# Instituto Tecnológico
# y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

**Maestría en Ciencia de Datos**



# Segmentación de prendas de vestir por medio de la utilización de algoritmos de *Computer Vision* y *Deep Learning*

---

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de
**MAESTRO EN CIENCIA DE DATOS**


Presenta: **ALEX MEDINA ANGUIANO**
Director: **DR. JAIME EMMANUEL ALCALÁ TEMORES**



Tlaquepaque, Jalisco, enero de 2024.

# AGRADECIMIENTOS

Mi más sincero agradecimiento al ITESO por abrirme las puertas a un nuevo panorama de conocimientos y experiencias. Gracias por el apoyo y la confianza que ha depositado en mí al otorgarme la invaluable beca que me ha permitido continuar con mis estudios de postgrado, alcanzar mis metas académicas y contribuir positivamente a mi comunidad.

Gracias a mis profesores de este programa de maestría, por compartir sus conocimientos y experiencias haciendo que los temas complejos fueran mucho más fáciles de comprender, en especial al Dr. Fernando Becerra, a la Dra. Rocío Carrasco, a la Dra. Paola Montoya y al Dr. Riemann Ruiz por contagiarme de su pasión y entusiasmo en cada una de las clases y despertar cada vez más mi interés en áreas de la ciencia de datos que aún desconocía.

Gracias también a Ángel, Elisa, Ale, Luis, Jesús y Aldo con los que inicié este programa académico como compañeros y que poco a poco se convirtieron en mis grandes amigos. Fueron para mí un soporte esencial durante las clases, proyectos, investigaciones y sesiones de tareas y/o terapias que se extendían hasta altas horas de la noche.

Agradecimiento especial también al Dr. Emmanuel Alcalá, quien fue un invaluable guía, un detallado mentor y apoyo fundamental durante el proceso de desarrollo de este trabajo. Aprecio enormemente el tiempo que dedicó a discutir y revisar este proyecto, brindando comentarios constructivos. Pero sobre todo por su paciencia y valiosas sugerencias para con mis bloqueos mentales y metodológicos.

Gracias a mis amigos Karen, More y Elena con quienes he conservado una amistad solida e inquebrantable a través de los años y a pesar de la distancia, siempre están presentes y al alcance de un WhatsApp para compartir nuestros problemas y penas, pero también nuestros éxitos y buenas noticias. Además de ser proveedores de imágenes de gatitos en caso de emergencia.

Agradecimiento a mis tías Claudia, Gabriela y Margarita, quienes me brindaron un hogar lleno de calidez y cariño, y se convirtieron en la mejor familia adoptiva que pude haber imaginado. Gracias por motivarme a creer más en mis capacidades, a pensar en grande y siempre dar lo mejor de mí.

Y finalmente agradecimiento infinito a toda mi familia. A mi abuelita Refu; a mis padres Cuquín y Martín; y mis hermanos Ulises, Jesús y Susana. Esta maestría la inicie con ustedes en mente y la he de finalizar de la misma manera. Han sido el pilar fundamental durante todo el trayecto de este programa. Este logro lleva impreso el amor y el respaldo de cada uno de ustedes. En especial a mi Madre que siempre ha sido mi refugio, mi fuente de fuerza y mi mayor motivo para alcanzar mis metas.

# RESUMEN

Este trabajo de investigación tiene un enfoque específico sobre los temas de *Computer Vision* y *Deep Learning*, específicamente el presente trabajo intenta realizar la segmentación de imágenes de prendas de vestir. El objetivo es la exploración, comparación y selección del mejor modelo de clasificación de imágenes utilizando Redes Neuronales Convolucionales (CNN, por sus siglas en inglés). Los modelos de CNN que se analizan son tres: el primero con una configuración de capas ocultas personalizadas, el segundo con configuración de un modelo *ResNet* pre entrenado y el tercero haciendo uso de *FastAI*, la cual es una librería de programación de alto nivel. Después de llevar a cabo un preprocesamiento de las imágenes para acondicionarlas a los modelos se lleva a cabo el entrenamiento, calibración y proceso de pruebas de estos sobre el conjunto de imágenes seleccionado. Haciendo uso de la precisión como métrica de evaluación, se lleva a cabo la selección del mejor modelo el cual resulta ser el que hace uso de *FastAI* con un desempeño del 93.5% de predicciones correctas.

# TABLA DE CONTENIDO

# 1. INTRODUCCIÓN

Resumen: Se describe el contexto de este trabajo el cual se basa en el reciente incremento de los repertorios de imágenes de todo tipo, los cuales pueden ser aprovechados por algoritmos de *Computer Vision* y *Deep Learning* en diversas áreas de la industria pública y privada, en este caso la investigación se lleva a cabo enfocada en repositorios de prendas de vestir.

## 1.1. Contexto

El esplendor de la era digital ha traído consigo un crecimiento exponencial de los repositorios del contenido multimedia digital. En 2022, la cantidad de usuarios de teléfonos inteligentes en el mundo actual es de 6648 millones [1], lo que se traduce en que el 83 % de la población mundial posee un teléfono inteligente, como consecuencia de esta distribución tan amplia y con la calidad cada vez mayor de las cámaras de los teléfonos inteligentes, la cantidad de fotografías tomadas en todo el mundo cada día se está disparando. En 2022, se toman 54.4 mil fotos por segundo y 1.7 billones por año [2].

Este desarrollo ha propiciado la creación de repositorios de imágenes bastante voluminosos de todos los campos semánticos imaginables. Estos conjuntos de datos son cada vez más y mejor aprovechados por la industria pública y privada dentro del área de visión por computadora y aprendizaje automático para entrenar algoritmos de inteligencia artificial, que luego pueden ser aprovechados para lograr la identificación de patrones y clasificación sobre otros conjuntos de imágenes que aumentan de forma creciente.

Actualmente, existen diferentes enfoques para realizar la segmentación. Además de la opción básica que implica que una persona realice manualmente la clasificación de lo que observa en las imágenes, ya se ha desarrollado alternativas más avanzadas. Estas consisten en utilizar herramientas de visión por computadora ya existentes, que pueden realizar la segmentación de manera automática. Un ejemplo de estas soluciones es Google Cloud AutoML Vision, una herramienta en línea de pago que puede entrenar modelos de aprendizaje automático para clasificar imágenes con o sin etiquetado previo [3].

En este trabajo, además de algunos modelos personalizados y pre entrenados, también implementará una herramienta de visión por computadora como la mencionadas anteriormente, llamada *FastAI*. Ésta consiste en una librería construida en Python que permite la creación de modelos de redes neurales de alto nivel, debido a que facilita la construcción de modelos de clasificación o segmentación de imágenes con muy pocas líneas de código. Esta herramienta será utilizada sobre un conjunto de datos de prendas de ropa para intentar clasificarlas en una de 20 clases.

## 1.2. Justificación

La habilidad de poder segmentar ropa representa una ventaja significativa para algoritmos de recomendación de compras que muchas compañías con comercio en línea pueden utilizar. Actualmente, muchas de estas compañías ya cuentan con sugerencia de compras y algoritmos de recomendación de modelos de atuendos (*outfits*) automatizados, sin embargo, estos algoritmos existentes están basados más en variables de comportamiento de los usuarios (como el historial de búsqueda o en función de las prendas que otros usuarios han comprado) que en los atributos de la ropa misma.

## 1.3. Problema

Para el caso particular de esta investigación, el área de aplicación con la que se trabajará es el de la moda y las prendas de vestir. Se pretende construir una solución sobre un conjunto de imágenes de prendas de vestir variadas que serán utilizadas para el entrenamiento de modelos de *Deep Learning* junto con sus respectivas etiquetas de categorías.

Una vez que el modelo de entrenamiento sea lo suficientemente capaz de realizar la identificación de las prendas en las imágenes, este mismo modelo será probado sobre un conjunto nuevo, este flujo de trabajo se ejemplifica en la Figura 1.

Para el entrenamiento de los modelos se utiliza un conjunto de datos que contienen etiquetas que identifican el tipo de prenda que está representada en la imagen analizar, por lo que se puede definir la investigación como un problema supervisado. Además, es posible catalogarlo como un problema *online*, debido a que se pretende tener los datos completos a utilizar ya almacenados e ir mejorando la solución y los modelos con nuevos datos.

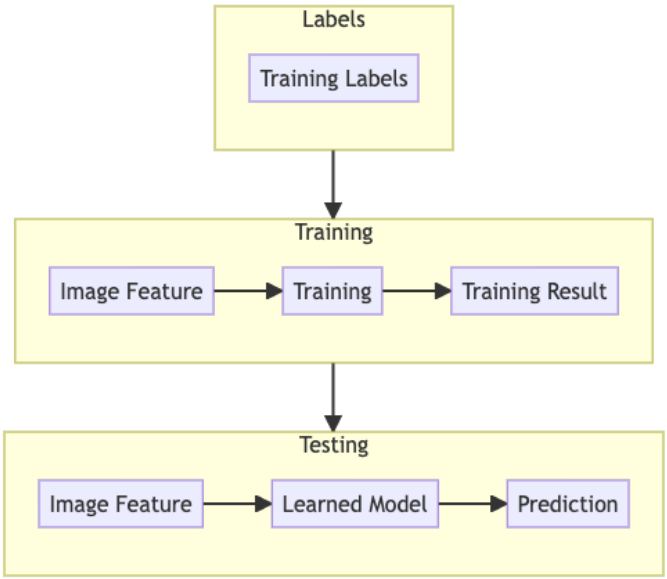


**Figura 1: Flujo de trabajo de la solución**

La manera en la que se estará midiendo el desempeño de los modelos es por medio de métricas básicas. Se estará midiendo principalmente a través del porcentaje de predicciones correctas que el modelo logre realizar sobre un conjunto de datos de prueba después de realizar el entrenamiento. Además, también se pretende tomar en cuenta el tiempo de entrenamiento de los modelos, ya que cada algoritmo puede tener un desempeño similar en cuanto a resultados, pero diferente en cuanto a poder computacional requerido. Por lo tanto, la selección del mejor modelo estará basado en el equilibrio de estos dos parámetros.

