respecto a la tasa de aprendizaje de la red neuronal convolucional propuesta. La Tabla 7, evidencia los resultados obtenidos acerca del tiempo de respuesta del entrenamiento.

Tabla 7.

*Tiempo de ejecución del modelo secuencial de la Red Neuronal Convolucional.*

Tasa	Tiempo de Ejecución
Tasa de Aprendizaje (ETA)	35737 s.

Fuente: Elaboración Propia.

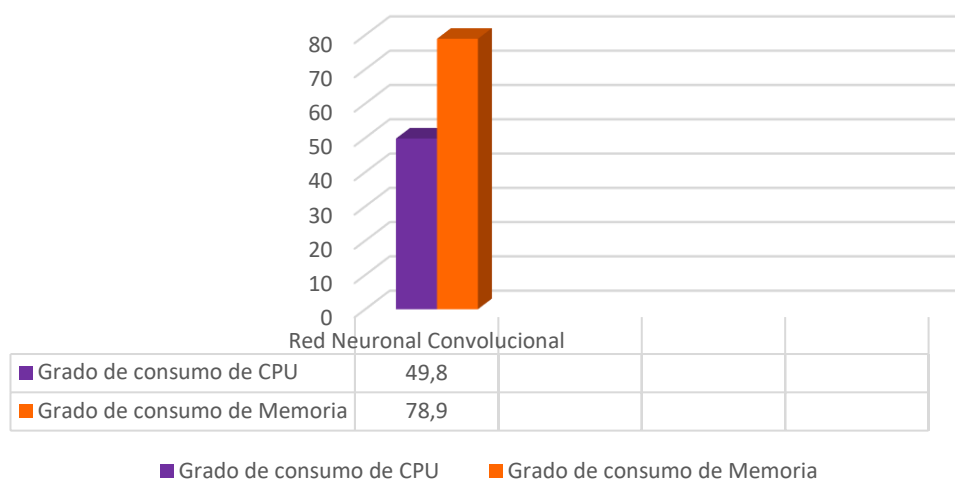
El tiempo de entrenamiento como se muestra su valor es de 35737 en segundos, esto en minutos es una aproximación de 9 horas con 56 minutos. Como última instancia, para concluir el acápite 3.1 Resultados, se delimitaron

los valores de CPU y Memoria, que arrojó la implementación de la Red Neuronal Convolutiva planteada, para el reconocimiento de la expresión facial de tristeza. Lo cual se evidencia en la Tabla 8 y la Figura 14.

Tabla 8.

*Medidas de consumo de recursos de la Red Neuronal Convolutiva planteada.*

Algoritmo de DL	Grado de consumo de CPU	Grado de consumo de Memoria
Red CNN	49,8	78,9%



### **3.2. Discusión de resultados.**

La mente humana es un ordenador altamente complicado, no lineal y paralelo. Dicho de otra manera, es un método de proceso de la indagación, que posee la capacidad de constituir sus componentes ordenados distinguidos como nervios con el propósito de producir a cabo ciertas suposiciones: trabajos de afirmación de modelos, registro de aparatos, codificación de cosas, etc.

Su cargo importante reside en suministrar un perfil del ambiente que nos envuelve, y lo que es más trascendental, nos suministra la indagación que precisamos para interactuar con dicho ambiente. De estas informaciones brota la razón que creyó el origen de las Redes Neuronales Artificiales o ANS.

¿Qué parámetros hacen viable la supremacía del cerebro en unas explícitas labores frente a los computadores? Resulta llamativo estar al tanto que las neuronas son mucho más fáciles, tardas y menos íntegras que una CPU, y sin embargo concurren labores que el cerebro soluciona eficientemente, contestaciones ante incitaciones del ambiente, reconocimiento del lenguaje, etc. Mientras que son laboriosamente llanos por un computador. En concluyente, la idea que subyace es la de chocar todos los inconvenientes que el cerebro soluciona con urgencia y eficacia. Logra implicar provechoso edificar técnicas que aparenten la distribución de las redes neuronales biológicas con la intención de lograr una funcionalidad pareja.

Redes Neuronales Convolucionales (CNN) es una clase de red neuronal, que incumbe a una habilidad de la enseñanza difícil, en sinopsis, es instalar un dibujo de ingreso en red, el retrato cruza la parte convolucional de la red y, en última instancia, la red anuncia el centro en retrato.

En la capa convolucional, se extrajeron las peculiaridades del objeto en el retrato y se somete el tamaño, logrando deducciones más resueltos, aprende patrones específicos dentro de la imagen.

### 3.3. Aporte práctico.

En razón, que se consideró necesario representar el progreso del actual trabajo de indagación, se formuló un esquema gráfico, el cual se delimitó por la Figura 15.

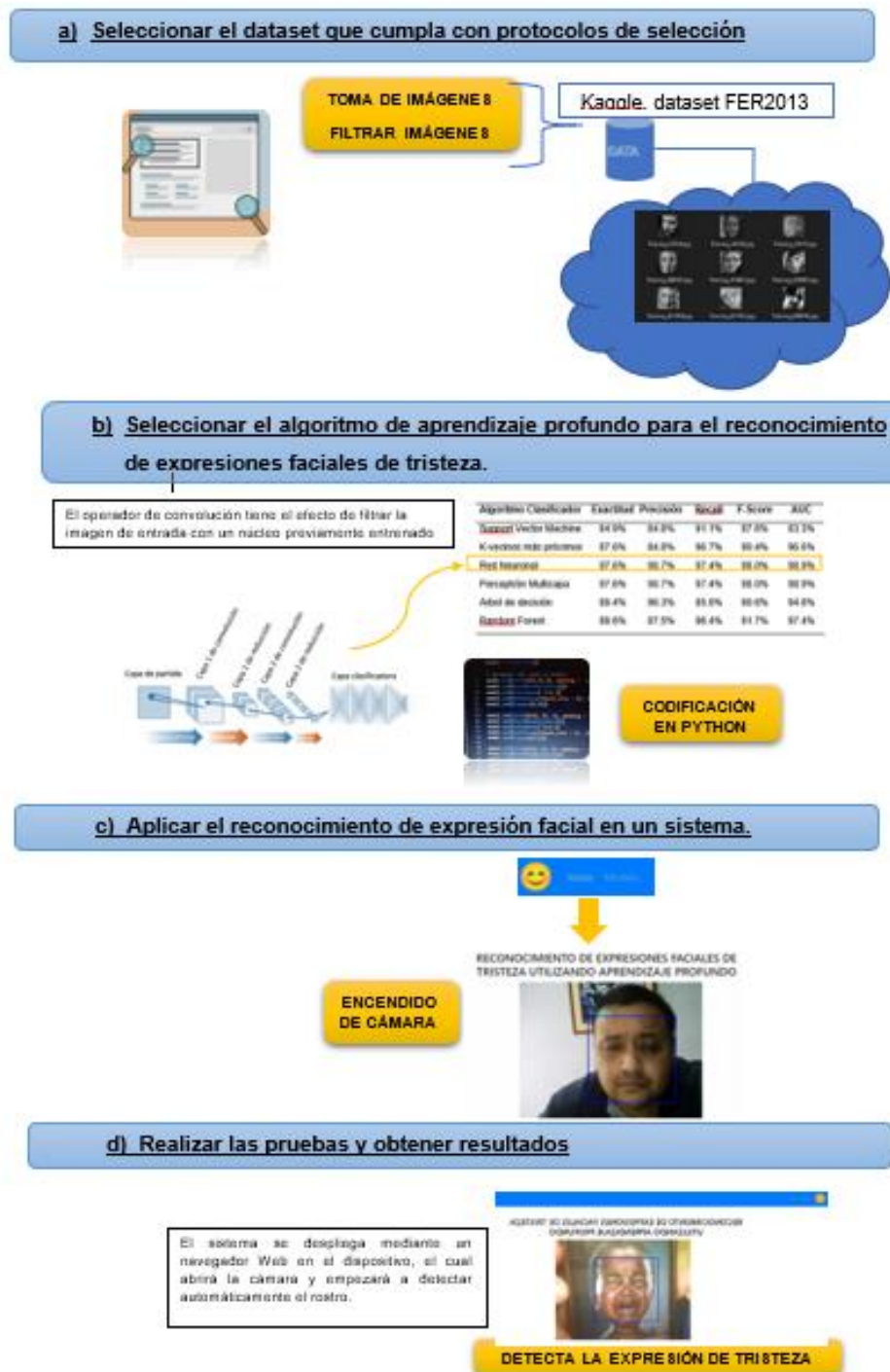


Figura 15. Representación del Método planteado a nivel general. Fuente: Elaboración propia.

A continuación, en los siguientes apartados, se explicó el paso a paso de cada uno de los objetivos específicos, para desarrollar el sistema que nos permita reconocer la expresión facial de tristeza haciendo uso de una red neuronal convolucional.

