Aprendizaje por transferencia usando redes neuronales convolucionales para la clasificación de emociones en rostros

Marco Tulio Flórez Flórez

Trabajo de grado presentado para optar al título de Ingeniero de Telecomunicaciones

Director:
Juan Rafael Orozco Arroyave, Doctor (PhD)

Asesor:
Cristian David Ríos Urrego, Ingeniero Electrónico

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Telecomunicaciones
Medellín, Antioquia, Colombia
2022

Cita	(Flórez Flórez Marco Tulio, 2022)
Referencia **Estilo APA 7 (2020)**	Flórez Flórez Marco Tulio, (2022). Aprendizaje por transferencia usando redes neuronales convolucionales para la clasificación de emociones en rostros [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia.

Grupo de Investigación GITA UdeA.

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Augusto Enrique Salazar Jiménez.

Agradecimientos

Primero que todo le agradezco a Dios por bendecirme y permitirme haber logrado este sueño. A mi asesor Juan Rafael Orozco y a mi co-asesor Cristian David Ríos, por guiarme en todo este proceso de formación. A mis padres José Julio Flórez y Mireya del Carmen Flórez, quienes, con su esfuerzo me han brindado todo el apoyo incondicional para salir adelante. A mis hermanos José Ángel y Julio Cesar, quienes, siempre me han aconsejado a tomar las mejores decisiones. A mi compañera Wendys Vanessa y mis hijas María Camila y Mariana, quienes se han convertido en la motivación más grande para alcanzar mis metas. A mis abuelos Ángel Rafael y Cruz María que desde el cielo siempre me están acompañando.

Índice

Índice de figuras

Índice de tablas

Resumen

En este trabajo se propone un enfoque basado en el aprendizaje por transferencia (del inglés, Transfer Learning, TL), este método consiste en usar y/o ajustar modelos previamente entrenados para mejorar el rendimiento de una tarea objetivo. Su eficiencia radica en el ahorro de tiempo y recursos al no tener que entrenar modelos desde cero. Esta técnica se implementó en la clasificación de cuatro emociones: neutro, enojado, feliz y triste, con el fin de comprender el comportamiento de una persona frente a los acontecimientos que se presentan en un entorno dado. Reconocer las emociones mencionadas anteriormente, resulta de gran utilidad a la hora de realizar aplicaciones dentro de este ámbito, por ejemplo, en el área de la educación, los profesores pueden identificar el nivel de atención de sus alumnos a través de sus expresiones faciales. Otra área de aplicación es la seguridad, donde a partir de cámaras de vigilancia se pueda obtener información útil del estado emocional de una persona, que refleje en sus expresiones posibles amenazas para la seguridad propia o de terceros. Motivados por esto, en este trabajo se propone abordar el problema de clasificación de emociones en rostros implementando tres arquitecturas de redes neuronales convolucionales usando la base de datos FER-2013 para obtener modelos base y emplearlos en el aprendizaje por transferencia hacia la base de datos Cohn-Kanade, esto con el fin de mejorar la eficiencia de los modelos implementados para la clasificación de cuatro emociones: neutro, feliz, triste y enojado. Particularmente, en este trabajo se realizarón 5 experimentos con el fin de comparar y comprobar si la técnica de aprendizaje por transferencia mejora diferentes métricas de desempeño: **I** Implementación y evaluación de diferentes redes neuronales convolucionales (VGG-16, AlexNet y LeNet) utilizando la base de datos FER-2013. **II** Implementación y evaluación de redes neuronales convolucionales (VGG-16, AlexNet y LeNet) utilizando la base de datos Cohn-Kanade. **III** Implementación y evaluación de aprendizaje por transferencia desde los modelos creados con la base de datos FER-2013 para la clasificación de Cohn-Kanade. **IV** Implementación y comparación de métodos clásicos de clasificación como: máquinas de soporte vectorial (del inglés, Support Vector Machines, SVM), bosques aleatorios (del inglés, Random Forest, RF) y el algoritmo de aumento de gradiente extremo (del inglés, Extreme Gradient Boosting, XGBoost) a partir de representaciones intermedias obtenidas de las CNNs usando la técnica de triple pérdida. **V** Clasificación biclase a partir del plano de Arousal y Valencia, utilizando métodos clásicos sobre la base de datos Cohn-Kanade.

En general los resultados muestran para los distintos experimentos realizados que la técnica de aprendizaje por transferencia incrementa el desempeño de la clasificación de emociones en rostros de personas en comparación a modelos entrenados desde cero. Al igual, se puede observar que la técnica triple pérdida en conjunto con métodos de clasificación clásicos, logran resultados comparables a los obtenidos a partir de redes neuronales profundas.

Palabras clave: Clasificación de emociones, redes neuronales convolucionales, aprendizaje por transferencia.

1. Introducción

1.1. Contexto

Las emociones representan una parte importante en la comunicación humana, son el reflejo de las ideas y pensamiento de las personas. Investigaciones psicológicas realizadas por Mehrabian [1] han demostrado que el lenguaje corporal contribuye en un 55% del reconocimiento de la comunicación no verbal de nuestro cuerpo, mientras que un 38% se manifiestan mediante propiedades acústicas como el tono y la energía del habla. Finalmente, un 7% de los mensajes es aportado por expresiones verbales. Con el rostro no solo se distingue las características y la identidad de una persona, también es posible realizar aplicaciones en procesos de ingeniería tales como: detección de patologías [2], generación de contenido multimedia [3] y seguridad mediante dispositivos electrónicos [4]. En la actualidad con el avance de la inteligencia artificial se destaca el reconocimiento de patrones como un procedimiento que se centra en la extracción automática de características de una señal de entrada, haciendo que los patrones permitan la detección, predicción, clasificación y toma de decisiones en un problema específico [5]. En el área de la visión por computadora el reconocimiento de patrones juega un papel muy importante a la hora de clasificar un conjunto de datos, ya que permite obtener información a partir de imágenes en su entrada y luego procesarla para obtener una descripción de la imagen analizada mediante el uso de diferentes operaciones matemáticas (convolución, promedio, máximos, mínimos, entre otros). Además se ha demostrado que las redes neuronales convolucionales (del inglés, Convolutional Neural Network, CNN) son efectivas en la tarea de detección y clasificación de imágenes, estas redes están constituidas principalmente por capas convolucionales, capas de agrupación y capas

totalmente conectadas [6]. Según Yann LeChun en [7], el reconocimiento de expresiones faciales basado en el aprendizaje profundo es uno de los métodos más utilizados para la detección de emociones del ser humano, donde las CNNs están entre las arquitecturas más usadas para la clasificación y el reconocimiento de imágenes. Una estrategia importante que se emplea en las CNNs es el aprendizaje por transferencia que permite a partir de un modelo base ajustar los parámetros para clasificar una base de datos objetivo, esta transferencia de conocimiento se puede hacer mediante técnicas como el ajuste fino o la extracción de características, con el fin de mejorar el desempeños de los sistemas implementados.

1.2. Estado del arte

En la literatura se ha demostrado que con la utilización de diferentes tipos de arquitecturas de CNNs que fueron diseñadas por comunidad científica (VGG-16, AlexNet y GoogleNet), se han logrado obtener buenos resultados en el reconocimiento de imágenes. Los autores en [8] reportaron resultados del 71.0% de exactitud para el modelo VGG-16, un 69.0% de exactitud para AlexNet y un 70.0% de exactitud para GoogleNet con el conjunto de datos FER-2013. Por otro lado en [9] se implementó la arquitectura VGG-16 obteniendo tasas de precisión del 84.6% utilizando la base de datos de expresiones faciales femenina japonesa (del inglés, Japanese Female Facial Expression, JAFFE) y una exactitud del 65.8% para imágenes de rostros de todo tipo de personas sin importar la edad (FER-3013). En [10] se propone un modelo hibrido de aprendizaje por transferencia basado en máquinas de Boltzmann restringida por convolución y un modelo CNN (Inception-v3) para la clasificación de imágenes, obteniendo resultados de hasta un 73.7% de precisión para el reconocimiento de emociones en el conjunto de datos FER-2013. En [11] se propone un método de aprendizaje profundo basado en el aprendizaje por transferencia implementando la técnica de ajuste fino usando la arquitectura AlexNet para el reconocimiento de emociones faciales, logrando una precisión del 99.4% y del 70.5% para el conjunto de datos Cohn-Kanade y FER-2013. En [12] los autores propusieron un modelo creado desde cero inspirado en la arquitectura VGG, utilizando la técnica de aprendizaje por transferencia obteniendo índices de reconocimientos de emociones de hasta 76.6% haciendo uso de la base de datos FER-2013, respectivamente. En [13] se llevó a cabo el reconocimiento de emociones faciales mediante el uso de técnicas de aprendizaje por transferencia utilizando cuatro tipos de CNNs (VGG-19, Resnet50, Inception-V3 y MobileNet), pre-entrenadas con la base de datos ImageNet [14]. Los autores utilizaron la base de datos Cohn-

Kanade logrando una precisión del 96% para VGG-19, 97.7% para Resnet50, 98.5% para Inception-V3 y el 94.2% para MobileNet. En [15] los autores propusieron el reconocimiento de emociones faciales mediante técnicas como histogramas de gradientes orientados (del inglés, Histogram of Oriented Gradients, HOG) y un algoritmo basado en patrones binarios locales (del inglés, Local Binary Pattern, LBP) como descriptores de características, centrándose en regiones de interés como los ojos, nariz y boca. Luego, para la clasificación de las emociones se emplearon SVM combinado con LBP obteniendo resultados del 95.2% de precisión sobre la base de datos Cohn-Kanade [16], mientras que para el algoritmo de HOG se obtuvieron resultados del 97.3% de precisión sobre la base de JAFFE [17]. Por otro lado en [18] se hizo una comparación de tres clasificadores binarios (k-vecinos más cercano, RF y SVM) para el reconocimiento de siete emociones (enojado, disgusto, miedo, feliz, neutro, triste y sorpresa) usando la base de datos Cohn-Kanade, en los resultados se mostró que el clasificador SVM fue el que presentó el mejor desempeño con una precisión del 80.0%. En [19] se implementó un método de aprendizaje profundo donde se combina un codificador automático con el algoritmo de clasificación XGBoost para el reconocimiento de expresiones faciales sobre las bases de datos JAFFE y Cohn-Kanade alcanzando una precisión del 90.9% y 94.1%, respectivamente.

1.3. Hipótesis

Es posible obtener un buen desempeño de las CNNs para el reconocimiento de emociones por medio del aprendizaje por transferencia.

1.4. Objetivos

1.4.1. Objetivo general

Implementar y evaluar el método de aprendizaje por transferencia aplicado sobre diferentes arquitecturas basadas en CNNs para la clasificación de emociones en rostros de la base de datos Cohn-Kanade.

1.4.2. Objetivos específicos

1. Diseñar algoritmos de pre-procesamiento y segmentación que permitan extraer rostros de la base de datos Cohn-Kanade.

2. Implementar y evaluar diferentes arquitecturas de CNNs usando la base de datos base Fer2013.

3. Implementar y evaluar el método de aprendizaje por transferencia para la clasificación de emociones en rostros de la base de datos Cohn-Kanade.

4. Implementar y evaluar el uso de la función de pérdida para la clasificación de rostros en la base de datos Cohn-Kanade.

5. Implementar y evaluar diferentes clasificadores binarios para la evaluación de emociones en rostros basados en el plano de Arousal y Valencia.