Si se toma como base el trabajo que una persona podría realizar al intentar llevar a cabo esta segmentación de prendas de vestir, entonces resulta bastante simple medir el desempeño que los modelos logren, tanto por el porcentaje de predicciones correctas como por el tiempo en el que logre completar la clasificación.

Un humano con desarrollo cognitivo normal es capaz de realizar la segmentación e identificación de prendas de vestir si observa una imagen de ella con relativa facilidad. Se estima que el ojo de una persona experta en el tema llega a lograr un error de aproximadamente el 5%, esto al realizar el trabajo manual de segmentación e identificación de los objetos que se muestran sobre una imagen sobre un conjunto de datos de 1500 imágenes y 1000 clases, estos resultados son obtenidos del *Large Scale Visual Recognition Challenge* (ILSVRC) en el cual se ponen a prueba la experticia humana contra los modelos de visión por computadora [4] [5].

Aunque los modelos presentados aquí no lograran compararse con la tasa de error de humanos expertos, existen otras características importantes a considerar. Una característica interesante de este tipo de soluciones es su escalabilidad y adaptabilidad para poder ser utilizado en problemas similares. La solución que se desarrollará debería ser posible de utilizarse con problemas similares. Tanto la experiencia obtenida y las herramientas que se obtengan en este problema se pueden trasponer relativamente fácil sobre conjuntos de datos similares. Esta es una ventaja ampliamente conocida en el área de visión por computadora, pues existen ya varios modelos e investigaciones que realizan un procedimiento similar pero aplicado sobre conjuros de datos diferentes o con contexto diferentes.

## 1.4. Objetivos

### 1.4.1. Objetivo General:

Lograr la segmentación de prendas de vestir variadas por medio de la utilización de algoritmos de *Computer Vision y Deep Learning*. Esto implica la creación y optimización de modelos de clasificación que serán aplicados sobre un conjunto de imágenes que contienen prendas de vestir variadas.

### 1.4.2. Objetivos Específicos:

1. Preparación y adaptación del conjunto de datos para ser utilizados por los modelos.
2. Limpieza del conjunto de datos para utilizar solo las muestras que sean realmente significativas.
3. Aplicación de métodos de *data augmentation* para contrarrestar el tamaño del conjunto de datos o el desbalanceo de las clases.

4. Diseño e implementación de las redes neurales y redes neurales convolucionales que serán utilizadas como modelos de clasificación de imágenes.
5. Evaluación y selección del mejor modelo de clasificación de imágenes basado en la precisión de los resultados al predecir.
6. Identificación de las características presentes en la imagen que más contribuyen en el modelo al momento de realizar la clasificación por medio de un análisis *Grad-CAM*.

# 2. METODOLOGÍA

Resumen: Se lleva a cabo una descripción detallada del conjunto de datos de imágenes que será utilizado, se explica también los procesos de limpieza de datos y las transformaciones aplicadas sobre el conjunto de datos. Además, se exponen los detalles de cada uno de los modelos de clasificación con redes neurales que serán comparados entre sí, desde su definición y características como los parámetros e hiper parámetros que serán utilizados.

## 2.1. Descripción de los datos

El conjunto de datos que será utilizado para entrenar los modelos que serán evaluados fue obtenido desde *Kaggle*, la cual es una plataforma para practicantes y profesionales del área de ciencia de datos donde los usuarios pueden publicar conjuntos de datos para realizar exploración, construcción de modelos, así como participar en competencias entre los mismos usuarios.

En este caso, el repositorio que será utilizado en esta investigación se puede descargar desde el siguiente enlace [https://www.kaggle.com/datasets/agrigorev/clothing-dataset-full]. Este conjunto de datos pertenece a Alexey Grigorev y fue creado como un esfuerzo de contribución comunitaria, es decir que muchas personas se involucraron aportando sus propias imágenes. Se encuentra publicado bajo una licencia *Creative Commons Zero* (CC0), esto significa que cualquier individuo puede utilizar estos datos para cualquier fin, incluso comercial [6].

El conjunto de datos consta de una compilación de imágenes de prendas de vestir en formato JPG. Dicho conjunto de datos tiene un tamaño de aproximadamente 7GB, el cual está constituido por 10000 imágenes catalogadas en 20 tipos diferentes de prendas de vestir. El conjunto se encuentra dividido en dos subconjuntos de 5000 imágenes cada uno, donde el primer conjunto son las imágenes en resolución original y el segundo en una versión comprimida.

Los datos ya están en un formato ampliamente accesible y listas para ser utilizados:

- Las imágenes se encuentran en formato JPG.
- Las etiquetas de las imágenes se encuentran en un archivo CSV.

## 2.2. Análisis exploratorio

A continuación, se realizará un análisis exploratorio del conjunto de datos de imágenes seleccionado, se lleva a cabo la descripción de los atributos del conjunto, las clases catalogan a las imágenes, el tipo de prendas que se analizan y las resoluciones de las imágenes que será utilizadas.

Todo este procedimiento se lleva a cabo dentro de la plataforma de desarrollo del lenguaje de programación Python, haciendo uso de librerías existentes enfocadas en procesos de análisis y visualización de datos como Numpy, Pandas, Matplotlib, Seaborn, Pillow, etcétera.

### *2.2.1. Atributos*

El conjunto de datos consta de dos subconjuntos de imágenes, uno correspondiente a todas las imágenes en resolución original y otro con las mismas imágenes, pero comprimidas, cada subconjunto cuenta con 5762 imágenes con extensión JPG. Además, se incluye un archivo en formato CSV que contienen la información descriptiva de las imágenes de los subconjuntos mencionados anteriormente, estos metadatos tienen los siguientes atributos:

- image: este atributo de tipo texto contiene el nombre de la imagen en los subconjuntos de imágenes a la que hace referencia.
- sender_id: este atributo de tipo entero contiene el un numero identificador del individuo que proporcionó esta imagen al conjunto.
- label: este atributo de tipo texto/categoría contiene la clase o categoría a la que pertenece la prenda de vestir en la imagen.
- kids: este atributo de tipo booleano indica si la prenda de vestir pertenece a una prenda para niño o no.

### *2.2.2. Clases*

El conjunto de datos contiene 20 diferentes clases que describen el contenido de la imagen, estas clases son las siguientes:

1. Blazer (109 imágenes)
2. Blouse (23 imágenes)
3. Body (69 imágenes)
4. Dress (357 imágenes)
5. Hat (171 imágenes)
6. Hoodie (100 imágenes)
7. Longsleeve (699 imágenes)
8. Not sure (228 imágenes)
9. Other (67 imágenes)
10. Outwear (312 imágenes)
11. Pants (692 imágenes)
12. Polo (120 imágenes)
13. Shirt (378 imágenes)
14. Shoes (431 imágenes)
15. Shorts (308 imágenes)
16. Skip (12 imágenes)

17. Skirt (155 imágenes)
18. T-Shirt (1011 imágenes)
19. Top (43 imágenes)
20. Undershirt (118 imágenes)

En este archivo de metadatos no existen valores perdidos, sin embargo, dentro de las etiquetas de las imágenes se pueden observar dos categorías que podrían tomarse como valores nulos en el contexto de que no se encuentra clasificado de manera correcta. Estas etiquetas son `Not sure`, `Other` y `Skip`, a las cuales 307 de las imágenes pertenecen. En conjunto estos valores representan el 5.7% de los valores del conjunto tal como se muestra en la Figura 2.

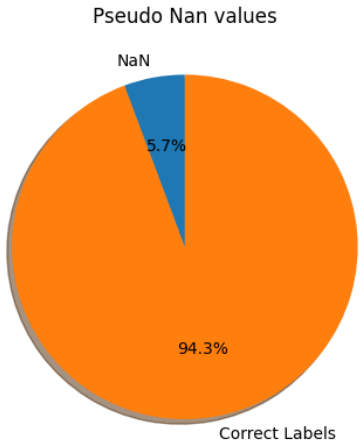


**Figura 2: Imágenes sin categoría definida**

Debido a que representan una categoría ambigua y que no coinciden con el contenido de las imágenes, tal como se ejemplifica en la Figura 3, se ha decidido no tomarlas en consideración ya que estas simplemente podrían confundir a los modelos durante el entrenamiento.


**Figura 3 Ejemplos de imágenes con clases no representativas**

Con un análisis simple, por medio de una gráfica de barras (véase la Figura 4), es posible identificar la distribución de las clases del conjunto de datos y observar que las clases se encuentran con un desbalance notorio.