**a) Seleccionar el dataset que cumpla con protocolos de selección.**

Se efectuó un análisis previo en base a papers de revisión, para luego proceder a la selección del dataset con el cual se trabajó. En esta oportunidad, se trata del dataset FER2013, un conjunto de datos muy usado en investigaciones de corte científico, referidas al reconocimiento de rostros o emociones. Se detalló que cantidad de imágenes se utilizaron, según la expresión, en esta ocasión se enfocó sólo la emoción tristeza. Para mayor comprensión se delimitó la Tabla 9.

Tabla 9.

*Cantidad de imágenes del dataset FER2013 para entrenamiento y validación.*

Emoción	Entrenamiento	Validación
Molesto	3995	958
Disgustado	436	111
De miedo	4097	1024
Feliz	7214	1774
Neutral	4965	1233
Triste	4830	1247
Sorprendido	3171	831

*Nota:* Imágenes de las distintas expresiones para entrenar y validar en modelo. Estas imágenes fueron sacadas de Kaggle, dataset FER2013.

Fuente: Elaboración propia.

Se utilizaron 4830 imágenes con características de tamaño de 48 x 48, en escala gris, con gestualidad de tristeza. A manera representativa, se evidencia la Figura 16, donde se visualizaron parte de los rostros adjuntos en el conjunto de datos del repositorio de Kaggle, con el cual se decidió trabajar debido a la veracidad de sus datos en base a la licencia del mismo y el nivel de usabilidad

que posee para ser empleado en investigaciones que requieran alto grado de validez y confiabilidad.



Figura 16. Captura de pantalla de las imágenes del repositorio de Kaggle, dataset FER2013. Fuente: Elaboración propia.

**b) Seleccionar el algoritmo de aprendizaje profundo para el reconocimiento de expresiones faciales de tristeza.**

Para escoger que tipo de clasificador trabajar se empleó la Tabla 10, donde se delimitó los valores según indicadores, que alguna vez hicieron uso del dataset FER2013 para el reconocimiento de expresiones faciales.

Tabla 10.

Medidas de rendimiento conforme a los indicadores según algoritmos de machine learning investigados.

Algoritmo	Exactitud	Precisión	Recall	F Score	AUC
Support Vector Machine	0,849	0,848	0,911	0,876	0,833
K Nearest Neighbor	0,876	0,848	0,967	0,904	0,966
Neuronal Network	0,976	0,987	0,974	0,980	0,989



Multilayer	0,976	0,987	0,974	0,980	0,989
Perceptron					
Decision Tree	0,894	0,963	0,856	0,906	0,946
Random Forest	0,896	0,875	0,964	0,917	0,974

*Nota:* Algunos porcentajes fueron buscados en trabajos de investigación.

Fuente: Elaboración Propia.

Ma et. al. (2017) en su estudio “Hand Gesture Recognition with Convolutional Neural Networks for the Multimodal UAV Control”, alegan que, la extracción de características de las imágenes faciales es la tarea más crucial en el reconocimiento facial, que se extrae mediante una red denominada neuronal convolucional, cuya técnica es una variación del aprendizaje profundo. Dado que, las redes neuronales de aprendizaje profundo son la mejor solución para la detección de objetos, el reconocimiento de patrones y el reconocimiento facial, la red neuronal convolucional es una de las redes neuronales profundas más utilizadas porque está bien adaptada a la invariancia de traducción de las imágenes.

Las redes neuronales convolucionales utilizan tres ideas básicas principales para la extracción de características de las imágenes faciales: campos receptivos locales, pesos compartidos y agrupación. Se eligió una red neuronal convolucional para clasificar las expresiones por dos razones: en primer lugar, se observó que la posición temporal de las funciones dentro de la ventana de gestos faciales identificados no era constante. Según lo investigado, la CNN puede superar esto asumiendo que las características próximas están conectadas independientemente de su posición general en el conjunto de datos.

La CNN logra esto conectando neuronas de entrada en el mismo campo receptivo para producir activación neuronal para la siguiente capa. En este contexto, se aplicaron cuatro capas convolucionales, cada una de ellas con el mismo filtro matriz de convolución 2D, y misma función de activación RELU. Sin embargo, si nos referimos a los kernels, si difieren, dado que la primera capa conlleva 64 k, la segunda 128 k, la tercera y cuarta ambas con 512k. Asimismo,

las 4 contienen un max pooling (2,2). Finalmente, el método de aplanamiento Flatten y la función softmax para predecir la emoción correcta, que, en este caso, sólo sería la expresión facial de tristeza. En segundo lugar, las imágenes que se van a tratar muestran señales del mismo gesto recopiladas de diferentes participantes. Se observó que cada sujeto producía patrones de señal únicos para cada una de las pruebas, por lo que era necesario que los participantes individuales calibraran la red antes de usarla. Se necesitaría ajustar una gran cantidad de parámetros para entrenar la red completa.

Se seleccionó como clasificador, a la red neuronal convolucional, este algoritmo convierte los datos de tal modo que las diferencias se tornan más imperiosos en el retrato de salida al poseer estas, un total numérico más valioso. El cual es asignado a los pixeles que las representan. Como se modela en la Figura 17.

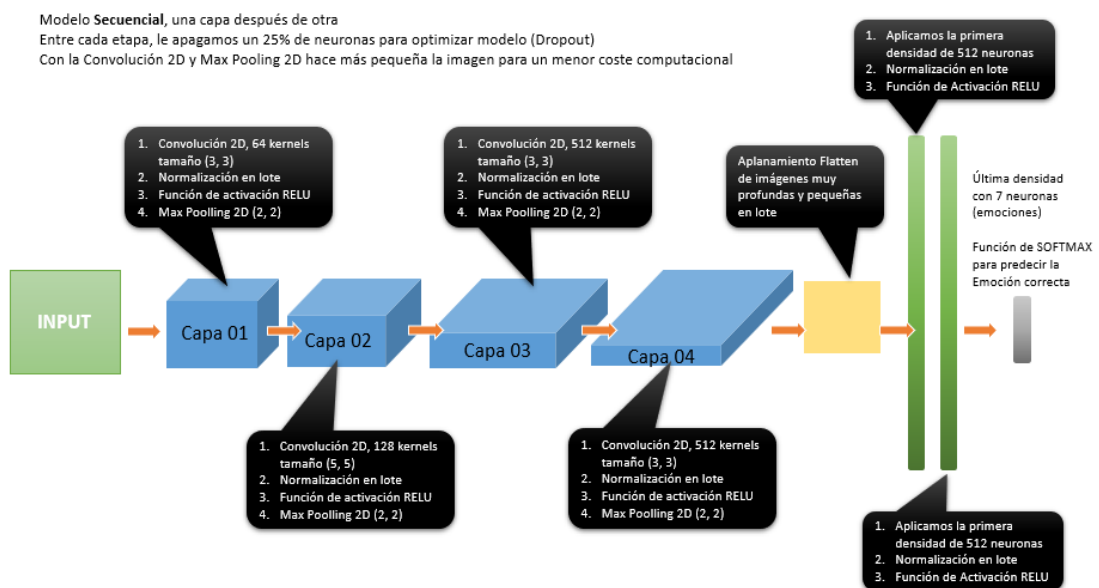
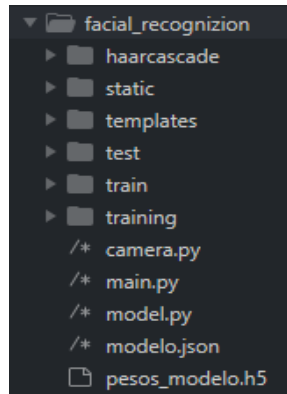


Figura 17. Modelo secuencial en 4 capas de la Red Neuronal Convolutiva.

Fuente: Elaboración propia.

En primera instancia, se crearon las carpetas de proyecto facial\_recognition, dónde se guarda la línea de código en cada subcarpeta, y las imágenes seleccionadas para la red neuronal convolucional.



*Figura 18.* Sistema estructurado en paquetes. Fuente: Elaboración propia.

Se procedió a explicar cuál fue el uso que se le dio a cada carpeta, cada clase o subclase, mediante descripciones por separado. Es por ello, que a continuación, se desglosa el funcionamiento de cada uno: a) haarcascade: contiene al haarcascade, que es un archivo XML preentrenado para ayudar en la precisión del reconocimiento facial. Fue sacado del repositorio oficial de la librería OpenCV en GitHub. b) static: contiene las imágenes usadas en la Web. c) templates: contiene los archivos HTML, necesarios para mostrar el funcionamiento del sistema en la Web. Asimismo, las carpetas para que sirvieron para realizar las pruebas y el entrenamiento, estas son: a) test y train: cuyos directorios contienen subcarpetas de imágenes de las distintas expresiones para entrenar y validar en modelo. Estas imágenes fueron sacadas de Kaggle, dataset FER2013. b) training: contiene al archivo escrito en el lenguaje Python que se encargará de entrenar y validar el modelo. Posterior a ello, en función a lo anterior, se delimitaron las clases utilizadas en el presente sistema, estas son: a) camera.py: cuyo archivo nos ayudó a encender la cámara y detectó la cara en tiempo real. b) main.py: cuyo file es necesario para iniciar el llamado de la web y correr todo el sistema. c) model.py: predice en tiempo real las expresiones del rostro, haciendo uso de los archivos modelo.json, cuyo archivo de consultas es la representación escrita en formato JSON del modelo. Finalmente, la clase pesos\_modelo.h5, cuyo contenido fue los pesos resultantes del modelo previsto.

