



# ARF - Armario Virtual

Jorge Haces Fuertes  
Sergio Madrid Pérez  
Carlos Turró Alcalá



# Introducción

---

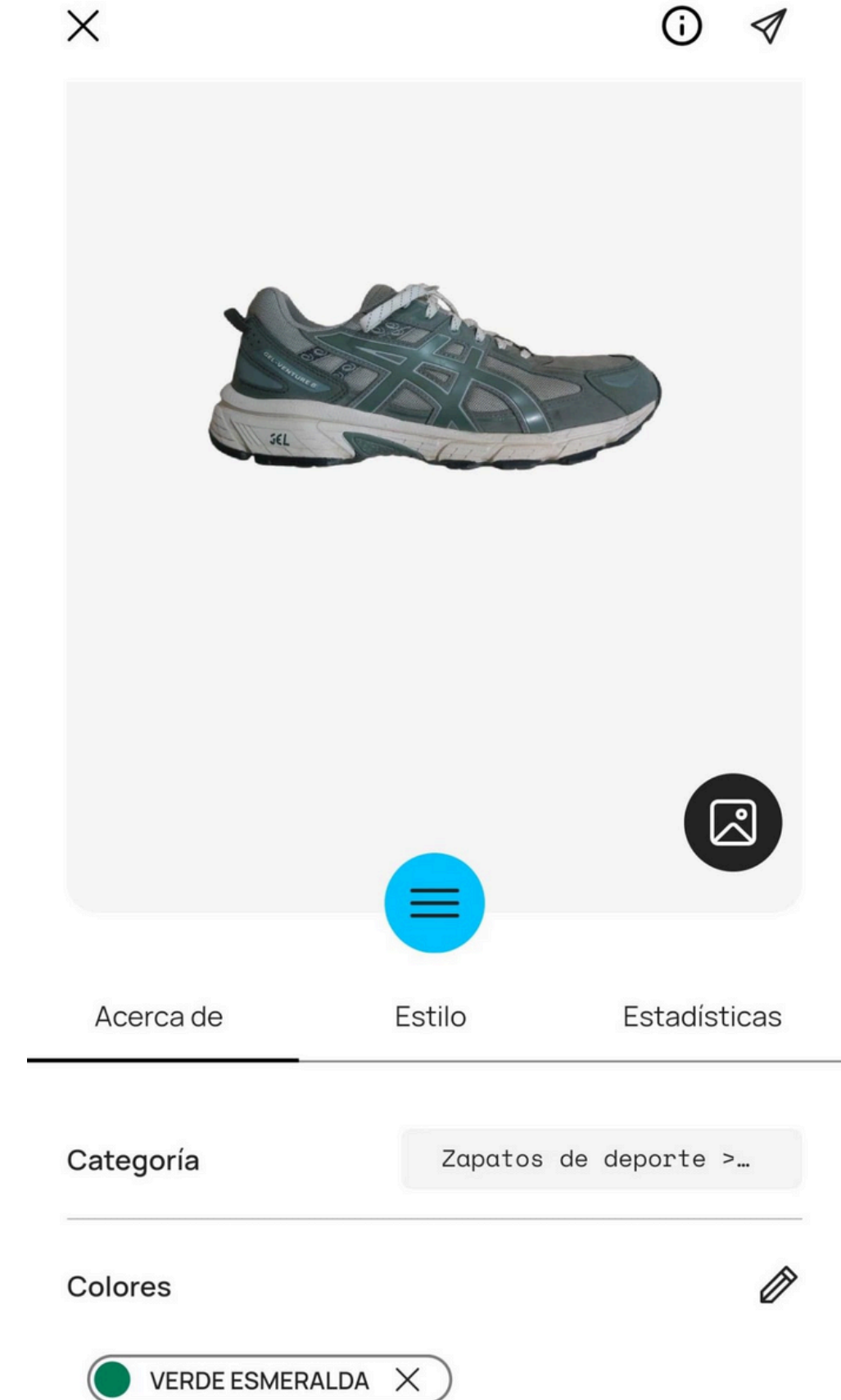
## Armario Virtual Inteligente:

- **Identificación y segmentación** de **todas** las prendas en una imagen.
- **Clasificación** de cada prenda por tipo, color y estilo.
- **Recomendación** de combinaciones de atuendos.
- **Organización** visual y estructurada del armario personal.

# Trabajos similares

## Whering:

- Elimina el fondo de fotos de prendas **únicas**.
- Etiqueta categoría y color automáticamente.
- Almacena en armario virtual.
- No clasifica el estilo.
- No proporciona recomendaciones.