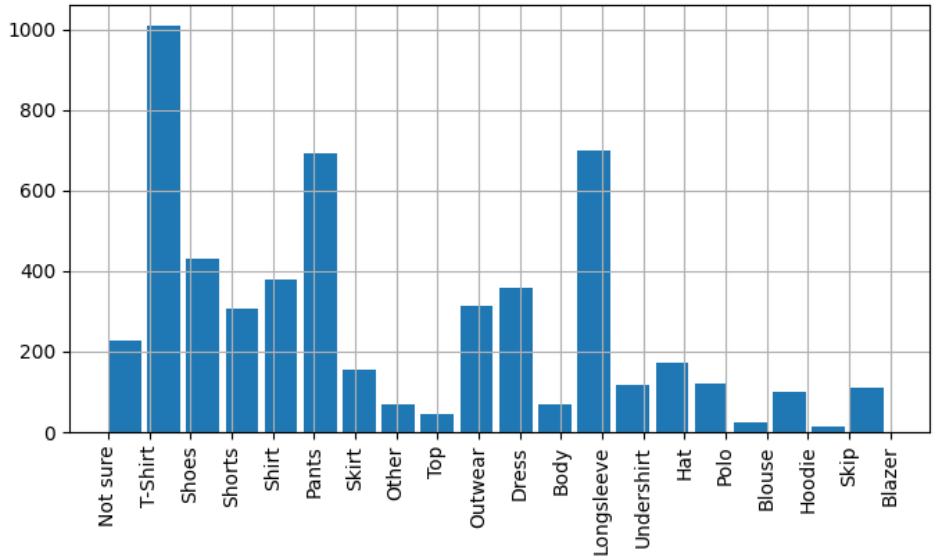


**Figura 4: Distribución de las clases en el conjunto de datos**

Se puede observar que aproximadamente un 60% de las imágenes se encuentran contenidas en solo 5 de las clases más extensas, lo cual podría representar una desventaja al momento de realizar el entrenamiento y significaría que se entrenaría con datos desbalanceados. El desglose de los porcentajes y los porcentajes acumulados pueden observarse en la Tabla 1.

Tabla 1: Porcentaje acumulado de las primeras 5 clases más extensas

| *Clases* | *Numero de muestras* | *Porcentaje* | *Porcentaje acumulado* |
|---|---|---|---|
| *T-Shirt* | 1011 | 18.711827 | 18.711827 |
| *Longsleeve* | 699 | 12.937257 | 31.649084 |
| *Pants* | 692 | 12.807699 | 44.456783 |
| *Shoes* | 431 | 7.977050 | 52.433833 |
| *Shirt* | 378 | 6.996113 | 59.429946 |

### *2.2.3.* Tipo

El conjunto de datos además se encuentra seccionado entre prendas para niños y prendas para adultos. De acuerdo con lo anterior, aproximadamente 9% (véase la Figura 5) de todas las imágenes representan imágenes de prendas de vestir para niños.

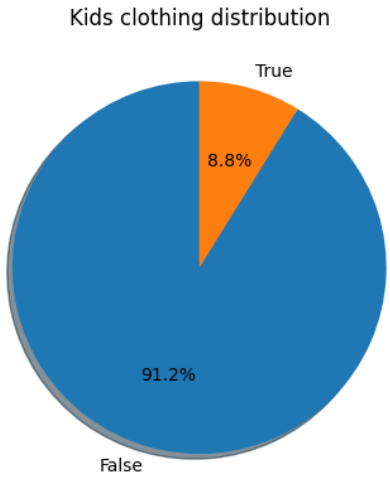


Figura 5: Distribución entre prendas para niños y prendas de adulto

### *2.2.4. Tamaño*

La resolución de las imágenes contenidas en el conjunto de datos está también se encuentra desbalanceado, esto debido a que los tamaños varían desde una altura de 400px y una anchura de 400px mínimo, hasta una altura de hasta 936px y una anchura de hasta 888px tal como se puede observar en la Figura 6. Habrá entonces que redimensionar todas estas imágenes para obtener una resolución uniforme para garantizar un mejor desempeño de los modelos.

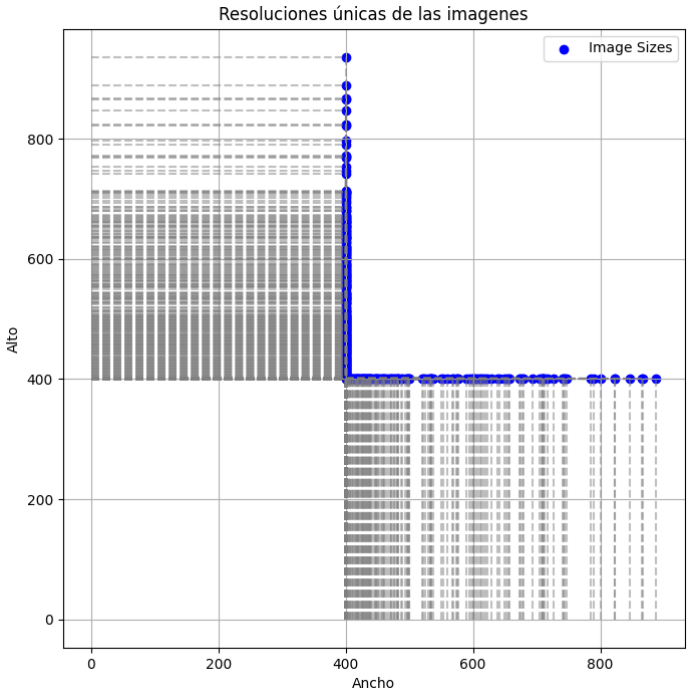


**Figura 6: Resoluciones de Imágenes**

## 2.3. *Descripción de los* modelos

Son tres los modelos que se utilizan en esta investigación. Los tres se tratan de modelos de clasificación con redes neurales convolucionales cuya característica diferencial son las capas ocultas y la modularidad.

Todo este procedimiento se lleva a cabo dentro de la plataforma de desarrollo del lenguaje de programación Python, haciendo uso de librerías, entre ellas las más relevantes son:

- scikit-learn, enfocadas en procesos de muestreo, división de conjuntos de datos y métricas de evaluación.
- Pytorch para todo lo referente a los modelos de redes neurales convolucionales (abreviadas con las siglas CNN por su nombre en inglés *Convolutional Neural Networks*), su entrenamiento, validación y posterior implementación de predicciones.
- *FastAI* como librería alternativa de alto nivel, enfocada también en el entrenamiento, validación y posterior implementación de predicciones.

El primer modelo es una red neural convolucional cuyas capas han sido personalizadas con el objetivo de generar un clasificador de imágenes.

### *2.3.1.* RED NEURAL CONVOLUCIONAL PERSONALIZADA

Esta red neural está estructurada de la siguiente manera:

1. Capa de entrada: Esta es la capa que recibe las imágenes es su estado original. Esta capa espera que las imágenes de entrada cuenten con una estructura 3 canales (RGB).
2. Capas convolucionales:
    a. self.conv1: 64 filtros con un tamaño de *kernel* de 3x3, seguido de activación ReLU y agrupación máxima (2x2).
    b. self.conv2: 128 filtros con un tamaño de *kernel* de 3x3, seguido de activación ReLU y agrupación máxima (2x2).
    c. self.conv3: 256 filtros con un tamaño de *kernel* de 3x3, seguido de activación ReLU y agrupación máxima (2x2).
3. Capas completamente conectadas:
    a. self.fc1: una capa completamente conectada con 256 * 28 * 28 características de entrada (resultantes de las dimensiones espaciales después de la agrupación máxima) y 512 características de salida, seguidas de la activación de ReLU.
    b. self.fc2: la capa final completamente conectada con 512 valores de entrada y la cantidad de clases como parámetro de salida. Esta capa incluye una función de activación lineal, que se utiliza para lograr la tarea de clasificación en el número de clases esperadas
4. Salida: El resultado final es el resultado de la última capa completamente conectada, que representa las puntuaciones/probabilidades de cada clase para cada imagen de entrada.

### *2.3.2.* RED NEURAL CONVOLUCIONAL PRE ENTRENADA

Esta red neural hace uso de la configuración de un modelo ResNet pre entrenado, el cual ha sido entrenado usando grandes conjuntos de imágenes previamente.

Este enfoque de redes neurales es muy común para realizar aprendizaje por transferencia, donde un modelo previamente entrenado en un conjunto de datos grande se adapta para dar resultados sobre un conjunto de datos más pequeño, aprovechando el conocimiento general aprendido por el modelo sobre otros datos, y lo refina para una tarea más específica.

La estructura que sigue esta red neural es la siguiente:

1. Modelo ResNet previamente entrenado: Esta red neural se basa en un modelo ResNet previamente entrenado, específicamente el modelo ResNet18, el cual es el modelo más básico de los modelos ResNet y que funciona bien para conjuntos de datos pequeños.
2. Modificación de la capa totalmente conectada: La capa completamente conectada al final del modelo ResNet se ajusta para adaptarse al número específico de clases que son utilizadas por el conjunto de datos.

### *2.3.3.* RED NEURONAL CONVOLUCIONAL PREENTRADA CON FASTAI

Esta red neuronal hace uso del mismo modelo del punto 2.3.2., con la diferencia que el entrenamiento y prueba del modelo se realiza por medio de la librería de FastAI, la cual implementa sus propias funciones para estas tareas. En el caso de los modelos en los puntos 2.3.1. y 2.3.2. ambos utilizan un método de entrenamiento y prueba personalizado.

El uso de esta librería significaría una simplificación de los procesos de creación, entrenamiento y prueba de redes neurales, pues hace uso de una metodología de alto nivel de abstracción donde no se es necesario contar con conocimientos tan profundos de programación, matemáticas o estructuras de datos.