Es necesario, precisar que para el desarrollo del trabajo se requirió instalar complementos y librerías para el sistema, las cuáles son: 1) Pip: Instrumento manejada para colocar librerías de Python de una manera más tocador, más eficiente y más resuelta. 2) Open CV: Librería libre de visión artificial básicamente perfeccionada por INTEL. Se usa para integrar cámara de vídeo en aplicaciones escritas en Python. 3) Tensorflow: Librería para reunir código de redes neuronales profundas. Si manejamos la ramificación GPU. Accedemos al computador que sea competente de resolver a través de la unidad de proceso grafico de la misma tarjeta gráfica. 4) Numpy: Librería de ocupaciones matemáticas de alto nivel para operaciones con vectores y/o matrices. 5) Seaborn: Es una librería para Python que permite generar fácilmente elegantes gráficos. 6) os: Módulo de Python para acceder a funcionalidades del Sistema operativo en el que corre la aplicación escrita en Python, por ejemplo, a funciones como leer y escribir ficheros. 7) Tensorflow > paquete Keras: Librería de redes neuronales artificiales escrita en Python. Es la librería que reduce las otras metodologías que se crearán uso en enseñanza profundo. 8) livelossplot: Es una librería que permite la visualización de gráficas durante el entrenamiento de redes neuronales. Además de la visualización, mantiene los datos referentes a los valores mínimos y máximos obtenidos durante las ejecuciones.

La segunda instancia, consistió en la descripción de código fuente del sistema, asimismo como la d de paquetes necesarios para la implementación del mismo. Se instalaron los paquetes por medio de “pip”, por ejemplo, para instalar Tensorflow se escribió en la consola: `[pip install tensorflow]`.

Luego de ello, se corrió los archivos del sistema desde consola, navegando hasta la carpeta del proyecto y ejecutándolo con Python, por ejemplo, para ejecutar “main.py”: `[cd Users/users/Desktop/facial-recognizion/python main.py]`

En los siguientes apartados, se describen los bloques de codificación de mayor relevancia para la implementación del sistema. Aquí mediante este párrafo de programación, fue que se importó los paquetes necesarios para entrenar el modelo colocar la subcarpeta. Se tomó como referencia parte anterior.

```
import numpy as np
```

```

import seaborn as sns
import matplotlib.pyplot as plt
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_i
mg
from tensorflow.keras.layers import Dense, Input, Dropout, Flatten, Conv2D
from tensorflow.keras.layers import BatchNormalization, Activation, MaxPoolin
g2D
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from livelossplot import PlotLossesKeras
from tensorflow.keras.utils import plot_model
from IPython.display import SVG, Image
from livelossplot import PlotLossesKerasTF
from tensorflow.keras import backend as K

```

Por consiguiente, una vez que se efectuó la importación, se consideró necesario llevar a cabo un recorrido de las imágenes para asegurarse que todas tengan la dimensión con la que se trabajó todo el proyecto (48x48) y un figsize = (12, 20)

```

def plot_example_images(plt):
img_size = 48
plt.figure(0, figsize=(12,20))
plt2.figure(0, figsize=(12,20))
ctr = 0

```

El siguiente bloque, que se consideró importante fue la carga de la data de entrenamiento y validación. La cantidad de imágenes estuvo definida en el mismo dataset. Un concepto clave fue establecer la cantidad de lote de imágenes con la que se entrenó al modelo. En esta ocasión, se consideró 64. Posterior, se estableció la tasa de aprendizaje del modelo como 0.0005 y por último la cantidad de épocas con las que se entrenó al modelo de red neuronal.

```

img_size = 48
batch_size = 64
clases = 7 # Cantidad de carpetas de emociones, posibles predicciones
lr = 0.0005 # learning_rate, ajustes de la red neuronal para acercarse a la
predicción óptima.
epochs = 50 # Cantidad de épocas que va a iterar el modelo

# Empezamos a preprocesar las imágenes
# Generando data para entrenamiento
datagen_train = ImageDataGenerator(horizontal_flip = True)
train_generator = datagen_train.flow_from_directory("../train/",
target_size = (img_size, img_size),
color_mode = "grayscale",
batch_size = batch_size,
class_mode = "categorical", # Clasificación categórica por 7 emociones
shuffle = True)

# Generando data para evaluación
datagen_validation = ImageDataGenerator(horizontal_flip = True)
validation_generator = datagen_validation.flow_from_directory("../test/",
target_size = (img_size, img_size),
color_mode = "grayscale",
batch_size = batch_size,
class_mode = "categorical", # Clasificación categórica por 7 emociones
shuffle = False)

```

Adicional a ello, se definió una clase que predecirá la expresión del rostro con ayuda de los archivos generados por el entrenamiento, [pesos\\_weight.h5](#) y [modelo.json](#). La predicción se ejecutó en cada fotograma del vídeo de la cámara del dispositivo y detectó primero el rostro, luego la expresión en él. La respuesta fue mediante un mensaje que se definió en una lista [LISTA\\_EMOCIONES](#), la cual contiene todas las posibles expresiones que el sistema podrá predecir. Mediante la línea `self.loaded_model.predict`, el sistema determinará el valor más

elevado y tomará su posición que coincide con la del elemento en la matriz `LISTA_EMOCIONES`. Por ejemplo, si la matriz de predicción es [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8], entonces el mayor valor será 0.8, el cual tiene la posición 7, luego el sistema tomará el valor en la lista `LISTA_EMOCIONES` con posición 7 que es `SORPRENDIDO`. Para fines de esta investigación, el sistema solo consideró la expresión de tristeza con posición 6. En caso que la predicción sea sobre otra expresión, simplemente retornará un valor vacío.

```
class FacialExpressionModel(object):

# LISTA_EMOCIONES = ["Molesto", "Disgustado", "De miedo", "Feliz",
"Neutral", "Triste", "Sorprendido"]

# En este caso, solo usaremos la tristeza
LISTA_EMOCIONES = ["", "", "", "", "", "Triste", ""]

# Inicializamos el modelo con los archivos obtenidos en el entrenamiento
def __init__(self, model_json_file, model_weights_file):
with open(model_json_file, "r") as json_file:
oaded_model_json = json_file.read()
self.loaded_model = model_from_json(loaded_model_json)
self.loaded_model.load_weights(model_weights_file)

# Función para predecir
def predict_emotion(self, img):
global session
set_session(session)
self.preds = self.loaded_model.predict(img)
return FacialExpressionModel.LISTA_EMOCIONES [np.argmax(self.preds)]
```

Las siguientes librerías fueron empleadas para abrir la cámara e importar el modelo.

```
#Librería OpenCV
import cv2

#Se importó el archivo model.py
from model import FacialExpressionModel
import numpy as np
```