1.5. Contribución de este trabajo

En el presente trabajo se realizó la clasificación de cuatro emociones en rostros de personas (neutro, enojado, feliz y triste) a partir de las bases de datos FER-2013 [20] y Cohn-Kanade [16], con el fin de poder comprender la conducta de una persona desde el punto de vista emocional. Diferentes arquitecturas se implementaron incluyendo algunas del estado del arte como AlexNet y VGG-16, además de una arquitectura LeNet creada experimentalmente. Luego del entrenamiento de cada arquitectura, se aplicó el aprendizaje por transferencia a la base de datos Cohn- Kanade, el conocimiento adquirido por la red de la base de datos FER-2013 mejoró el aprendizaje de los patrones en la base de datos Cohn-Kanade que contiene menor cantidad de muestras. También se evaluó la función de pérdida de la mejor arquitectura utilizando la técnica triple pérdida para obtener representaciones intermedias de la red para cada emoción. Posteriormente, a partir de cada representación obtenida tres clasificadores binarios fueron implementados (SVM, RF y XGBoost) para la clasificación de las cuatro emociones. Para este experimento, se unificaron las emociones de neutro y feliz en una clase positiva y las emociones de triste y enojado en una clase negativa

basados en el plano de Arousal y Valencia [21], cada experimento se evaluó a partir de diferentes medidas de desempeño, de esta manera se pudo concluir cual fue el modelo que obtuvo un mejor desempeño para solucionar el problema de clasificación de emociones en rostros.

2. Marco teórico

2.1. Detección de rostro

2.1.1. Algoritmo de Viola-Jones

El algoritmo de Viola-Jones es un método que proporciona de forma rápida y eficiente la detección de rostros en una imagen determinada. Consiste en recorrer una imagen mediante el uso de ventanas deslizantes de izquierda a derecha y de arriba abajo a través de toda la imagen objetivo, esta transformación da como resultado una imagen integral, proporcionando de forma rápida y eficiente la detección de un rostro en una imagen determinada. Este proceso se basa en el uso de características tipo Haar y un clasificador AdaBoost en cascada. La detección de rostros utilizando el algoritmo de Viola-Jones se divide en tres pasos fundamentales (ver Figura 1). El primer paso consiste en extraer las características de la imagen mediante un clasificador Haar utilizando miles de imágenes positivas, es decir imágenes que contienen caras, e imágenes negativas que son aquellas que no contienen rostro. Todo este conjunto de imágenes actuarán como plantillas para la detección del rostro. El segundo paso se basa en el entrenamiento del clasificador en cascada AdaBoost que permite una detección facial eficiente durante el tiempo de ejecución. Por último se hace un recorrido de la imagen para detectar rostros en diferentes ubicaciones y tamaño [22].

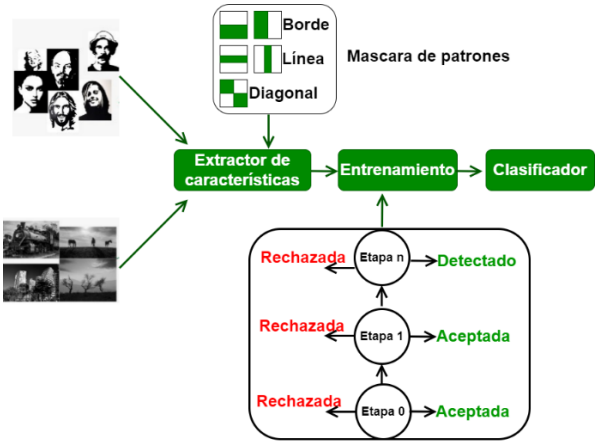


Figura 1. Detección de objetos mediante el algoritmo de Viola-Jones.

2.1.2 Histograma de gradientes orientados (HOG)

Los histogramas de gradientes orientados son descriptores utilizados principalmente en el reconocimiento de patrones y procesamiento de imágenes. Este tipo de descriptores tienen muchas aplicaciones tales como, sistemas de vigilancia, control de acceso, cámaras digitales e interacción entre personas y computadoras [23]. HOG es un método que es utilizado principalmente para extraer características de objetos en una imagen, este proceso es mostrado en la Figura 2 y consiste en dividir la imagen en un número de celdas y para cada una de las celdas se construye un histograma de gradientes orientados, este conjunto de histogramas a su vez es conectado para construir un vector de características que es la representación global de la imagen, luego este vector lo analizará una SVM que se encarga de determinar si la imagen de entrada contiene o no un rostro [24].

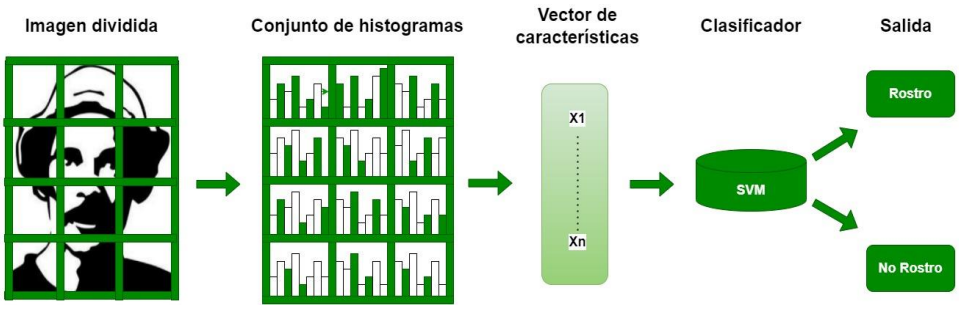


Figura 2. Cálculo de histogramas de gradientes orientados.

2.2. Redes neuronales convolucionales (CNN)

Una CNN es un tipo de red multicapa que está diseñada principalmente para el reconocimiento de patrones en imágenes y videos. Las CNNs son utilizadas en varios campos como: en el reconocimiento de objetos, clasificación de imágenes, procesamiento del lenguaje natural, clasificación de imágenes diagnósticas, entre otros [25]. La estructura básica de este tipo de red está compuesta por varias etapas: (ver Figura 3) una entrada, capas convolucionales, capas de agrupación y una capa totalmente conectada encargada de determinar la salida de la red.

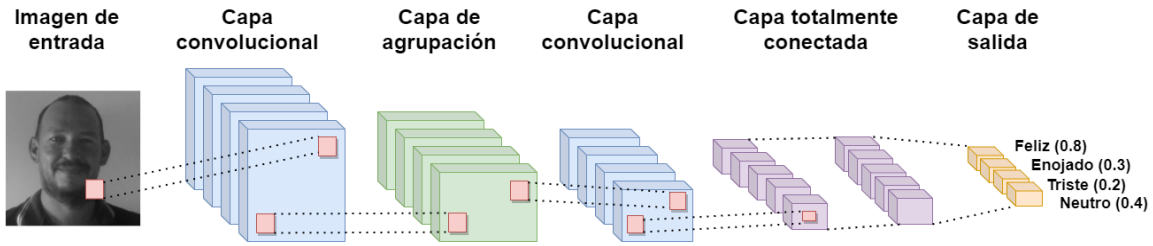


Figura 3. Estructura general de una CNN.

2.2.1. Capa convolucional

Las capas convolucionales realizan las operaciones de convolución a una imagen de entrada mediante un conjunto de filtros para crear mapas de características como se observa en la Figura 4. La operación de convolución consiste en colocar un filtro en la parte superior izquierda de la imagen y realizar el producto punto entre el filtro y los datos de entrada, posteriormente se desliza el filtro hacia la derecha sobre cada posición de la imagen de forma horizontal hasta llegar al borde de la imagen, luego el filtro se desplaza hacia abajo y se repite este proceso hasta recorrer toda la imagen dando como resultado un mapa de características.

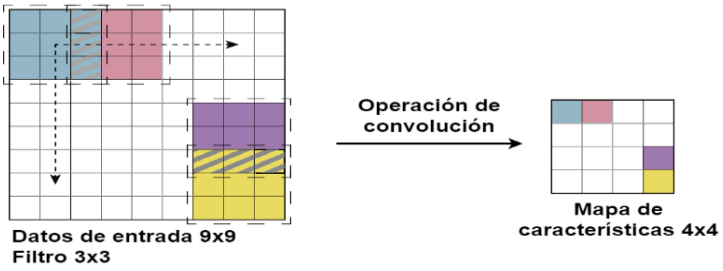


Figura 4. Operación de convolución [26].

Matemáticamente la operación de convolución para tiempos discretos (ver Ecuación 1) combina dos funciones y genera una tercera en la salida, en el caso de una CNN el primer argumento $x(t)$ sería la entrada, seguido por el operador de convolución representado por (*), el segundo argumento $w(t)$ son los pesos, y la salida $s(t)$ la denominamos mapa de características. De igual forma la operación de convolución también es representada como una sumatoria de convolución con (a) variando desde $(-\infty)$ hasta (∞) del producto de $x(a)$ con $w(t-a)$ [27].

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (1)$$

Las CNNs comúnmente usan una función de activación no lineal, este cálculo se realiza frecuentemente después de cada capa convolucional. Sin embargo, a veces las transformaciones no lineales se implementan como una capa independiente para permitir una mayor flexibilidad en la arquitectura de la red. Entre las posibles funciones no lineales está la función ReLU, tangente hiperbólica o sigmoide.

2.2.2. Capa de agrupación

El propósito de la capa de agrupación es reducir el tamaño espacial de la representación capturada por la capa convolucional. Principalmente simplifica la información recopilada y crea una versión condensada de la misma información. Un ejemplo correspondiente a esta capa es el valor máximo (max-pooling) o el valor promedio (average-pooling) mostrado en la Figura 5.

Figura 5. Operación de max-pooling y average-pooling.

Para el caso del max-pooling, se divide una matriz de entrada en bloques de un tamaño específico y se selecciona por cada región el valor máximo de cada bloque. El cálculo de este método se realiza a partir de la Ecuación 2, donde $A_{0,0}$ representa el primer bloque de la matriz de entrada, la función máxima es representa por max y los valores $f_{0,0} \dots f_{1,1}$ son los índices de cada bloque de agrupación [28].

$$A_{0,0} = \max(f_{0,0}, f_{0,1}, f_{1,0}, f_{1,1}) \quad (2)$$

En el caso del average-pooling, el método funciona de forma similar, a diferencia que en la salida devuelve el valor promedio de cada bloque seleccionado. El cálculo del valor promedio para nuestro ejemplo viene dado por la Ecuación 3.

$$A_{0,0} = \frac{1}{4}(f_{0,0} + f_{0,1} + f_{1,0} + f_{1,1}) \quad (3)$$

2.2.3. Capa completamente conectada

La última etapa de una red neuronal convolucional es comúnmente una red totalmente conectada, su función principal es interpretar y realizar la clasificación de las características detectadas y extraídas a partir de las capas convolucionales y de agrupación. Finalmente, se aplica una función softmax para obtener la probabilidad de cada una de las clases que se quiere analizar [26].

2.2.4. Arquitecturas de redes neuronales convolucionales

AlexNet

En el año 2012, Krizhevsky, creador de AlexNet, ganó el concurso ILSVRC (ImageNet Large Scale Visual Recognition Competition) [29], competencia que se realiza anualmente evaluando los algoritmos para la detección de objetos y clasificación de imágenes a gran escala. En la Figura 6 se puede observar el diagrama simplificado de la arquitectura AlexNet el cual está compuesto por una imagen de entrada que tiene una dimensión de 224x224x3 (RGB) , 5 capas convolucionales, 3 capas Max pooling y 3 capas totalmente conectadas. El tamaño de los filtros de cada capa convolucional se especifica a continuación: la primera capa tiene 96 filtros con un tamaño de 11x11, la segunda capa cuenta con 256 filtros con un tamaño de 5x5, la tercera y cuarta capa tienen 384 filtros cada uno con un tamaño de 3x3 y la quinta capa posee 256 filtros con un tamaño de 3x3. Este modelo fue entrenado con aproximadamente 1.2 millones de imágenes de la base de datos ImageNet, utilizando una función softmax en la capa de salida para la clasificación de 1.000 objetos diferentes. AlexNet es una red que contiene 60 millones de parámetros y 650.000 neuronas. Este diseño está dividido en dos grupos de bloques para aprovechar el uso de unidades de procesamiento gráfico o GPU (Graphics processing unit) en paralelo que se utilizaron para el entrenamiento [30].

