

**Desarrollo y evaluación de un sistema inteligente para la identificación de posturas  
ergonómicas inadecuadas en ambientes de trabajo remoto**

Mario Alberto Triana Aguilar

Asesor

Jorge Luis Quintero López

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería - ECBTI

Especialización en Ciencia de Datos e Ingeniería

2025

## **Agradecimientos**

Agradezco al tutor Jorge Luis Quintero por la idea de proyecto, su acompañamiento y asesoría durante la realización de este, a Dios, mis padres y todos los que me han apoyado durante este proceso.

## Resumen

En entornos laborales actuales de oficina y trabajo remoto, la detección de malas posturas se ha convertido en un aspecto de gran importancia para evitar problemas de salud y preservar el bienestar de los empleados. Las jornadas extensas de trabajo sentado en una misma posición con posturas inadecuadas acarrearán problemas físicos y dolencias, las cuales afectan el desempeño y también la calidad de vida de las personas. El objetivo general de este estudio es realizar un sistema de detección de posturas, a través de técnicas de Machine Learning, utilizando la biblioteca Pytorch. Esto con el fin de identificar posturas inadecuadas presentes en varias secuencias de imágenes; en aras de prevenir los problemas de salud generados por las malas posturas. Se trata de un estudio cuantitativo y comparativo ya que se utilizan distintas métricas estadísticas como precisión, sensibilidad, F1-score, entre otras para evaluar la eficiencia del modelo realizado mediante la librería Pytorch y el ya establecido por Teachable Machine.

Para ello, se utilizó un conjunto de datos que contiene la captura de varias imágenes de distintos sujetos adoptando varias posturas. Gracias a este conjunto de datos es que se generan los modelos de entrenamiento y posteriormente se validan. Adicional a esto se realizó una evaluación de la eficiencia y precisión de los modelos, utilizando métricas como precisión, sensibilidad y especificidad, para garantizar un desempeño adecuado en la detección de posturas.

Los resultados de lo anterior fueron comparados con los que arroja la herramienta Teachable Machine, que es una plataforma web que emplea modelos de redes neuronales pre entrenados y permite a los usuarios entrenar modelos de Machine Learning sin necesidad de programación.

**Palabras clave:** Machine Learning, Pytorch, detección de posturas, Teachable Machine, visión computacional.

## Abstract

In current office and remote work environments, the detection of poor posture has become a very important aspect to avoid health problems and preserve the well-being of employees. Long working hours sitting in the same position with inadequate postures lead to physical problems and ailments, which affect performance and also the quality of life of people. The general objective of this study is to develop a posture detection system, through Machine Learning techniques, using the Pytorch library. This, in order to identify inadequate postures present in several sequences of images; in order to prevent health problems caused by bad postures. This is a quantitative and comparative study since different statistical metrics such as accuracy, sensitivity, F1-score, among others, are used to evaluate the efficiency of the model made using the Pytorch library and the one already established by Teachable Machine.

For this purpose, a dataset containing the capture of several images of different subjects adopting various postures was used. Thanks to this data set, training models are generated and subsequently validated. In addition to this, an evaluation of the efficiency and accuracy of the models was performed, using metrics such as accuracy, sensitivity and specificity, to ensure adequate performance in the detection of postures.

The results of the above were compared with those of the Teachable Machine tool, which is a web platform that uses pre-trained neural network models and allows users to train Machine Learning models without the need for programming.

**Keywords:** Machine Learning, Pytorch, posture detection, Teachable Machine, computer vision.

## Tabla de Contenido

Introducción .....	8
Justificación .....	10
Objetivos.....	11
Objetivo General.....	11
Objetivos Específicos.....	11
Marco de Referencia.....	12
Marco Teórico.....	12
Marco Conceptual.....	14
Marco Normativo.....	15
Metodología.....	17
Tipo de Investigación y Enfoque .....	17
Resultados.....	24
Recolección y Procesamiento de Datos .....	24
Data Augmentation .....	27
Entrenamiento y Modelo CNN Propio.....	28
Modelo de Predicción Hecho en Teachable Machine.....	37
Conclusiones.....	41
Recomendaciones para Trabajos Futuros .....	42
Referencias Bibliográficas .....	43

**Lista de Tablas**

<b>Tabla 1</b> <i>Objetivo Específico 1</i> .....	18
<b>Tabla 2</b> <i>Objetivo Específico 2</i> .....	19
<b>Tabla 3</b> <i>Objetivo Específico 3</i> .....	21
<b>Tabla 4</b> <i>Principales Características de los Datasets Usados</i> .....	24
<b>Tabla 5</b> <i>Resultados del Modelo Propio</i> .....	33
<b>Tabla 6</b> <i>Resultados del Modelo de Teachable Machine</i> .....	38

## Lista de Figuras

<b>Figura 1</b> <i>Imagen de Ejemplo Dataset Roboflow</i> .....	25
<b>Figura 2</b> <i>Imagen Propia de Buena Postura</i> .....	26
<b>Figura 3</b> <i>Imagen Propia de Mala Postura</i> .....	26
<b>Figura 4</b> <i>Imagen procesada</i> .....	28
<b>Figura 5</b> <i>Función Relu (Cifuentes et al., 2019)</i> .....	29
<b>Figura 6</b> <i>Matriz de Confusión Datos de Entrenamiento</i> .....	30
<b>Figura 7</b> <i>Curva ROC Datos de Entrenamiento</i> .....	31
<b>Figura 8</b> <i>Matriz de Confusión Datos de Prueba</i> .....	32
<b>Figura 9</b> <i>Curva ROC Datos de Prueba</i> .....	32
<b>Figura 10</b> <i>Gráfica de Pérdida Durante el Entrenamiento y Validación</i> .....	34
<b>Figura 11</b> <i>Gráfica de Precisión en la Validación</i> .....	35
<b>Figura 12</b> <i>Predicciones del Modelo Hecho en Pytorch para Imágenes Externas</i> .....	36
<b>Figura 13</b> <i>Matriz de Confusión Modelo Teachable Machine</i> .....	37
<b>Figura 14</b> <i>Curva ROC Datos de Prueba Modelo Teachable Machine</i> .....	38
<b>Figura 15</b> <i>Predicciones del Modelo de Techable Machine para Imágenes Externas</i> .....	39

## Introducción

El aprendizaje automático (Machine Learning) como rama de la Inteligencia Artificial, se enfoca en identificar y clasificar patrones presentes en un conjunto de datos o información visual, lo que lo hace un elemento clave de la visión computacional. El avance en visión computacional y en procesamiento de imágenes ha hecho que las técnicas de reconocimiento de la actividad humana (HAR) sean más accesibles (Solà, 2020). A su vez, las redes neuronales convolucionales (CNNs) permiten analizar varias capas de datos visuales y reconocer distintos patrones visuales complejos que están presentes en imágenes. Lo anterior ha sido de gran importancia para la detección de posturas (Deng et al., 2022) y la identificación de emociones mediante la observación de cambios temporales en la expresión facial (Fasiha & Prasadu, 2022).

Los modelos de Redes Neuronales Convolucionales (CNN) han sido utilizados en la identificación de posturas por su capacidad para ofrecer representaciones visuales complejas en múltiples escalas (Ogundokun et al., 2022). A través del uso de técnicas basadas en aprendizaje automático (ML) se puede llegar a entender la relación que hay entre los factores de riesgo ergonómicos y la salud de los trabajadores (MassirisFernández et al., 2020). Estos modelos pueden examinar tendencias a largo plazo en los datos ergonómicos, previendo problemas de salud antes de que surjan. Dicha información puede ser el inicio para tomar medidas que mejoren los puestos de trabajo.

Se ha visto que en los entornos laborales actuales, especialmente en el trabajo remoto han aumentado los problemas de salud, tales como las lesiones resultantes de trabajos repetitivos; que generan trastornos musculoesqueléticos (Chan et al., 2020), los cuales se manifiestan como dolores de cuello y espalda (Wu, 2021). Estos problemas se encuentran asociados a las malas posturas, la ausencia de pausas activas y a espacios de trabajo inadecuados. A pesar de que

existen normas como la ISO 11226:2000 que establece recomendaciones para distintas posturas estáticas, y herramientas tecnológicas como el modelo propuesto por (Noor & Park, 2023), que utiliza estimación de poses para identificar puntos claves del cuerpo tomando como referencia las secuencias de video y crea mapas de calor conjuntos para representar las posturas; su implementación en un contexto real es limitada. Por ende, surge la necesidad de desarrollar soluciones de monitoreo basadas en Machine Learning que permitan identificar posturas inadecuadas en distintos entornos de oficina y trabajo remoto. También se ha demostrado que cuando existen intervenciones ergonómicas en el entorno de trabajo, se ve un aumento en la satisfacción de trabajo por parte de los empleados y disminución de fatiga muscular, la cual en entornos de trabajo remoto tiende a aumentar, debido a las limitaciones del mismo (Gumasing et al., 2023).