En las siguientes líneas de código, se importó la librería Flask para ver el sistema mediante un navegador web. El sistema se debe abrir en una página, donde automáticamente empezará a detectar rostros y a predecir las emociones, haciendo uso de la cámara del computador. Los rostros detectados se enmarcaron y sobre ese cuadro, apareció el mensaje de la emoción que el modelo haciendo uso de la predicción, infirió.

```
from flask import Flask, render_template, Response
from camera import VideoCamera

# Inicializamos Flask
app = Flask(__name__)

# Llamamos a la cámara del sistema
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```



### c) Aplicar el reconocimiento de expresión facial en un sistema.

El modelo se ha entrenado y se ha aplicado a la cámara web integrada al dispositivo en donde se despliega el sistema. En primer lugar, se detectó el rostro del vídeo mediante la librería OpenCV y se aplicó en tiempo real el modelo entrenado al rostro para predecir expresiones de tristeza. Mientras más caminos tenga el modelo (expresiones) e imágenes relacionadas de distintos rostros, más efectividad alcanzará.

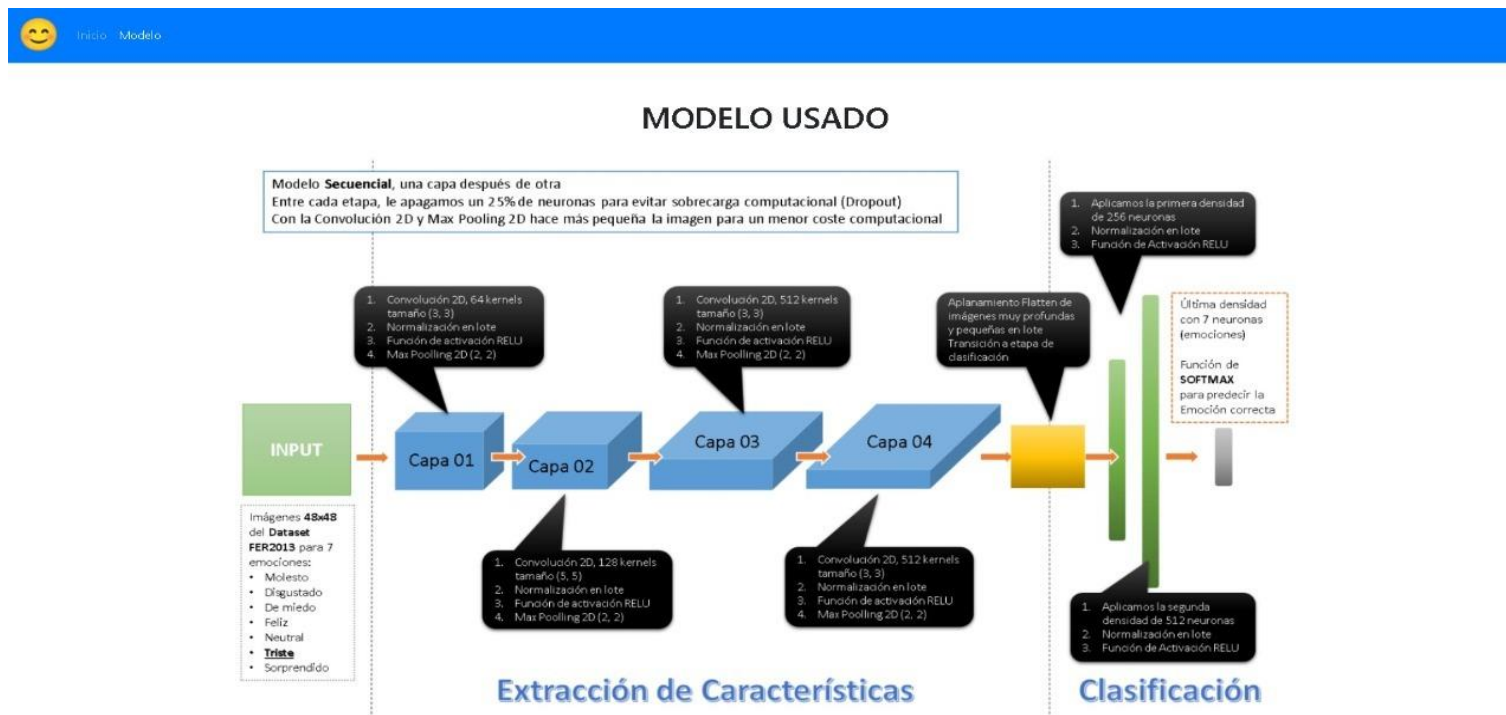


Figura 19. Interfaz Principal del Sistema haciendo referencia al modelo utilizado, el cual es modelo secuencial.

Fuente: Elaboración propia.

El sistema presenta dos opciones: [Modelo], que muestra el Modelo Secuencial, e [Inicio] donde se enciende la cámara web del ordenador y puede detectar la expresión facial. Lo comentado, se especifica mediante un print en la Figura 20.



Figura 20. Menú de Opciones del Sistema de Reconocimiento Facial. Fuente: Elaboración propia.

En consecuencia, luego se mostró el marco azul donde se enfoca el rostro, para efectuar la detección del mismo, y así reconocer la expresión facial que refleje la persona, para luego proceder con el entrenamiento de emociones.



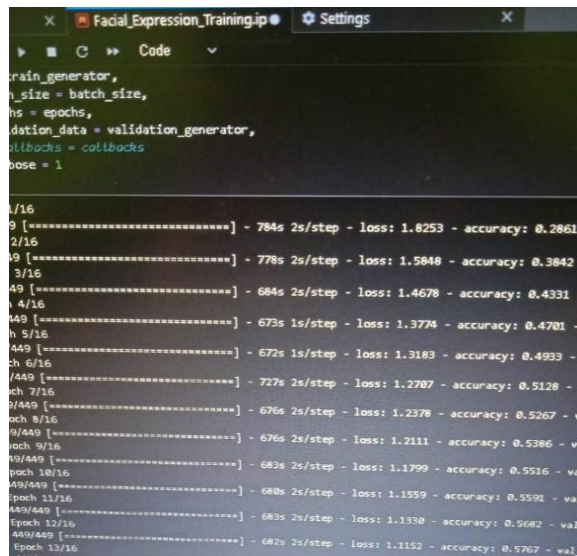
Figura 21. Evidencia de rostro triste, niño llorando. Fuente: Elaboración propia.

Como evidencia del archivo: [modelo.json] y [peso\_modelo.h5]. Los cuales se describieron previamente en las secciones anteriores, se agrega una imagen para evidenciar la creación y peso de los archivos. Lo cual se indicó en la Figura 22.



Figura 22. Evidencia de rostro triste, niño llorando. Fuente: Elaboración propia.

d) Realizar las pruebas y obtener resultados.



