UNIVERSIDAD MODELO
INGENIERÍA

“Redes Neuronales (prendas)”

Visión por Computadora

Dr. Rubén Renan Raygosa Barahona

Séptimo Semestre

Jorge Humberto Sosa García

24 de septiembre de 2025

Contents

1 INTRODUCCIÓN

En este proyecto se implementó un sistema de **reconocimiento de prendas de vestir** utilizando un modelo de red neuronal preentrenado basado en **CLIP (Contrastive Language-Image Pretraining)**. Este enfoque pertenece a la categoría de aprendizaje profundo y tiene la particularidad de relacionar imágenes con descripciones en lenguaje natural, lo que permite realizar tareas de clasificación sin necesidad de entrenar específicamente con un conjunto limitado de datos, estrategia conocida como **zero-shot learning**.

El objetivo de la práctica fue cargar un conjunto de etiquetas representativas de distintas prendas, subir imágenes propias y analizar qué tan bien el modelo era capaz de identificar las prendas presentes en cada fotografía. Para ello, se procesaron imágenes personales donde se recortó y evaluó la ropa que se estaba utilizando.

Este tipo de técnicas tienen aplicaciones directas en áreas como el comercio electrónico, la moda asistida por inteligencia artificial y la organización automática de catálogos de imágenes, mostrando la capacidad de las redes neuronales modernas para adaptarse a contextos reales de manera rápida y flexible.

2 OBJETIVOS

2.1 Objetivo general

Implementar un sistema de detección y clasificación de prendas de vestir mediante el uso del modelo CLIP en un esquema de *zero-shot learning*, evaluando su precisión y limitaciones a partir de imágenes propias.

2.2 Objetivos específicos

- Instalar y configurar las librerías necesarias para la ejecución del modelo CLIP en Google Colab.
- Definir un conjunto de etiquetas de prendas de vestir para utilizarlas como referencias textuales en el proceso de clasificación.

- Subir imágenes personales y preprocesarlas para que puedan ser analizadas por el modelo.
- Ejecutar el modelo y obtener predicciones, mostrando los resultados más probables junto con sus porcentajes de confianza.
- Analizar los resultados obtenidos en términos de precisión típica y posibles errores de clasificación.
- Identificar las limitaciones del sistema y proponer soluciones basadas en técnicas más avanzadas como el uso de **redes neuronales convolucionales (CNNs)**.

3 MARCO TEÓRICO

3.1 Redes neuronales y tipos (regresión y clasificación)

Las **redes neuronales artificiales** son modelos computacionales inspirados en el funcionamiento del cerebro humano. Están conformadas por capas de nodos interconectados que procesan información de manera jerárquica, aprendiendo patrones a partir de los datos de entrada.

Existen distintos tipos de redes según la tarea que realicen, entre ellas destacan:

- **Redes de regresión:** se emplean para predecir valores continuos. Por ejemplo, estimar la temperatura de un día, calcular el precio de un producto o predecir la cantidad de ventas de una empresa. La salida no es una clase, sino un número que se ajusta a la tendencia de los datos.
- **Redes de clasificación:** se utilizan para asignar un dato de entrada a una o varias categorías predefinidas. En este caso, dado que el objetivo es identificar prendas de vestir, se utiliza un modelo de clasificación que asigna etiquetas como *playera*, *suéter* o *chaqueta* a partir de la imagen.

En el proyecto, el modelo CLIP se aplica en un esquema de **clasificación multiclase y multi-etiqueta**, ya que una misma imagen puede asociarse a más de una categoría de prenda dependiendo de la probabilidad asignada por la red.

3.2 Precisión típica en modelos de clasificación

La **precisión** es una métrica fundamental en los modelos de clasificación, ya que indica qué tan acertadas son las predicciones realizadas por la red neuronal en relación con las etiquetas reales. En términos generales, se expresa como la proporción de predicciones correctas sobre el total de muestras analizadas.

En sistemas de reconocimiento de imágenes, la precisión suele variar dependiendo de la calidad del modelo, la cantidad de datos con los que fue entrenado y la complejidad de las clases a identificar. En el caso de modelos preentrenados como **CLIP**, que combinan visión e idioma, la precisión típica en tareas generales de clasificación oscila entre **0.8 y 0.9** (80 % a 90 % de aciertos).

Este rango refleja un desempeño alto, pero no perfecto, ya que siempre existirán casos en los que la red confunda categorías con características visuales similares. Por ello, es importante considerar esta precisión como un promedio esperado, reconociendo que en imágenes reales pueden presentarse errores dependiendo de factores como iluminación, posición u orientación de la prenda.