Este trabajo responde a una necesidad creciente en los entornos laborales remotos y de oficina: la detección automatizada de posturas inadecuadas que afectan la salud de los empleados. Para ello, se emplean algoritmos de Machine Learning y una plataforma interactiva, Teachable Machine, que permite realizar predicciones sin requerir conocimientos de programación.

### **Justificación**

Este proyecto aplicado es un estudio exploratorio enfocado al área de salud ocupacional, en ambientes de oficina y trabajo remoto. Se pretende clasificar diferentes tipos posturas de las personas con el fin de detectar las que son malas y así se puedan prevenir lesiones musculoesqueléticas y otros problemas de salud. Ya sea en un contexto de trabajo remoto o en una oficina, es de gran importancia asegurar que los trabajadores tengan una postura adecuada, ya que si no esto produce consecuencias malas al empleado y a la empresa. A título personal, como ingeniero biomédico, mi principal interés es presentar soluciones a problemas relacionados con la salud, en este caso, enfocado a la salud ocupacional, donde se puede aplicar el conocimiento en Big Data y Machine Learning para brindar una solución a los inconvenientes generados por las malas posturas.

## Objetivos

### Objetivo General

Diseñar y validar un modelo de aprendizaje automático para la identificación de posturas corporales inadecuadas en entornos de trabajo remoto, con el fin de prevenir trastornos musculoesqueléticos asociados a malas prácticas posturales.

### Objetivos Específicos

Recolectar y procesar un conjunto de imágenes que representen diferentes posturas corporales adecuadas e inadecuadas, utilizando una base de datos pública y registros propios.

Desarrollar e implementar un modelo de redes neuronales convolucionales utilizando Pytorch para la clasificación de posturas correctas e incorrectas.

Comparar los resultados arrojados por el modelo propio frente a uno ya desarrollado (Teachable Machine) mediante métricas de precisión, sensibilidad y especificidad.

## Marco de Referencia

### Marco Teórico

El aprendizaje supervisado es una de las ramas más importantes del aprendizaje automático, se basa en entrenar un modelo, tomando como referencia un conjunto de datos previamente etiquetado. Consiste en modelar una función  $f : X \rightarrow Y$  utilizando pares  $(x_i, y_i)$  donde  $x_i$  son entradas (en este caso imágenes) y  $y_i$  son las etiquetas conocidas. El objetivo de este tipo de aprendizaje es encontrar una función  $f$  que modele la dependencia  $P(x, y)$  entre la entrada  $x$  y la salida  $y$ . A esto se le suma el concepto de error o pérdida para medir la concordancia entre la predicción  $f(x)$  y la salida esperada. Esta pérdida es evaluada por la función  $L(f(x), y) \rightarrow R^+$  (McNicholas & Tait, 2019).

En visión computacional, donde se busca que las máquinas puedan analizar datos visuales; las redes neuronales convolucionales (CNN) han mostrado ser de gran utilidad en aplicaciones vinculadas con la salud y el monitoreo a personas de la tercera edad. Actualmente, son los modelos más importantes del Deep Learning y están especialmente diseñadas para procesar imágenes como valores de entrada. Una red neuronal simple sigue la siguiente estructura: la primera capa, conocida como capa de convolución, extrae las características de la imagen por medio de un filtro (Kernel). Este filtro se va desplazando por la imagen, haciendo un barrido y a medida que hace esto realiza una serie de operaciones que generan un mapeo de activación con las características de mayor peso o importancia. La segunda capa se encarga de extraer la información más importante de la capa de convolución y reducir el tamaño de la imagen por medio de una operación llamada max pooling. Posteriormente, las capas de clasificación procesan las características extraídas en las etapas anteriores y generan la predicción final. Esta se interpreta mediante una función de activación que transforma las salidas

en probabilidades asociadas a cada clase (Soffer et al., 2019). En resumen, estas redes están diseñadas para extraer automáticamente características visuales relevantes de imágenes mediante filtros, capas convolucionales y operaciones de reducción de dimensionalidad, lo que permite una clasificación más precisa (Lubinus Badillo et al., 2021).

Según (Hasib et al., 2021), el uso de CNN en sistemas de visión computacional facilita la detección de caídas al procesar datos de imágenes y video en tiempo real. Esta investigación hace hincapié en que la precisión de estos sistemas puede aumentar implementando técnicas de preprocesamiento de imágenes y modelos de aprendizaje profundo; lo cual mejora la habilidad del algoritmo para detectar caídas.

Las arquitecturas de las CNN pueden ser desarrolladas en Pytorch o TensorFlow, que son librerías de código abierto especializada en modelos de Machine Learning. Estas librerías son útiles para analizar imágenes y videos mediante CNN y técnicas de pose estimation, como por ejemplo la estimación de puntos clave, la cual fue hecha en (Xie & Guo, 2019); donde vemos que por medio de TensorFlow y OpenCV se puede realizar el reconocimiento y análisis de estos puntos clave en imágenes del cuerpo humano y a partir de esto identificar parámetros específicos. En este trabajo se utilizaron imágenes de diferentes posturas, tanto adecuadas como inadecuadas durante una jornada laboral como lo son: postura hacia atrás, postura hacia adelante y postura erguida, tomando como referencia el estudio piloto que hizo (Kulikajevs et al., 2021). Con estimaciones de este tipo se pueden prevenir desórdenes musculoesqueléticos e identificar distintos patrones de postura que afectan la salud, como lo plantea (Sohrabi et al., 2024).

Adicional a lo anterior, se realizó un modelo de aprendizaje supervisado utilizando una CNN desarrollada en Pytorch, con el fin de clasificar imágenes de distintos sujetos en entornos laborales según su postura: correcta o incorrecta. Se utilizó como conjunto de datos una serie de

imágenes previamente etiquetadas, donde cada imagen fue clasificada manualmente en una de las dos categorías ya dichas. Lo anterior constituyó la base del entrenamiento del modelo; permitiéndole ajustar sus parámetros internos en cada época (iteración), para minimizar el error de predicción.

Se utilizaron etiquetas binarias para cada clase (0 = correcta, 1 = incorrecta) y métricas como precisión, sensibilidad, especificidad y F1 score para la evaluación del modelo, la cual se llevó a cabo con imágenes distintas a las del conjunto de entrenamiento.

Otra técnica útil para los algoritmos de clasificación es data augmentation, la cual consiste en generar nuevas imágenes modificadas a partir de las originales, esto para aumentar el número de imágenes de nuestro conjunto de entrenamiento. Esto también permite ampliar la variabilidad del conjunto de datos.

Además del modelo propio realizado en Pytorch, se utilizó la herramienta web Teachable Machine, que también implementa un enfoque de aprendizaje supervisado. Esta plataforma permite crear modelos de Machine Learning sin necesidad de utilizar código o un conocimiento sólido en aprendizaje automático. Nos brinda una interfaz amigable que provee a los usuarios clasificación, entrenamiento del modelo y evaluación para cualquier número de clases (Carney et al., 2020). Se pueden introducir imágenes y realizar un modelo de predicción de estas, que es de gran utilidad y se puede emplear como referencia para compararlo con un modelo de predicción de desarrollo propio.

### **Marco Conceptual**

- **Píxel:** unidad mínima que forma una imagen digital, puede ser negro, blanco o tener matices de gris.

- Kernel: matriz de números que se utiliza para enforzar, grabar y detectar características de la imagen.
- Clasificación de imágenes: asignar una etiqueta o categoría a una imagen.
- Reconocimiento de patrones: capacidad que tiene el algoritmo para identificar estructuras dentro de los datos.
- Entrenamiento: proceso donde un modelo de Machine Learning aprende a realizar una tarea específica.