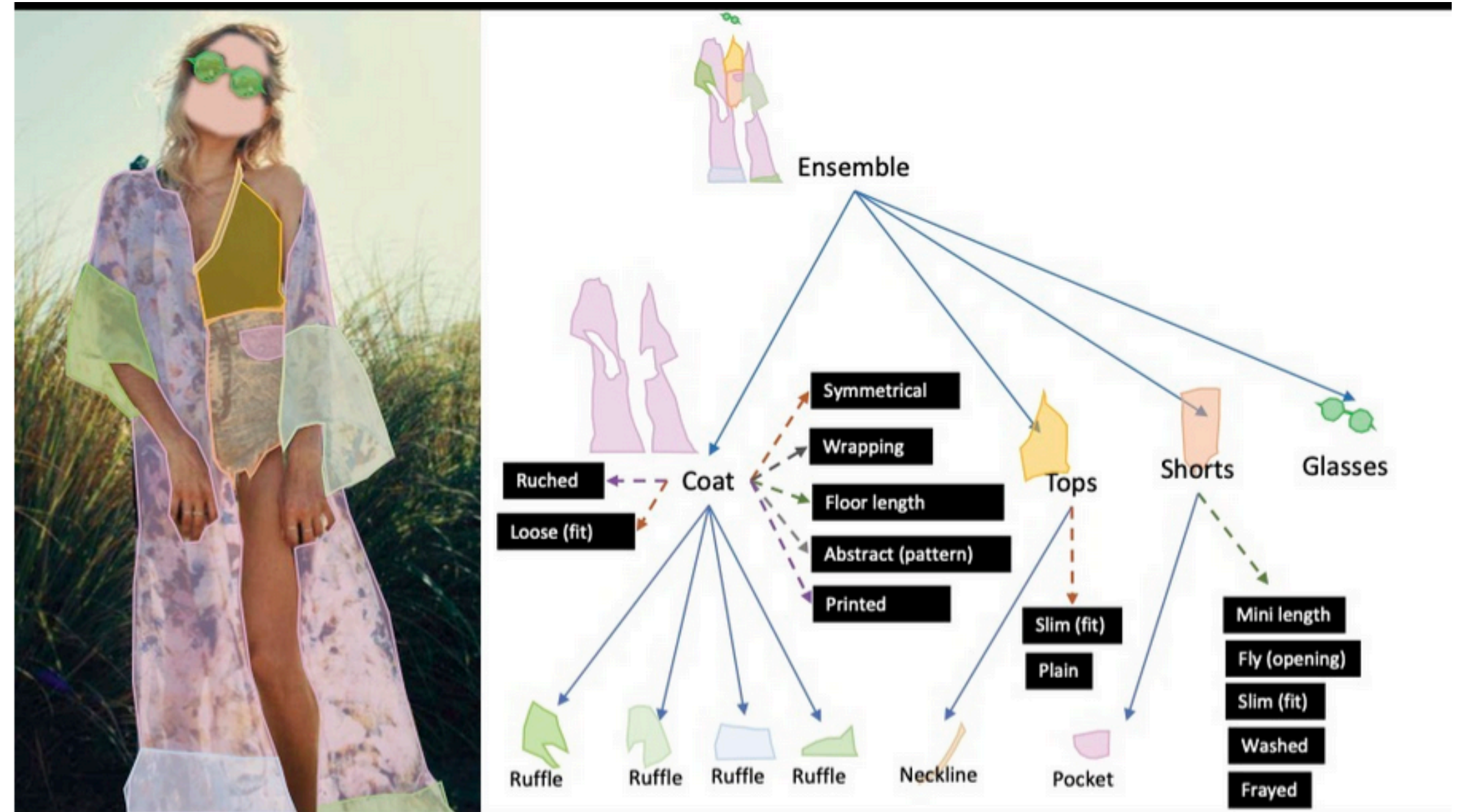


# Descripción de la tarea iMaterialist

## iMaterialist (Fashion) 2021 at FGVC8

Fine-grained segmentation task for fashion and apparel

- **50K** imágenes
- **46** Clases
  - 29 piezas principales
  - 16 complementos
- **294** atributos (no hacemos uso)

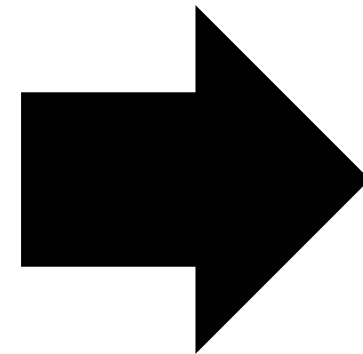




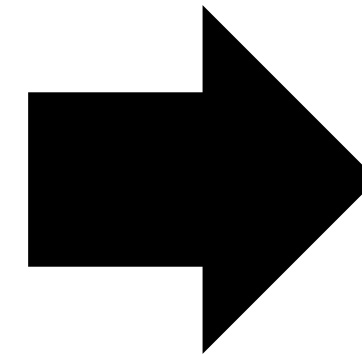
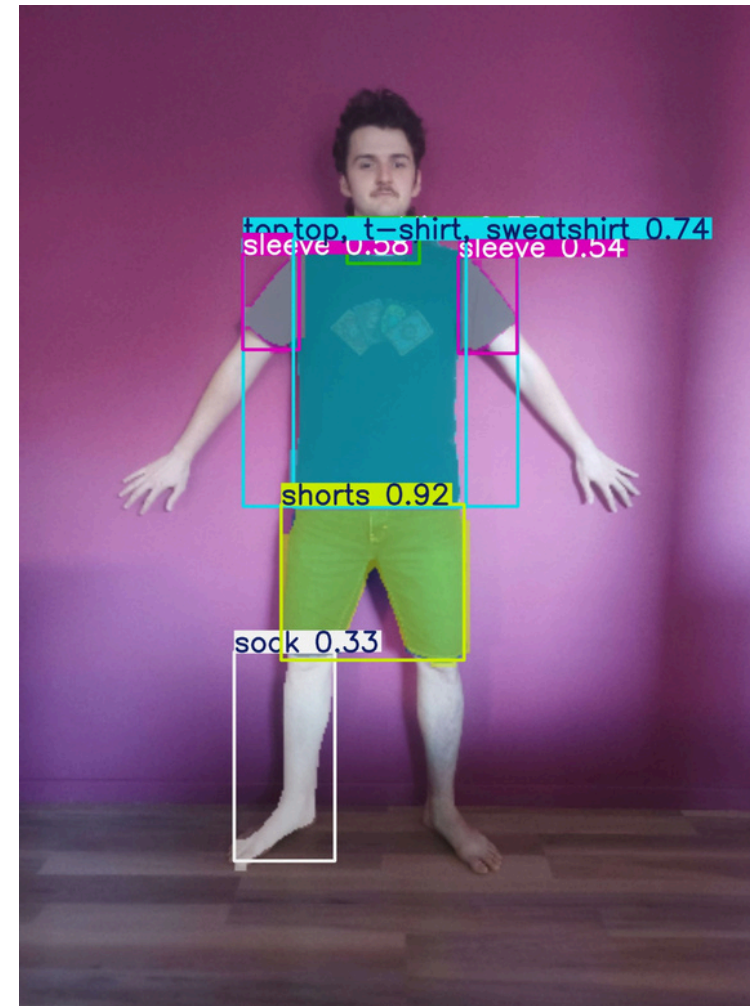
# ¿Cual es el input del sistema y su output?