```
train_generator,  
batch_size = batch_size,  
steps_per_epoch = steps_per_epoch,  
validation_data = validation_generator,  
callbacks = callbacks,  
verbose = 1  
1/16  
9 [=====] - 784s 2s/step - loss: 1.8253 - accuracy: 0.2861  
2/16  
9 [=====] - 778s 2s/step - loss: 1.5848 - accuracy: 0.3842  
3/16  
49 [=====] - 684s 2s/step - loss: 1.4678 - accuracy: 0.4331 -  
4/16  
49 [=====] - 673s 1s/step - loss: 1.3774 - accuracy: 0.4701 -  
5/16  
449 [=====] - 672s 1s/step - loss: 1.3183 - accuracy: 0.4933 -  
6/16  
/449 [=====] - 727s 2s/step - loss: 1.2707 - accuracy: 0.5128 -  
7/16  
9/449 [=====] - 676s 2s/step - loss: 1.2378 - accuracy: 0.5267 -  
8/16  
9/449 [=====] - 676s 2s/step - loss: 1.2111 - accuracy: 0.5386 -  
9/16  
9/449 [=====] - 683s 2s/step - loss: 1.1799 - accuracy: 0.5516 -  
10/16  
49/449 [=====] - 688s 2s/step - loss: 1.1559 - accuracy: 0.5591 -  
11/16  
49/449 [=====] - 683s 2s/step - loss: 1.1338 - accuracy: 0.5682 -  
12/16  
449/449 [=====] - 682s 2s/step - loss: 1.1152 - accuracy: 0.5767 -  
Epoch 12/16
```

Figura 23. Etapa de entrenamiento en la época 12. Fuente: Elaboración propia.

Como se observó en la figura 23, en la etapa de entrenamiento la época 12, el accuracy es de 0.5682, lo cual es aceptable e indica que el sistema está funcionando. El sistema se desplegó mediante un navegador Web en el dispositivo, el cual se abrió la cámara y empezó a detectar automáticamente el rostro. Aquí tenemos algunos resultados: Imágenes de evidencia, en tiempo real: evidenciando el trabajo está la Figura 24, 25, 26 y 27. Las cuales se exponen en las siguientes líneas.

#### RECONOCIMIENTO DE EXPRESIONES FACIALES DE TRISTEZA UTILIZANDO APRENDIZAJE PROFUNDO

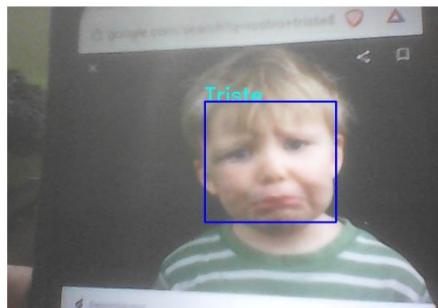
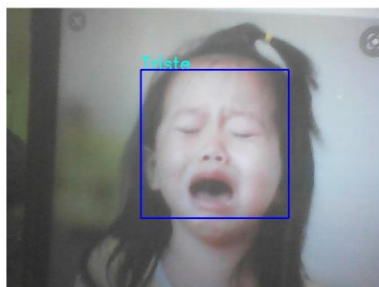


Figura 24. Evidencia de cara triste. Fuente: Elaboración propia.

RECONOCIMIENTO DE EXPRESIONES FACIALES DE TRISTEZA  
UTILIZANDO APRENDIZAJE PROFUNDO



*Figura 25.* Evidencia de cara triste de una niña. Fuente: Elaboración propia.

RECONOCIMIENTO DE EXPRESIONES FACIALES DE TRISTEZA  
UTILIZANDO APRENDIZAJE PROFUNDO



*Figura 26.* Evidencia de cara triste de un varón adulto. Fuente: Elaboración propia.

RECONOCIMIENTO DE EXPRESIONES FACIALES DE  
TRISTEZA UTILIZANDO APRENDIZAJE PROFUNDO



*Figura 27.* Evidencia de cara triste de Coronel Caján Erick Arturo. Fuente: Elaboración propia.

## IV. CONCLUSIONES Y RECOMENDACIONES.

### 4.1. Conclusiones.

- Se seleccionó el dataset apoyado en la investigación de corte científico, Facial emotion recognition on a dataset using convolutional neural network, donde se valieron de dicha data para desarrollar un sistema de reconocimiento de expresiones faciales basado en Redes neuronales convolucionales (CNN) que permitió clasificar automáticamente las expresiones faciales. Asimismo, se verificó el grado de usabilidad y si contaba con licencia de uso, para evidenciar la validez del conjunto de datos.
- Se efectuó una búsqueda de los algoritmos de deep learning, aplicados al tema de reconocimiento de expresiones faciales, y en base a los resultados de los indicadores de los algoritmos trabajados de las investigaciones leídas, se procedió a seleccionar la red neuronal convolucional, debido a su buen performance para la identificación de emociones humanas.
- Se desarrolló un sistema web en Python, haciendo uso de la librería OpenCV y la herramienta TensorFlow para detectar el rostro en tiempo real, donde se empleó el modelo convolucional de cuatro capas de conv2D y MaxPooling2D junto a 2 fases de Dense, con la finalidad de predecir la emoción en el rostro. Asimismo, se crearon las carpetas “test” y “train”, para el entrenamiento y prueba de la CNN respectiva.
- Se efectuaron las pruebas correspondientes, buscando observar si el entrenamiento dado fue correcto, donde se encontró que la CNN tuvo una exactitud, precisión, f score, superior al 90%. Sin embargo, en aspectos de consumo de memoria se identificó que requiere de un alto consumo, y en tiempos de ejecución la primera vez demandó muchos minutos, lo cual es lógico al contrastar con la cantidad de imágenes utilizadas. Finalmente, se obtuvo que en la época 50 alcanzó una precisión mayor a 96% debido al operador de convolución.

## 4.2. Recomendaciones.

- Se sugiere utilizar las librerías de Tensorflow paquete Keras, compilar código de redes neuronales profundas, Numpy, librería de funciones matemáticas con vectores y/o matrices, livelossplot, librería que permite la visualización de gráficas durante el entrenamiento de redes neuronales. (Anexo 03)
- Se recomienda efectuar nuevas investigaciones sobre el reconocimiento de expresiones faciales no sólo considerando la expresión tristeza sino también otras expresiones faciales.
- Se sugiere utilizar para desbloqueo de dispositivos con expresiones faciales: Realizar un sistema empático, si te detecta triste alertar emociones y trate de motivarte, así evitar alteraciones emocionales.

## REFERENCIAS.

- Alonso Sierra, J. D., & Castaño Saavedra , D. L. (2019). *SISTEMA DE RECONOCIMIENTO FACIAL PARA CONTROL DE ACCESO A VIVIENDAS*. Universidad Católica de Colombia. Obtenido de <https://repository.ucatolica.edu.co/bitstream/10983/24032/1/Final%20Trabajo%20de%20grado.pdf>
- Álvarez, V., Sánchez, C., Gutiérrez, S., Domínguez-Soberanes, J., & Velázquez, R. (2018). Facial Emotion Recognition: A Comparison of Different Landmark-Based Classifiers. El Salvador. doi:10.1109/RICE.2018.8509048
- Andrade, A. G. (20 de Febrero de 2019). The Neuroscience of Emotions: Society Seen from the Point of View of the Individual. An Approximation to the Link between Sociology and Neuroscience. México. Obtenido de [http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S0187-01732019000100039](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-01732019000100039)
- BlogThinkBig. (abril de 2019). *¿Sabes en qué se diferencian las redes neuronales del Deep Learning?* (P. Recuero, Editor) Obtenido de <https://blogthinkbig.com/redes-neuronales-deep-learning>
- De, A., & Saha, A. (2015). A comparative study on different approaches of real time human emotion recognition based on facial expression detection. India. doi:10.1109/ICACEA.2015.7164792
- Deng, L., & Yu, D. (2014). Aprendizaje profundo: métodos y aplicaciones", Fundamentos y tendencias® en el procesamiento de señales. *Aprendizaje profundo: métodos y aplicaciones, VII(3-4)*, 197-387. doi:<http://dx.doi.org/10.1561/20000000039>
- Ekman, P., & L. Rosenberg, E. (2012). *What the Face reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System*

(FACS) (SERIES IN AFFECTIVE SCIENCE) (Segunda ed.). New York: Oxford University Press.

González, E. J. (2014). *DETECCIÓN DE EMOCIONES DEL USUARIO*. PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO. Obtenido de [http://opac.pucv.cl/pucv\\_txt/txt-0000/UCD0248\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-0000/UCD0248_01.pdf)

He Li, J., Buenaposada, M., & Baumela, L. (2009). Real-time facial expression recognition with illumination-corrected image sequences. Netherlands. doi:10.1109/AFGR.2008.4813328

Kim, S., Hwan An, G., & Kang, S.-J. (2017). Facial expression recognition system using machine learning. Corea del Sur. doi:10.1109/ISOCC.2017.8368887

Liu, K.-C., Hsu, C.-C., Wang, W.-Y., & Chiang, H.-H. (2019). Real-Time Facial Expression Recognition Based on CNN. Vietnam. doi:10.1109/ICSSE.2019.8823409

M. Chawan, P., Manan, J., & Jinal, M. (2013). Real Time Emotion Recognition through Facial Expressions for Desktop Devices. Obtenido de [https://www.researchgate.net/publication/316511050\\_Real\\_Time\\_Emotion\\_Recognition\\_through\\_Facial\\_Expressions\\_for\\_Desktop\\_Devices](https://www.researchgate.net/publication/316511050_Real_Time_Emotion_Recognition_through_Facial_Expressions_for_Desktop_Devices)

Moreano, J. A. (2021). *Técnica eficiente para reconocimiento facial global utilizando wavelets y máquinas de vectores de soporte en imágenes 3D*. Universidad Nacional Mayor de San Marcos. Obtenido de [https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/16552/Cadena\\_mj.pdf?sequence=1](https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/16552/Cadena_mj.pdf?sequence=1)

Navarro Briones, J. D. (2020). *Sistema de Reconocimiento de Expresión Facial para la Detección de Emociones: Prototipo para Medir el Nivel de Satisfacción de Servicio al Cliente en la Recepción de la Carrera de Computación de la Universidad Católica Santiago de Guayaquil*. Trabajo de Titulación, Universidad Católica de Santiago de Guayaquil, FACULTAD DE



INGENIERIA, Guayaquil. Obtenido de  
<http://repositorio.ucsg.edu.ec/bitstream/3317/14344/1/T-UCSG-PRE-ING-CIS-252.pdf>

Pellejero, N. F., Grinblat, G., & Uzal, L. (2018). Semantic analysis on faces using deep neural networks. 14-29. Rosario, Argentina: IBERAMIA. doi:doi:10.4114/intartif.vol21iss61pp14-29

Peng, Z.-y., Zhu, Y.-h., & Zhou, Y. (2010). Real-time Facial Expression Recognition Based on Adaptive Canny Operator Edge Detection. China. doi:10.1109/MMIT.2010.100

Srivastava, S. (2012). Real Time Facial Expression Recognition Using a Novel Method. India. doi:10.5121/ijma.2012.4204

Towards Data Science. (2019). *Support Vector Machines for Classification*. Obtenido de <https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3>

Tümen, V., Faruk Söylemez, Ö., & Ergen, B. (2017). Facial emotion recognition on a dataset using convolutional neural network. Turquía. doi:10.1109/IDAP.2017.8090281

Whitehill, J., Littlewort, G., Wu, T., Fasel, I., Frank, M., Movellan, J., & Bartlett, M. (2011). The computer expression recognition toolbox (CERT). USA. doi:10.1109/FG.2011.5771414

Yang, J., Zhang, F., Chen, B., & Khan, S. (2019). Facial Expression Recognition Based on Facial Action Unit. EE.UU. doi:10.1109/IGSC48788.2019.8957163