### **Marco Normativo**

La norma ISO 11226:2000 es una norma internacional que tiene como objetivo evaluar las posturas de trabajo estáticas. Esta norma establece que las labores deben proporcionar suficiente variación física y mental, de tal manera que las tareas estén distribuidas equilibradamente; facilitando la comunicación y el aprendizaje. Así mismo en el apartado 7.1 del libro “Posturas de trabajo evaluación del riesgo”, (Fernández, 2015) se declaran unas directrices para identificar si una postura es aceptable o no, Estas pueden resumirse en dos aspectos principales:

1. Parámetros de referencia:
  - a. Ángulos: rangos articulares aceptables para las articulaciones (cuello, tronco, hombros, codos, muñecas, caderas, rodillas y tobillos). Están medidos en grados y varían según la articulación.
    - i. Cuello: la inclinación hacia adelante máxima permitida ese de  $20^{\circ}$ .
    - ii. Tronco: inclinación de  $20^{\circ}$  es aceptable, más de  $60$  se considera de alto riesgo para la postura.
    - iii. Hombros: elevación mayor a  $60^{\circ}$  respecto al plano horizontal es riesgosa.

- iv. Codos: el rango de flexión recomendado es entre 70° y 120°.
- v. Muñecas: las desviaciones laterales mayor a 15° deben evitarse.
- vi. Caderas y rodillas: en la posición de sentado, el rango de flexión recomendado es entre 90° y 120°.
- vii. Tobillos: no se recomienda una flexión plantar mayor a 20°.
  - b. Duración: se especifican intervalos de tiempo específicos para mantener una misma postura, diferenciando las aceptables y las incorrectas.
  - c. Frecuencia: repetitividad de las posturas y su impacto en la salud del trabajador.
- 2. Evaluación:
  - a. Desviaciones: se analiza el grado de desviación respecto a una postura cómoda.
  - b. Tiempo de mantenimiento: el tiempo en el que se mantiene una postura específica y si supera el tiempo recomendado.
  - c. Correcciones: medidas para modificar una postura considerada incorrecta o reducir su duración.

En complemento con lo anterior, se establece un procedimiento para determinar las posturas, que incluye lo siguiente:

- a. Registro: documentar las posturas durante la jornada laboral.
- b. Medición de ángulos: medir los valores de los ángulos articulares.
- c. Duración: evaluar el tiempo y frecuencia con que se mantiene una misma postura.
- d. Clasificación de la postura: determinar si es aceptable, tolerable o no aceptable.
- e. Medidas: cambios en la distribución del puesto de trabajo o en los métodos de ejecución de las labores, en aras de mejorar la postura.

## Metodología

### Tipo de Investigación y Enfoque

Este proyecto de investigación se basa en una metodología no experimental, comparativa, cuantitativa y aplicada, que se enfoca en el desarrollo y validación de un modelo de aprendizaje automático, orientado a la detección de posturas corporales en ambientes de trabajo remoto. El enfoque no experimental está basado en la observación y análisis de datos existentes previamente etiquetados, que fueron recolectados a partir de imágenes que reflejan posturas reales (correctas e incorrectas), sin intervenir directamente sobre los sujetos o las condiciones de captura de las imágenes del dataset. El estudio utilizó un modelo de aprendizaje automático supervisado, el cual fue elaborado a partir de redes neuronales convolucionales (CNN) e implementado con la librería Pytorch, con evaluación basada en métricas de rendimiento (precisión, sensibilidad, especificidad, F1-score). Además, se utilizó la herramienta Teachable Machine como modelo de referencia. Las imágenes fueron preprocesadas y aumentadas mediante técnicas de data augmentation para reducir el sobreajuste y mejorar la capacidad de clasificación. Ambos modelos fueron entrenados con datasets equivalentes para garantizar condiciones iguales y controladas en la comparación.

En las siguientes tablas vamos a detallar las actividades correspondientes durante el desarrollo de esta investigación, teniendo en cuenta cada uno de los objetivos específicos

**Tabla 1***Objetivo Específico 1*

Actividad	Detalle
Tomar fotos frontales de posturas adecuadas e inadecuadas de 4 sujetos, en un entorno de trabajo remoto. Seleccionar un dataset de una base de datos pública que contenga fotografías en vista lateral de personas sentadas en trabajo remoto con posturas correctas e incorrectas	La captura de las imágenes propias se realizó con una cámara Dji Osmo Pocket 3 a una resolución de 3840 x 2160 pixeles en formato 16:9 en dos entornos distintos y simulando entornos remotos. También se utilizó la herramienta Kinovea, la cual permite realizar un análisis biomecánico de videos e imágenes por medio de la medición de ángulos articulares. Con esta herramienta se verificó que los ángulos del cuello, hombro y codo en las fotos tomadas fueran de acuerdo con los parámetros de la norma ISO 11226:2000, clasificando cada imagen como correcta o incorrecta. El dataset de la base de datos pública utilizado fue extraído de Roboflow, una plataforma web que ofrece datasets públicos especialmente diseñados para proyectos de detección de objetos y clasificación de imágenes. Este conjunto de datos está compuesto por un total de 1477

---

imágenes de personas en posición sentada dentro de ambientes de trabajo remoto u oficina. Las imágenes están organizadas en tres subconjuntos de datos: entrenamiento (1034 imágenes), validación (297 imágenes) y evaluación (146 imágenes).

---

*Nota. Obtención y recolección de datos propios y de base de datos pública*

## **Tabla 2**

### *Objetivo Específico 2*

Actividad	Detalle
Elaboración de un modelo de aprendizaje automático, basado en redes neuronales convolucionales, utilizando Pytorch	Habiendo previamente cargado las imágenes, estas se transformaron en tensores, que son estructuras de datos multidimensionales utilizadas en Pytorch para representar imágenes. Luego, se organizan los datos en tres subconjuntos de datos acuerdo con las carpetas del conjunto de datos original (“train”, “test” y “validation”) en las cuales el modelo almacenará cada uno de los elementos según corresponda. A cada uno se le asigna un dataloader, que se encarga de agrupar los datos en lotes (batches) para

---

---

optimizar el flujo de datos en el transcurso del entrenamiento. Además, las imágenes fueron redimensionadas a 112x112 píxeles en escala de grises, para asegurar la homogeneidad de los datos de entrada. Después de esto, con el fin de mejorar la capacidad de clasificación del modelo y reducir el sobreajuste, se aplicó “data augmentation” o aumento de datos para el entrenamiento. Estas nuevas imágenes usadas en el aumento de datos fueron generadas utilizando transformaciones de las imágenes originales como giros horizontales, ajustes de brillo y contraste, rotación y desenfoco; esto mediante la librería albumentations, donde se triplicó el número de imágenes de entrenamiento. El modelo CNN empleado consta de tres capas convolucionales, encargadas de la extracción de características de las imágenes. Cada capa incluye distintas operaciones como: convolución, normalización por lotes (BatchNorm), activación ReLU, y MaxPooling, además de

---

---

capas de Dropout que reducen la probabilidad de sobreajuste. La salida de estos bloques se aplanó y pasó por una capa densa con 64 neuronas, seguida de una capa de salida con dos neuronas, correspondiente a las clases "correcta" e "incorrecta". El modelo fue entrenado utilizando el optimizador Adam, función de pérdida CrossEntropyLoss, y política de reducción de tasa de aprendizaje mediante ReduceLROnPlateau.

---

*Nota. Descripción del modelo de Machine Learning*

### **Tabla 3**

#### *Objetivo Específico 3*

Actividad	Detalle
Pruebas y comparación de resultados	Para evaluar y comparar del desempeño del modelo y analizar estos resultados, se emplearon dos fuentes de datos que fueron la base de datos pública Roboflow y las imágenes propias capturadas. El conjunto de imágenes propias se dividió equitativamente, se utilizaron dos sujetos para entrenamiento y dos para prueba. También se utilizó la

---

---

inicialización de pesos mediante el método de He (Kaiming), que ayuda a mantener una varianza constante a lo largo de redes profundas. En el entrenamiento, se utilizó la función de pérdida CrossEntropyLoss junto con el optimizador Adam, configurado con una tasa de aprendizaje inicial de 0.01 y un weight decay de  $1e-4$ . En aras de obtener una mejor generalización del modelo, se utilizaron técnicas como

ReduceLROnPlateau que ajusta la tasa de aprendizaje cuando no hay mejoras y Early Stopping, que finaliza el entrenamiento si la pérdida de validación no mejora tras 5 épocas consecutivas.

El rendimiento del modelo se evaluó a través de métricas como precisión, sensibilidad, especificidad y F1-score, junto con unas matrices de confusión y curvas ROC.