---

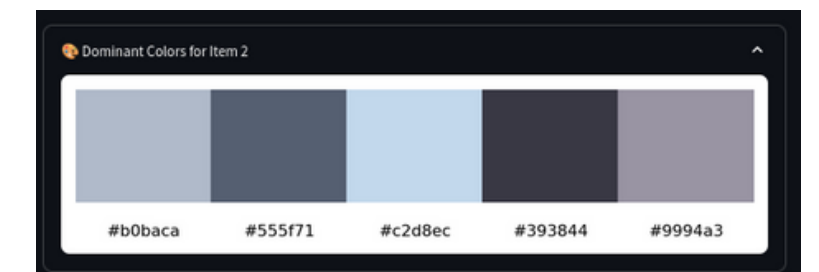
Imagen de entrada



- Clases
- Máscaras de bits



- Pieza de ropa
- Estilo
- Colores



# Arquitectura del sistema

---

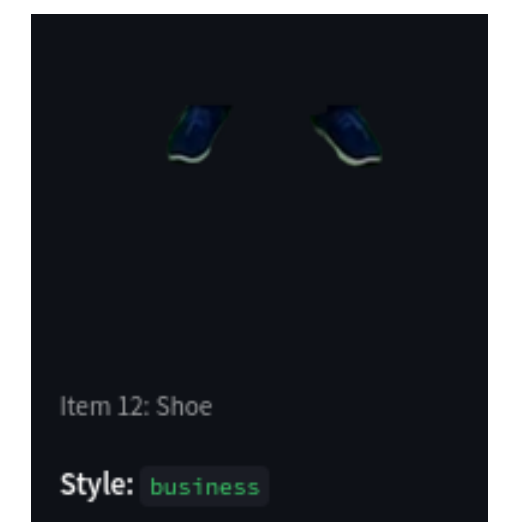
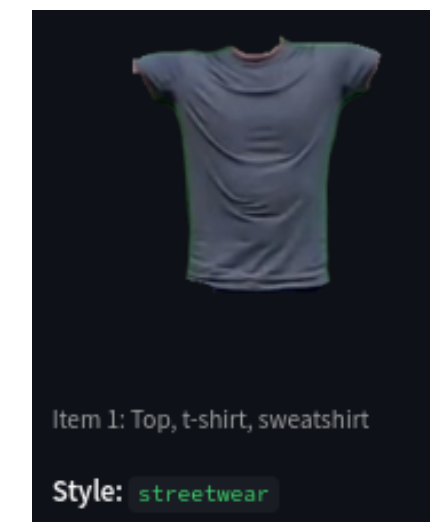
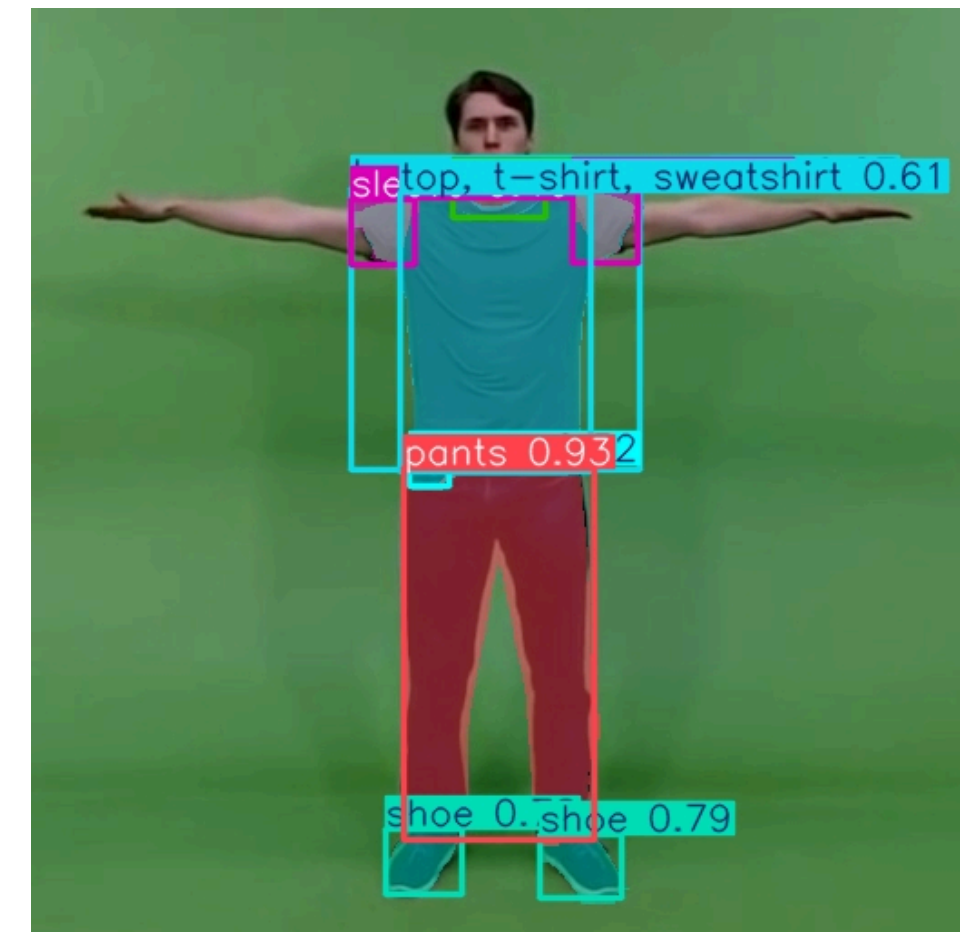
**Segmentación:** YOLOv11 nano