## 2.4. Descripción de las métricas

Para la evaluación de estos modelos se hace uso de la métrica de precisión. Esta es una de las métricas más comunes y sencillas para evaluar un modelo de clasificación y se define como la fracción de predicciones correctas que el modelo obtuvo [7], esta definición también puede ser expresada por medio de la ecuación (2-1):

$$\text{Precision} = \frac{\text{Numero de predicciones correctas}}{\text{Numero total de predicciones}} \qquad (2\text{-}1)$$

## 2.5. Descripción de los experimentos / simulaciones

Haciendo uso de modelo con la red neural convolucional personalizada (2.3.1) se realiza un proceso de *fine-tunning* o ajustes del modelo que servirá como punto de referencia para calibrar los parámetros y opciones de las redes neurales restantes, y probar configuraciones específicas en los experimentos. Debido a esto, previo a la experimentación propiamente dicha, en esta sección se muestran algunas pruebas preliminares que sirvieron de calibración

para elegir los parámetros con los cuales experimentar. En la sección de Resultados se muestran ahora sí los resultados de los experimentos.

Antes de comenzar con la experimentación sobre el conjunto de datos se llevan a cabo un conjunto de transformaciones básicas sobre las imágenes.

- Normalización de los canales de las imágenes: se realiza una normalización de 0.5 sobre la media y 0.5 sobre la desviación estándar, esto significa que se ajusta la escala de los valores de los canales de colores de tal manera que los nuevos valores de la media y la desviación estándar sean ahora de 0.5. La normalización tiene el beneficio de proporcionar una convergencia más rápida al momento de entrenar, además de asegurar que la red no dará más importancia a un canal de color especifico en lugar de a todos.
- Redimensionamiento de las imágenes: Se lleva a cabo el redimensionamiento de las imágenes que, como ya se mencionó anteriormente, no tienen una resolución uniforme entre ellas. El tamaño final de las imágenes será de 224x224, esto debido a que este es un tamaño utilizado generalmente por modelos pre entrenados de redes neurales convolucionales al momento de su propio entrenamiento.

Además de las transformaciones anteriores se llevan a cabo algunas transformaciones extras que tienen el objetivo de aumentar el número de muestras de las clases con menos imágenes. Estas transformaciones son las siguientes:

- *RandomVerticalFlip*: invierte la imagen verticalmente
- *RandomRotation*: rota la imagen $\pm 30°$ respecto a su estado original
- *RandomGrayscale*: transforma la imagen aplicando un filtro en escala de grises
- *RandomPerspective*: transforma la imagen aplicando una distorsión de la perspectiva

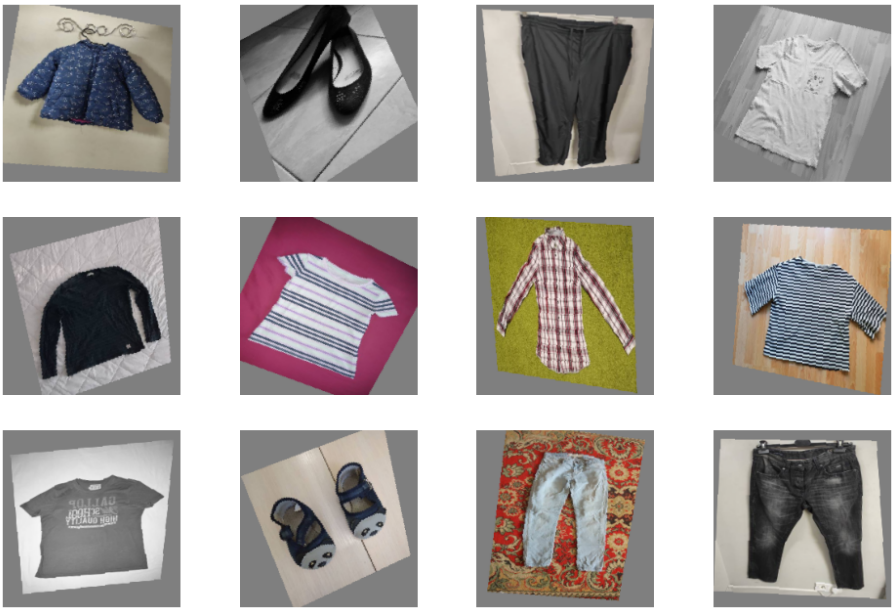

**Figura 7: Imágenes transformadas para el balanceo de las clases**

Una representación de las transformaciones aplicadas al conjunto de imágenes puede ser observada en la Figura 7.

### *2.5.1. Primera prueba con CNN Personalizada: Identificación de una hiperclase*

Se realiza el entrenamiento del modelo con los siguientes parámetros mostrados en la Tabla 2.

**Tabla 2: Parámetros de la primera con CNN Personalizada**

| *Parámetro* | *Valor* | *Detalle* |
|---|---|---|
| *optimizer* | Adam() | |
| *loss_function* | CrossEntropyLoss() | learning_rate = 0.003 |
| *clases* | 16 | 0: 'Blazer',<br>1: 'Blouse',<br>2: 'Dress',<br>3: 'Hat',<br>4: 'Hoodie',<br>5: 'Longsleeve',<br>6: 'Outwear',<br>7: 'Pants',<br>8: 'Polo',<br>9: 'Shirt',<br>10: 'Shoes',<br>11: 'Shorts',<br>12: 'Skirt', |

| | | |
|---|---|---|
| | | 13: 'T-Shirt',<br>14: 'Top',<br>15: 'Undershirt' |
| *epocs* | 8 | |
| *lambda1* | 0.001 | Parámetro de regularización L1 |
| *lambda2* | 0.001 | Parámetro de regularización L2 |

### *2.5.2. Segunda prueba con CNN Personalizada: exclusión de clases*

Se realiza el entrenamiento del modelo con los mismos parámetros de la prueba anterior (2.5.1), sin embargo, se ha removido las clases cuyo número de observaciones es menor al 30% de la clase con mayores observaciones. La Figura 8 muestra las clases que cumplen con esta condición.

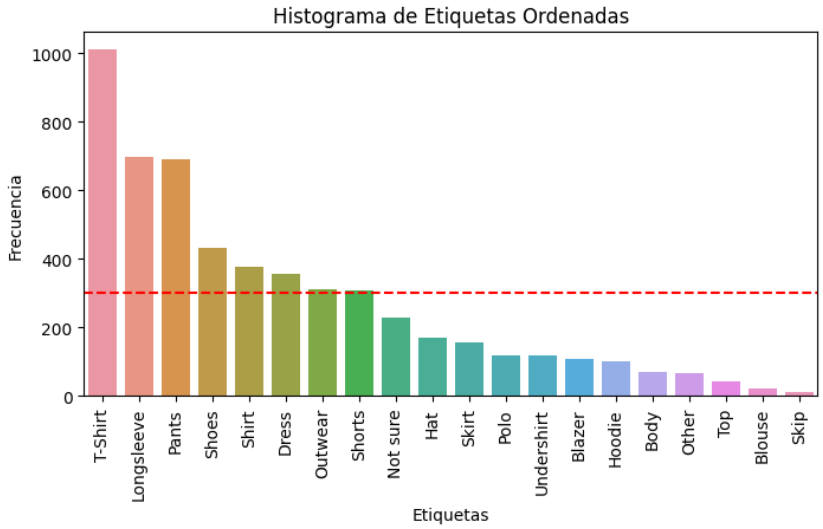


**Figura 8: Clases con menos del 30% de muestras que la clase mayor**

Se realiza entonces el entrenamiento del modelo con los parámetros mostrados en la Tabla 3.

**Tabla 3: Parámetros de la segundo prueba con CNN Personalizada**

| *Parámetro* | *Valor* | *Detalle* |
|---|---|---|
| *optimizer* | Adam | |
| *loss_function* | CrossEntropyLoss | Learning rate = 0.003 |
| *clases* | 8 | 0: 'Dress',<br>1: 'Longsleeve',<br>2: 'Outwear',<br>3: 'Pants',<br>4: 'Shirt',<br>5: 'Shoes',<br>6: 'Shorts', |

| | | |
|---|---|---|
| | 7: 'T-Shirt' | |
| *epocs* | 8 | |
| *lambda1* | 0.001 | Parámetro de regularización L1 |
| *lambda2* | 0.001 | Parámetro de regularización L2 |

### *2.5.3. Experimento: CNN Personalizada*

Dado que las pruebas preliminares arrojaron resultados poco favorables (ver la sección de Resultados), y de acuerdo con el análisis preliminar de los resultados sobre ambos (2.5.1. y 2.5.2.) se decide entonces modificar la cantidad de clases a utilizar para este y todos los modelos (2.3.2. y 2.3.3.) de los experimentos posteriores. Para este experimento se realizará entonces el entrenamiento del modelo con los mismos parámetros del experimento anterior (2.5.2.), sin embargo, se ha removido las clases cuyo número de observaciones es menor al 30% y también las clases que pudieran representar superclases o que su etiqueta pudiera funcionar mejor en una solución de categorización multiclase, en este caso las clases extras removidas son “longsleeve” y “outwear”. Los parámetros para utilizar son los mostrados en la Tabla 4.