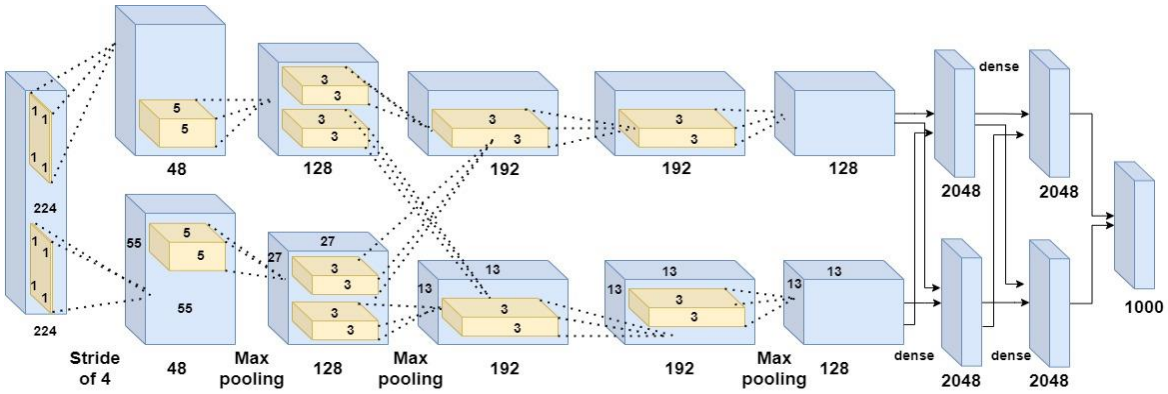


Figura 6. Arquitectura AlexNet [30].

VGG-16

VGG-16 es una arquitectura de CNNs propuesta por K. Simonyan y A. Zisserman en 2014. En la Figura 7 se observa un ejemplo del modelo VGG-16 el cual está compuesto por 13 capas convolucionales, 5 capas Max pooling, 3 capas totalmente conectadas y una capa softmax para la decisión final. Esta arquitectura en su configuración cuenta con más de 138.000.000 de parámetros. El tamaño de los filtros utilizados en las capas convolucionales son relativamente pequeños (3x3) y un paso de 1 para garantizar la misma dimensión espacial en cada mapa de activación. Además las capas ocultas están equipadas con la función de activación ReLU. La agrupación espacial se lleva a cabo mediante 5 capas de Max pooling utilizando filtros de (2x2). Finalmente VGG-16 tiene 3 capas totalmente conectadas que tienen una profundidad diferente, las dos primeras generan un vector de 4096 dimensiones y la última realiza la clasificación de 1.000 clases debido a su entrenamiento con la base de datos ImageNet [31].

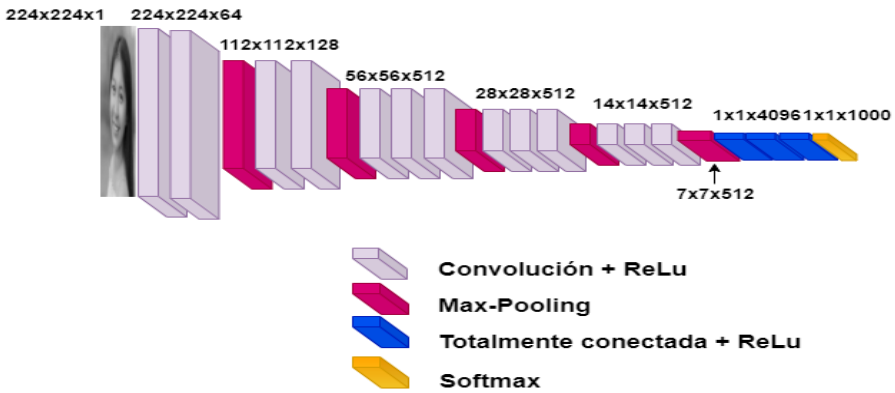


Figura 7. Arquitectura VGG-16 [31].

2.3 Aprendizaje por transferencia

Cuando hablamos de aprendizaje por transferencia nos referimos a utilizar modelos previamente entrenados usando parte del conocimiento que han adquirido a través de su entrenamiento, con el fin de aprender y mejorar el rendimiento de nuevos modelos. La Figura 8 muestra una comparación de un proceso tradicional de aprendizaje automático donde los algoritmos de este tipo de aprendizaje se utilizan para resolver tareas específicas de manera independiente. Sin embargo, cuando se aplica el aprendizaje por transferencia (parte b) el modelo puede aprender las características de una tarea y aplicarla a otra [32], [33]. Existen diferentes métodos que son utilizados en el aprendizaje por transferencia para lograr una mayor precisión en las CNNs, a continuación mencionaremos dos de estas técnicas: el ajuste fino, el cual consiste en utilizar modelos previamente entrenados, retirando la última capa completamente conectada y remplazado esta por una nueva, que tiene el mismo número de clases en la tarea de clasificación. El otro método es la extracción de características que consiste en utilizar los parámetros aprendidos de una red pre-entrenada para luego extraer las características y ser procesadas a través de un clasificador que es entrenado desde cero.

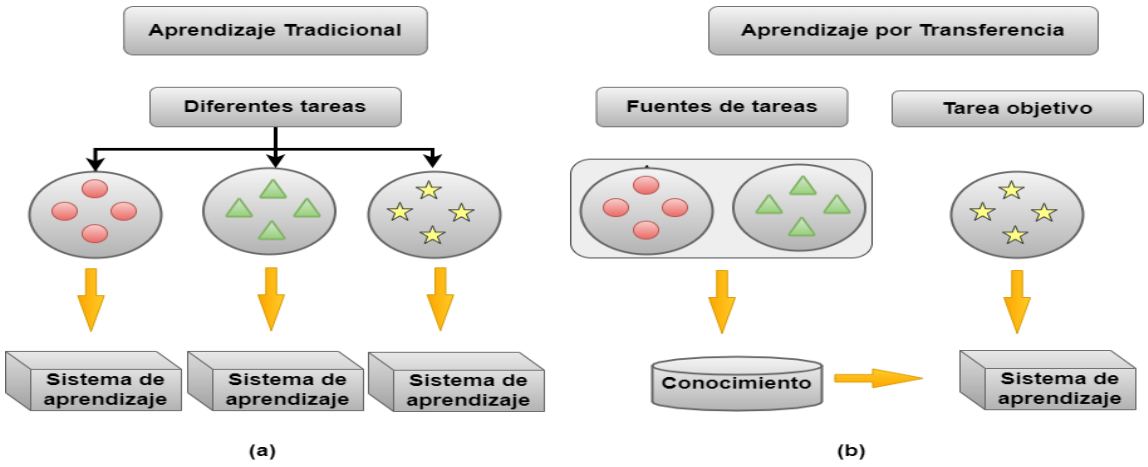


Figura 8. Comparación método tradicional y método de aprendizaje por transferencia [34].

Matemáticamente el aprendizaje por transferencia puede ser explicado a partir de algunas notaciones introducidas por Pan e Yang [34].

Dominio: Dado un conjunto de datos específicos $X = \{X_{1,} X_{2,} \dots X_{n,}\} \in \chi$, donde χ es un espacio de características y $P(X)$ es la distribución de probabilidad marginal del conjunto de datos, se define un domino como $D = \{\chi, \mathrm{P(X)}\}$.

Tarea: Dado un conjunto de datos específicos $X = \{X_1, X_2, \dots X_n\} \in \chi$ y sus etiquetas $Y = \{Y_1, Y_2, \dots Y_n\} \in \mathrm{Y}$, donde Y está dentado por un espacio de etiquetas. Una tarea se define como $\mathrm{T} = \{\mathrm{Y}, f(\mathrm{X})\}$, donde f es una función predictiva objetiva, que puede verse como una distribución condicional $\mathrm{P}(Y|\mathrm{X})$.

Aprendizaje por transferencia: Dado un domino de origen D_s y su tarea correspondiente T_s, donde la función predictiva f_s se interpreta como un conocimiento obtenido de D_s y T_s. El objetivo es obtener una función predictiva f_t para una tarea destino T_t con domino destino D_t. En general el aprendizaje por transferencia tiene como objetivo mejorar el desempeño de f_t al utilizar el conocimiento de f_s, donde $D_s \neq D_t$ o $\mathrm{T}_s \neq \mathrm{T}_t$. De forma resumida el aprendizaje por transferencia se define como: $D_s, \mathrm{T}_s \rightarrow D_t, \mathrm{T}_t$ [35].

2.4. Triple pérdida

Esta es una función de pérdida utilizada para predecir y mejorar la distancia entre las muestras de entrada. Esta función utiliza tripletas (ancla, positivo y negativo) y calcula la distancia euclidiana entre pares positivos y pares negativos (ver Ecuación 4), donde $f(x)$ significa las representaciones de las muestra de entrada (x) en un espacio de características, $|| \dots ||_2^2$ representa el cuadrado de la norma $L2$, además (x_i^a) es el ancla, (x_i^p) la muestra positiva, (x_i^n) la muestra negativa y alfa $(\propto)$ es el margen que separa la distancia entre pares positivos y negativos. Luego de obtener esta diferencia se representa esta información en un espacio de característica de tal forma que la distancia entre las muestras de la misma clase sea pequeña, mientras que la distancia entre un par de muestras de diferentes clases sea grande.

$$\sum_i^N [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \propto] \quad (4)$$

En otras palabras para aplicar la técnica triple pérdida es necesario hacer una comparación entre pares de muestras positivas y negativas. En la parte izquierda de la Figura 9 se muestra un ejemplo de esta técnica, aquí se puede observar que el ancla (registro actual de la imagen) se encuentra más alejado del dato positivo (imagen con la misma etiqueta del ancla) respecto al dato negativo (imagen con una etiqueta diferente al ancla). Lo que se pretende con esta técnica es

disminuir esta distancia entre el ancla y el dato positivo, del mismo modo aumentar la distancia entre el ancla y el dato negativo (parte derecha de la Figura 9) [36].

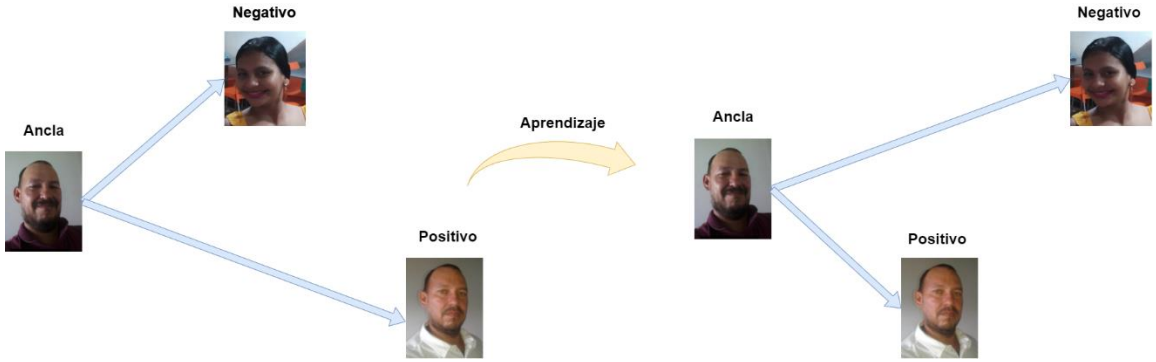


Figura 9. Funcionamiento Triple Pérdida.

2.5. Clasificadores

2.5.1. Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial hacen parte de las técnicas de aprendizaje supervisado que pueden ser aplicadas para resolver problemas de clasificación o de regresión. Su principal objetivo es construir un hiperplano que permita separar una clase de otra, maximizando el margen entre dos clases diferentes de forma óptima [37]. En la Figura 10 se muestra la representación gráfica de una SVM con margen duro la cual hace referencia a los datos que son linealmente separables, cabe destacar que aquí existen infinitos hiperplanos que pueden separar una clase de otra de forma lineal.