**Post-procesamiento:**

- Fusionar mangas y cuello con parte superior.
- Unificación de pares: zapatos, calcetines...
- Centrado y escalado de cada prenda.

**Colores predominantes:**

- Extraer píxeles visibles.
- Aplicar K-Means para agrupar colores.
- Se obtienen los colores predominantes (centroides).



# Arquitectura del sistema

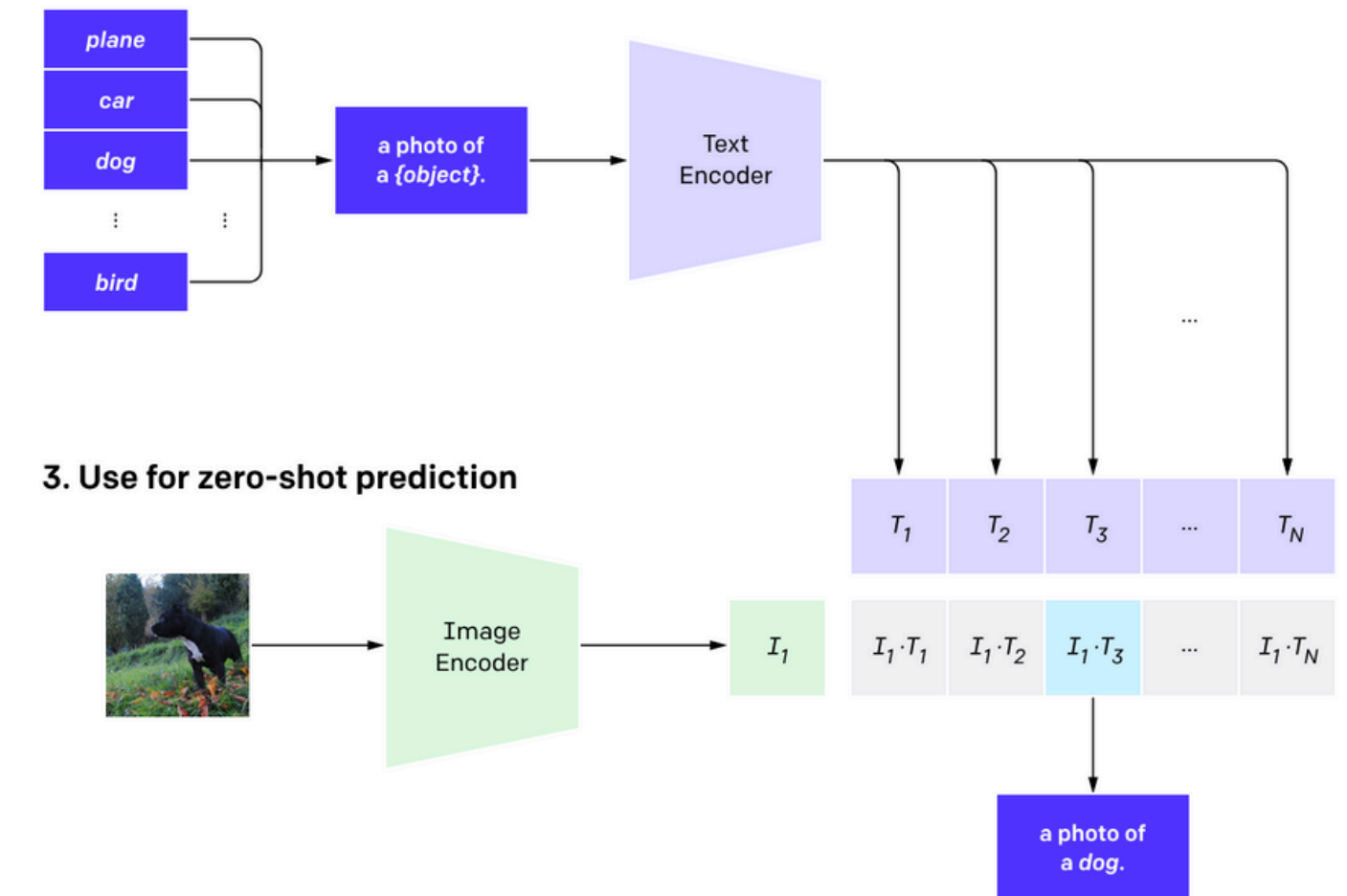
## Estilos:

- Conjunto de **etiquetas predefinidas**.
- Modelo CLIP: **zero-shot** classification.
- Se obtiene una probabilidad para cada estilo: (ej. casual, vintage, sporty)

## Recomendación de outfits:

- Se identifica la categoría de la prenda.
- Se determinan los grupos complementarios.
- Se filtran las prendas por grupo.
- Se calcula la **similitud estilo-color**.
- Se recomiendan las prendas con mayor similitud.