Yolcu, G., Oztel, I., Kazan, S., Oz, C., Palaniappan, K., E. Lever, T., & Bunyak, F. (2017). Deep learning-based facial expression recognition for monitoring neurological disorders. USA. doi:10.1109/BIBM.2017.8217907

Yu, K., & Yin, J. (2018). The Study of Facial Gesture Based on Facial Features. China. doi:10.1109/IISR.2018.8535785

Ahonen, A. Hadid, and M. Pietikäinen, "Face Recognition with Local Binary Patterns.

Dai, J Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," May 2016.

García, B. (2015). Implementación de Técnicas de Deep Learning [Trabajo de Fin de Grado, Universidad de la Laguna]. <http://riull.ull.es/xmlui/handle/915/1409>

Gori, M. (2017). Machine Learning: A Constraint-Based Approach. Morgan Kaufmann

Goodfellow, I., Bengio, Y., y Courville, A. (2016). Deep Learning. MIT Press.

Hearty, J. (2016). Advanced Machine Learning with Python (Primera edición). Packt Publishing Ltd.

Hurwitz y Kirsch, (2018). Machine Learning For Dummies. John Wiley y Sons

Lecun et al. (1998) Lecun, y., Bottou, L., Bengio, Y., y Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. <https://doi.org/10.1109/5.726791>

Ponce, (2010). Inteligencia Artificial con aplicaciones a la ingeniería. (Primera Edición). Alfaomega

Rouse, Margaret. Aprendizaje profundo. Search data Center. (2'17)

Russell y Norvig, 2008 Russell. Inteligencia artificial: Un enfoque moderno. Pearson Educación.

Sáez, Adrina (2019) Deep learning para el reconocimiento facial de emociones básicas. Universidad Politécnica de Catalunya.

<https://github.com/opencv/opencv/tree/master/data/haarcascades>.

<https://www.kaggle.com/msambare/fer2013>.

## ANEXOS.

### ANEXO 01: Resolución de Aprobación de Proyecto de Investigación.



UNIVERSIDAD  
SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°2363-2020/FIAU-USS

Pimentel, 16 de diciembre de 2020

#### VISTO:

El Acta de reunión N°1012 - 2020/FIAU-ISS del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante mensaje de correo electrónico del 15 de diciembre de 2020, y;

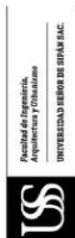
#### CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la Facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, según documentos de Visto el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerda aprobar los temas de las Tesis a cargo de los egresados que se detallan en el anexo de la presente Resolución.



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO**

**RESOLUCIÓN N°2363-2020/FIAU-USS**

Pimentel, 16 de diciembre de 2020

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

**SE RESUELVE:**

**ARTÍCULO 1°: APROBAR**, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los egresados del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

**ARTÍCULO 2°: ESTABLECER**, que la inscripción del Tema de la Tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

**ARTÍCULO 3°: DEJAR SIN EFECTO**, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

**REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE**

  
**USS** Dr. Mario Fernando Ramos Mescol  
Decano - Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

  
**USS** MBA. María Noelia Sialer Rivera  
Secretaria Académica / Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

**ANEXO**

Orden	Apellidos y Nombres	TITULO DEFINITIVO PROYECTO DE INVESTIGACIÓN
1	ALMENDRAS FLORES RONALD ELADIO	COMPARACIÓN DEL RENDIMIENTO DE PROTOCOLOS DE COMUNICACIÓN DE DISPOSITIVOS IOT PARA ASEGURAR LOS DATOS EN EL ÁMBITO DE LA SALUD
2	BOCANEGRA PINCHI YAN CARLOS	DESARROLLO DE UNA METODOLOGÍA PARA LA ADQUISICIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN ORIENTADA A PEQUEÑAS ORGANIZACIONES BASADA EN ESTÁNDARES
3	BRENIS LLAGENTO JULIO ANTONIO	COMPARACIÓN DE ALGORITMOS DE AUTENTICACIÓN EN CONEXIONES DE REDES PRIVADAS VIRTUALES PARA USO EN TRABAJO REMOTO. CASO DE ESTUDIO: MUNICIPALIDAD PROVINCIAL DE FERREÑAFE
4	BUSTAMANTE GUERRERO FERNANDO LUCERO CHUCAS LUIS ALBERTO	EVALUACIÓN DE LA CALIDAD DE SOFTWARE DE LAS APLICACIONES EMBEBIDAS EN DISPOSITIVOS PUNTOS DE VENTA BASADOS EN ESTANDARES
5	CABANILLAS TORRES ALVARO PAUL HARBERT FARRO VARGAS LUIS ADEMAR	DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA EVALUACIÓN DE LOS PERFILES POR COMPETENCIA LABORAL DE UN PUESTO GERENCIAL
6	CABRERA GUEVARA EDSON OMAR ROQUE IZQUIERDO ADELMO	DESARROLLO DE UNA METODOLOGÍA ÁGIL AD HOC PARA LA CREACIÓN DE APLICACIONES MÓVILES EN PEQUEÑAS EMPRESAS. CASO DE ESTUDIO: AMÉRICA PERÚ S.A.C, LIMA.
7	CAMPOS BARRERA SANDRO PAUL PASTOR OLIVA CESAR AUGUSTO	DETECCIÓN DE PATRONES DE COMPORTAMIENTO EN LA DESERCIÓN DE ESTUDIANTES DE LA ESCUELA PROFESIONAL DE INGENIERÍA DE INDUSTRIAS ALIMENTARIAS DE LA UNIVERSIDAD NACIONAL DE JAÉN UTILIZANDO MINERÍA DE DATOS
8	CARRASCO MARTINEZ LUIS MARLON	EVALUACIÓN DE MÉTODOS DE DETECCIÓN DE INTRUSOS PARA BLOQUEAR Y NOTIFICAR ATAQUES EN LA ZONA DESMILITARIZADA DE LA RED
9	CARRERÑO CASTILLO JORGE LUIS SALAZAR AGUILAR LUIS	EVALUACIÓN DE ALGORITMOS PARA MEDIR EFICIENCIA EN EL TRÁFICO OCULTO DE VOZ IP
10	CARRERA PEÑA LUIS ARMANDO	COMPARACIÓN DE TÉCNICAS DE CLASIFICACIÓN EN EL RECONOCIMIENTO DE CARACTERES DE DÍGITOS DE IMAGEN DIGITAL DE MEDIDORES ELÉCTRICOS
11	CARRERA SANCHEZ JOSE ROSVEL	DESARROLLO DE LA ARQUITECTURA EMPRESARIAL BASADO EN MARCOS DE TRABAJO PARA EL ALINEAMIENTO ESTRATÉGICO DE TECNOLOGÍAS DE INFORMACIÓN. CASO DE ESTUDIO: COLEGIO RAMÓN CASTILLA – BAGUA GRANDE
12	CARRILLO CAJO CYNTHIA EFIGENIA PISCOYA INOÑAN SANDRA PAOLA	DESARROLLO DE UNA PLATAFORMA MÓVIL COMO APOYO A LA ATENCIÓN DE PACIENTES EN HOSPITALES Y CLÍNICAS UTILIZANDO CLOUD COMPUTING Y GEOREFERENCIACIÓN
13	CELIS SANCHEZ SUJEILY PEREZ ROJAS FRANKLIN ALEXI	MODELO DE EVALUACIÓN DE PROCESOS TI CON BASE EN EL MARCO DE REFERENCIA COBIT 5 PAM. CASO DE ESTUDIO SIEMPRESOFT E.I.R.L. 2020 - CHICLAYO
14	CONDOR ROJAS JHAIR	DESARROLLO DE UNA METODOLOGÍA DE GESTIÓN DE RIESGOS DE CIBERSEGURIDAD PARA UNA MUNICIPALIDAD PERUANA. CASO DE ESTUDIO: MUNICIPALIDAD DE CHICLAYO
15	CORONEL CAJAN ERICK ARTURO	RECONOCIMIENTO DE EXPRESIONES FACIALES DE TRISTEZA UTILIZANDO APRENDIZAJE PROFUNDO
16	CORTEZ BURGOS JOHANDER ENRIQUE MEDRANO MORI JOSE LUIS	DESARROLLO DE UN MODELO DE PROCESOS PARA UNA FÁBRICA PERUANA DE SOFTWARE BASADA EN METODOLOGÍAS ÁGILES CASO DE ESTUDIO CONASTEC S.R.L.
17	GASTELO FERNANDEZ EDIN JONNY RODRÍGUEZ FLORES ALFREDO HIPÓLITO	DESARROLLO DE UN MODELO DE GESTIÓN DE RIESGOS BASADO EN LA METODOLOGÍA MAGERIT PARA MINIMIZAR LOS RIESGOS DE ADQUISICIÓN Y USO DE TI EN UNA MUNICIPALIDAD DEL PERÚ. CASO DE ESTUDIO: MUNICIPALIDAD DISTRITAL DE CUPISNIQUE - CAJAMARCA
18	GUERRERO SERNAQUE LUIGGI EMILIO LOZA VARGAS FERNANDO AUGUSTO	DESARROLLO DE UNA APLICACIÓN MIDDLEWARE PARA EL ENVÍO Y VALIDACIÓN DE COMPROBANTES ELECTRÓNICOS A SUNAT
19	HUAMAN CASAS JUNIOR ALDAIR SERRATO VILCHERRES FERNANDO JOSE	DESARROLLO DE UN MÉTODO PARA DETECCIÓN DE FRAUDES DE PAGOS EN LÍNEA UTILIZANDO APRENDIZAJE AUTOMÁTICO
20	IDROGO CORNEJO LEONCIO FERNANDEZ SALDAÑA CARLOS	DESARROLLO DE UN MÉTODO DE CLASIFICACIÓN AUTOMÁTICA DE CITRUS AURANTIFOLIA USANDO PROCESAMIENTO DIGITAL DE IMÁGENES
21	LOPEZ BARRETO ALDO HASSAN	COMPARACIÓN DE PROTOCOLOS DE REDES PRIVADAS VIRTUALES PARA AUMENTAR LA SEGURIDAD EN EL CONTEXTO DE TELETRABAJO

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO**
**RESOLUCIÓN N° 2363-2020/FIAU-USS**

Pimentel, 16 de diciembre de 2020

22	MENDOZA RENGIFO GENARO	DESARROLLO DE UNA METODOLOGÍA ÁGIL AD HOC PARA LA CREACIÓN DE APLICACIONES WEB EN PEQUEÑAS EMPRESAS. CASO DE ESTUDIO: SOLTI S.A.C.
23	MONTALVO SANDOVAL JOSE LUIS RUBIO OTERO DANIEL	DESARROLLO DE UN MÉTODO DE IDENTIFICACIÓN DE DEFECTOS EXTERNOS DEL MANGIFERA INDICA L USANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y APRENDIZAJE DE MÁQUINA
24	ÑAÑEZ PEREDO ALEXIS JOEL	COMPARACIÓN DE ALGORITMOS DE DETECCIÓN DE BORDES PARA UNA EFECTIVA VECTORIZACIÓN DE IMÁGENES DIGITALES DE MOLDES TEXTILES
25	ORTEGA PUENTE OSCAR FERNANDO	DESARROLLO DE UNA APLICACIÓN MÓVIL DE REPORTE CIUDADANO UTILIZANDO IMÁGENES Y GEOLOCALIZACIÓN
26	PARDO VILLEGAS JOSE DIEGO SALINAS VELEZ BRYAN JOSE RICARDO	COMPARACIÓN DE TÉCNICAS DE ESTIMACIÓN DEL GRADO DE SALINIDAD EN SUELOS DE ESCASA VEGETACIÓN MEDIANTE EL PROCESAMIENTO DE IMÁGENES MULTIESPECTRALES DE SATÉLITE
27	RAVELO RUIZ ALLEN MARCEL	EVALUACIÓN DE ALGORITMOS DE PREDICCIÓN APLICADOS AL MANTENIMIENTO PREDICTIVO DE MÁQUINAS ELÉCTRICAS ROTATIVAS
28	REQUEJO NAVARRO JERSONS EXFRANSHER	EVALUACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR SEGURIDAD EN UNA RED PRIVADA VIRTUAL
29	SALAZAR DÁVILA GIANFRANCO STEVEN	COMPARACIÓN DE TÉCNICAS DE VALIDACIÓN DE REQUERIMIENTOS DE SOFTWARE EN PEQUEÑAS EMPRESAS
30	SANTISTEBAN VALDERA ELVIS ALBERTO VILLANUEVA CASTRO JOSE SEBASTIAN	DESARROLLO DE UN MODELO DE GESTIÓN DE INCIDENCIAS BASADO EN MARCOS DE REFERENCIA DE BUENAS PRÁCTICAS PARA MEJORAR EL SERVICIO DE TI. CASO DE ESTUDIO MUNICIPALIDAD DISTRITAL DE MÓRROPE – LAMBAYEQUE
31	THEOLOGITIS SANCHEZ DIMITRIS IOANNIS	CONTROL EN EL ABASTECIMIENTO DEL INVENTARIO BASADO EN REDES NEURONALES PARA PEQUEÑAS Y MEDIANAS EMPRESAS

**ANEXO 02:** Formatos de Instrumentos de recolección de datos del Proyecto de Investigación.

<b>CONSUMO DE RECURSOS</b>			
<b>USO DE CONSUMO DE CPU</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	49,8

<b>USO DE MEMORIA</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	78,9%

<b>TIEMPO DE EJECUCIÓN</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	35737 s.

<b>RENDIMIENTO</b>			
<b>INDICADOR EXACTITUD</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	0,9120

<b>INDICADOR PRECISION</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	0,9050

<b>INDICADOR RECALL</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	0,8383

<b>INDICADOR F1 SCORE</b>			
Algoritmo	Fecha	Hora	Valor
Red Neuronal CNN	29-06-2021	09:50:10	0,8449

<b>INDICADOR AUC SCORE</b>			
Algoritmo	Fecha	Hora	Valor

**ANEXO 03:** Codificación del sistema de detección facial utilizando red CNN.

training/trainer.py: Archivo necesario para entrenar el modelo.

# Importación de librerías requeridas para la implementación del sistema.