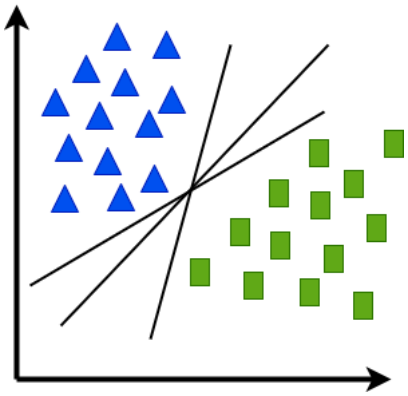


Figura 10. SVM con margen duro.

Partiendo de un conjunto de datos de entrenamiento S = $\{(x_1; y_1), \dots, (x_n; y_n)\}$, donde cada muestra $(x_i; y_i)$ para $i = 1 \dots n$ forma un vector de n características $x_i \in R^n$ con sus correspondientes etiquetas $y_i \in \{+1, -1\}$ se puede definir un hiperplano de separación a partir de la Ecuación 5, siendo b el sesgo y w el vector normal del hiperplano.

$$w^T \ x_i + b = 0 \tag{5}$$

Dicho de otra forma el hiperplano de separación deberá cumplir las siguientes inecuaciones:

$$\begin{aligned} w^T x_i - b \geq 1 \ para \ y_i = 1 \\ w^T x_i - b \leq -1 \ para \ y_i = -1 \end{aligned} \tag{6}$$

Las anteriores expresiones definen dos hiperplanos que permite separar una clase positiva (+1) o negativa (-1). Estas condiciones se pueden re-escribir brevemente como se muestra en la Ecuación 7.

$$y_i(w^T \ x_i - b) \geq 1, \ i = 1, \dots, n \tag{7}$$

Sin embargo, la optimización de $\frac{1}{2}\|w\|^2$ está sujeta a esta restricción (7) buscando obtener un hiperplano óptimo de decisión, luego la función objetivo puede expresarse a partir de la Ecuación 8.

$$\underset{w,b}{argmin} \ \frac{1}{2}\|w\|^2 \tag{8}$$

Matemáticamente estas restricciones (7) se pueden escribir como la función de decisión de una SVM con margen duro a partir de la Ecuación 8 donde se introducen los multiplicadores de Lagrange $\alpha_i, \ i = 1, 2, \dots, N$ para resolver el problema de optimización planteado.

$$\underset{w,b}{argmin} \ \underset{\alpha_i \geq 0}{max} \left(\frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [\, y_i(w^T \ x_i - b) - 1]\right) \tag{9}$$

Por otro lado, en la Figura 11 tenemos una SVM con margen blando, es decir, los datos no son linealmente separables. Esto implica introducir una variable de holgura ε_i con $i = 1, \dots, n$ que permita tener errores en el sistema con el fin de no sobreajustarlo.

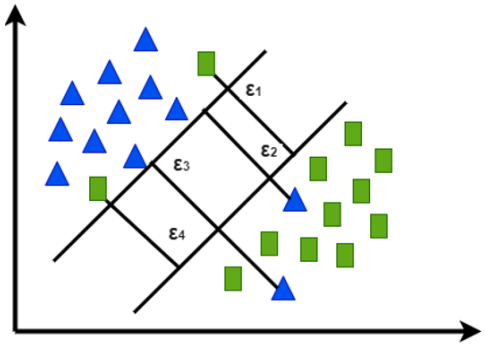


Figura 11. SVM con margen blando.

Al tener una variable de holgura, se introduce un parámetro C como variable de regularización el cual controla el tamaño del margen y la penalización de los puntos ubicados al otro lado del margen de decisión, dando lugar a las siguientes restricciones.

$$\begin{aligned} w^T\, x_i + b \geq 1 - \varepsilon_i \;\; si\; y_i \;\; = +1 \\ w^T\, x_i + b \leq -1 + \varepsilon_i \;\; si\; y_i \;\; = -1 \end{aligned} \tag{10}$$

Ahora, el problema primal de la optimización es minimizar $\|w\|$ y la cantidad de desviación descrita por la variable de holgura.

$$\min \frac{1}{2}\, \|w\|^2 + C \sum_i \varepsilon_i \tag{11}$$

En consecuencia, la función de decisión para la clasificación de una SVM con margen blando está dada por la Ecuación 12 donde se incorporan los coeficientes de Lagrange α y β para solucionar el problema de optimización [38].

$$\underset{w,\varepsilon,b}{argmin}\ \underset{\alpha,\beta}{max} \left(\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \varepsilon_i - \sum_{i=1}^{N} \alpha_i [\, y_i (w^T\; x_i - b) - 1 + \varepsilon_i] - \sum_{i=1}^{N} \beta_i \varepsilon_i\right) \tag{12}$$

2.5.2. Bosques aleatorios (RF)

El algoritmo de bosques aleatorios también hace parte de las técnicas de aprendizaje supervisado muy utilizado para regresión y clasificación, este método es basado en la generación de múltiples árboles de decisión seleccionando las muestras de forma aleatoria sobre un conjunto de datos. En la figura 12 se muestra un ejemplo del algoritmo RF de clasificación donde realiza combinaciones en paralelo y aprovecha la independencia de cada árbol de decisión para determinar

si una muestra a evaluar pertenece a una determinada clase. Luego, se define una única salida a partir de una regla de votación [39].

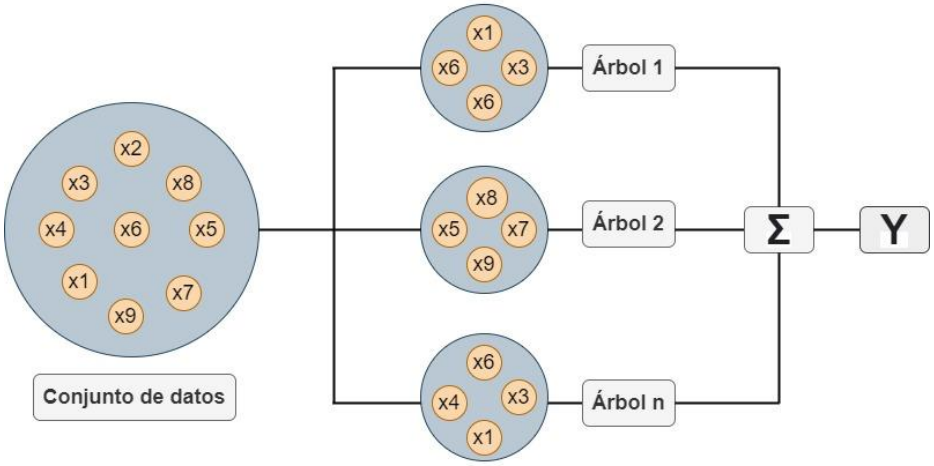


Figura 12. Funcionamiento del algoritmo Random Forest.

Dado un conjunto de datos S = $\{(x_1; y_1), \dots, (x_m; y_m)\}$ con M números de instancia donde $x_i \in R^n$ con N números de atributos y sus correspondientes etiquetas $y_i \in \{y_1, y_2, \dots, y_n\}$. A partir de ese conjunto de datos tenemos varios árboles de decisión que dividen la base de datos hasta alcanzar la condición de detección. El valor de cada clase se asigna en función de la proporción de las etiquetas presentes en cada nodo [40].

La proporción para la j_{-esima} clase en el nodo terminal h del árbol de decisión t para un caso de prueba X se representa como:

$$P_{j,h}^{t} = \frac{1}{n_h} \sum_{i \in h} \mathbb{I}(Y_i = j) \, . \qquad (13)$$

Donde n_h el número total de instancias en el nodo terminal h y la función indicadora se define con la letra $\mathbb{I}$. Asumiendo estos datos el valor de la clase j está representado por la Ecuación 14.

$$\hat{Y}_j^t = \underset{1 \le j \le y}{max} \{P_{j,h}^t\} \, . \qquad (14)$$

Ahora, para asignar el valor final de las clases, el algoritmo de RF en clasificación se basa en la votación mayoritaria contando las clases predichas por cada árbol de decisión.

$$Y(Y_i = j) = \sum_{t=1}^{n-arbol} \mathbb{I}.\mathbb{J}(\hat{Y}_j^t) \, . \tag{15}$$

Finalmente, para de decidir cuál será la salida, el algoritmo de RF utiliza la Ecuación 16

$$\hat{Y} = \max_{1 \le j \le y} \{Y(Y_i = j)\} \, . \tag{16}$$

2.5.3. *Extreme Gradient Boosting* (XGBoost)

El algoritmo XGBoost consiste en generar múltiples árboles de decisión, de forma que cada árbol aprenda del resultado de los árboles previos. Este algoritmo se optimiza a partir del gradiente descendente con el objetivo de ser eficiente y preciso a la hora de resolver un problema de clasificación o regresión [41], [42]. En la Figura 13 es posible apreciar n árboles de decisión conectados secuencialmente donde cada árbol está corrigiendo lo que el anterior no puedo predecir correctamente. Finalmente la salida será la principal mejora incorporada en todo este proceso secuencial.

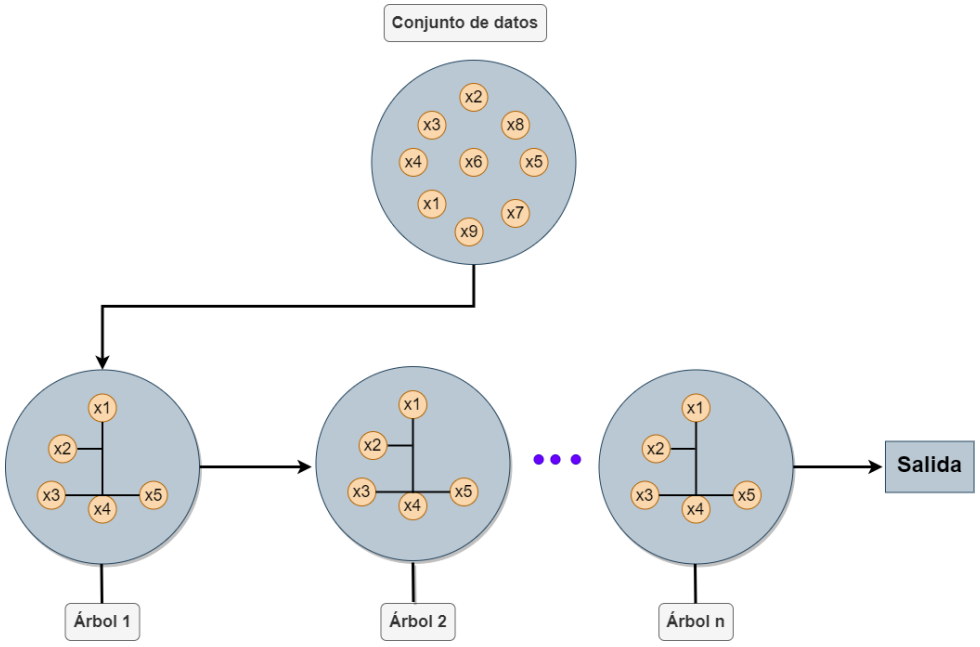


Figura 13. Funcionamiento del algoritmo XGBoost.

Dado el conjunto de datos S = $\{(x_1; y_1), \dots, (x_n; y_n)\}$, donde cada muestra $(x_i; y_i)$ para $i = 1 \dots n$ forma un vector de n características $x_i \in R^n$ con sus correspondientes etiquetas $y_i \in \{+1, -1\}$. Se define la función objetivo J como:

$$J(\Theta) = L(\Theta) + \Omega(\Theta) \, . \tag{17}$$

Donde *L* es la función de pérdida para el entrenamiento, que mide qué tan exacto es nuestro modelo y Ω es el término de regularización, que mide la complejidad del modelo y ayuda a reducir el sobreajuste. Dado que el modelo base del algoritmo XGBoost son los árboles de decisión, la salida $\widehat{y_i}$ del modelo se puede expresar así:

$$\widehat{y_i} = \sum_{i=1}^{k} f_k \, (x_i), f_k \, \epsilon \, F \, . \tag{18}$$

En la Ecuación 18, k corresponde al número de árboles y $f_k(x_i)$ representa la puntuación dada de todas las observaciones del conjunto de datos F. Ahora nuestra función objetivo en cada instante t estará representada por la Ecuación 19, donde n es el número de predicciones de nuestro modelo.