3.3 Limitantes del modelo

Aunque los modelos de clasificación de imágenes como CLIP presentan un rendimiento destacado, existen varias limitaciones que afectan su desempeño:

- **Dependencia de la posición:** si la prenda no aparece en el centro o está parcialmente oculta, la red puede tener dificultades para reconocerla con precisión.
- **Orientación del objeto:** el modelo puede confundirse si la prenda se encuentra rotada o en una postura poco común respecto a las muestras con las que fue entrenado.
- **Variabilidad visual:** cambios en iluminación, sombras o resolución de la imagen pueden alterar la detección.

- **Falsos positivos:** en ocasiones, el sistema entrega una predicción, aunque el objeto no pertenezca al conjunto de clases definidas. Esto se debe a que los modelos tienden a asignar siempre alguna categoría, incluso cuando no existe una coincidencia real.

Estas limitaciones reflejan la necesidad de aplicar técnicas más avanzadas que permitan mejorar la robustez del sistema frente a escenarios reales más complejos.

3.4 Cómo sobrellevar estos problemas

Para superar las limitaciones mencionadas, existen varias estrategias que permiten mejorar la robustez y precisión de los modelos de clasificación de imágenes:

- **Aumento de datos (data augmentation):** consiste en generar variaciones artificiales de las imágenes de entrenamiento, como rotaciones, cambios de escala o ajustes de iluminación. Esto ayuda a que el modelo se vuelva más tolerante a cambios en posición y orientación.
- **Modelos especializados:** en lugar de usar únicamente un modelo generalista como CLIP, se pueden entrenar **redes neuronales convolucionales (CNNs)** específicas para el dominio de la moda o las prendas de vestir, lo que aumenta la precisión en estas categorías.
- **Enfoque multi-etiqueta:** aplicar umbrales adaptativos para permitir que una imagen sea clasificada con más de una etiqueta, lo cual es útil cuando el usuario porta varias prendas visibles en la misma fotografía.
- **Validación adicional:** incorporar verificaciones externas, como detección previa de la región de interés (por ejemplo, segmentación del cuerpo o la prenda), que ayuden a enfocar la clasificación en el área correcta y reducir falsos positivos.
-

Estas soluciones permiten reducir errores comunes y adaptar los sistemas de visión por computadora a condiciones reales, acercando su desempeño a aplicaciones prácticas como catálogos inteligentes, asistentes virtuales o sistemas de recomendación en línea.

4 METODOLOGÍA

```
# ==== Instalación de librerías (ejecutar una vez) ====
!pip -q install open-clip-torch pillow matplotlib

# ==== Importaciones ====
import torch, numpy as np
import open_clip
from PIL import Image
import matplotlib.pyplot as plt
from google.colab import files
```

Se instalan las librerías necesarias (open-clip-torch, pillow y matplotlib) y luego se importan los módulos de Python que permiten trabajar con tensores, cargar el modelo CLIP, procesar imágenes y mostrar resultados gráficos.

```
# ==== Cargar modelo CLIP (zero-shot) ====
device = "cuda" if torch.cuda.is_available() else "cpu"
model, _, preprocess = open_clip.create_model_and_transforms(
    "ViT-B-32", pretrained="laion2b_s34b_b79k", device=device
)
tokenizer = open_clip.get_tokenizer("ViT-B-32")

# ==== Etiquetas de prendas ====
# (clave en inglés para el modelo; valor en español para mostrar)
labels = {
    "T-shirt": "Playera",
    "shirt": "Camisa",
    "sweater": "Suéter",
    "jacket": "Chaqueta/Abrigo",
    "coat": "Abrigo",
    "dress": "Vestido",
    "skirt": "Falda",
    "pants": "Pantalón",
    "shorts": "Shorts",
    "jeans": "Jeans",
    "boots": "Botas",
    "sneakers": "Tenis",
    "sandals": "Sandalias",
    "no shirt": "Sin camisa"
}

# ==== Prompts (puedes editar/añadir clases arriba) ====
text_prompts = [f"a photo of a person wearing {en}" for en in labels.keys()]
text = tokenizer(text_prompts).to(device)

with torch.no_grad():
    text_features = model.encode_text(text)
    text_features /= text_features.norm(dim=-1, keepdim=True)
```