**Tabla 4: Parámetros del experimento - CNN Personalizada**

| *Parámetro* | *Valor* | *Detalle* |
|---|---|---|
| *optimizer* | Adam | |
| *loss_function* | CrossEntropyLoss | Learning rate = 0.003 |
| *clases* | 6 | 0: 'Dress',<br>1: 'Pants',<br>2: 'Shirt',<br>3: 'Shoes',<br>4: 'Shorts',<br>5: 'T-Shirt' |
| *epocs* | 8 | |
| *lambda1* | 0.001 | Parámetro de regularización L1 |
| *lambda2* | 0.001 | Parámetro de regularización L2 |

### *2.5.4. CNN Pre entrenada*

Ahora que se tiene una configuración básica, se lleva a cabo el experimento con un modelo pre entrenado. Los parámetros mostrados en la Tabla 5 corresponden tal como se mencionó con anterioridad a los parámetros mostrados en la Tabla 4.

**Tabla 5: Parámetros del experimento - CNN Pre entrenada**

| *Parámetro* | *Valor* | *Detalle* |
|---|---|---|
| *optimizer* | Adam | |
| *loss_function* | CrossEntropyLoss | Learning rate = 0.003 |

| | | |
|---|---|---|
| *clases* | 6 | 0: 'Dress',<br>1: 'Pants',<br>2: 'Shirt',<br>3: 'Shoes',<br>4: 'Shorts',<br>5: 'T-Shirt' |
| *epocs* | 8 | |
| *lambda1* | 0.001 | Parámetro de regularización L1 |
| *lambda2* | 0.001 | Parámetro de regularización L2 |
| *pretrained_model* | ResNet18 | |

### *2.5.5. Experimento: CNN FastAI*

El modelo que utiliza la librería FastAI, utiliza su propia función de entrenamiento, sin embargo, se intenta utilizar los mismos parámetros utilizados en los experimentos anteriores. Los parámetros de este modelo son los mostrados en la Tabla 6.

De igual manera la librería FastAI puede hacer uso de modelos pre entrenados. En este caso en aras de mejorar los resultados se decide hacer uso del mismo modelo utilizado anteriormente (2.3.2) que hace uso del modelo ResNet18.

**Tabla 6; Parámetros del experimento - CNN FastAI**

| *Parámetro* | *Valor* | *Detalle* |
|---|---|---|
| *optimizer* | Adam | |
| *loss_function* | NLLLoss | Learning rate = 0.003 |
| *clases* | 6 | 0: 'Dress',<br>1: 'Pants',<br>2: 'Shirt',<br>3: 'Shoes',<br>4: 'Shorts',<br>5: 'T-Shirt' |
| *epocs* | 8 | |
| *pretrained_model* | ResNet18 | |

# 3. RESULTADOS Y DISCUSIÓN

Resumen: Se llevan a cabo los experimentos descritos en la sección de Metodología y se comparan los resultados obtenidos por cada modelo. Se evalúan las métricas y los resultados del desempeño que lograron durante el proceso de implementación de predicciones sobre un conjunto de datos prueba. Además, se lleva a cabo un análisis *Grad-CAM* (*Gradient weighted Class Activation Map*) con el objetivo de descubrir y analizar los patrones que los modelos utilizaron para llevar a cabo la clasificación.

## 3.1. Resultados

### *3.1.1. Primera prueba con CNN personalizada*

Bajo estas configuraciones, el modelo fue deficiente. La precisión lograda al momento de predecir sobre el subconjunto de datos de validación fue de apenas 20.7%. Sin embargo, el resultado arrojado por la función de perdida (véase la Figura 9) parece ser favorable lo cual podría indicar que los parámetros del modelo no están perjudicando el proceso de entrenamiento.

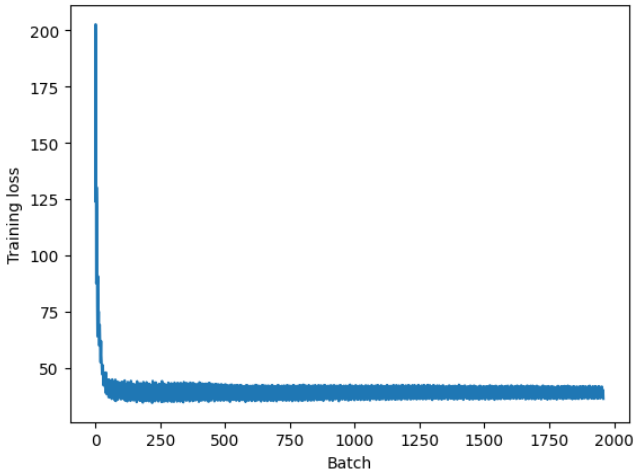


**Figura 9: Función de perdida de la primera prueba con una CNN Personalizada**

Al realizar un análisis más profundo haciendo uso de la matriz de confusión (véase la Figura 10) resultante al realizar las predicciones de valores sobre el subconjunto de datos de prueba resaltan principalmente los resultados de una clase en específico.


**Figura 10: Matriz de confusión de la primera prueba con CNN Personalizada**

Se observa notoriamente que la clase de “longsleeve” acapara las predicciones de la mayoría de las demás clases. La **Figura 11** muestra algunas muestras de la clase en cuestión, donde es posible observar que efectivamente la categoría que representa pudiera ser interpretada más como una super clase (o una clase más general en una jerarquía de clases), ya que muchas de las imágenes catalogadas bajo esta etiqueta posiblemente podrían ser también catalogadas en clases más específicas como por ejemplo 'Blazer', 'Blouse', 'Hoodie', 'Outwear', 'Shirt', o 'T-Shirt'.


**Figura 11: Muestro de la clase "longsleeve"**

### *3.1.2. Segunda prueba con CNN personalizada*

Bajo estas configuraciones el modelo arrojó resultados un poco mejores que los obtenidos en el primer experimento. La precisión que se logró al momento de predecir sobre el subconjunto de datos de validación fue de 51%. La función de perdida mostrada en la Figura 12 muestra al igual que en el experimento uno (2.5.1), el aprendizaje se mantiene estable.

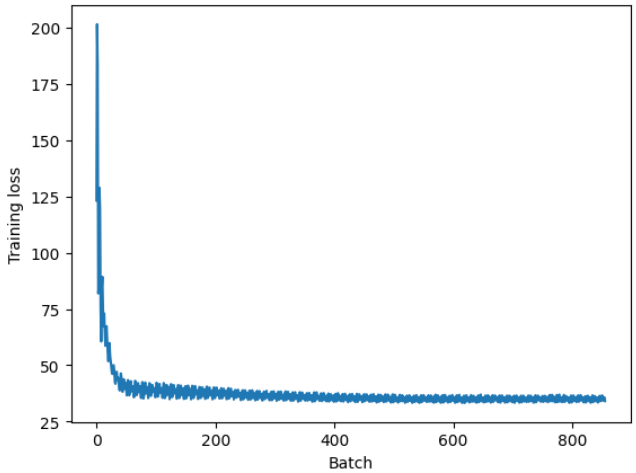


**Figura 12: Función de perdida de la segunda prueba con CNN Personalizada**

Al realizar un análisis más profundo haciendo uso de la matriz de confusión resultante al realizar las predicciones de valores sobre el subconjunto de datos de prueba resaltan nuevamente los resultados de la clase “longsleeve”. Se observa claramente en la Figura 13 que dicha clase es incorrectamente predicha por el modelo cuando la imagen de entrada es una clase distinta.

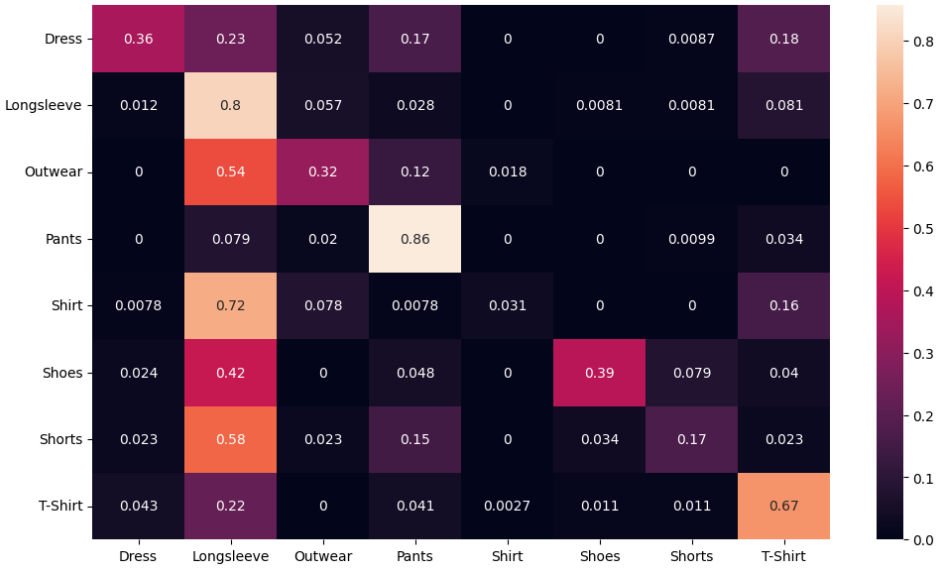


**Figura 13: Matriz de confusión de la segunda prueba con CNN Personalizada**

### *3.1.3. CNN Personalizada*

Bajo estas configuraciones el modelo arrojó resultados muy poco satisfactorios. La precisión que se logró al momento de predecir sobre el subconjunto de datos de validación fue de apenas 62%. La grafica de función de perdida mostrada en la Figura 14 continúa siendo similar a las pruebas preliminares anteriores donde se observa una gráfica que muestra un aprendizaje rápido y que se estabiliza también rápidamente.