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \widehat{y_i}) + \sum_{k=1}^{t} \Omega(f_k) \, . \tag{19}$$

Finalmente una vez entrenado el modelo, re-escribimos la salida en función de los valores predichos en cada paso t como:

$$\widehat{y_i}^{\,t} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y_i}^{\,(t-1)} + f_t(x_i) \tag{20}$$

2.6. Medidas de desempeño

En este trabajo se utilizaron diferentes medidas de desempeño para evaluar el rendimiento de las diferentes arquitecturas y clasificadores implementados para el reconocimiento de emociones en rostros. Los resultados se evaluaron y se utilizaron de manera individual mediante el cálculo de las siguientes métricas: sensibilidad, especificidad, precisión general, precisión balanceada (Unweighted Average Recall, UAR), F1 score y coeficiente Kappa.

La Tabla 1 muestra una matriz de confusión para un sistema de clasificación binario, esta matriz es una herramienta útil de la cual se derivan diferentes medidas de desempeño con el fin de evaluar los algoritmos implementados. Cada fila de la matriz representa la clase predicha, mientras que cada columna representa la clase real [43].

		Clase real	
		Positivo	Negativo
Clase predicha	Positivo	Verdaderos positivo (TP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (TN)

Tabla 1. Matriz de confusión.

- **Verdaderos positivos** (del inglés, True positive, TP): Corresponden a los valores que el modelo predice como positivos y pertenecen a la clase positiva.

- **Falsos positivos** (del inglés, False positive, FP): Corresponden a los valores que el modelo predice como positivo y pertenecen a la clase negativa.

- **Verdaderos negativos** (del inglés, True negative, TN): Corresponde a los valores que el modelo predice como negativos y pertenecen a la clase negativa.

- **Falsos negativos** (del inglés, False negative, FN): Corresponde a los valores que el modelo predice como negativo y pertenecen a la clase positiva.

Sensibilidad: Se conoce también como tasa de verdaderos positivos y representa el porcentaje de muestras positivas que fueron identificadas correctamente por el clasificador. (Ecuación 21)

$$Sensibilidad = \ TP/(TP + FN) \tag{21}$$

Especificidad: Se conoce también como tasa de verdaderos negativos y representa el porcentaje de muestras negativas que fueron identificadas correctamente por el clasificador. (Ecuación 22)

$$Especificidad = \ TN/(TN + FP) \tag{22}$$

Precisión general: Esta métrica representa el porcentaje de muestras predichas correctamente sobre el total de los datos de prueba. (Ecuación 23)

$$Precisión\ general = \ (TP + TN)/(TP + TN + FP + FN) \quad (23)$$

Precisión balanceada (del inglés Unweighted Average Recall, UAR): Es una métrica útil para evaluar modelos cuando las clases están desequilibrada. La precisión balanceada está representada por el promedio asimétrico de la sensibilidad y la especificidad. (Ecuación 24)

$$Precisión\ balanceada = \ (Sensibilidad + Especificidad)/2 \quad (24)$$

F1 score: Esta métrica representa el promedio armónico entre la precisión y la sensibilidad, el F1 score puede tomar valores entre 0 y 100, donde el mejor valor es representado por 100 y la peor puntuación es 0. (Ecuación 25)

$$F1\ score = \ 2 * ((Precisión * Sensibilidad)/(Precisión + Sensibilidad)) \quad (25)$$

Coeficiente kappa: Esta métrica mide la proporción de concordancia observada sobre el total de observaciones correspondientes a un conjunto de datos, excluyendo las concordancias atribuibles al azar. El coeficiente kappa toma valores entre -1 y +1, donde +1 representa el grado de mayor concordancia, por el contrario el valor de -1 representa el mayor grado de discordancia. (Ecuación 26)

$$Coeficiente\ kappa = \ 2 * ((Po - Pe)/(1 - Pe)) \quad (26)$$

Donde Po es la proporción de concordancia observada y Pe es la proporción de concordancia esperada por azar. [44]

Otra herramienta que se utilizó en este trabajo para evaluar y comparar el rendimiento de los distintos algoritmos de aprendizaje supervisado fue la curva ROC y el área bajo la curva AUC, a partir del cual se define que tan bueno es un modelo para distinguir una de las dos clases evaluadas.

Curva ROC: La curva ROC (del inglés, Receiver Operating Characteristic, ROC) es un gráfico que representa la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1 - especificidad) para un sistema de clasificación binario. En la figura 14 parte a, podemos observar

una función de distribución de una clase positiva y una clase negativa, a partir de aquí se empieza a mover un umbral o puntos de corte por toda la distribución dando como resultado la curva ROC (parte b) la cual permite comparar el desempeño de los diferentes modelos basados en el área bajo la curva AUC.

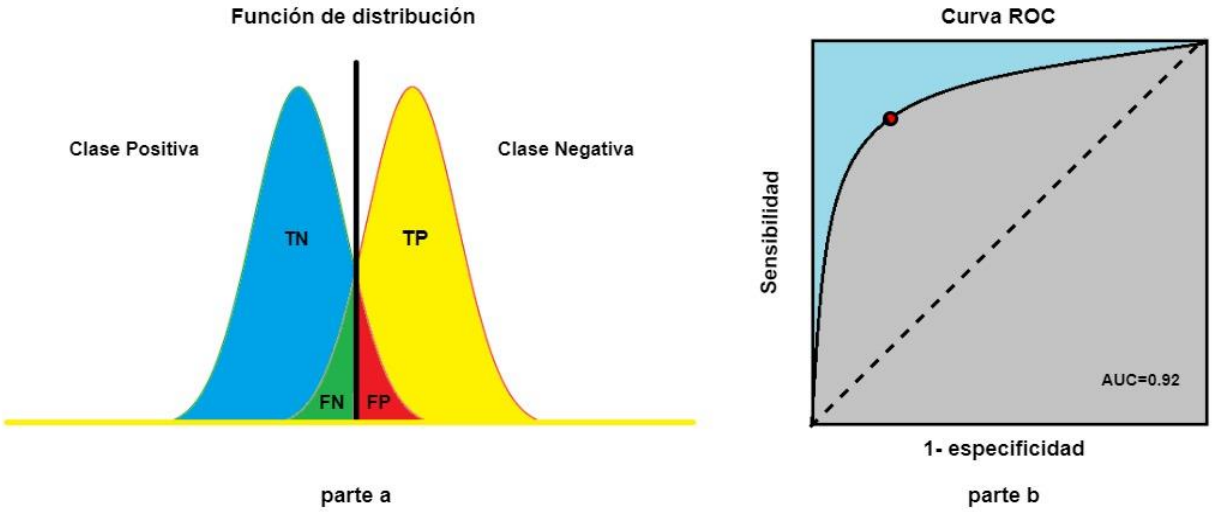


Figura 14. Función de distribución que genera una curva ROC.

Área bajo la curva: El área bajo la curva (del inglés, Area Under the Curve, AUC) es una medida que se utiliza para identificar el rendimiento general de un modelo. El rango de valores del AUC varía entre 0 y 1.0, donde 0.5 es una puntuación que no diferencia los valores de la prueba entre dos clases, por lo tanto podemos decir que es una mala puntuación y el valor de 1.0 nos indica el mejor desempeño del clasificador. En general el AUC nos permite comparar el rendimiento de los clasificadores a la vez y así poder determinar cuál fue el que mejor resolvió el problema de clasificación [45].

3. Metodología

La metodología que se utilizó en este trabajo de grado para alcanzar los objetivos propuestos se presenta en la Figura 15. Inicialmente se definen las bases de datos que se van a utilizar en la clasificación de emociones en rostros de personas, luego se implementaron los algoritmos de Viola Jones y HOG para segmentar y seleccionar los rostros de cada imagen de la base de datos Cohn-Kanade y escoger el algoritmo que presente el mejor desempeño. A continuación, se implementó la técnica de validación cruzada con independencia de usuario sobre la base de datos Cohn-Kanade,

posteriormente se implementan tres arquitecturas basadas en CNNs, dos del estado del arte tales como: AlexNet y VGG16. Además, se crea una experimentalmente denotada LeNet, luego, se realizó la implementación de la técnica del aprendizaje por transferencia y la técnica triple perdida. Además de las CNNs implementadas, también se evaluaron tres clasificadores clásicos (SVM, RF, XGBoost) a partir de representaciones obtenidas usando la técnica de triple pérdida. Finalmente, se realiza una evaluación de todos los resultados a través de diferentes medidas de desempeño.

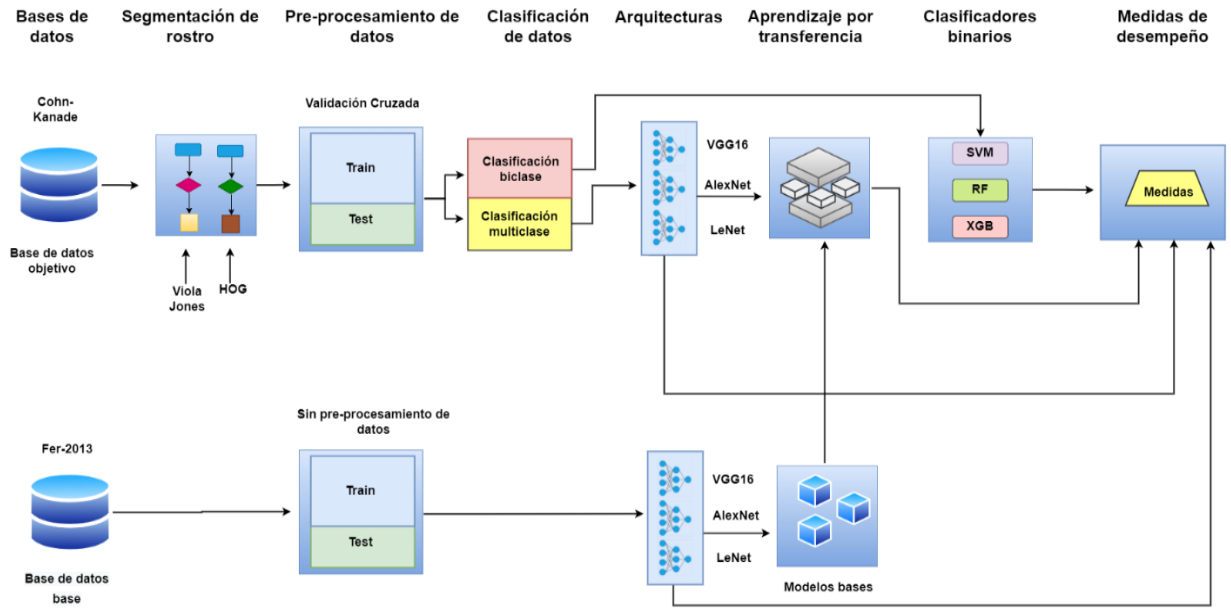


Figura 15. Metodología general.

3.1 Bases de datos

Para el desarrollo de este trabajo se utilizaron dos bases de datos, FER-2013 para el entrenamiento de modelos base y Cohn-Kanade que es nuestra base de datos objetivo.

3.1.1 FER-2013

La base de datos FER-2013 está conformada por 35.887 imágenes de rostros de 48x48 píxeles en escala de grises etiquetadas en 7 categorías de emociones: enojado, disgusto, miedo, feliz, tristeza, sorpresa y neutro, sin embargo para el desarrollo de este trabajo sólo se utilizaron las siguientes emociones: neutro, enojado, feliz y triste, con el fin de que los resultados sean comparables con los del estado del arte. Esta base de datos se utilizó para entrenar las CNNs con

el objetivo de crear modelos base y aplicar aprendizaje por transferencia hacia la base datos Cohn-Kanade. En la Tabla 2 se puede observar la cantidad de imágenes por emoción en el conjunto de entrenamiento y de prueba.

Emoción	Entrenamiento	Prueba
Neutro	4.965	607
Enojado	3.995	467
Feliz	7.215	895
Triste	4.830	653

Tabla 2. Base de datos FER-2013.