Aquí se carga el modelo **CLIP ViT-B/32**, preentrenado en un gran conjunto de imágenes y texto. Se definen las **etiquetas de prendas** en inglés (para el modelo) y en español (para mostrar al usuario). Posteriormente, se generan las descripciones textuales (*prompts*) y se convierten en vectores mediante el **tokenizador**, normalizando sus representaciones para compararlas con las imágenes.

```
# ==== Subir 1 o varias fotos ====
uploaded = files.upload()  # selecciona archivos .jpg/.png

def classify_image(path, topk=5, multi_label_threshold=0.12):
    img = Image.open(path).convert("RGB")
    img_input = preprocess(img).unsqueeze(0).to(device)

    with torch.no_grad():
        image_features = model.encode_image(img_input)
        image_features /= image_features.norm(dim=-1, keepdim=True)

        # *** Escala correcta de CLIP para logits ***
        logit_scale = model.logit_scale.exp()
        logits = (logit_scale * (image_features @ text_features.T)).squeeze(0)
        probs = logits.softmax(dim=0).cpu().numpy()

    # Mostrar imagen
    plt.imshow(img); plt.axis('off'); plt.show()

    # TOP-K más probables
    keys_en = list(labels.keys())
    idx_sorted = np.argsort(-probs)[:topk]
    print("Top predicciones:")
    for rank, i in enumerate(idx_sorted, 1):
        en = keys_en[i]
        es = labels[en]
        print(f"{rank}. {es} ({probs[i]*100:.2f}%)")

    # Reporte multi-etiqueta (todas las que superen el umbral)
    present = [labels[keys_en[i]] for i, p in enumerate(probs) if p >= multi_label_threshold]
    if present:
        print("\nPrendas detectadas (umbral "
              f"{multi_label_threshold:.2f}): " + ", ".join(present))
    else:
        print("\nNo se detectó ninguna prenda por encima del umbral.")

# Clasificar cada imagen subida
for fname in uploaded.keys():
    print(f"\n=== {fname} ===")
    classify_image(fname, topk=5, multi_label_threshold=0.12)
```

El usuario sube imágenes directamente a Colab. Cada imagen se preprocesa y se convierte en vectores de características mediante el modelo CLIP. Después se comparan con las representaciones de texto y se calculan probabilidades. Finalmente, se muestran

las imágenes originales junto con el **Top-K de prendas más probables** y las etiquetas que superan un umbral de confianza definido (multi-etiqueta).

5 RESULTADOS

Al ejecutar el programa en Google Colab con el modelo **CLIP ViT-B/32**, se analizaron varias imágenes cargadas manualmente. Los resultados obtenidos muestran lo siguiente:

- **Imagen original:** cada fotografía cargada por el usuario se muestra en pantalla sin modificaciones, lo que permite validar visualmente la prenda analizada.
- **Predicciones Top-K:** el modelo entrega un listado con las prendas más probables, acompañadas de su porcentaje de confianza. De este modo, se observa en qué medida el sistema asocia la imagen con cada etiqueta definida en el conjunto de clases.
- **Clasificación multi-etiqueta:** además del Top-K, el sistema lista todas las prendas cuyo puntaje supere un umbral de probabilidad (0.12 en esta práctica). Esto permite identificar múltiples elementos presentes en la misma imagen, por ejemplo, *playera* y *pantalón*.