También, se realizó una prueba con 10 imágenes que no pertenecían al dataset para observar la predicción del modelo con imágenes ajenas al entrenamiento y test. A

---

---

las predicciones se sumó una probabilidad de confianza, para detectar situaciones de baja precisión y saber dónde se requiere enfocar las posibles mejoras del algoritmo. Además, se hizo uso de la herramienta Teachable Machine, a esta se le suministró el mismo dataset que al modelo anterior y se fijaron los mismos valores en los parámetros tamaño del lote, tasa de aprendizaje y épocas. Luego, Teachable Machine realizó el entrenamiento y arrojó un código que contiene un modelo de Machine Learning ya establecido, basado en Keras que es una biblioteca de Tensor Flow. Este código fue adaptado para que realizara la evaluación del modelo utilizando los mismos datos y las mismas métricas del modelo anterior. Lo anterior, para garantizar que ambos modelos tuvieran las mismas condiciones y así obtener una comparación más precisa.

---

*Nota. Pruebas y análisis realizado*

## Resultados

### Recolección y Procesamiento de Datos

El dataset empleado en este proyecto proviene de dos fuentes. La fuente de datos 1 que son imágenes obtenidas de la base de datos pública (Roboflow) y la fuente de datos 2 que son fotografías capturadas por una cámara propia.

#### Tabla 4

##### *Principales Características de los Datasets Usados*

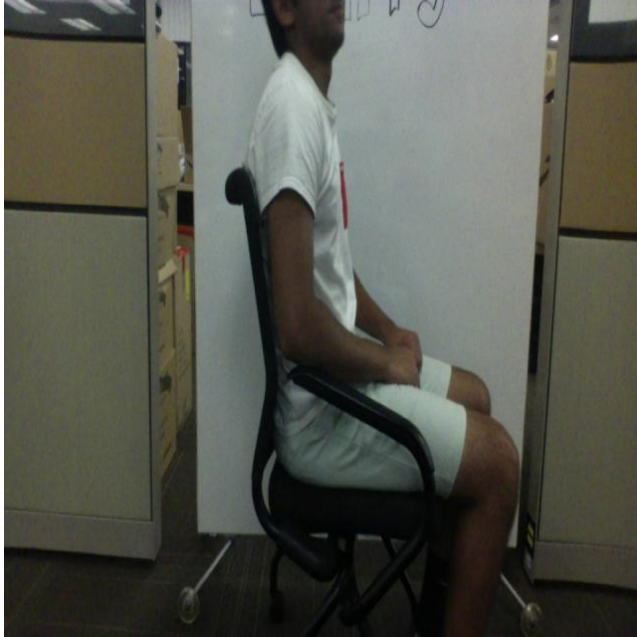
Atributo	Fuente de datos 1	Fuente de datos 2
Origen	Base de datos pública (Roboflow)	Fotografías captadas por cámara
Clases	Correcta / Incorrecta	Correcta / Incorrecta
Cantidad total de imágenes	1477	1641
Cantidad imágenes train	515 / 519	557 / 570
Cantidad imágenes test	72 / 74	108 / 109
Cantidad imágenes validación	160 / 137	160 / 137
Vista de la postura	Lateral	Frontal
Resolución	640x640	640x640 y 3840x2160

*Nota. Descripción de los datasets empleados en este estudio.*

A continuación se muestra un ejemplo de las imágenes de la base de datos pública y de elaboración propia, estas últimas con la medición de los ángulos de posturas correctas e incorrectas.

**Figura 1**

*Imagen de Ejemplo Dataset Roboflow*

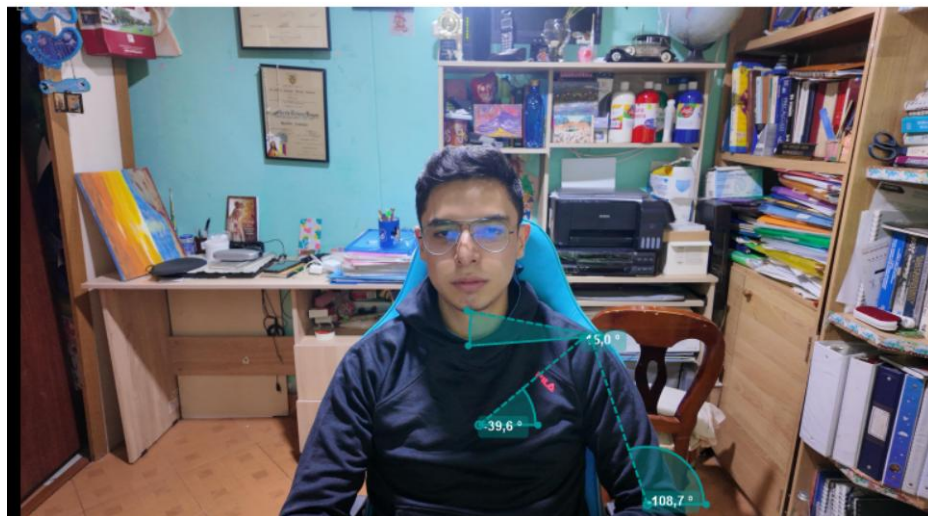


*Nota. Imagen extraída del dataset Roboflow.*

Para asegurarnos que las imágenes propias capturadas correspondieran a una buena o mala postura, se hizo uso de la herramienta Kinovea para medir los ángulos articulares de cuello, hombro y codo, como se ve en las figuras 2 y 3. De tal manera que los valores medidos de una buena o mala postura estuvieran dentro de los umbrales establecidos por la norma ISO 11226:2000.

**Figura 2**

*Imagen Propia de Buena Postura*



*Nota Imagen de elaboración propia*

**Figura 3**

*Imagen Propia de Mala Postura*



*Nota. Imagen de elaboración propia*

Las imágenes fueron organizadas en un diccionario que contiene tres claves principales: “train”, “valid” y “test”, cada una con sus respectivas imágenes y etiquetas (0 para postura correcta y 1 para postura incorrecta). Las imágenes fueron leídas utilizando OpenCV, se convirtieron a formato RGB y sus dimensiones fueron ajustadas a 112x112 píxeles.

### **Data Augmentation**

Con el fin de mejorar el rendimiento del modelo se incorporó un algoritmo de data augmentation al conjunto de datos de entrenamiento para triplicar la cantidad de datos y facilitar el aprendizaje del modelo. Para ello, se utilizó la librería albumentations, la cual aplica transformaciones aleatorias a las imágenes de entrenamiento, lo cual aumenta la diversidad del dataset y mejora la capacidad de generalización del modelo. Las transformaciones realizadas a las imágenes fueron las siguientes:

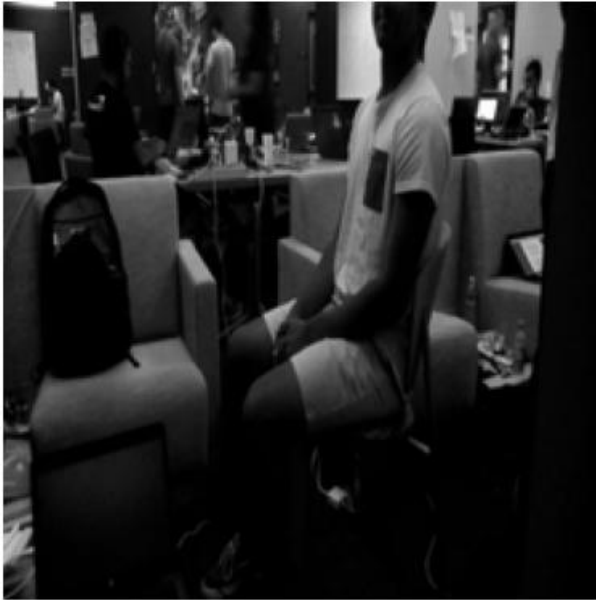
1. Giro horizontal (HorizontalFlip): para posturas simétricas
2. Ajuste de brillo/contraste (RandomBrightnessContrast) en un 50% de los casos, para robustecer el modelo ante cambios de iluminación.
3. Rotaciones (Rotate,  $\pm 15^\circ$ ): simula variaciones naturales en la postura.
4. Desenfoque gaussiano (GaussianBlur, 30% de probabilidad): para evitar que el modelo se fije en detalles irrelevantes.

Por cada imagen del conjunto de entrenamiento, se producen dos imágenes adicionales, dando como resultado un dataset de entrenamiento triplicado con respecto al original. Esto fue implementado para prevenir el sobreajuste y mejorar el rendimiento del modelo. Luego, usando torchvision.transforms, las imágenes se convirtieron a escala de grises, se volvieron a redimensionar a una escala de 112x112 para luego transformarlas en tensores, los cuales son estructuras de datos multidimensionales utilizadas para representar los datos visuales en el

ambiente computacional. Después, los datos se organizaron en tres subconjuntos: “train”, “test” y “validation” y normalizaron dejando como media 0.5 y desviación estándar 0.5, para que las imágenes estén en un rango que facilite el entrenamiento del modelo.