### 2. Create dataset classifier from label text



$$\text{sim}(t, i) = \alpha \cdot \cos(\vec{s}_t, \vec{s}_i) + \beta \cdot \cos(\vec{c}_t, \vec{c}_i)$$

# Diseño experimental

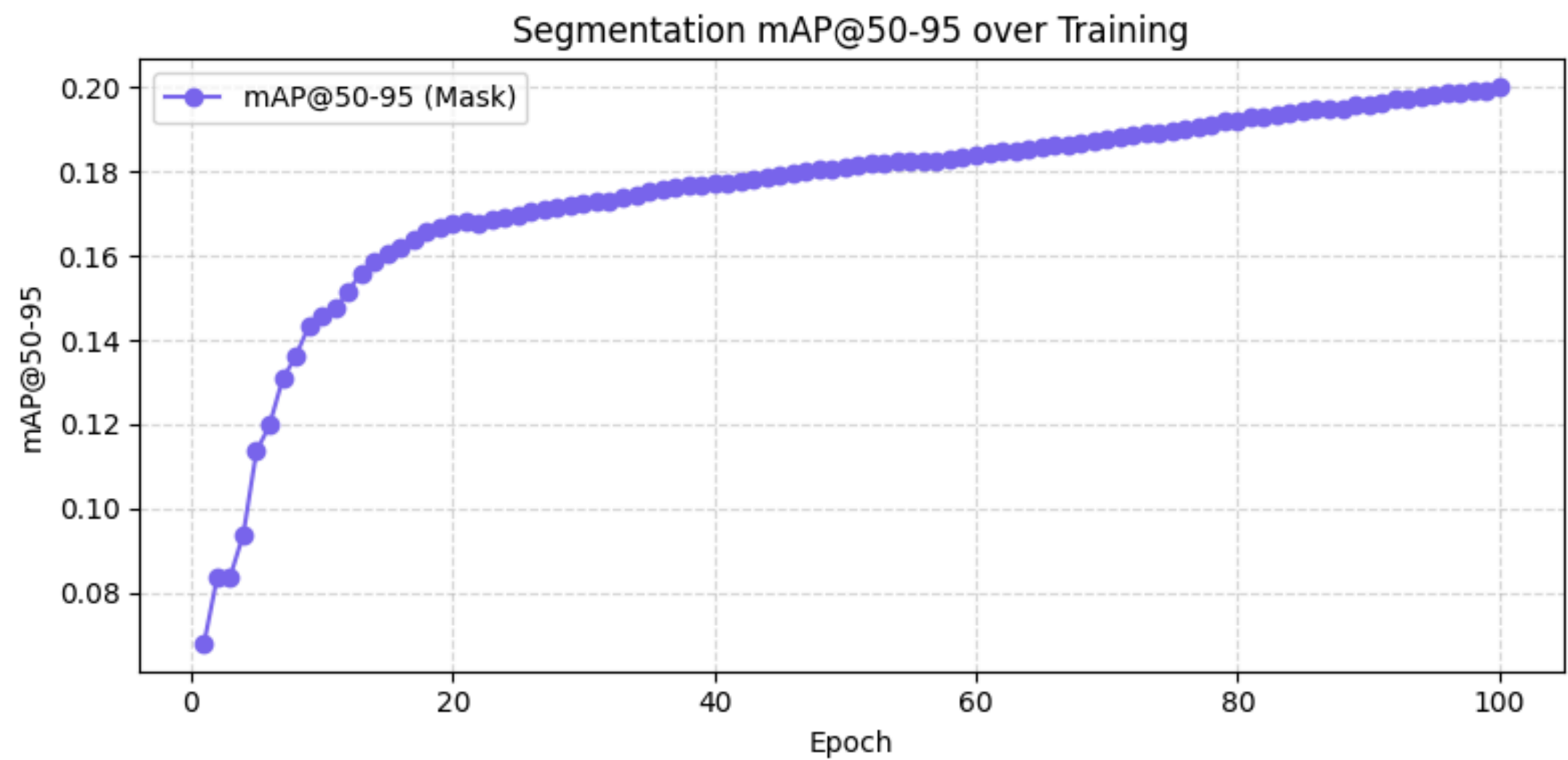
---

- **Fine-tuning** YOLO11n: preentrenado con COCO
- 95% Train, 5% Validación
- Conjunto de test no disponible.
- Optimizador SGD(LR = 0.01, momentum = 0.9)
- 100 epochs  $\approx$  9 horas