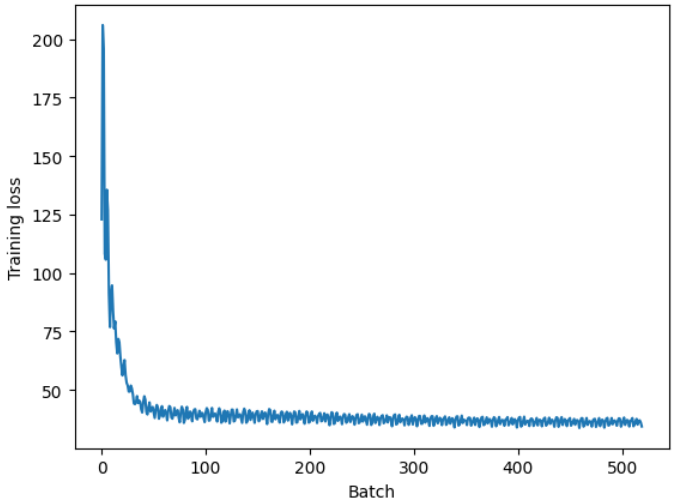


**Figura 14: Función de perdida en el experimento CNN Personalizada**

Ahora al analizar la matriz de confusión de este experimento, mostrada en la Figura 15, se puede observar que finalmente el aprendizaje se realiza de una manera mejor distribuida entre todas las clases (a comparación con las pruebas preliminares), sin embargo, el desempeño del modelo para predecir sigue siendo un tanto deficiente.

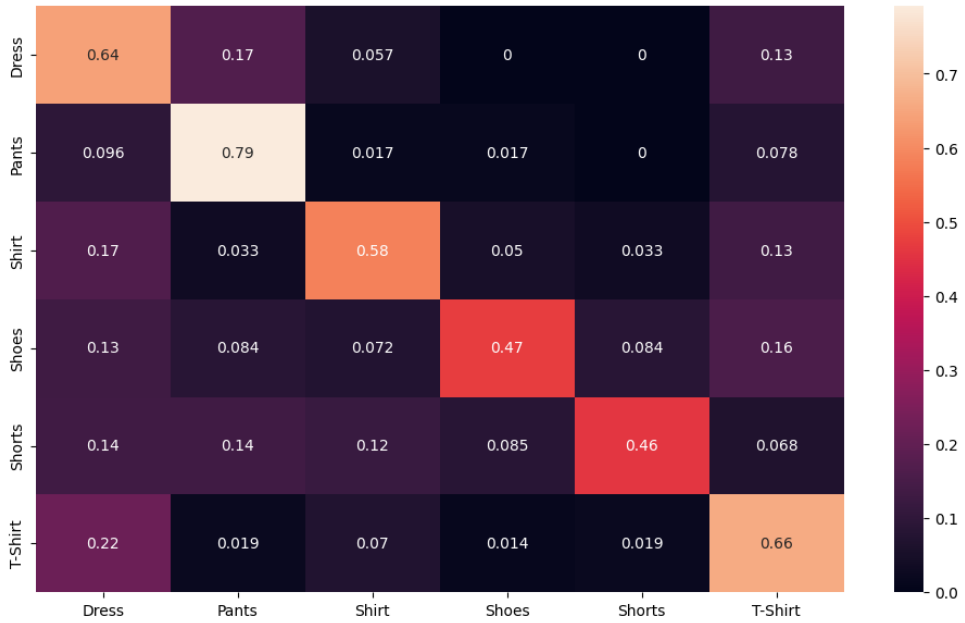


**Figura 15: Matriz de confusión del experimento - CNN Personalizada**

#### 3.1.3.1. CNN Pre entrenada

Bajo estas configuraciones el modelo arrojó resultados más satisfactorios, con una precisión sobre el subconjunto de datos de validación de 72%. Después del entrenamiento la gráfica de función de perdida igual muestra un aprendizaje estable (véase la Figura 16), sin embargo, algo que resalta de los resultados de las funciones de perdida anteriores es que los valores no varían tanto en cada *batch* lo cual se representa por una línea más suavizada.

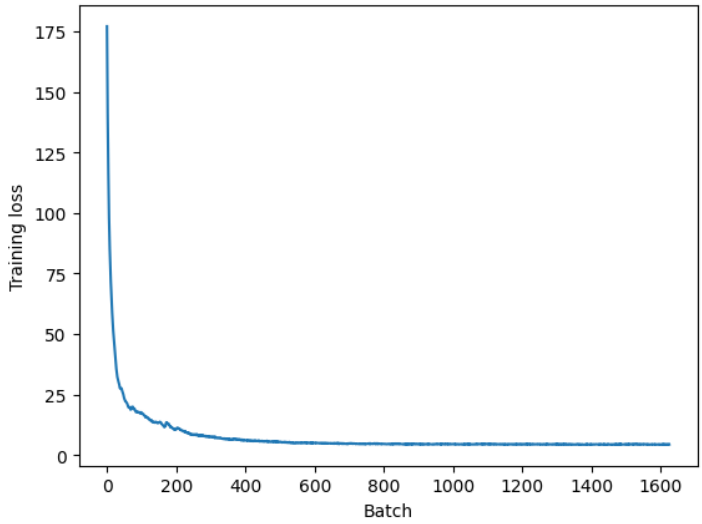


**Figura 16: Función de perdida en el experimento -CNN Pre entrenada**

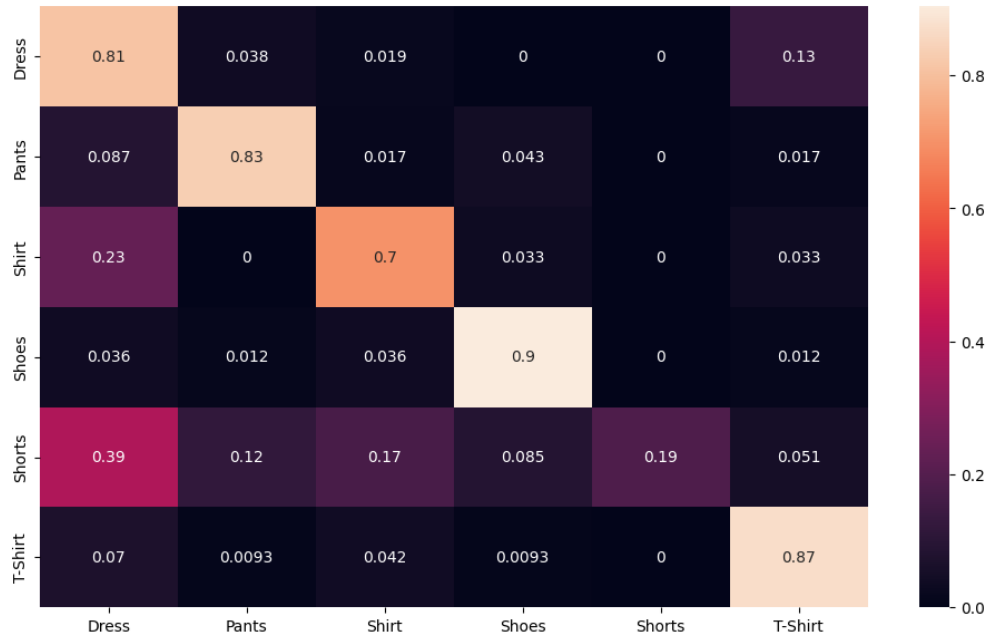


**Figura 17: Matriz de confusión del experimento – CNN Pre entrenada**

De la matriz de confusión (véase la Figura 17) es posible observar que el desempeño del modelo mejoró bastante en los porcentajes de predicción por cada una de las clases. Sin embargo, hubo un problema notable donde la clase “dress” afectó bastante las predicciones de la clase “short”, sin embargo, el resto de las clases obtuvieron un resultado muy consistente y aceptable.

#### 3.1.3.2. CNN FastAI

Bajo estas configuraciones el modelo arrojó los resultados más satisfactorios hasta el momento. La precisión que se logró al momento de predecir sobre el subconjunto de datos de validación fue de 93.5%

En este caso la matriz de confusión muestra resultados muy favorables y de igual manera las predicciones de cada una de las clases se distribuyen uniformemente en las clases correspondientes. Los resultados de la matriz de confusión de este modelo se observan en la Figura 18.