3.1.2 Cohn-Kanade

La base de datos Cohn-Kanade está conformada por 5.876 imágenes etiquetadas de 123 individuos, esta surge a partir de un experimento en el cual se pedía a cada participante que hiciera una transición desde una expresión neutra a una emoción solicitada, haciendo del primer fotograma una muestra de 640x490 píxeles en escala de grises que contiene un rostro neutro y los últimos fotogramas con las mismas dimensiones una de las siete emociones (feliz, triste, enojado, asustado, sorpresa, disgusto y desprecio). Para el desarrollo de esta propuesta se trabajó con cuatro emociones básicas: feliz, neutro, enojado y triste, con el fin de obtener resultados comparables con los reportados en el estado del arte. Los algoritmos de Viola Jones y HOG fueron usados en esta base de datos con el fin de segmentar y seleccionar los rostros de cada imagen. Finalmente, el algoritmo que se eligió fue el HOG obteniendo resultados del 98% de eficiencia en la detección de rostros. En la tabla 3 se puede observar la cantidad de imágenes por emoción.

Emoción	Pre-procesamiento (HOG)
Neutro	327
Enojado	135
Feliz	207
Triste	84

Tabla 3. Base de datos Cohn-Kanade.

3.2 Validación cruzada de K-particiones (Cohn-Kanade)

Se implementó la técnica validación cruzada de k-particiones a la base de datos Cohn-Kanade, con 5 grupos y garantizando independencia de usuarios. Esta estrategia consiste en dividir nuestro conjunto de datos en k partes, utilizando k-1 parte para el entrenamiento y las restante para validar el modelo. Este proceso se repite k veces permitiéndonos evaluar por completo la base de datos. En la Figura 16 se puede observar una representación del conjunto de datos dividido en las 5 particiones.

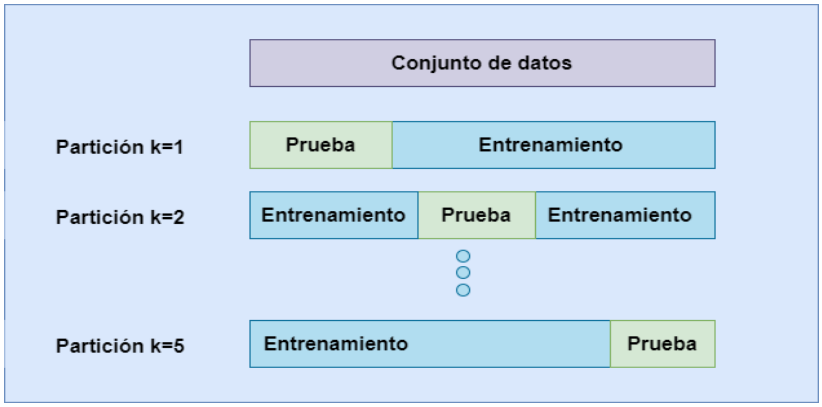


Figura 16. Validación cruzada.

3.3 Experimentos

Para el desarrollo de este trabajo se realizaron 5 experimentos, utilizando dos bases de datos, FER-2013 y Cohn-Kanade. Se implementaron varias técnicas de clasificación definiendo la cantidad de los parámetros de entrenamiento utilizados en cada arquitectura propuesta. Además se implementó la técnica de aprendizaje por transferencia con el fin de comprobar si hubo mejoras en los porcentajes de rendimiento en la clasificación de emociones en rostros. De igual forma se implementó y evaluó la función de triple pérdida en la arquitectura con el mejor desempeño, finalmente se implementaron clasificadores clásicos para la clasificación de las cuatros emociones.

3.3.1 Entrenamiento y clasificación de CNNs usando la base datos FER-2013

En esta etapa se realizaron 5 experimentos utilizando las arquitecturas LeNet, VGG16 y AlexNet sobre la base de datos FER-2013, con la finalidad de obtener los modelos base para luego ser utilizados en el aprendizaje por transferencia. La primera arquitectura implementada fue la

LeNet la cual se creó desde cero variando el número y tamaño de las capas convolucionales, filtros y capas lineales, esta red cuenta con tan solo 492.524 parámetros. En la Tabla 4 se muestra la configuración final de esta red para la clasificación de emociones en rostro de personas. Se seleccionará como ejemplo algunas capas para comprender mejor esta configuración, para Conv (1x8x5x1), 1 representa el canal de entrada (escala de grises), 8 el número de filtros, 5 el tamaño del filtro cuadrado y 1 el paso del filtro. Max Pool (2,2), 2 representa el tamaño del filtro y 2 el paso del filtro. Lineal (100,4), 100 representa el número de neuronas de entrada y 4 las neuronas de salida.

Para el entrenamiento de esta red se usó una función de pérdida de entropía cruzada, además se optimizó a partir del gradiente descendente estocástico y se utilizó una técnica de regulación llamada parada anticipada. Finalmente, la arquitectura LeNet fue entrenada con la base de datos FER-2013 con sus pesos inicializados aleatoriamente.

Tipo de capa	**Forma de salida**
Conv1 (1x8x5x1)	(8,44,44)
Max Pool (2,2)	(8,22,22)
Conv2 (8x12x5x1)	(12,18,18)
Max Pool (2,2)	(12,9,9)
Lineal (972,400)	(1,400)
Lineal (400,200)	(1,200)
Lineal (200,100)	(1,100)
Lineal (100,4)	(1,4)

Tabla 4. Arquitectura LeNet implementada. Conv: Convolución, Max Pool: Max Pooling, Lineal: Capa totalmente conectada.

Los demás experimentos fueron implementados a partir de las arquitecturas VGG16 y AlexNet, en la configuración de cada red lo primero que se hizo fue hacer una transformación a las imágenes de la base de datos FER-2013 de 48x48x1 a 224x224x3 que es la entrada de cada red a partir de una interpolación bilineal. Para el entrenamiento se usó la misma función de pérdida y optimizador de la red LeNet. Finalmente, se realizaron dos experimentos para cada arquitectura,

los cuales fueron entrenar la arquitectura a partir de una inicialización aleatoria y ajustar los pesos pre-entrenados con la base de datos de ImageNet.

3.3.2 Entrenamiento y clasificación de CNNs usando la base datos Cohn-Kanade

En esta etapa se realizaron 5 experimentos, entrenando las arquitecturas LeNet, VGG16 y AlexNet con la base de datos Cohn-Kanade de forma aleatoria y pre-entrenada con ImageNet. La configuración de estas arquitecturas fue la misma del experimento anterior. Finalmente estos resultados tienen como fin tener una referencia y poderlos comparar con los resultados obtenidos utilizando la técnica del aprendizaje por transferencia.

3.3.3 Transferencia de aprendizaje para la clasificación de la base de datos Cohn-Kanade

En esta etapa se implementó al aprendizaje por transferencia desde el modelo base entrenado con FER-2013 para la base de datos destino Cohn-Kanade, para esto se hizo un ajuste fino a la arquitectura previamente entrenada, con el fin de comprobar si esta técnica mejora el rendimiento de las CNNs que fueron entrenadas de forma aleatoria. Al igual que en los anteriores experimentos se realizaron las respectivas configuraciones para el ajuste de las diferentes arquitecturas y se validaron los resultados a partir de las diferentes medidas de desempeño.

3.3.4 Implementación y clasificación de la función triple pérdida para la clasificación de la base datos Cohn-Kanade.

En esta etapa se realizaron 4 experimentos diferentes teniendo como referencia la arquitectura LeNet que fue la que obtuvo el mejor desempeño en la clasificación de emociones utilizando la base de datos Cohn-Kanade sin emplear el aprendizaje por transferencia. Estos experimentos se hicieron con el fin de comparar si la técnica triple pérdida supera los resultados obtenidos con el aprendizaje por transferencia. Inicialmente se implementó la técnica triple pérdida para sacar una representación intermedia de 128 muestras de cada imagen de la base de datos Cohn-Kanade. Luego se emplearon métodos clásicos (SVM, RF y XGBoost) para la clasificación de las cuatro emociones. Para el caso de la SVM se utilizó un kernel lineal y un kernel Gaussiano, además se variaron los parámetros de C entre {0.001, 0.005, 0.01,…,1000} y gamma entre {0.0001, 0.001, 0.01,…,1000} a partir de una malla de búsqueda. Para RF se variaron los parámetros de máxima profundidad de cada árbol entre {1, 3, 5, 8, 10, 15, 20} y número de estimadores {1, 2, 5, 10, 20,

50, 100, 200}. Por último, para XGBoost se variaron los parámetros de máxima profundidad entre {1, 3, 5, 8, 10, 15, 20} y una tasa de aprendizaje de {0.1, 0.001, 0.0001}. Además se eligieron los mejores valores a partir de la moda. Finalmente se evaluó cada clasificador con las respectivas medidas de desempeño.

3.3.5 Clasificación biclase de la base datos Cohn-Kanade basados en el plano de Arousal y Valencia.

En esta etapa se realizaron 4 experimentos basados en el plano Arousal y Valencia con el fin de realizar una clasificación biclase, utilizando los métodos clásicos (SVM, RF y XGBoost) sobre la base de datos Cohn-Kanade. Inicialmente, se juntaron las emociones de neutro y feliz en una clase positiva y las emociones de triste y enojado en una clase negativa. Además, se le asignó una etiqueta con un valor de "1" a cada muestra de la clase positiva y una etiqueta con un valor de "0" a cada muestra de la clase negativa. Adicionalmente se graficó la curva ROC con su correspondiente AUC y las distribuciones de densidad de probabilidad con sus respectivos histogramas con el fin de conocer el rendimiento de cada modelo de clasificación biclase. Por último se evaluaron los modelos a partir de las diferentes medidas de desempeño.

4. Resultados

4.1. Resultados modelos base (FER-2013)

En la Tabla 5 se presentan los resultados obtenidos al implementar las tres arquitecturas sobre la base de datos Fer-2013, es de aclarar que esta base de datos no se aplicó ninguna estrategia de división de datos, debido a que la base de datos ya tiene conjunto de entrenamiento, validación y prueba (estandarizados en el estado del arte), por lo tanto no se reportó desviación estándar en los resultados.

Como se puede apreciar el mejor resultado se logró utilizando la arquitectura AlexNet pre-entrenada con ImageNet, con un 67.4% en su precisión balanceada. Por otra parte, en la precisión por clases la emoción enojado fue la que obtuvo el mejor desempeño con un 86.0%. En cambio la arquitectura VGG16 implementada con una inicializando aleatoria, obtuvo el desempeño más bajo con un 50.2% de precisión balanceada y en la precisión por clases, la emoción neutro solo alcanzó una puntuación del 21.0%.

Arquitecturas	Precisión balanceada (UAR)	F1 score	Coeficiente kappa	Precisión por clases
LeNet	58.8 %	58.8 %	0.44	Neutro: 55.0% Enojado: 42.0% Feliz: 69.0% Triste: 44.0%
VGG16 (True)	62.9 %	64.6 %	0.52	Neutro: 60.0% Enojado: 67.0% Feliz: 64.0% Triste: 51.0%
VGG16 (False)	50.2 %	52.8 %	0.36	Neutro: 21.0% Enojado: 64.0% Feliz: 48.0% Triste: 52.0%
AlexNet (True)	67.4 %	67.1 %	0.59	Neutro: 56.0% Enojado: 86.0% Feliz: 71.0% Triste: 48.0%
AlexNet (False)	61.1 %	64.1 %	0.52	Neutro: 36.0% Enojado: 85.0% Feliz: 50.0% Triste: 55.0%

Nota: VGG-16 y AlexNet fueron implementadas con una inicialización aleatoria (False) y pre-entrenadas con ImageNet (True)

Tabla 5. Resultados modelos base (Fer-2013)

4.2. Resultados Cohn-Kanade con inicialización aleatoria

En la Tabla 6 se presentan los resultados de implementar las tres arquitecturas sobre la base de datos Cohn-Kanade, en esta tabla se reporta desviación estándar resultado de aplicar una estrategía de validación cruzada.