#### **Figura 4**

*Imagen procesada*



*Nota. Imagen extraída del dataset Roboflow*

#### **Entrenamiento y Modelo CNN Propio**

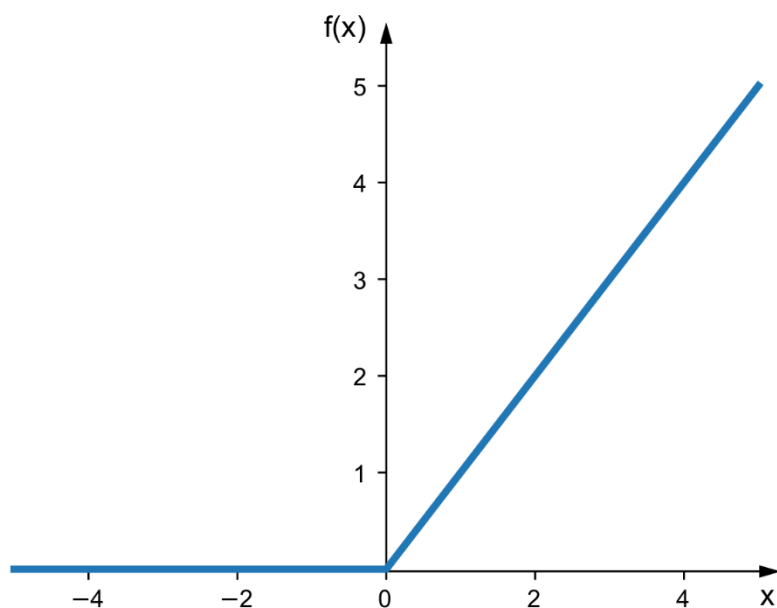
Se utilizó un modelo de Redes Neuronales Convolucionales (CNN) realizado con la librería de código abierto Pytorch, la cual es muy utilizada para el desarrollo de algoritmos de Deep Learning. Antes del entrenamiento, utilizando “DataLoaders”, cada conjunto fue dividido en lotes para optimizar la alimentación de datos durante el entrenamiento. Este modelo CNN está formado en primera instancia por tres bloques de convolución, cada uno con las siguientes operaciones:

- Convolución en dos dimensiones, fijando un kernel 3x3 y padding de 1
- Normalización por lotes
- Función de activación ReLU (Figura 4), para introducir no linealidad al modelo
- Max Pooling (2x2) para reducir las dimensiones a la mitad
- Dropout, que consiste en desactivar aleatoriamente canales de neuronas para

prevenir el sobreajuste

### Figura 5

*Función Relu (Cifuentes et al., 2019)*



*Nota. Imagen extraída de (Cifuentes et al., 2019).*

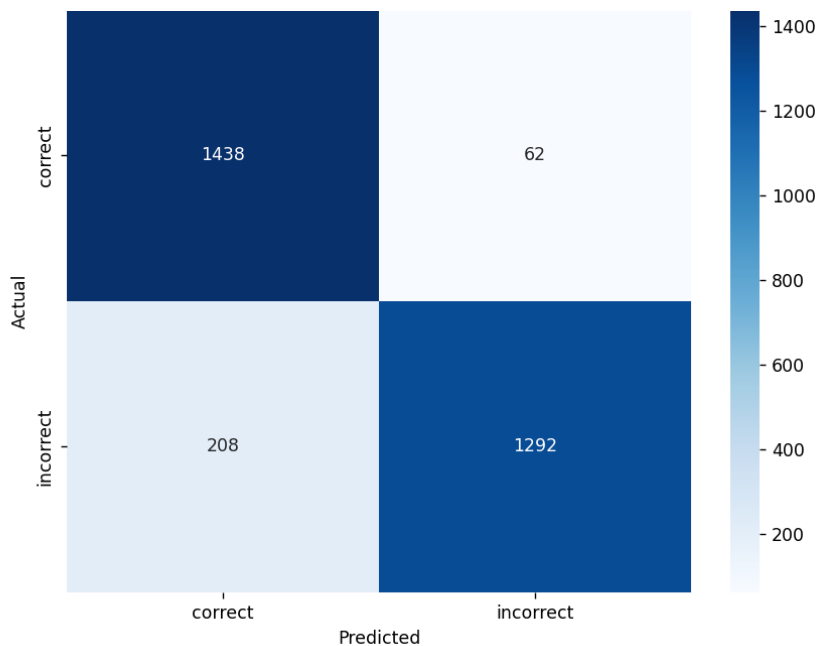
Después, el tensor que se obtiene como salida se aplana y se transforma en un vector unidimensional de 6272 valores, que pasa por dos capas lineales. En la primera este vector es transformado a un vector de 64 neuronas en donde se aplica normalización por lotes, función de activación ReLU y dropout; para luego pasar por una segunda capa lineal donde la salida de 64

dimensiones fue transformada en una salida de 2 dimensiones y pasó a través de dos capas lineales, que corresponde a las dos clases a predecir (correcta e incorrecta).

Los resultados para este modelo fueron los siguientes:

### Figura 6

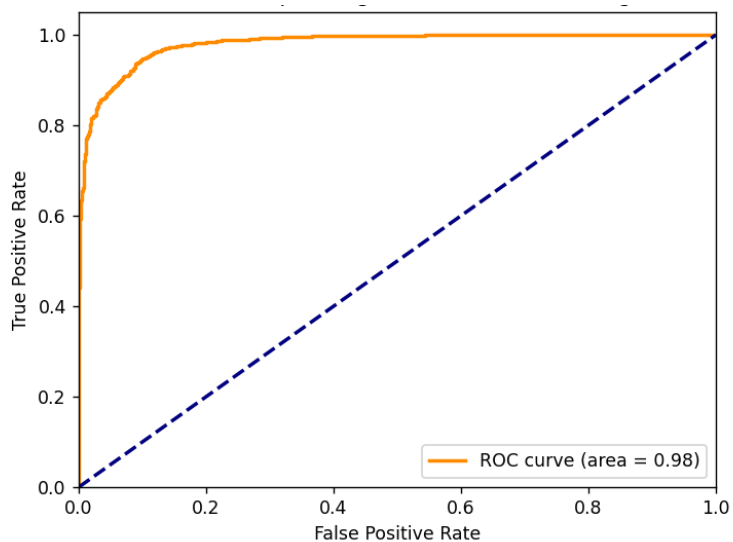
*Matriz de Confusión Datos de Entrenamiento*



*Nota. Imagen de elaboración propia.*

**Figura 7**

*Curva ROC Datos de Entrenamiento*

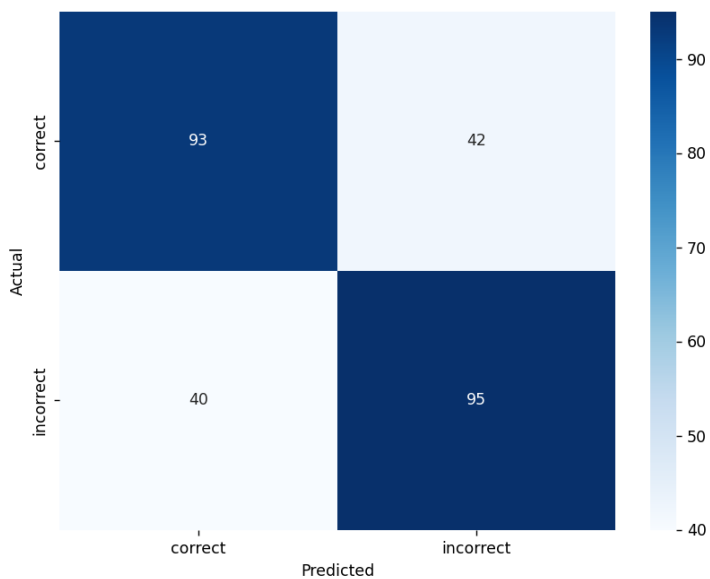


*Nota. Imagen de elaboración propia.*

Para el subconjunto de prueba los resultados fueron los siguientes:

**Figura 8**

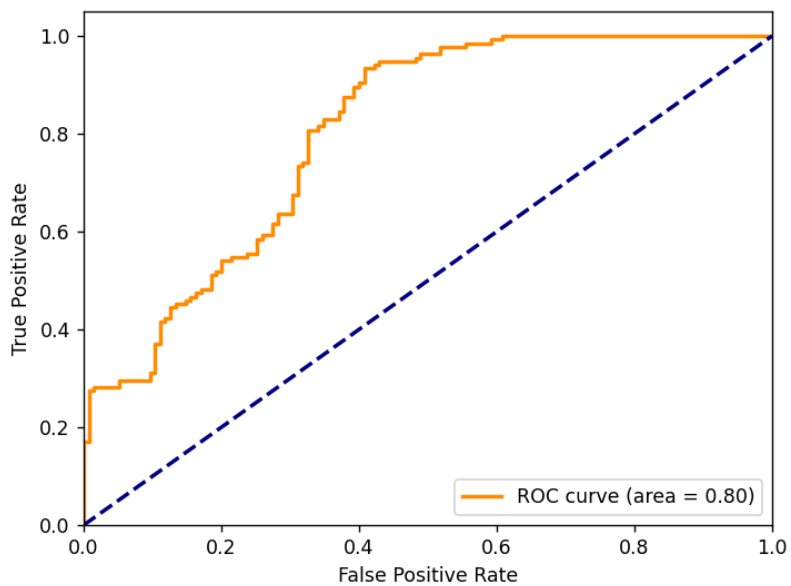
*Matriz de Confusión Datos de Prueba*



*Nota. Imagen de elaboración propia*

**Figura 9**

*Curva ROC Datos de Prueba*



*Nota. Imagen de elaboración propia.*

**Tabla 5***Resultados del Modelo Propio*

Métrica	Datos de entrenamiento	Datos de prueba
Accuracy	0.91	0.69
Precisión	0.91	0.69
Recall	0.91	0.69
F1-score	0.90	0.69
ROC AUC	0.98	0.80

*Nota. Se muestra cada una de las métricas de evaluación para el conjunto de datos de prueba.*

El valor accuracy nos indica la precisión general del modelo durante el aprendizaje, la cual para los datos de entrenamiento es bastante alta (91%) cuando se trata de los datos con los que fue entrenado. No obstante, este valor se reduce en gran medida cuando se pone a prueba con los datos de testing, arrojando un valor accuracy del 70%, esto nos indica que se está presentando sobreajuste (overfitting) en el modelo; lo cual se traduce en que el modelo se ajusta demasiado a los patrones del conjunto de datos de entrenamiento y no generaliza bien cuando se trata de datos nuevos.

En cuanto a la precisión los valores obtenidos son los mismos que en accuracy y las matrices de confusión nos indican que para los datos de entrenamiento el modelo tiene mayor cantidad de aciertos en la predicción para la clase “correcta” donde logra identificar correctamente 1438 posturas correctas y 62 incorrectas, con solo 26 falsos positivos, es decir posturas incorrectas clasificadas como correctas y 1292 imágenes de la clase “incorrecta”

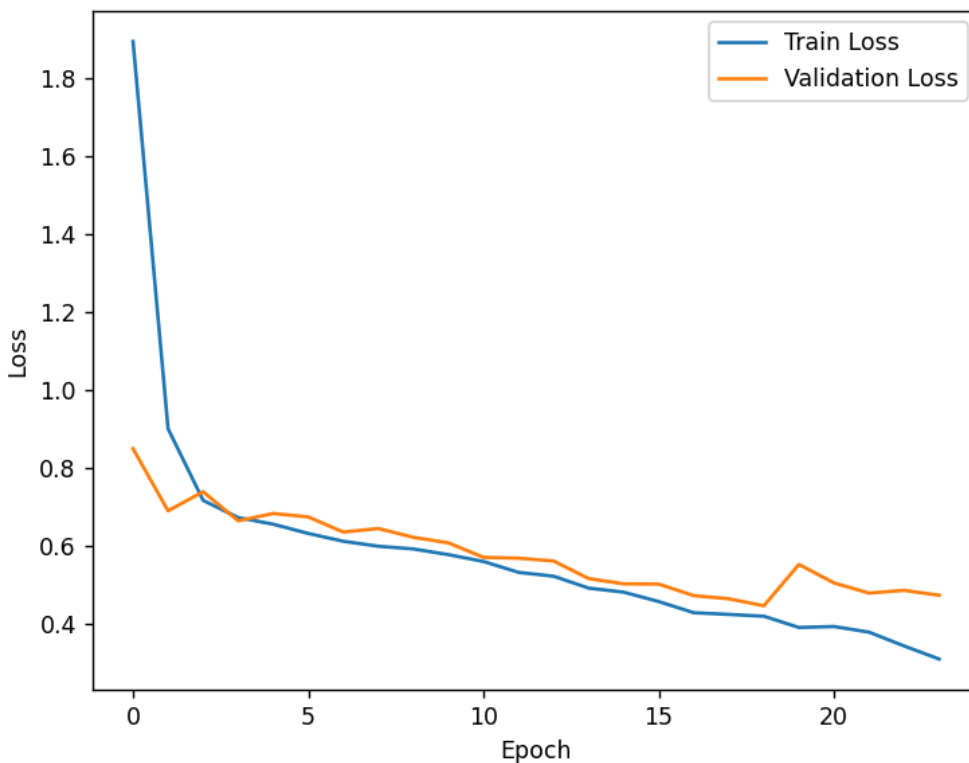
predichas adecuadamente con 208 falsos negativos. En el conjunto de prueba se observa una disminución en el rendimiento. El modelo acierta en 93 posturas correctas y 95 incorrectas con 42 falsos positivos y 40 falsos negativos.

Las medidas de sensibilidad (Recall), F1-score y el área bajo la curva ROC (AUC ROC) reafirman que existe un sobreajuste, puesto que también se observa una disminución en estas para los datos del conjunto de prueba con respecto a los de entrenamiento, es decir una pérdida del modelo en su capacidad de clasificar.

Los resultados generales fueron los que se muestran a continuación:

### Figura 10

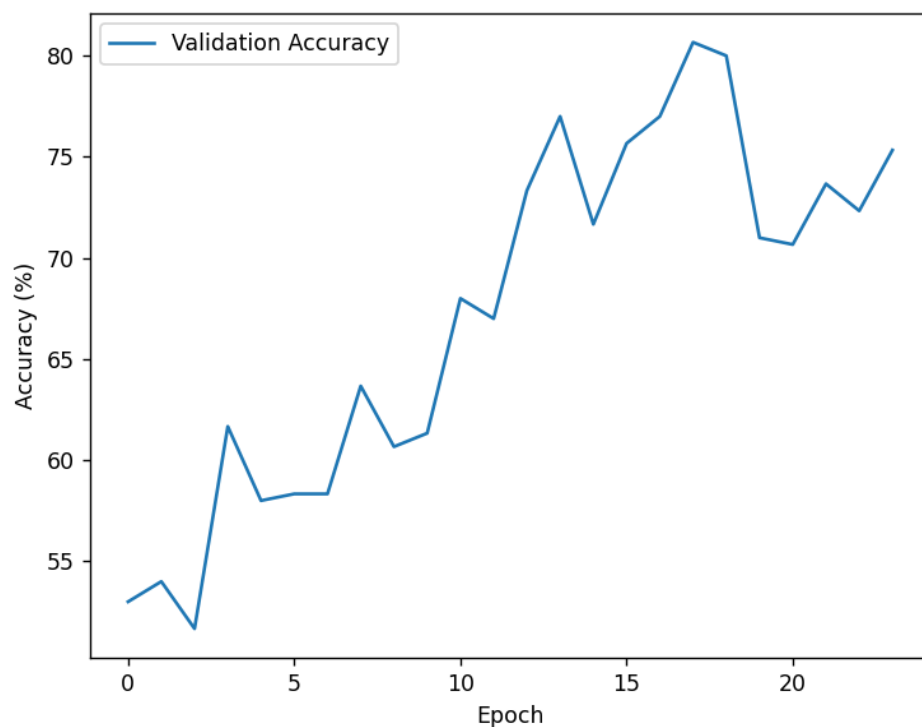
*Gráfica de Pérdida Durante el Entrenamiento y Validación*



*Nota. Imagen de elaboración propia.*

**Figura 11**

*Gráfica de Precisión en la Validación*



*Nota. Imagen de elaboración propia.*

En las gráficas de la Figura 10 podemos ver que hay una disminución en el sobreajuste, ya que ambas curvas (train loss y validation loss) disminuyen progresivamente. Por su parte en la gráfica de la validación (Figura 11 vemos que la precisión global (Accuracy) se mantiene alrededor del 70%. Lo anterior se refleja en la predicción del modelo, donde para las 10 imágenes ajenas al dataset predijo consistentemente 7 de ellas, como se verá a continuación:

**Figura 12***Predicciones del Modelo Hecho en Pytorch para Imágenes Externas*

*Nota. Imagen de elaboración propia.*

Como se puede evidenciar, el algoritmo presenta algunas imprecisiones respecto a las variaciones de los entornos de las imágenes, esto se encuentra asociado al sobreajuste (overfitting), ya que en las imágenes donde el algoritmo falla sus predicciones son las que tienen fondos oscuros o condiciones de iluminación poco favorables en comparación a las imágenes del dataset de entrenamiento donde predominan una buena iluminación.