```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
import os
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
```

```
from tensorflow.keras.layers import Dense, Input, Dropout, Flatten, Conv2D
```

```
from tensorflow.keras.layers import BatchNormalization, Activation, MaxPooling2D
```

```
from tensorflow.keras.models import Model, Sequential
```

```
from tensorflow.keras.optimizers import Adam
```

```
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
```

```
from livelossplot import PlotLossesKeras
```

```
from tensorflow.keras.utils import plot_model
```

```
from IPython.display import SVG, Image
```

```
from livelossplot import PlotLossesKerasTF
```

```
from tensorflow.keras import backend as K
```

# Nos aseguramos de finalizar todas las sesiones de Keras existentes para empezar con una sesión nueva.

```
K.clear_session()
```

# Ploteando imágenes. Aquí recorreremos las imágenes de entrenamiento y evaluación y los redimensionamos a escala de 48 x 48 píxeles.

```
def plot_example_images(plt):
```

```
img_size = 48
```



```

plt.figure(0, figsize=(12,20))
plt2.figure(0, figsize=(12,20))
ctr = 0

for expression in os.listdir("train/"):
for i in range(1,6):
ctr += 1
plt.subplot(7,5,ctr)
img = load_img("train/" + expression + "/" +os.listdir("train/" + expression)[i], target
_size=(img_size, img_size))
plt.imshow(img, cmap="gray")

for expression in os.listdir("test/"):
for i in range(1,6):
ctr += 1
plt2.subplot(7,5,ctr)
img = load_img("test/" + expression + "/" +os.listdir("test/" + expression)[i], target_s
ize=(img_size, img_size))
plt2.imshow(img, cmap="gray")

plt.tight_layout()
plt2.tight_layout()
return plt

# Imprimiendo carpetas de entrenamiento y validación.
for expression in os.listdir("../train/"):
print(str(len(os.listdir("../train/" + expression))) + " - Emoción de entrenamiento: " +
expression)

for expression in os.listdir("../test/"):
print(str(len(os.listdir("../test/" + expression))) + " - Emoción de validación: " + expr
ession)

```

# Inicializamos el tamaño de las imágenes a 48 x 48 y la cantidad de lote de imágenes para procesar.

```
img_size = 48
```

```
batch_size = 64
```

# Cantidad de carpetas de emociones, posibles predicciones.

```
clases = 7
```

# learning\_rate, ajustes de la red neuronal para acercarse a la predicción óptima.

```
lr = 0.0005
```

# Cantidad de épocas que va a iterar el modelo.

```
epochs = 50
```

# Empezamos a preprocesar las imágenes. Generando data para entrenamiento.

```
datagen_train = ImageDataGenerator(horizontal_flip = True)
```

```
train_generator = datagen_train.flow_from_directory("../train/",
```

```
target_size = (img_size, img_size),
```

```
color_mode = "grayscale",
```

```
batch_size = batch_size,
```

```
class_mode = "categorical", # Clasificación categórica por 7 emociones.
```

```
shuffle = True)
```

# Generando data para evaluación.

```
datagen_validation = ImageDataGenerator(horizontal_flip = True)
```

```
validation_generator = datagen_validation.flow_from_directory("../test/",
```

```
target_size = (img_size, img_size),
```

```
color_mode = "grayscale",
```

```
batch_size = batch_size,
```

```
class_mode = "categorical", # Clasificación categórica por 7 emociones.
```

```
shuffle = False)
```

# Creación del modelo de red neuronal convolucional. Inicializamos en modelo.

# Secuencial porque va en varias capas apiladas, una después de otra.