# Resultados

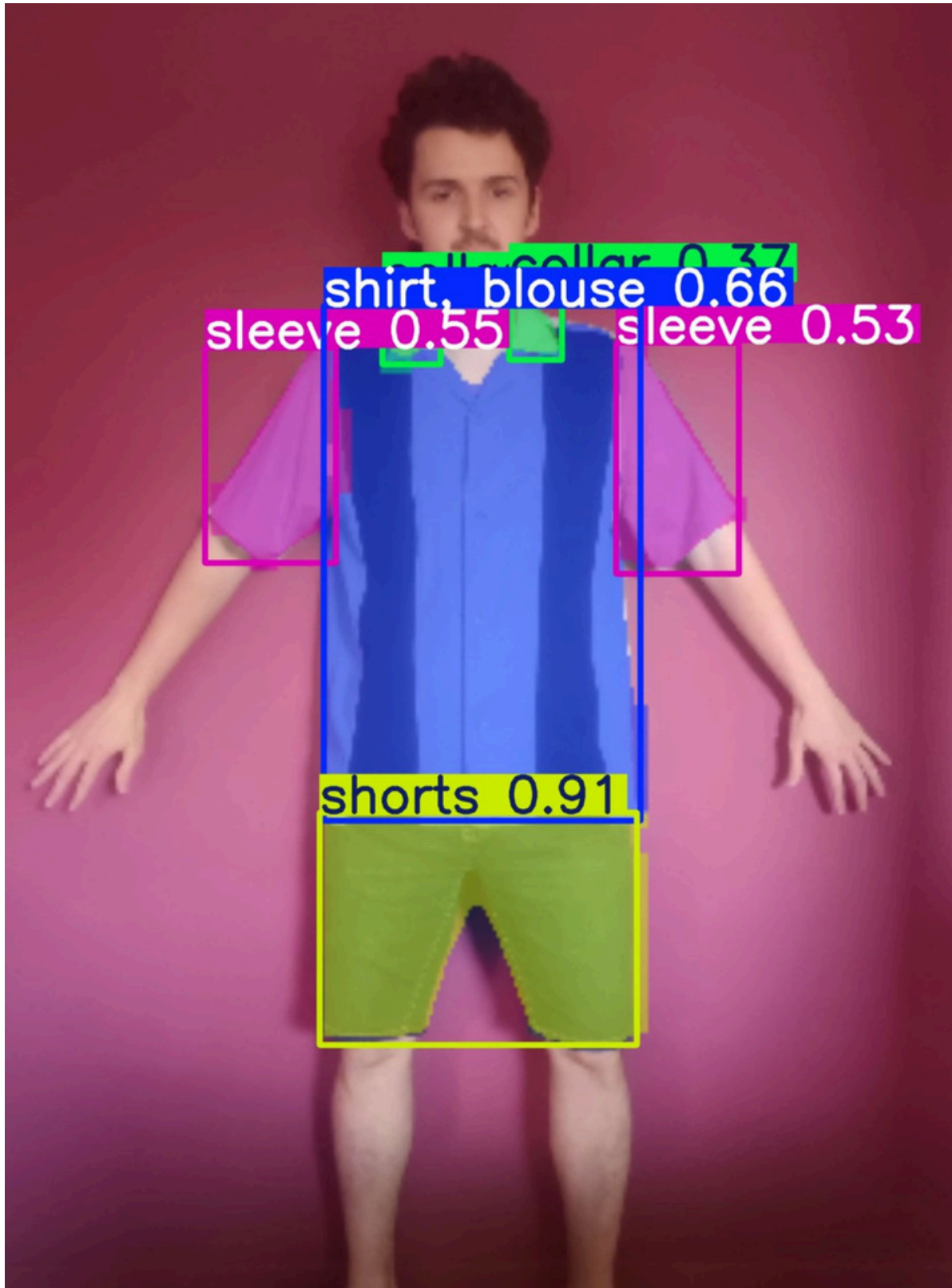
---



Epoch	mAP@50	mAP@50-95	Precission	Recall
100	0.3132	0.2000	0.5194	0.3314

# Análisis de los resultados.

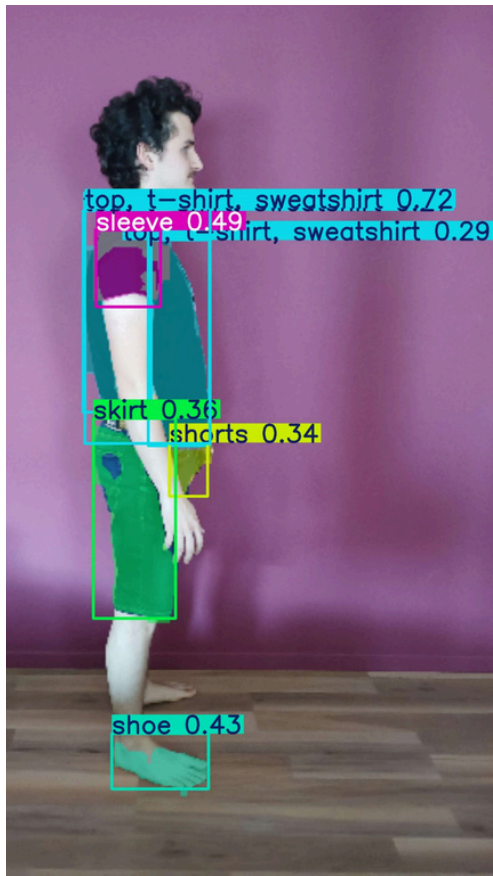
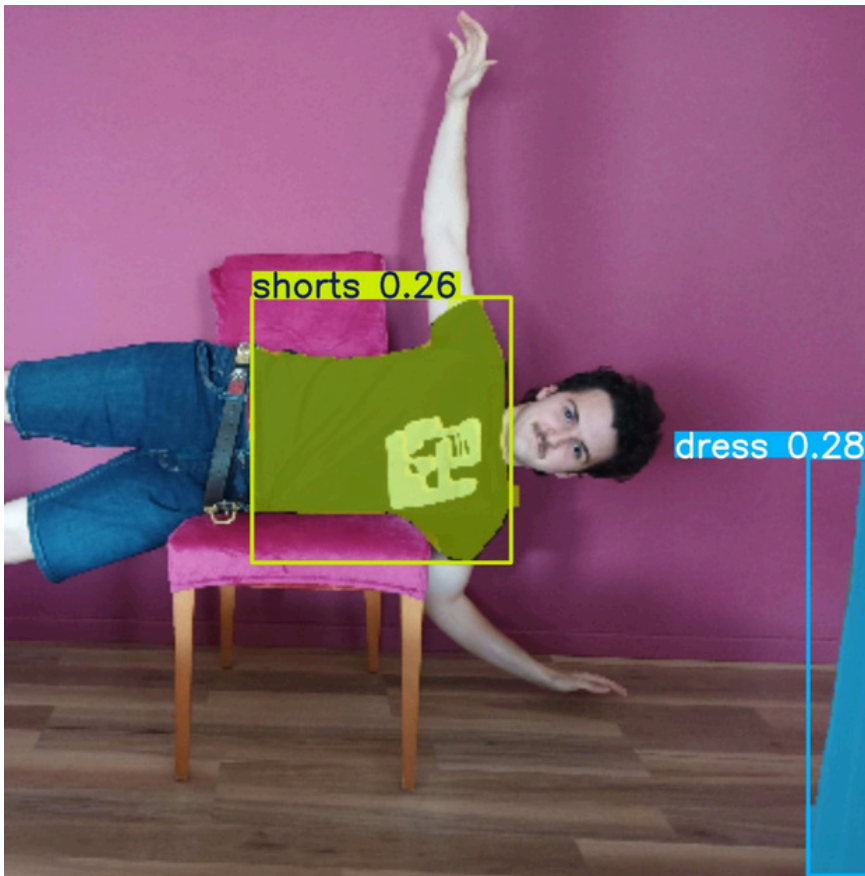
---