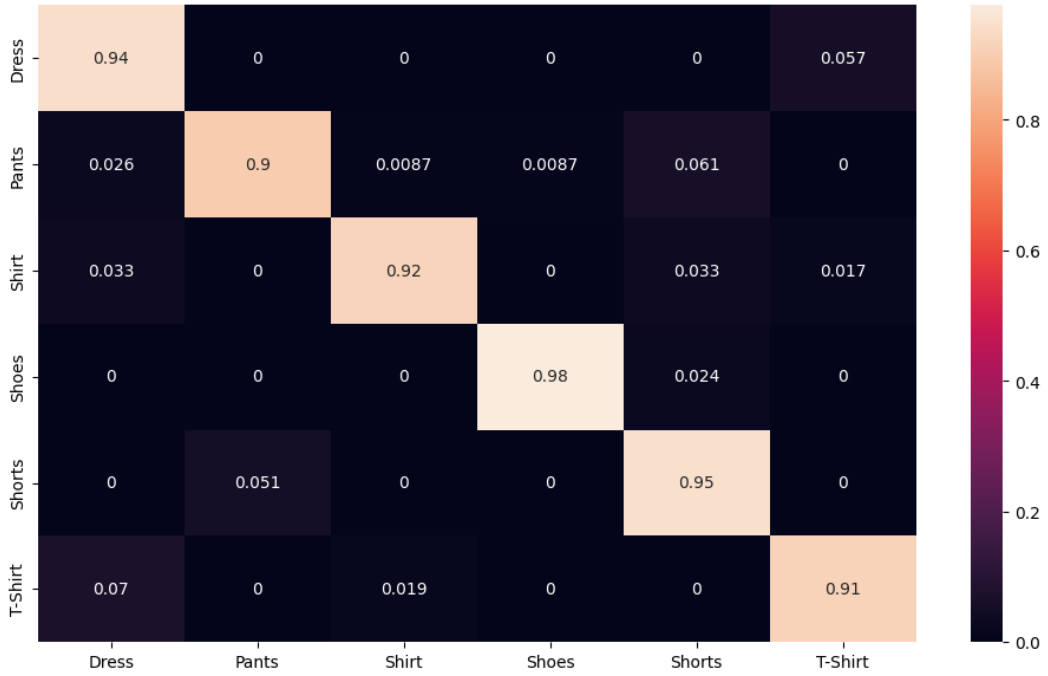


**Figura 18: Matriz de confusión del primer experimento – CNN FastAI**

### *3.1.4. Análisis de las métricas de los modelos*

En la Tabla 7 se hace un desglose simple del resultado obtenido por cada una de los de los modelos que fueron implementados. El modelo que mejor desempeño obtuvo de acuerdo con la métrica definida fue el que se construyó haciendo uso de la librería FastAI con casi 94% de precisión al momento de predecir nuevos valores. En segundo lugar, está el modelo que hace uso de un modelo pre entrenado con una precisión de 72%, y finalmente en último lugar se encuentra el modelo personalizado con una precisión del 62%.

**Tabla 7: Comparación de los resultados de las CNN**

| *MODELO* | *METRICA*<br>*Precisión* |
|---|---|
| *CNN Personalizado* | 62% |
| *CNN Pre entrenado* | 72% |
| *CNN FastAI* | 93.5% |

El tiempo de entrenamiento y prueba de los modelos fue descartado como métrica ya que los resultados obtenidos difieren en 10 y 20 puntos porcentuales entre cada modelo, por lo tanto, no se consideró necesario realizar la comparación de la duración de ejecución ya que al final solo un modelo proporciono un resultado satisfactorio.

## 3.2. Discusión

### *3.2.1. Análisis Grad-CAM del mejor modelo*

Como complemento a los resultados, se realizó un análisis Grad-CAM (*Gradient-weighted Class Activation Mapping*). Este método permite examinar la última capa oculta convolucional de la CNN que utiliza el modelo y logra identificar los puntos de activación que presentan un mayor peso en el proceso de la toma de decisión del modelo. Los resultados se representan por medio de un mapa de calor donde las zonas más rojas representan las zonas en las imágenes que más importancia tienen para llevar a cabo la clasificación [8].

En la Figura 19 se presenta una muestra de cómo el modelo con mejor desempeño (la CNN FastAI). Se presentan 3 imágenes por muestra, una con la imagen original que recibió la red neural del modelo, otra con el mapa de calor con los puntos o zonas de activación con mayor peso para el modelo y una tercera imagen con las dos imágenes previamente descritas sobrepuestas en una sola.

En dicho análisis, algo que resalta a la vista de manera considerable es que el mapa de calor de este modelo muestra zonas de intensidad fuerte representadas por los tonos rojos en la mayoría del área de la imagen. Esto podría ser interpretado como que toda la imagen, y no solo la prenda, tiene un peso importante para el modelo al momento de realizar la decisión. Esto resulta inquietante debido a que podría implicar que no solo la prenda presente en la imagen es importante para el modelo, sino que también el fondo de la imagen tiene un peso similar.


**Figura 19: Grad-CAM análisis en muestra de imágenes - CNN FastAI**

### *3.2.2. Análisis Grad-CAM del segundo mejor modelo*

El resultado del análisis *Grad-CAM* del segundo mejor modelo arrojo resultados en los mapas de calor de las imágenes que fueron más fáciles de interpretar. Fue posible identificar patrones en las zonas de mayor peso y por lo tanto facilitando también los rasgos que el modelo utilizó para realizar la categorización.

Para ejemplificar lo anterior a continuación se presentan algunas muestras del resultado del mapa de calor de algunas clases especificas donde los patrones de las zonas de activación de la imagen resaltan fácilmente. Estas son las zonas que el modelo utiliza para realizar la clasificación.

#### 3.2.2.1. Clase T-Shirt Grad-CAM

En la Figura 20 se muestran algunos ejemplos de los mapas de calor de la clase T-Shirt.

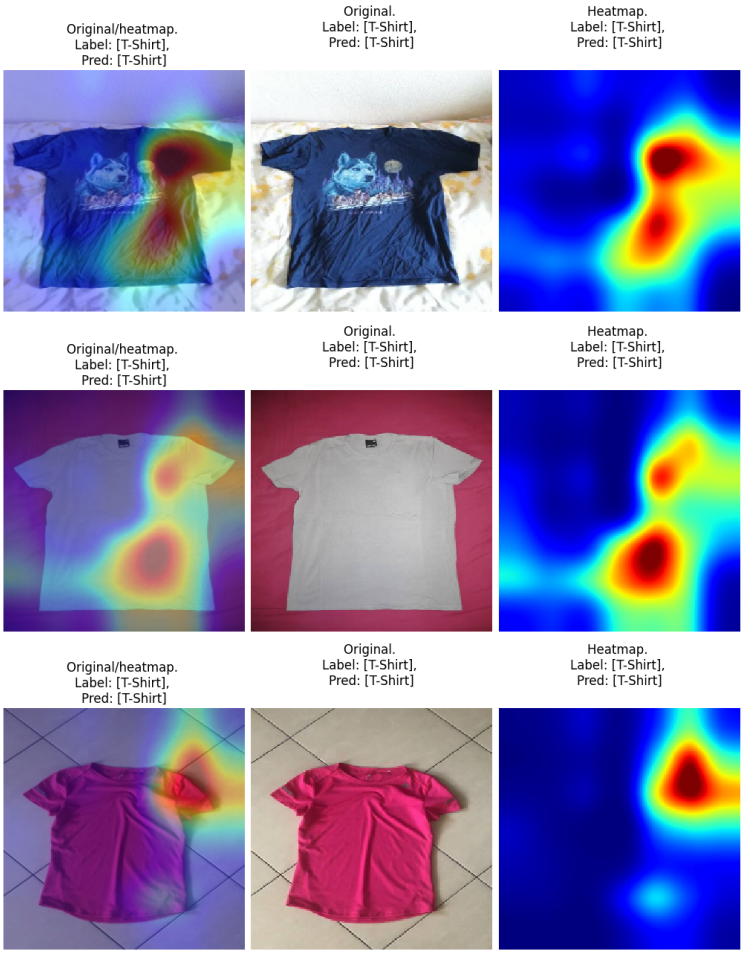


**Figura 20: Análisis Grad-CAM de la clase Shirt**

Pareciera entonces que lo que el modelo busca en la imagen para clasificarla como una T-Shirt es una la presencia de una manga corta y un ángulo recto en la parte inferior de la prenda.

#### 3.2.2.2. Clase Pants Grad-CAM

Otro ejemplo de patrones fáciles de identificar en los mapas de calor son los que proporciona el modelo sobre la clase Pants. En la Figura 21 se observan las zonas de activación de más significativas que el modelo utiliza para clasificar una imagen como un pantalón. Pareciera entonces que el modelo busca en la imagen los ángulos agudos presentes en el tiro del pantalón, además del largo y el ancho de las piernas.

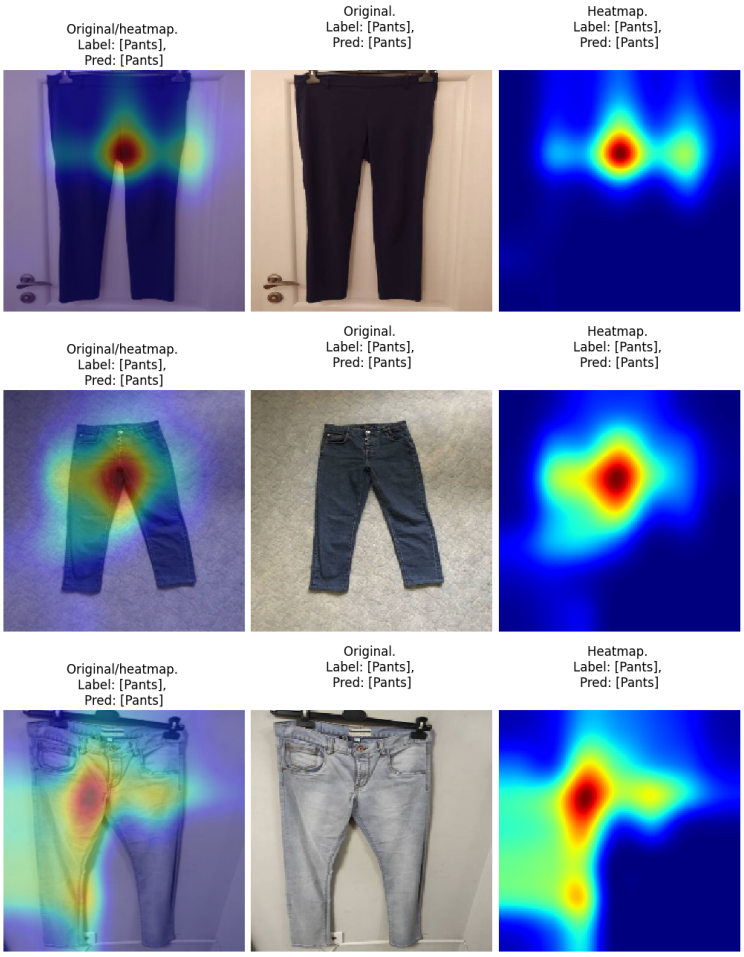


**Figura 21: Análisis Grad-CAM de la clase Pants**

Un patrón similar es el que se utiliza también para la identificación de imágenes pertenecientes a la clase Shorts tal como puede observarse en la Figura 22.