```
model = Sequential()
```

```
# Agregamos las capas al modelo.
```

```
# Capa 01, el modelo agrega la clase Conv2D es decir altura y ancho, 64 kernels (nro filtros) cuyas dimensiones son (3,3) que representan el tamaño de los kernels, padding same intenta rellenar uniformemente hacia la izquierda y hacia la derecha, batch normalization es un método que normaliza cada lote de datos (batch_size), normalizando así los datos las distancias de los datos van de 0 a 1 y esto ayuda a la red neuronal a trabajar mejor. La función RELU, convierte cualquier número negativo en 0, dejando los positivos intactos. Max pooling (2,2) para reducir las dimensiones espaciales del volumen de salida, con el fin de reducir el coste computacional y minimizando la posibilidad de overfitting. Finalmente, Dropout de (0.25) para reducir el sobreajuste.
```

```
model.add(Conv2D(64, (3, 3), padding = 'same', input_shape = (img_size, img_size, 1)))  
model.add(BatchNormalization())  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool_size = (2, 2))) #  
model.add(Dropout(0.25))
```

```
# Capa 02, el modelo agrega la clase Conv2D es decir altura y ancho, 128 kernels (nro filtros) cuyas dimensiones son (5,5) que representan el tamaño de los kernels, batch normalization para normalizar cada lote de datos (batch_size) en función de 0 a 1 para un mejor desempeño. La función RELU, para la obtención de positivos. Max pooling (2,2) para reducir las dimensiones espaciales del volumen de salida, con el fin de reducir el coste computacional y minimizando la posibilidad de overfitting. Finalmente, Dropout de (0.25) para reducir el sobreajuste.
```

```
model.add(Conv2D(128, (5, 5), padding = 'same'))  
model.add(BatchNormalization())  
model.add(Activation('relu'))
```

```
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(0.25))
```

# Capa 03, el modelo agrega la clase Conv2D es decir altura y ancho, 512 kernels (nro filtros) cuyas dimensiones son (3,3) que representan el tamaño de los kernels, batch normalization para normalizar cada lote de datos en función de 0 a 1 para un mejor desempeño. La función RELU, para la obtención de positivos. Max pooling (2,2) para reducir las dimensiones espaciales del volumen de salida, con el fin de reducir el coste computacional y minimizando la posibilidad de overfitting. Finalmente, Dropout de (0.25) para reducir el sobreajuste.

```
model.add(Conv2D(512, (3, 3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(0.25))
```

# Capa 04, el modelo agrega la clase Conv2D es decir altura y ancho, 512 kernels (nro filtros) cuyas dimensiones son (3,3) que representan el tamaño de los kernels, batch normalization para normalizar cada lote de datos en función de 0 a 1 para un mejor desempeño. La función RELU, para la obtención de positivos. Max pooling (2,2) para reducir las dimensiones espaciales del volumen de salida, con el fin de reducir el coste computacional y minimizando la posibilidad de overfitting. Finalmente, Dropout de (0.25) para reducir el sobreajuste.

```
model.add(Conv2D(512, (3, 3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(0.25))
```

# Aquí aplanamos las imágenes muy profundas y pequeñas, una dimensión con toda la información.

```

model.add(Flatten())
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))

# Bajamos el 25% de las neuronas para aprender caminos alternos, para adaptarse
# mejor a la nueva información, con diferentes emociones.
model.add(Dropout(0.25))
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))

# Reducimos nuevamente el 25% de las neuronas.
model.add(Dropout(0.25))

# Esta capa sirve para predecir entre las clases o emociones en el modelo. La
# probabilidad más alta es la predicción correcta.
model.add(Dense(classes, activation = 'softmax'))
opt = Adam(lr = lr)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['accuracy'])

# Imprimimos las estadísticas del modelo.
model.summary()

# Entrenando y evaluando el modelo.
reduce_lr = ReduceLROnPlateau(monitor = 'val_loss', factor = 0.1, patience = 2, min_lr = 0.00001, mode = 'auto')

# Guardamos los pesos obtenidos en cada época.
checkpoint = ModelCheckpoint("../pesos_modelo.h5", monitor = 'val_accuracy', save_weights_only = True, mode = 'max', verbose = 1)

```

```

# Esto para imprimir la gráfica de pérdida y ganancia.
callbacks = [PlotLossesKerasTF(), checkpoint, reduce_lr]
history = model.fit(
batch_size = batch_size,
x = train_generator,
epochs = epochs,
validation_data = validation_generator,
callbacks = callbacks)
# Representación de modelo entrenado en JSON, es la estructura del modelo, lo
guardamos en model.json
model_json = model.to_json()
with open("../modelo.json", "w") as json_file: json_file.write(model_json)

```

model.py: Archivo para predecir la emoción.

```

# Importamos paquetes necesarios.
from tensorflow.keras.models import model_from_json
from tensorflow.python.keras.backend import set_session
import numpy as np
import tensorflow as tf

# Configuración necesaria para optimizar la sesión de la cámara.
config = tf.compat.v1.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.15
session = tf.compat.v1.Session(config=config)
set_session(session)

# Clase para predecir la emoción.
class FacialExpressionModel(object):

# Todas las posibles salidas, por orden de lectura de carpetas.
# LISTA_EMOCIONES = ["Molesto", "Disgustado", "De miedo", "Feliz", "Neutral", "
Triste", "Sorprendido"]

```

```
# En esta oportunidad, sólo usaremos la tristeza.
```

```
LISTA_EMOCIONES = ["", "", "", "", "", "Triste", ""]
```

```
# Inicializamos el modelo con los archivos obtenidos en el entrenamiento.
```

```
def __init__(self, model_json_file, model_weights_file):  
    with open(model_json_file, "r") as json_file:  
        loaded_model_json = json_file.read()  
        self.loaded_model = model_from_json(loaded_model_json)  
        self.loaded_model.load_weights(model_weights_file)
```

```
# Función para predecir.
```

```
def predict_emotion(self, img):  
    global session  
    set_session(session)
```

```
# En este paso, vamos a obtener una matriz con los porcentajes de las predicciones  
como [0, 0, 0, 0, 1, 0, 0], el 1 representa el valor más alto.
```

```
self.preds = self.loaded_model.predict(img)
```

```
# Los valores coinciden con el arreglo de emociones, seleccionamos la más alta y  
la retornamos, por ejemplo, [0, 0, 0, 0, 1, 0, 0] => "Neutral"
```

```
return FacialExpressionModel.LISTA_EMOCIONES[np.argmax(self.preds)]
```

```
camera.py: Archivo para abrir la cámara del sistema.
```

```
# Importamos paquetes necesarios, Librería OpenCV.
```

```
import cv2.  
from model import FacialExpressionModel  
import numpy as np
```

```
# Cargamos el haarcascade.
```

```
facec = cv2.CascadeClassifier('./haarcascade/haarcascade_frontalface_default.xml')
```

```
# Cargamos el archivo para predecir y le pasamos los archivos generados por el entrenamiento.
```

```
model = FacialExpressionModel("modelo.json", "pesos_modelo.h5")
```

```
# Cargamos el tipo de font de la cámara.
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
class VideoCamera(object):
```

```
def __init__(self):
```

```
self.video = cv2.VideoCapture(0)
```

```
def __del__(self):
```

```
self.video.release() # Terminar la sesión de cámara
```

```
def get_frame(self):
```

```
_, fr = self.video.read() # Leemos la cámara
```

```
gray_fr = cv2.cvtColor(fr, cv2.COLOR_BGR2GRAY)
```

```
faces = facec.detectMultiScale(gray_fr, 1.3, 5)
```

```
# Algoritmo para abrir la cámara y retornarlo en archivo .jpeg
```

```
for (x, y, w, h) in faces:
```

```
fc = gray_fr[y:y+h, x:x+w]
```

```
roi = cv2.resize(fc, (48, 48))
```

```
pred = model.predict_emotion(roi[np.newaxis, :, :, np.newaxis])
```

```
cv2.putText(fr, pred, (x, y), font, 1, (255, 255, 0), 2)
```

```
cv2.rectangle(fr, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

```
_, jpeg = cv2.imencode('.jpg', fr)
```

```
return jpeg.tobytes()
```

```
main.py: Archivo para empezar a llamar al sistema.
```



```

# Flask para interpretar código python en la web mediante un puerto.
from flask import Flask, render_template, response
from camera import VideoCamera

# Inicializamos Flask.
app = Flask(__name__)
# Llamamos a la cámara del sistema.
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

# Creamos las rutas para ser leídas por la web.
@app.route('/') # Ruta principal
def index():
    return render_template('index.html')

@app.route('/modelo') # Ruta para ver el modelo en un gráfico
def modelo():
    return render_template('modelo.html')

@app.route('/video_feed') # Ruta para llamar a la cámara
def video_feed():
    return Response(gen(VideoCamera()), mimetype = 'multipart/x-mixed-
replace; boundary=frame')

# Definimos puerto para desplegar sistema en la web y le definimos como en desa
rollo True para detección de cambios.
if __name__ == '__main__':
    app.run(host='localhost', debug = True)

```