Además, se identificaron errores en la predicción de las imágenes que contienen ruido, es decir un fondo u objetos alrededor que dificultan la identificación de los patrones presentes en las posturas correctas e incorrectas. También se observa que el modelo es más propenso a predecir errores cuando hay una flexión del tronco de la persona (imágenes 2, 3 y 5) y cuando el sujeto interactúa con un dispositivo diferente a un computador, como se ve en la imagen 3 donde la persona está usando un celular. Esto nos indica que la predicción del modelo puede ser

afectada por las variaciones que haya en el entorno de las fotos a predecir, lo cual sugiere que si hay más diversidad de imágenes del dataset de entrenamiento esto puede mejorar.

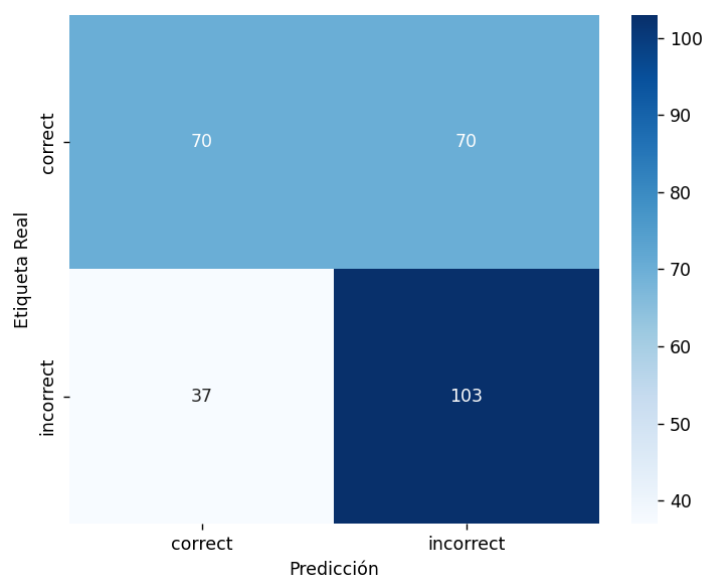
### Modelo de Predicción Hecho en Teachable Machine

El modelo realizado en la aplicación web Teachable Machine fue configurado con los mismos parámetros que el modelo hecho con la librería Pytorch (25 épocas, lote de 32 y tasa de aprendizaje de 0,01), la diferencia es que la lógica de Teachable Machine se basa en la librería Keras y no hace falta programar el código del algoritmo, sino que la aplicación lo hace por sí misma. Teachable Machine cuenta con la opción de exportar un código del modelo cuando ya ha sido entrenado por la aplicación, este código fue adaptado para que procesara las imágenes de prueba utilizadas en el modelo anterior y evaluara el rendimiento del modelo utilizando las mismas métricas que fueron accuracy, precisión, recall, F1 score y AUC-ROC.

Para el modelo de Teachable Machine se obtuvieron los siguientes resultados:

### Figura 13

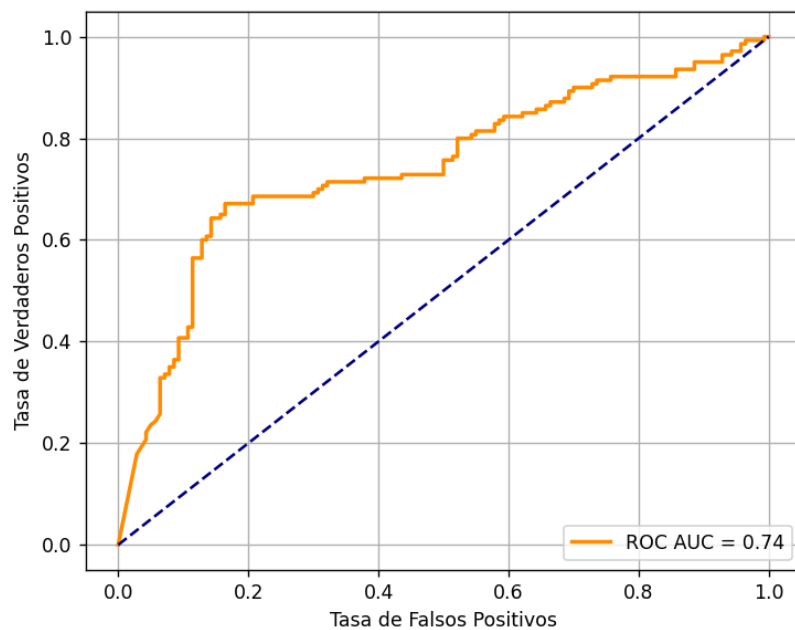
*Matriz de Confusión Modelo Teachable Machine*



*Nota. Imagen de elaboración propia.*

**Figura 14**

*Curva ROC datos de prueba modelo Teachable Machine*



*Nota. Imagen de elaboración propia.*

**Tabla 6**

*Resultados del Modelo de Teachable Machine*

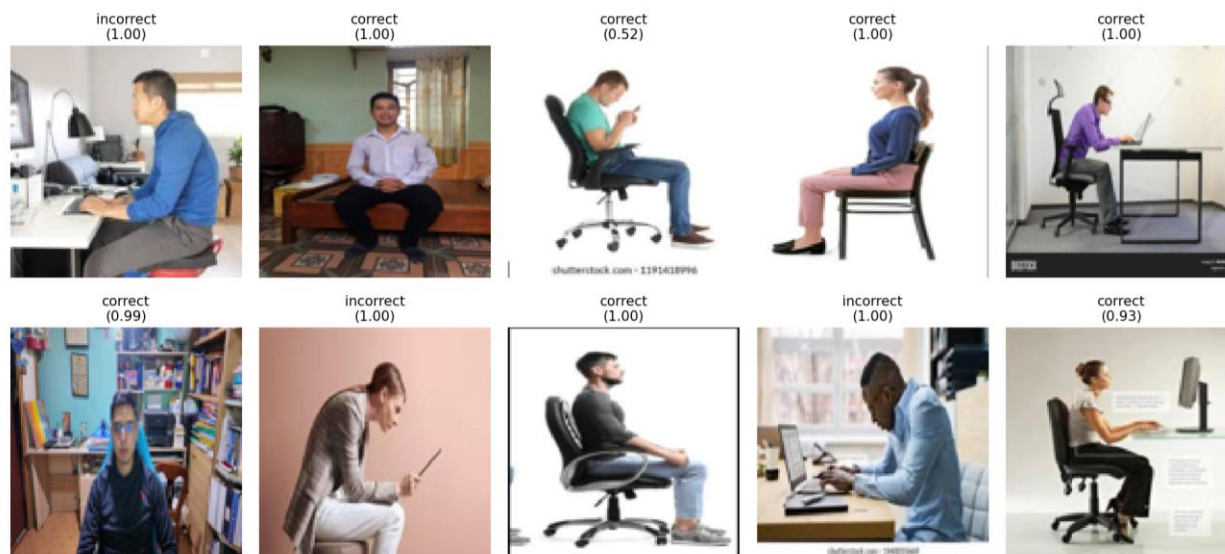
Métrica	Puntaje
Accuracy	0.62
Precisión	0.60
Recall	0.74
F1-score	0.66
ROC AUC	0.74

*Nota. Se muestra cada una de las métricas de evaluación para el modelo de Teachable Machine.*

En comparación con el modelo desarrollado en Pytorch, el de Teachable Machine tuvo un desempeño más bajo, el cual se vio reflejado en las métricas de evaluación y en la curva ROC. Este modelo presenta más dificultades a la hora de clasificar la clase “correcta” de las posturas, ya que clasificó 70 imágenes adecuadamente y 70 incorrectamente. En cuanto a la clase “incorrecta” acertó en 103 clasificaciones y falló en 37, como vemos en la figura 13. Vemos también en la gráfica de la curva ROC que el rendimiento de este modelo es menor que el del anterior, esta gráfica en términos generales muestra un buen comportamiento, pero en varios tramos tiende a acercarse más a la línea diagonal, lo que indica menor capacidad de discriminación a la hora de predecir.