Como se puede observar el mejor resultado se obtuvo con la arquitectura LeNet siendo esta del 72.8% en su precisión balanceada. Una hipótesis que podemos mencionar de porque la arquitectura LeNet fue la mejor de las tres arquitecturas en este experimento, estaría relacionada con la poca cantidad de parámetros de esta red para realizar el entrenamiento y clasificación de emociones usando la base de datos Cohn-Kanade. De hecho, si comparamos la arquitectura LeNet con AlexNet y VGG16, encontramos que la primera red tiene tan solo 492.524 parámetros, mientras que la segunda tiene 60.000.000 y la última tiene 138.000.000 parámetros respectivamente.

Por otra parte, en la precisión por clase se obtuvo un porcentaje ligeramente bajo de la emoción triste en todos los experimentos, el cual no superó el 40% de precisión. Quizás este resultado se debe a la poca cantidad de imágenes que tiene esta emoción en la base de datos Cohn-Kanade.

Arquitecturas	Precisión balanceada (UAR)	F1 Score	Coeficiente Kappa	Precisión por clases
LeNet	72.8 % +/- 3.5	78.8 %+/- 0.1	0.69+/- 0.13	Neutro: 86.6 +/- 10.6 Enojado: 65.4 +/-14.3 Feliz: 99.2 +/- 1.0 Triste: 39.1 +/- 18.1
VGG16 (True)	48.9% +/- 9.4	59.4 % +/- 0.2	0.43 +/- 0.20	Neutro: 78.4+/- 14.1 Enojado: 41.1 +/- 13.4 Feliz: 73.6 +/- 21.0 Triste: 2.6 +/- 3.4
VGG16 (False)	43.8 % +/- 8.0	54.0 % +/- 0.1	0.37+/- 0.14	Neutro: 85.0 +/- 10.3 Enojado: 41.1 +/-13.4 Feliz: 73.6 +/- 21.0 Triste: 2.6+/- 3.4
AlexNet (True)	51.3 % +/- 8.5	61.1 % +/- 0.2	0.47 +/- 0.19	Neutro: 82.1 +/- 8.0 Enojado: 25.9 +/-14.4 Feliz: 88.8 +/- 8.565 Triste: 7.5 +/- 6.9
AlexNet (False)	52.0 % +/- 7.0	59.1 %+/- 0.1	0.42 +/- 0.11	Neutro: 69.4 +/- 10.0 Enojado: 28.2 +/- 9.9 Feliz: 82.7 +/- 10.6 Triste: 27.6 +/- 24.0

Nota: VGG-16 y AlexNet fueron implementadas con una inicialización aleatoria (False) y pre-entrenadas con ImageNet (True)

Tabla 6. Resultados con inicialización aleatoria (Cohn-Kanade)

4.3. Resultados aplicando la técnica de aprendizaje por transferencia (Cohn-Kanade)

En la Tabla 7 se puede apreciar los resultados de utilizar la técnica de aprendizaje por transferencia aplicando el método de ajuste fino hacia la base de datos Cohn-Kanade.

Como se puede observar el mejor desempeño se obtuvo aplicando el aprendizaje por transferencia a la arquitectura LeNet siendo esta del 78.9% en su precisión balanceada y se comprobó que hubo una mejora en el rendimiento de la red de hasta un 6.1% en comparación con la misma arquitectura sin usar aprendizaje por transferencia (inicialización aleatoria de pesos). En

cuanto a la métrica f1 score se obtuvo un porcentaje del 84.5%, puntuación que nos permite considerar que el modelo es bueno en la clasificación de las cuatro emociones (neutro, enojado, feliz y triste). De igual forma se presentaron mejoras significativas en las demás arquitecturas con los datos inicializados de forma aleatoria y pre-entrenados con ImageNet.

Arquitecturas	Precisión balanceada (UAR)	F1 Score	Coeficiente Kappa	Precisión por clases
LeNet	78.9 % +/- 5.1	84.5 % +/- 0.1	0.78 +/- 0.12	Neutro: 92.8 +/- 7.5 Enojado: 71.6 +/- 8.9 Feliz: 98.3 +/- 3.3 Triste: 51.5 +/- 17.8
VGG16 (True)	51.6 % +/- 8.6	61.2 % +/- 0.2	0.47 +/- 0.19	Neutro: 76.7 +/- 10.4 Enojado: 44.5+/- 17.3 Feliz: 83.1 +/- 12.7 Triste: 2.2 +/- 4.4
VGG16 (False)	50.4 % +/- 7.6	59.4 % +/- 0.1	0.44 +/- 0.18	Neutro: 76.7+/- 16.1 Enojado: 42.6+/- 23.5 Feliz: 80.1+/- 9.2 Triste: 2.2 +/- 4.4
AlexNet (True)	52.6 % +/- 7.3	62.5 % +/- 0.1	0.47 +/- 0.14	Neutro: 86.3 +/- 6.8 Enojado: 27.3 /- 9.9 Feliz: 77.5 +/- 18.0 Triste: 19.2 +/- 19.7
AlexNet (False)	62.5 % +/- 5.6	67.5 % +/- 0.1	0.53 +/- 0.09	Neutro: 70.0 +/- 12.4 Enojado: 48.8+/- 19.3 Feliz: 88.7 +/- 11.5 Triste: 42.5 +/- 16.0

Nota: VGG-16 y AlexNet fueron implementadas con una inicialización aleatoria (False) y pre-entrenadas con ImageNet (True)

Tabla 7. Resultados con Aprendizaje por Transferencia (Cohn-Kanade)

4.4. Resultado con la función triple pérdida (Cohn-Kanade)

En la Tabla 8 se presentan los resultados de los tres clasificadores implementados utilizando la técnica triple pérdida sobre la base de datos Cohn-Kanade. El propósito de este experimento es comparar los resultados de este método con los resultados obtenidos en el aprendizaje por transferencia.

En los resultados es posible observar que la implementación de una SVM con kernel lineal obtuvo el mejor resultado, reportando una precisión balanceada del 78.4%. Nótese también que en

la precisión por clase, las emociones de neutro y feliz tienen una mejor eficiencia con el 94.0% y 98.8% respectivamente, mientras que las emociones de triste y enojado tuvieron una eficiencia más baja con un porcentaje del 49.7% y 71.3%. Por otro lado, también es posible observar que este método obtuvo resultados muy similares a los resultados reportados aplicando la técnica de aprendizaje por transferencia sobre la arquitectura LeNet, presentando una diferencia de sólo 0.5% en su precisión balanceada sobre el modelo con mejor desempeño.

Clasificador	Precisión balanceada (UAR)	F1 Score	Coeficiente Kappa	Precisión por clase
SVM (Linear)	78.4% +/- 6.6	85.2 % +/- 0.1	0.79 +/- 0.07	Neutro: 94.0+/- 6.1 Enojado: 71.3+/- 17.7 Feliz: 98.8 +/- 1.7 Triste: 49.7 +/- 14.8
SVM (RBF)	77.3% +/- 8.3	84.2 % +/- 0.1	0.78 +/- 0.08	Neutro: 93.7 +/- 5.9 Enojado: 70.2+/- 19.5 Feliz: 98.8 +/- 1.7 Triste: 46.3 +/- 19.7
Random Forest	76.0% +/- 9.0	83.0 %+/- 0.1	0.80 +/- 0.10	Neutro: 91.4+/- 8.7 Enojado: 69.9 +/- 16.2 Feliz: 97.5 +/- 2.4 Triste: 45.1 +/- 17.6
XGBoost	77.0% +/- 8.5	83.0 %+/- 0.1	0.75 +/- 0.10	Neutro: 88.0 +/- 11.5 Enojado: 71.7 +/- 15.7 Feliz: 98.3 +/- 2.4 Triste: 49.8 +/- 16.5

Tabla 8. Clasificación Multiclase basado en tres clasificadores a partir de una representación obtenida usando el método de triple pérdida (Cohn-Kanade)

4.5. Resultados de la clasificación biclase basados en el plano Arousal y Valencia (Cohn-Kanade)

En la Tabla 9 es posible observar los resultados de los diferentes clasificadores binarios: SVM, RF y XGBoost, con el fin de hacer una comparación y análisis del rendimiento de cada uno de ellos, basándonos en el plano Arousal y Valencia para agrupar las cuatro emociones de la base de datos Cohn-Kanade de la siguiente forma: las emociones de neutro y feliz en una clase positiva y las emociones de triste y enojado en una clase negativa.

Los resultados muestran que el mejor resultado corresponde al clasificador RF, reportando una precisión balanceada del 82.7%, mientras que la SVM con kernel Gaussiano presentó el mayor porcentaje de sensibilidad, con un 95.0%.

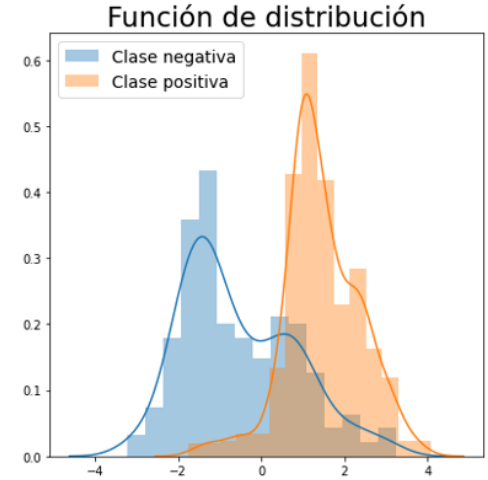
Clasificador	Precisión balanceada (UAR)	Sensibilidad	Especificidad	F1 Score
SVM (Linear)	80.5 % +/- 4.2	91.5 % +/- 7.4	69.6 % +/- 12.2	86.0 % +/- 3.2
SVM (RBF)	81.1 % +/- 8.0	95.0 % +/- 5.1	67.1 % +/- 18.3	86.6 % +/- 6.1
Bosques aleatorios	82.7 % +/- 7.9	92.2 % +/- 7.3	73.3 % +/- 16.1	86.6 % +/- 6.4
XGBoost	82.1 % +/- 5.8	92.2 % +/- 7.9	71.9 % +/- 13.2	86.5 % +/- 5.7

Tabla 9. Clasificación Biclase basado en tres clasificadores (Cohn-Kanade)

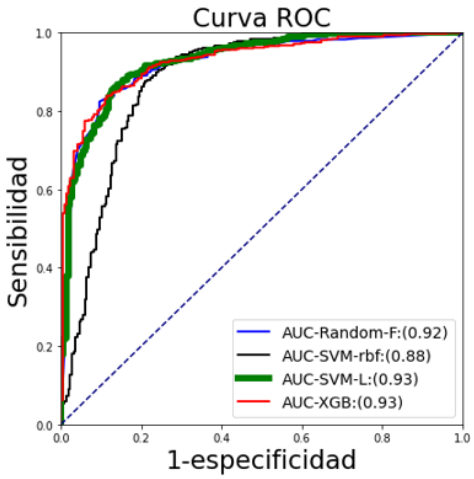
Función de distribuciones y Curva ROC

En la Figura 17 se observa en la parte izquierda, una gráfica de función de distribución de una SVM con kernel Gaussiano la cual fue construida a partir de los scores de la clase positiva (color naranja) como los scores de la clase negativa (color azul) y representa la distancia de cada muestra al hiperplano de separación. En esta figura se puede detallar que la clase positiva no tiene muchos errores, lo cual concuerda con una alta sensibilidad, mientras que la clase negativa tiene más errores lo que conlleva a tener una baja especificidad.