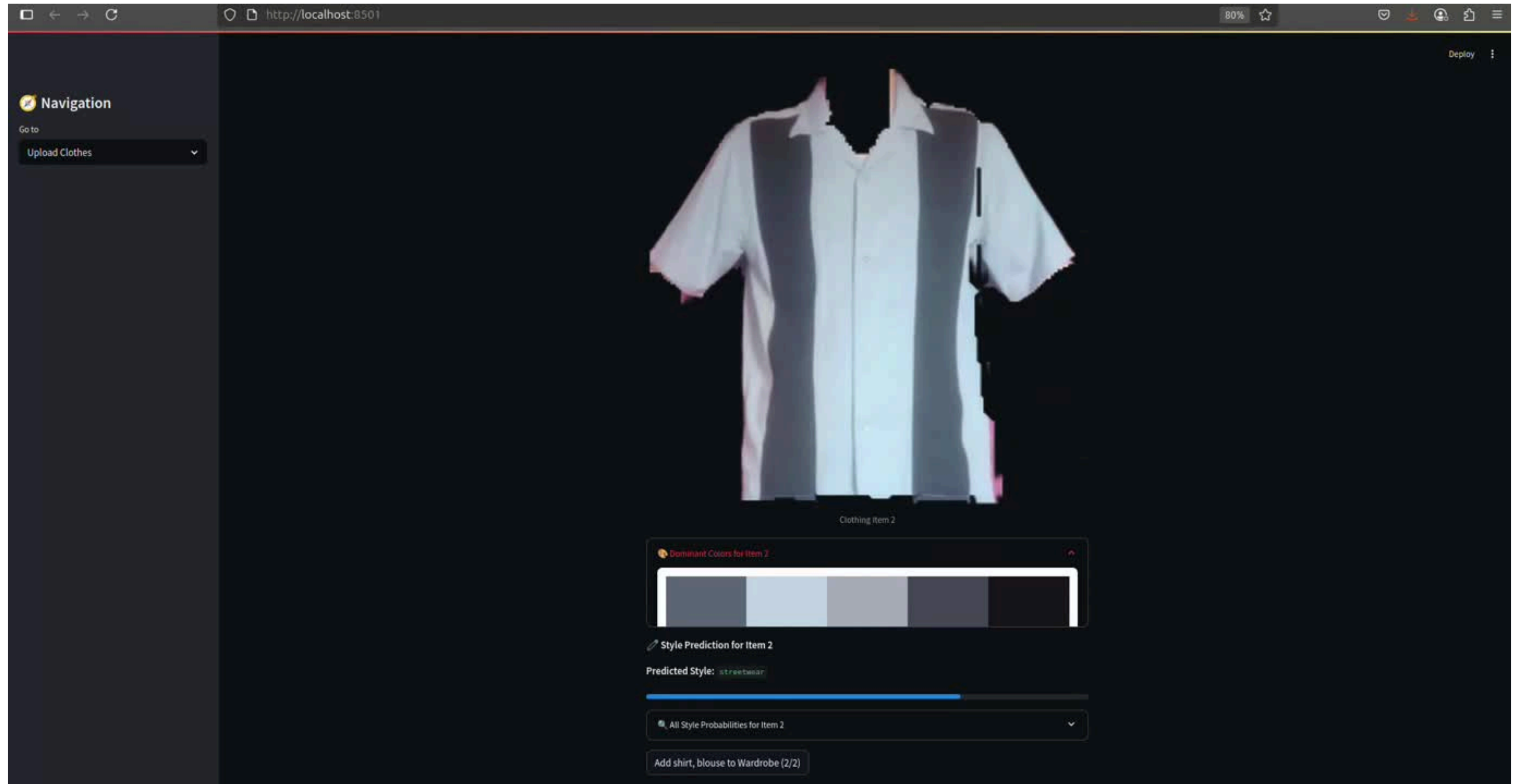
Resultados correctos  
Correcta pose

## Limitaciones

---



# Demo



# Conclusión

---

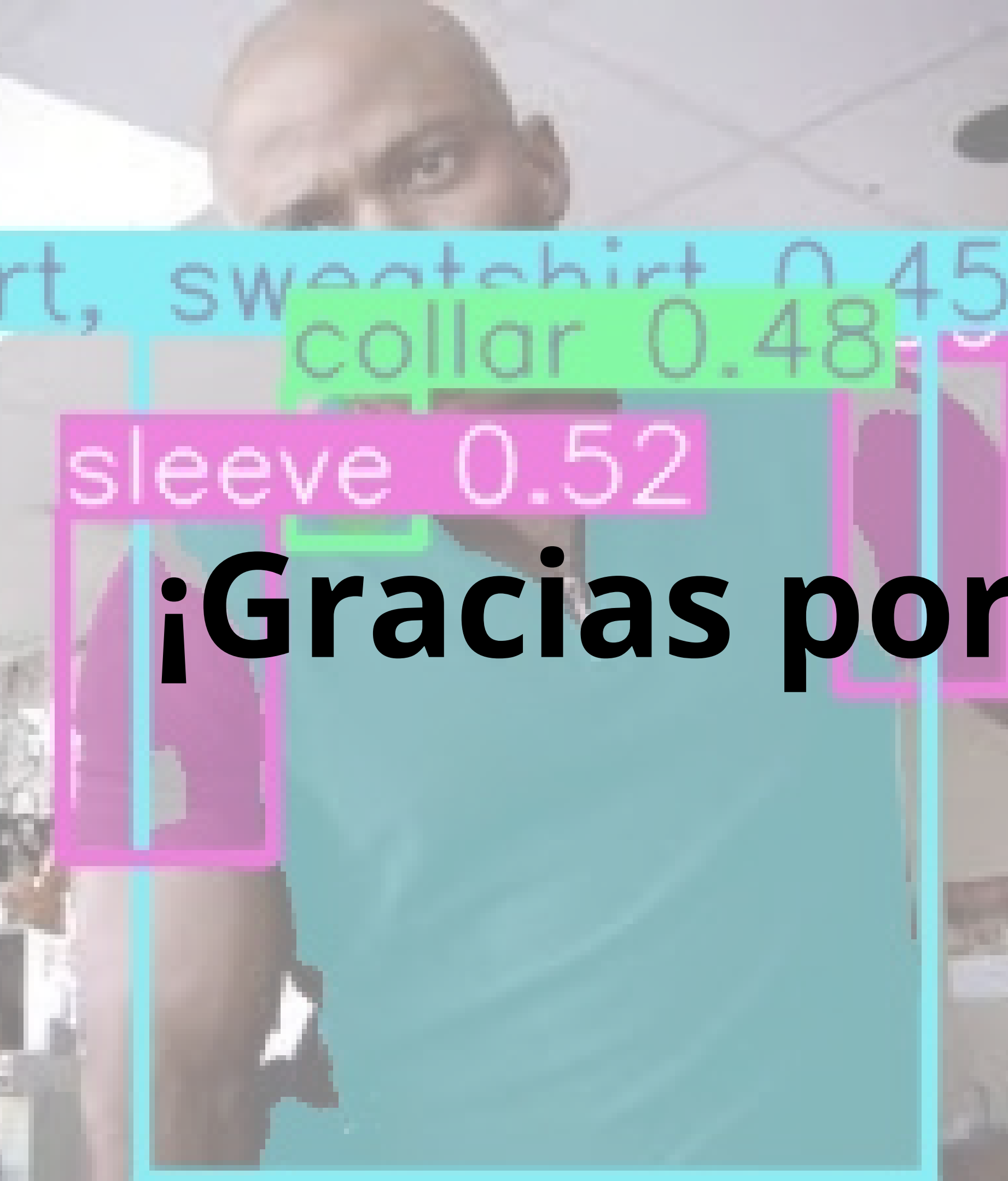
- Trabajo sobre tarea **iMaterialist**, de segmentación de ropa.
- Empleo de modelo **Yolo 11 Nano** y **CLIP**.
- Resultados buenos dependiendo de foto y pose.
- Creación de un armario inteligente basado en esta tarea.

# Trabajo futuro

---

- Proporcionar recomendaciones de piezas **fuera del armario** (catálogo).
- Detección mejorada para **poses** no estándar.
- Mejorar la **precisión** del modelo: mayor tamaño.
- Emplear **embeddings visuales** para la recomendación de outfits.
- Entrenar un modelo para clasificar estilos.
- Visualizar la ropa sobre una imagen del usuario.





**¡Gracias por su atención!**