### Figura 15

#### *Predicciones del Modelo de Teachable Machine para Imágenes Externas*



*Nota. Imagen de elaboración propia.*

En la figura 14 vemos que el modelo presenta un sobreajuste muy alto, ya que a pesar de realizar adecuadamente 9 de las 10 predicciones, la distribución de probabilidades obtenida en cada imagen de prueba muestra que la predicción es muy sesgada ya que en la mayoría de las imágenes tiene valor 1 (100%), el cual es muy alto, reflejando que el modelo tiene un exceso de seguridad en las predicciones.

El sobreajuste indica que tiende a adaptarse a un patrón específico de imágenes; lo cual dificulta que clasifique adecuadamente imágenes totalmente diferentes a las suministradas en el entrenamiento. Es decir que el modelo aprendió a memorizar características específicas de las imágenes de entrenamiento. Debido a esto, la clasificación se ve afectada cuando el modelo se enfrenta a fotos que difieren de las utilizadas en el entrenamiento.

Para esta prueba no fue posible realizar técnicas de sobreajuste como data augmentation o regularización mediante dropout, debido a que el modelo utilizado fue el previamente desarrollado por Teachable Machine, que está basado en Keras y cuenta con una arquitectura ya definida. Como el propósito de este estudio es observar cómo se comporta este modelo tal y como es, no se le realizaron modificaciones que afectaran su arquitectura. Únicamente se añadieron imágenes de prueba y porcentajes de confianza que muestran probabilidad de certeza según la predicción realizada.

## Conclusiones

El modelo propio basado en CNN y desarrollado en Pytorch obtuvo mejores resultados en las métricas de precisión en comparación al modelo de Teachable Machine. Con un accuracy de 91% frente a 69% del modelo de Teachable Machine.

La ventaja del modelo hecho por cuenta propia es que al realizarle personalizaciones para mejorar la predicción como data augmentation y dropout, se obtuvo una mejora significativa en los resultados de predicción del modelo, ya que esto fue lo que evitó que hubiera un sobreajuste alto.

El modelo de Teachable Machine tuvo un sobreajuste alto, debido a que no se le realizaron técnicas de mejora como las realizadas al modelo CNN y en la aplicación web únicamente le podemos ajustar los parámetros de tamaño del lote, tasa de aprendizaje y número de épocas

La evaluación realizada permitió evidenciar que ambos modelos fueron afectados por el sobreajuste, más que todo el realizado en Teachable Machine.

A pesar de que este sistema requiere mejoras brinda los fundamentos para una solución a un problema real en un contexto laboral, para empresas que estén interesadas en monitorear y corregir las posturas corporales de sus empleados.

### **Recomendaciones para Trabajos Futuros**

Se recomienda utilizar un dataset con gran variabilidad que tenga fotografías en diferentes entornos de oficina, de diferentes sujetos, con diferentes condiciones y que las imágenes hayan sido captadas desde diferentes perspectivas.

Realizar una detección de puntos clave en el cuerpo humano, seleccionando las articulaciones mencionadas en la norma ISO 112226:2000 de detección de posturas, utilizando librerías como MediaPipe u OpenPose.

Hacer una evaluación a largo plazo en distintos entornos labores reales para validar el desempeño del sistema en un contexto práctico.

### Referencias Bibliográficas

- Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <http://arxiv.org/abs/2004.10934>
- Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable machine: Approachable web-based tool for exploring machine learning classification. *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3334480.3382839>
- Chan, Y. W., Huang, T. H., Tsan, Y. T., Chan, W. C., Chang, C. H., & Tsai, Y. Te. (2020). The Risk Classification of Ergonomic Musculoskeletal Disorders in Work-related Repetitive Manual Handling Operations with Deep Learning Approaches. *Proceedings - 2020 International Conference on Pervasive Artificial Intelligence, ICPAI 2020*, 268–271. <https://doi.org/10.1109/ICPAI51961.2020.00057>
- Cifuentes, A., Mendoza, E., Lizcano, M., Santrich, A., & Moreno-Trillos, S. (2019). Desarrollo de una red neuronal convolucional para reconocer patrones en imágenes. *Revista I+D En TIC*, 10(2), 7–17. <http://revistas.unisimon.edu.co/index.php/identific>
- Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., & Zafeiriou, S. (2022). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 5962–5979. <https://doi.org/10.1109/TPAMI.2021.3087709>
- Fasiha, A., & Prasadu, P. (2022). Non-intrusive Stress detection based on temporal emotion analysis in videos applying machine learning. *Paper Knowledge . Toward a Media History of Documents*, 3(April), 49–58.
- Fernández, M. F. V. (2015). Posturas de trabajo y evaluación de riesgos. *Instituto Nacional de Seguridad e Higiene En El Trabajo (INSHT)*, 15, 54.

<https://www.insst.es/documents/94886/96076/Posturas+de+trabajo.pdf/3ff0eb49-d59e-4210-92f8-31ef1b017e66>

- Gumasing, M. J. J., Rendon, E. R. A., & German, J. D. (2023). Sustainable Ergonomic Workplace: Fostering Job Satisfaction and Productivity among Business Process Outsourcing (BPO) Workers. *Sustainability (Switzerland)*, *15*(18). <https://doi.org/10.3390/su151813516>
- Hasib, R., Khan, K. N., Yu, M., & Khan, M. S. (2021). Vision-based Human Posture Classification and Fall Detection using Convolutional Neural Network. *2021 International Conference on Artificial Intelligence, ICAI 2021*, 74–79. <https://doi.org/10.1109/ICAI52203.2021.9445263>
- Kang, J. H., Hsieh, E. H., Lee, C. Y., Sun, Y. M., Lee, T. Y., Hsu, J. B. K., & Chang, T. H. (2023). Assessing Non-Specific Neck Pain through Pose Estimation from Images Based on Ensemble Learning. *Life*, *13*(12), 1–16. <https://doi.org/10.3390/life13122292>
- Kulikajevs, A., Maskeliunas, R., & Damaševičius, R. (2021). Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Computer Science*, *7*, 1–20. <https://doi.org/10.7717/peerj-cs.442>
- Lubinus Badillo, F., Rueda Hernández, C. A., Marconi Narváez, B., & Arias Trillos, Y. E. (2021). Redes neuronales convolucionales: un modelo de Deep Learning en imágenes diagnósticas. Revisión de tema. *Revista Colombiana de Radiología*, *32*(3), 5591–5599. <https://doi.org/10.53903/01212095.161>
- MassirisFernández, M., Fernández, J. Á., Bajo, J. M., & Delrieux, C. A. (2020). Ergonomic risk assessment based on computer vision and machine learning. *Computers and Industrial Engineering*, *149*(September). <https://doi.org/10.1016/j.cie.2020.106816>
- McNicholas, P. D., & Tait, P. A. (2019). Supervised Learning. *Data Science with Julia*, 93–128. <https://doi.org/10.1201/9781351013673-5>

- Noor, N., & Park, I. K. (2023). A Lightweight Skeleton-Based 3D-CNN for Real-Time Fall Detection and Action Recognition. *Proceedings - 2023 IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2023*, 2171–2180. <https://doi.org/10.1109/ICCVW60793.2023.00232>
- Ogundokun, R. O., Maskeliūnas, R., & Damaševičius, R. (2022). Human Posture Detection Using Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms. *Applied Sciences (Switzerland)*, 12(19). <https://doi.org/10.3390/app121910156>
- Soffer, S., Ben-Cohen, A., Shimon, O., Amitai, M. M., Greenspan, H., & Klang, E. (2019). Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide. *Radiology*, 290(3), 590–606. <https://doi.org/10.1148/radiol.2018180547>
- Sohrabi, M. S., Khotanlou, H., Heidarimoghadam, R., Mohammadfam, I., Babamiri, M., & Soltanian, A. R. (2024). Modeling the Impact of Ergonomic Interventions and Occupational Factors on Work-Related Musculoskeletal Disorders in the Neck of Office Workers with Machine Learning Methods. *Journal of Research in Health Sciences*, 24(3), e00623–e00623. <https://doi.org/10.34172/jrhs.2024.158>
- Solà, R. F. (2020). *REAL-TIME MULTI-HUMAN POSE ESTIMATION AND ACTIVITY CLASSIFICATION SYSTEM* by Advisor : Albino Nogueiras Rodriguez Tutor : Oriol Hernán Galobart.
- Wu, M. (2021). *Detecting Posture Quality with Real-time Pose Estimation*.
- Xie, L., & Guo, X. (2019). Object detection and analysis of human body postures based on TensorFlow. *Proceedings - 2019 IEEE International Conference on Smart Internet of Things, SmartIoT 2019*, 397–401. <https://doi.org/10.1109/SmartIoT.2019.00070>