También podemos observar en la parte derecha de la misma gráfica la curva ROC, que nos indica de manera visual el desempeño de los tres clasificadores implementados. Se puede observar que los clasificadores que obtuvieron un mejor desempeño para nuestro experimento fueron los de la SVM con kernel lineal y el algoritmo de XGBoost, ambos resultados de un área bajo la curva del 0.93



Distribucion de probabilidad de las puntuaciones para SVM (RBF)



Curva ROC para los clasificadores implementados

Figura 17. Función de distribuciones y Curva ROC

5. Conclusiones

En este trabajo se llevó a cabo la implementación de técnicas de aprendizaje por transferencia aplicado a modelos de redes neuronales convolucionales para la clasificación de cuatro emociones: neutro, enojado, feliz y triste. En los experimentos realizados se demostró que utilizar el aprendizaje por transferencia mejoró las medidas de desempeño de todas las arquitecturas implementadas, principalmente la arquitectura LeNet diseñada experimentalmente, obteniendo resultados de hasta 78.9% en su precisión balanceada. Igualmente, se pudo comprobar una mejora en el rendimiento de la red de un 6.1% en comparación con la misma arquitectura sin usar aprendizaje por transferencia. Además se observaron desempeños muy similares al del aprendizaje por transferencia cuando se implementó la función triple pérdida en una SVM con kernel lineal, alcanzando resultados del 78.4% en su precisión balanceada.

Por otro lado, también se llevó a cabo un estudio comparativo entre diferentes clasificadores binarios: SVM, RF y XGBoost, basados en el plano Arousal y Valencia para la clasificación biclase de la base de datos Cohn-Kanade. La evaluación de las cuatro emociones mencionadas anteriormente, se unificaron de la siguiente forma: neutro-feliz en una clase positiva y triste-enojado en una clase negativa. De acuerdo a los resultados reportados, se pudo comprobar que el clasificador RF obtuvo el mejor desempeño con un 82.7% en su precisión balanceada, mientras que la SVM con kernel Gaussiano presentó el mayor porcentaje de sensibilidad con un 95.0%.

Finalmente, en este trabajo ha quedado demostrado, que para una base de datos pequeña, el aprendizaje por transferencia es una alternativa eficiente a la hora de clasificar emociones a partir de imágenes, logrando mejorar su resultado considerablemente. Sin embargo, también es posible concluir que con una base de datos más numerosa y bien robusta, es posible obtener una mejor adaptación para resolver una tarea en específico.

6. Referencias

[1] Mehrabian A. Communication without words. Psychol Today. 1968; 2 (4):53e56.

[2] Marco-Garcia S, Ferrer-Quintero M, Usall J, Ochoa S, Del Cacho N, Huerta-Ramos E. Facial emotion recognition in neurological disorders: a narrative review. Rev Neurol. 2019 Sep 1;69(5):207-219. Spanish, English. doi: 10.33588/rn.6905.2019047. PMID: 31364150.

[3] Mamonto, N.E., Maulana, H., Liliana, D.Y., & Basaruddin, T. (2018). Multimedia Content Development as a Facial Expression Datasets for Recognition of Human Emotions. IOP Conference Series: Materials Science and Engineering, 306.

[4] Priyanka A. Abhang, Bharti W. Gawali, Suresh C. Mehrotra, Chapter 5 – Emotion Recognition, Editor(s): Priyanka A. Abhang, Bharti W. Gawali, Suresh C. Mehrotra, Introduction to EEG- and Speech-Based Emotion Recognition, Academic Press, 2016, Pages 97-112, ISBN 9780128044902, https://doi.org/10.1016/B978-0-12-804490-2.00005-1.

[5] Shamir, L., Delaney, J. D., Orlov, N., Eckley, D. M., & Goldberg, I. G. (2010). Pattern recognition software and techniques for biological image analysis. PLoS computational biology, 6(11), e1000974. https://doi.org/10.1371/journal.pcbi.1000974

[6] I. Lasri, A. R. Solh and M. E. Belkacemi, "Facial Emotion Recognition of Students using Convolutional Neural Network," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019, pp. 1-6, https://doi: 10.1109/ICDS47004.2019.8942386.

[7] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh and A. Akan, "Real Time Emotion Recognition from Facial Expressions Using CNN Architecture," 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4, https://doi: 10.1109/TIPTEKNO.2019.8895215.

[8] Caroppo, A., Leone, A. & Siciliano, P. Comparison Between Deep Learning Models and Traditional Machine Learning Approaches for Facial Expression Recognition in Ageing

Adults. J. Comput. Sci. Technol. 35, 1127–1146 (2020). https://doi.org/10.1007/s11390-020-9665-4

[9] Jason C. Hung, Kuan-Cheng Lin, Nian-Xiang Lai, Recognizing learning emotion based on convolutional neural networks and transfer learning, Applied Soft Computing, Volume 84, 2019, 105724, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2019.105724

[10] Y. Wang, Y. Li, Y. Song and X. Rong, "The Application of a Hybrid Transfer Algorithm Based on a Convolutional Neural Network Model and an Improved Convolution Restricted Boltzmann Machine Model in Facial Expression Recognition," in IEEE Access, vol. 7, pp. 184599-184610, 2019, https://doi:10.1109/ACCESS.2019.2961161.

[11] S. A. -P. Raja Sekaran, C. Poo Lee and K. M. Lim, "Facial Emotion Recognition Using Transfer Learning of AlexNet," 2021 9th International Conference on Information and Communication Technology (ICoICT), 2021, pp. 170-174, doi: 10.1109/ICoICT52021.2021.9527512.

[12] S. Pandey, S. Handoo and Yogesh, "Facial Emotion Recognition using Deep Learning," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 248-252, doi: 10.1109/MECON53876.2022.9752189.

[13] Chowdary, M.K., Nguyen, T.N. & Hemanth, D.J. Deep learning-based facial emotion recognition for human–computer interaction applications. Neural Comput & Applic (2021). https://doi-org.udea.lookproxy.com/10.1007/s00521-021-06012-8

[14] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[15] D. Lakshmi, R. Ponnusamy, Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders, Microprocessors and Microsystems, Volume 82, 2021, 103834, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2021.103834.

[16] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101, https://doi:10.1109/CVPRW.2010.5543262.

[17] The Japanese Female Facial Expression (JAFFE) Database. Available: http://www.kasrl.org/jaffe.html

[18] R. A. Nugrahaeni and K. Mutijarsa, "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification," 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), 2016, pp. 163-168, https://doi:10.1109/ISEMANTIC.2016.7873831.

[19] J. Yang, D. Zhang, Z. Pan, D. Liu and J. Chen, "Facial Expression Recognition Based on Convolutional Denoising Autoencoder and XGBoost," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2019, pp. 149-154, https://doi: 10.1109/ITAIC.2019.8785596.

[20] Challenges in Representation Learning: Facial Expression Recognition Challenge 2013 dataset. https://www.kaggle.com/c/challenges-in-representation-learning-facialexpression-recognition-challenge/data

[21] Tseng, A., Bansal, R., Liu, J., Gerber, A. J., Goh, S., Posner, J., Colibazzi, T., Algermissen, M., Chiang, I. C., Russell, J. A., & Peterson, B. S. (2014). Using the circumplex model of affect to study valence and arousal ratings of emotional faces by children and adults with autism spectrum disorders. Journal of autism and developmental disorders, 44(6), 1332–1346. https://doi.org/10.1007/s10803-013-1993-6

[22] H. Sharma, S. Saurav, S. Singh, A. K. Saini and R. Saini, "Analyzing impact of image scaling algorithms on viola-jones face detection framework," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 1715-1718, doi: 10.1109/ICACCI.2015.7275860.

[23] Cerná, L., Cámara-Chávez, G., & Menotti, D. (2013). Face Detection: Histogram of Oriented Gradients and Bag of Feature Method.

[24] Carcagnì, P., Del Coco, M., Leo, M. et al. Facial expression recognition and histograms of oriented gradients: a comprehensive study. SpringerPlus 4, 645 (2015). https://doi.org/10.1186/s40064-015-1427-3

[25] A. Verma, P. Singh and J. S. Rani Alex, "Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition," 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), 2019, pp. 169-173, https://doi:10.1109/IWSSIP.2019.8787215.

[26] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, Andrea Mechelli, Chapter 10 - Convolutional neural networks, Editor(s): Andrea Mechelli, Sandra Vieira, Machine Learning, Academic Press, 2020, Pages 173-191, ISBN 9780128157398, https://doi.org/10.1016/B978-0-12-815739-8.00010-9

[27] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning (adaptive Computation And Machine Learning Series) MITpress, 2016.

[28] Hyun, Junhyuk & Seong, Hongje & Kim, Euntai. (2019). Universal Pooling -- A New Pooling Method for Convolutional Neural Networks.

[29] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015. paper | bibtex | paper content on arxiv.

[30] F. R. Mashrur, A. Dutta Roy and D. K. Saha, "Automatic Identification of Arrhythmia from ECG Using AlexNet Convolutional Neural Network," 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2019, pp. 1-5, doi: 10.1109/EICT48899.2019.9068806.

[31] Danish Vasan, Mamoun Alazab, Sobia Wassan, Babak Safaei, Qin Zheng, Image-Based malware classification using ensemble of CNN architectures (IMCEC), Computers & Security, Volume 92, 2020, 101748, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.101748

[32] P. Pooyoi, P. Borwarnginn, J. H. Haga and W. Kusakunniran, "Snow Scene Segmentation Using CNN-Based Approach With Transfer Learning," 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Chonburi, Thailand, 2019, pp. 97-100, https://doi: 10.1109/ECTI-CON47248.2019.8955140.

[33] N. Aneja and S. Aneja, "Transfer Learning using CNN for Handwritten Devanagari Character Recognition," 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 2019, pp. 293-296, https://doi:10.1109/ICAIT47043.2019.8987286.

[34] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions onKnowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, https://doi: 10.1109/TKDE.2009.191.

[35] Xu, W., He, J., & Shu, Y. (2020). Transfer Learning and Deep Domain Adaptation. In (Ed.), Advances and Applications in Deep Learning. IntechOpen. https://doi.org/10.5772/intechopen.94072

[36] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823, https://doi:10.1109/CVPR.2015.7298682.

[37] J. Suriya Prakash, K. Annamalai Vignesh, C. Ashok and R. Adithyan, "Multi class Support Vector Machines classifier for machine vision application," 2012 International Conference on Machine Vision and Image Processing (MVIP), 2012, pp. 197-199, https://doi: 10.1109/MVIP.2012.6428794.

[38] Peter Wittek, 7 - Supervised Learning and Support Vector Machines, Editor(s): Peter Wittek, Quantum Machine Learning, Academic Press, 2014, Pages 73-84, ISBN 9780128009536, https://doi.org/10.1016/B978-0-12-800953-6.00007-4.

[39] Q. Zhou, W. Lan, Y. Zhou and G. Mo, "Effectiveness Evaluation of Anti-bird Devices based on Random Forest Algorithm," 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), 2020, pp. 743-748, https://doi: 10.1109/ICCSS52145.2020.9336891.

[40] V. Jain and A. Phophalia, "Exponential Weighted Random Forest for Hyperspectral Image Classification," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 3297-3300, doi: 10.1109/IGARSS.2019.8897862.

[41] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang and Y. Si, "A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost," in IEEE Access, vol. 6, pp. 21020-21031, 2018, doi: 10.1109/ACCESS.2018.2818678.

[42] Y. Niu, "Walmart Sales Forecasting using XGBoost algorithm and Feature engineering," 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), 2020, pp. 458-461, https://doi: 10.1109/ICBASE51474.2020.00103.

[43] Contreras, Leonardo E., Fuentes, Héctor J., & Rodríguez, José I.. (2020). Academic performance prediction by machine learning as a success/failure indicator for engineering students. Formación universitaria, 13(5), 233-246. https://dx.doi.org/10.4067/S0718-50062020000500233

[44] Borja-Robalino, Ricardo & Monleon-Getino, Antonio & Benedé, Jose. (2020). Estandarización de Métricas de Rendimiento para Clasificadores Machine y Deep Learning.

[45] Kumar R, Indrayan A. Receiver operating characteristic (ROC) curve for medical researchers. Indian Pediatr. 2011 Apr; 48(4):277-87. https://doi: 10.1007/s13312-011-0055-4. PMID: 21532099.