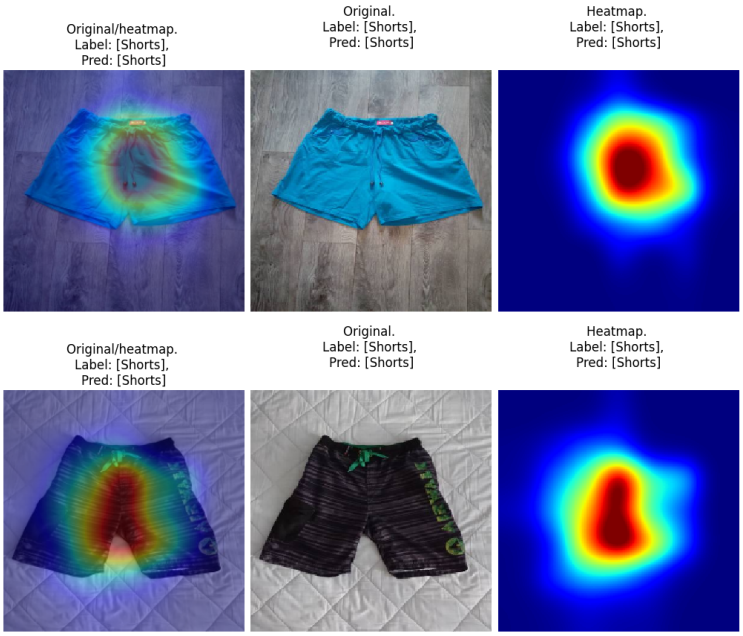


**Figura 22: Análisis Grad-CAM de la clase Shorts**

### *3.2.3. Comparación del análisis Grad-CAM de ambos modelos*

Los resultados del análisis *Grad-CAM* del modelo CNN FastAI y el modelo CNN pre entrenada presentaron resultados muy diferentes en sus mapas de calor. Resalta principalmente que los mapas de calor de la CNN pre entrenada presentan zona con puntos de activación muy bien delimitados en las imágenes y con patrones en cada clase de prenda que son fácilmente identificados.

Esto en contraste de con la CNN FastAI cuyos mapas de calor resultaron ser más uniformes sin zonas con puntos de activación definidos. Mas bien resalta demasiado la prevalencia del color rojo en varias intensidades. Esto indicaría que todo el contenido de la imagen está siendo utilizada por el modelo para realizar la categorización, y como ya se explicó en el punto 3.1.4., las zonas rojas en el mapa de calor representan los puntos de activación más relevantes para el modelo.

Este resultado resulta un tanto controversial ya que a pesar de que el modelo CNN FastAI arrojó la mejor métrica (93% de precisión) en cuanto a desempeño, significaría también que para este modelo no solo la prenda en la imagen es relevante, sino que también el fondo de esta. Esto representa un resultado inesperado ya que la prenda debería ser el objeto principal para analizar, independientemente de lo que aparezca en el resto de la imagen.

Estos resultados prevalentemente rojos en los mapas de calor y la poca capacidad para identificar zonas de puntos de activación relevantes en las imágenes, podrían representar también un *overfitting* o un entrenamiento excesivo sobre el conjunto de datos. Esta percepción de *overfitting* podría cobra aún más sentido ya que la CNN FastAI no hace uso

explícito de los parámetros lambda1 y lambda2 (2.5.5.) que fungen con parámetros de regularización L1 y L2 respectivamente los cuales sí se encuentran presentes en el modelo CNN pre entrenado (2.5.4.). Estos parámetros tienen precisamente la función de prevenir el sobreajuste del modelo al penalizar los pesos más altos en la red neural. En contraste con los resultados del modelo CNN pre entrenado donde todas las zonas de puntos de activación más relevantes se encuentran solo casi exclusivamente dentro del perímetro de la prenda.

# 4. CONCLUSIONES

Resumen: Se describen las conclusiones obtenidas de acuerdo con los objetivos planteados al inicio del trabajo de investigación. Se selecciona el modelo ganador de acuerdo con las métricas propuestas y se lleva a cabo una discusión sobre los posibles trabajos futuros que pueden sen aplicados en este conjunto de datos para mejor los resultados.

## 4.1. *Conclusiones*

El conjunto de datos de imágenes tuvo llevar un conjunto de transformaciones, limpieza y acondicionamiento previo para facilitar que los modelos aprendan de manera más rápida y eficiente, esto debido a que conjunto de datos original presentaba una cantidad significativa de clases que, de haber sido utilizadas sin ningún tratamiento previo, los modelos no habrían obtenido los resultados deseados.

Algunos pasos previos constaron de varias transformaciones cuyo objetivo principal fue la de fungir como un preproceso de *data augmentation*, que tuvo la función de aumentar la cantidad de muestras presentes en cada clase, debido a que existía un desbalance significativo entre ellas. Además, se realizó un proceso de limpieza donde se pretendía acondicionar el conjunto para que los datos de entrada fueran los más significativos de cada clase. Este consistió principalmente en la eliminación de clases con muy pocas muestras dentro del conjunto de datos, así como la remoción de algunas clases que representaban categorías más globales que podrían ser clasificadas como superclases o clases padres.

Finalmente se llevó a cabo un análisis Grad-CAM satisfactorio donde se descubrieron las zonas de puntos de activación que cada modelo con buen desempeño utiliza al momento de clasificar. El descubrimiento más significativo fue que el modelo con mejor desempeño de los tres que fueron evaluados presentó una gran cantidad de puntos de activación distribuidos uniformemente a través de toda el área de la imagen, mientras que el segundo modelo con mejor desempeño presento puntos de activación más aislados en zonas clave dentro de cada imagen de acuerdo la prenda que mostraba.

En general, el objetivo se cumplió satisfactoriamente, esto debido a que se logró llevar a cabo la segmentación de prendas de vestir presentes en el conjunto de datos de imágenes seleccionado. De los tres modelos de redes neurales convolucionales, dos de ellos presentaron un desempeño aceptable con una precisión mayor al 70%. Ambos modelos con buen desempeño hacen uso de la red ResNet18 pre entrenada.

## 4.2. *Trabajo Futuro*

Uno de los problemas principales que resaltaron durante esta investigación fue la aparente clasificación errónea que existía originalmente en el conjunto de datos, así como la existencia de clases redundantes que afectaron el desempeño de los modelos preliminares. Como trabajos futuros para este mismo conjunto de datos de imágenes posiblemente se podrían realizar las siguientes actividades:

- Para mejorar el etiquetado de las clases podría ser implementado algún algoritmo de clusterización para identificar las imágenes que presentan prendas similares y

después manualmente introducir la etiqueta correspondiente de cada clúster generado.

- Ahora que, si se desea continuar con el etiquetado actual, se podría realizar experimentos con modelos de clasificación de imágenes de clases múltiples. Con esto sería posible conservar las clases redundantes o superclases que existen actualmente en el conjunto de datos y realizar predicciones donde las prendas pudieran ser clasificadas en más de una clase. Por ejemplo, las clases “longsleeve” y “shirt”, o “longsleeve” y “outwear”, cuyas prendas podrían ser clasificadas en ambas clases.

# BIBLIOGRAFIA


[1] A. Turner, «HOW MANY SMARTPHONES ARE IN THE WORLD?,» 2023. [En línea]. Available: https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.

[2] M. Broz, «How many pictures are there (2023): Statistics, trends, and forecasts,» 2023. [En línea]. Available: https://photutorial.com/photos-statistics/.

[3] Google Cloud, «Documentación de AutoML Vision,» 2023. [En línea]. Available: https://cloud.google.com/vision/automl/docs?hl=es-419. [Último acceso: January 2024].

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg y L. Fei-Fei, «ImageNet Large Scale Visual Recognition Challenge,» *International Journal of Computer Vision,* vol. 115, p. 211–252, 2015.

[5] Statista, «Computer vision large-scale visual recognition challenge (ILSVRC) winning error rates, from 2020 to 2017,» 17 March 2022. [En línea]. Available: https://www.statista.com/statistics/808190/worldwide-large-scale-visual-recognition-challenge-error-rates/. [Último acceso: 2023].

[6] A. Grigorev, «Clothing Dataset,» Medium, 21 October 2020. [En línea]. Available: https://medium.com/data-science-insider/clothing-dataset-5b72cd7c3f1f. [Último acceso: 2023].

[7] Google for Developers, «Classification: Accuracy,» Google, 18 Julio 2022. [En línea]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy. [Último acceso: Octubre 2023].

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh y D. Batra, «Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,» 2019. [En línea]. Available: https://arxiv.org/abs/1610.02391. [Último acceso: October 2023].