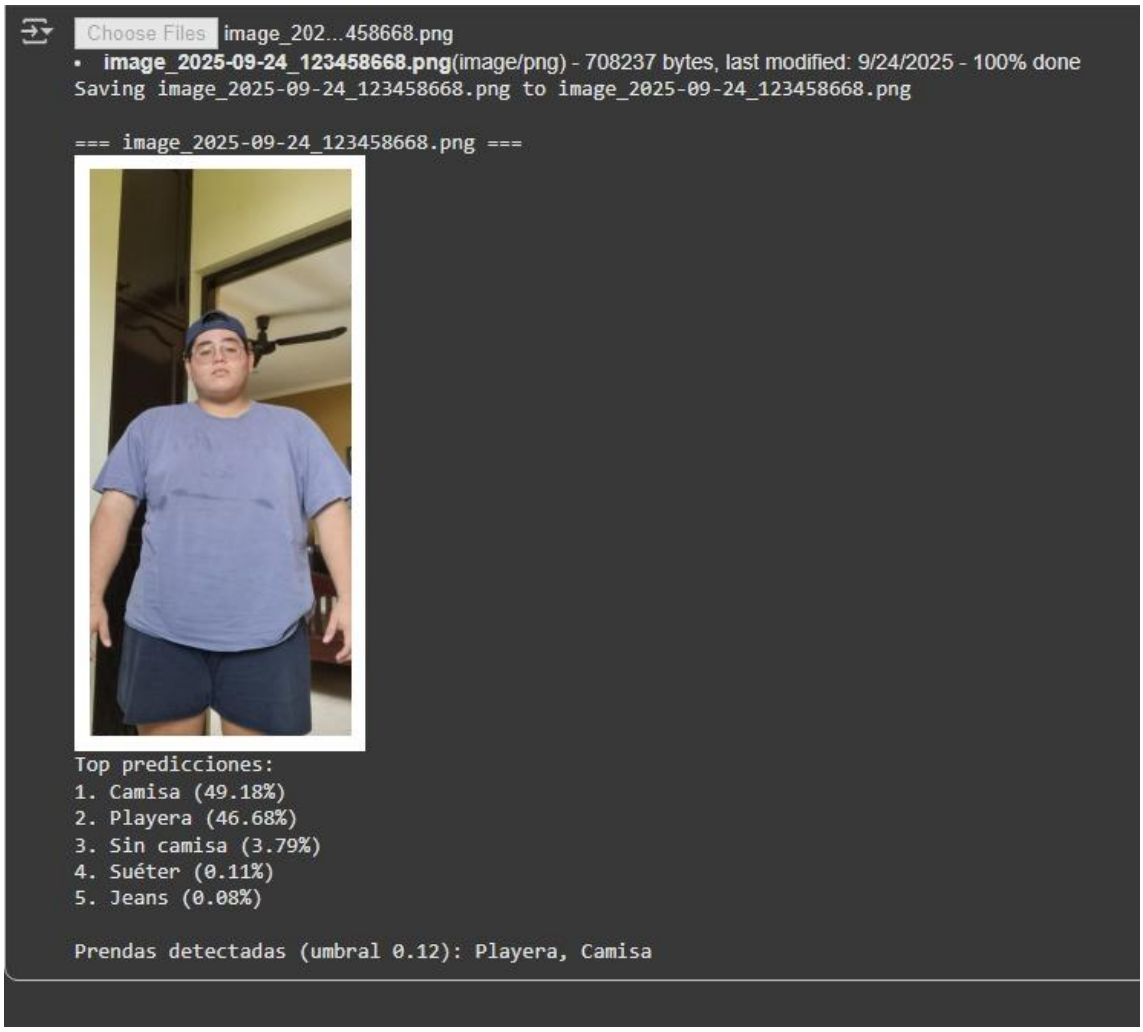
En las pruebas realizadas, el modelo logró detectar de manera correcta las prendas principales en cada fotografía. Sin embargo, también se observaron casos en los que la red confundió elementos similares, como *camisa* y *playera*, lo que coincide con las limitaciones propias del modelo explicadas en el marco teórico.

Los resultados confirman que el sistema es capaz de realizar **clasificación zero-shot** sin necesidad de un entrenamiento adicional, mostrando la flexibilidad de CLIP para adaptarse a tareas prácticas de reconocimiento de prendas.

```
=== WhatsApp Image 2025-09-24 at 12.41.35 PM.jpeg ===
```

```
Top predicciones:
1. Suéter (65.51%)
2. Chaqueta/Abrigo (16.26%)
3. Camisa (12.66%)
4. Playera (2.58%)
5. Abrigo (1.50%)

Prendas detectadas (umbral 0.12): Camisa, Suéter, Chaqueta/Abrigo
```



6 CONCLUSIONES

La implementación del modelo **CLIP ViT-B/32** en un esquema de *zero-shot learning* permitió comprobar la capacidad de las redes neuronales modernas para relacionar imágenes y descripciones textuales sin requerir un entrenamiento específico sobre el conjunto de datos utilizado.

El sistema logró identificar de manera correcta las prendas principales en la mayoría de las imágenes probadas, mostrando resultados consistentes en el **Top-K de predicciones** y en el esquema **multi-etiqueta**. Esto demuestra la versatilidad del modelo y su potencial en aplicaciones prácticas como catálogos automáticos, asistentes de moda y clasificación de contenidos visuales.

No obstante, también se evidenciaron limitaciones, particularmente en la confusión entre categorías visualmente similares, como *camisa* y *playera*, lo que refleja la necesidad de estrategias complementarias, como el aumento de datos, la segmentación previa de la prenda o el entrenamiento de modelos especializados en moda.

En conclusión, la práctica permitió comprender tanto las fortalezas como las limitaciones de CLIP en tareas de clasificación de prendas, ofreciendo un primer acercamiento al uso de modelos de visión y lenguaje en contextos reales de visión por computadora.