

Universidad Las Palmas de Gran Canaria

FACULTAD DE INGENIERÍA INFORMÁTICA

ASESOR INTELIGENTE DE ROPA EN FUNCIÓN DEL CLIMA

Autora:
Andrea Santana López

Enero 2026

Índice general

0.1.	Introducción	2
0.2.	Motivación	3
0.3.	Objetivos del proyecto	4
0.3.1.	Objetivo general	4
0.3.2.	Objetivos específicos	4
0.4.	Descripción técnica del trabajo realizado	6
0.5.	Descripción técnica del sistema	6
0.5.1.	Arquitectura del sistema	6
0.5.2.	Segmentación semántica de prendas	6
0.5.3.	Clasificación de prendas de vestir	7
0.5.4.	Clasificación de calzado	7
0.5.5.	Obtención de contexto	7
0.5.6.	Sistema de recomendación	8
0.5.7.	Interfaz gráfica de usuario	8
0.5.8.	Servidor de recomendación de vestimenta	8
0.5.9.	Modelo de datos de entrada	9
0.5.10.	Integración con modelos de lenguaje (Ollama)	9
0.5.11.	Gestión de errores y tolerancia a fallos	10
0.5.12.	Formato de respuesta	10
0.6.	Fuentes y tecnologías utilizadas	11
0.7.	Conclusiones y propuestas de ampliación	12
0.8.	Indicación de herramientas/tecnologías con las que les hubiera gustado contar	13
0.9.	Diario de reuniones	14
0.10.	Creditos materiales no originales del grupo	15
0.10.1.	Modelos de aprendizaje profundo	15
0.10.2.	APIs y servicios externos	15
0.10.3.	Librerías y frameworks de software	16
0.11.	Bibliografía	17

0.1. Introducción

Gracias a la gran ayuda de la humanidad nuestro planeta sufré provocando muchos problemas como los siguientes: 1. La perdida de especies 2. El calentamiento global 3. La contaminación de agua y aire 4. La deforestación y degradación de ecosistemas 5. la sobreexplotación de los recursos primarios 6. El cambio climático

Este último es el que muchas personas están notando ya que a la hora de decidir el conjunto de prendas y accesorios que van a llevar no solo piensan en el ambiente que van a estar, con quien van a estar, si me quedará bien por el físico, etc, sino que piensan el clima que podría estar y claro aunque los meteorólogos intenten hacer predicciones precisas gracias a las ayudas humanas estos no logran grandes precisiones provocando que la gente este muy indecisa antes de salir.

Estas indeciones hacen que muchas personas no quieran salir a la calle o que les de igual que llevar y acaben poniéndose enfermos o que lleven demasiadas prendas o accesorios para cuidarse del frío, calor, lluvia o nieve o cualquier tipo de clima. Es por tanto que se me ha ocurrido desarrollar una aplicación móvil donde, los usuarios al sacarse una foto puedan ver si las prendas que llevan puestas están bien para salir.

0.2. Motivación

La motivación de este proyecto surgió cuando se observó que muchos adolescentes no prestan atención a las indicaciones del clima que dice los meteorólogos en las cadenas de noticias. Este suceso ha provocado la inspiración de crear este asesor inteligente de vestimenta en función del clima para que sepan si sus prendas son adecuadas para todo el día o para el tiempo en que estén en el instituto o para cualquier tipo de ocasión en las que vayan a salir.

0.3. Objetivos del proyecto

El objetivo principal del presente proyecto es desarrollar un sistema inteligente capaz de analizar la vestimenta de una persona a partir de una imagen y generar recomendaciones personalizadas de outfit en función del contexto climático y temporal, utilizando técnicas de visión por computador, aprendizaje profundo y modelos de lenguaje.

0.3.1. Objetivo general

Diseñar e implementar un asesor inteligente de vestimenta que integre segmentación y clasificación de prendas, obtención de información contextual en tiempo real y generación automática de recomendaciones mediante modelos de lenguaje.

0.3.2. Objetivos específicos

- Implementar un sistema de segmentación semántica que permita identificar y localizar prendas de vestir en imágenes de personas.
- Desarrollar un módulo de clasificación de prendas de ropa capaz de determinar el tipo de vestimenta presente en cada región segmentada.
- Implementar un sistema de clasificación de calzado que identifique el tipo de zapato utilizando modelos especializados.
- Integrar servicios externos para la obtención automática de la ubicación geográfica del usuario.
- Obtener información meteorológica en tiempo real a partir de las coordenadas geográficas del usuario.
- Incorporar la hora actual como variable contextual para mejorar la personalización de las recomendaciones.
- Diseñar e implementar un servidor backend que centralice la lógica de recomendación mediante una API REST.
- Integrar un modelo de lenguaje de gran escala para la generación de recomendaciones textuales coherentes y contextuales.
- Desarrollar una interfaz gráfica interactiva que permita al usuario cargar imágenes y visualizar los resultados del análisis.

- Garantizar la robustez del sistema mediante la gestión de errores y tolerancia a fallos en servicios externos.

0.4. Descripción técnica del trabajo realizado

0.5. Descripción técnica del sistema

El sistema desarrollado implementa un asesor inteligente de outfits basado en técnicas de visión por computador, aprendizaje profundo y obtención de información contextual en tiempo real. A partir de una imagen de entrada, el sistema es capaz de segmentar prendas de vestir, clasificar los distintos tipos de ropa y calzado, obtener información meteorológica según la ubicación del usuario y generar una recomendación personalizada.

0.5.1. Arquitectura del sistema

La arquitectura del sistema se organiza en un pipeline modular compuesto por los siguientes componentes:

- Segmentación semántica de prendas
- Clasificación de ropa
- Clasificación de calzado
- Obtención de contexto (ubicación, clima y hora)
- Sistema de recomendación
- Interfaz gráfica de usuario

Cada módulo opera de forma secuencial y desacoplada, facilitando la escalabilidad y el mantenimiento del sistema.

0.5.2. Segmentación semántica de prendas

Para la segmentación de las prendas se emplea un modelo de aprendizaje profundo basado en la arquitectura *SegFormer B2*, entrenado específicamente para la detección de ropa:

`mattmdjaga/segformer_b2_clothes`

El modelo realiza una segmentación semántica a nivel de píxel, generando un mapa de clases que identifica distintas categorías de prendas como parte superior, pantalones, faldas, vestidos y calzado. Posteriormente, el mapa de segmentación se reescalía al tamaño original de la imagen y se visualiza mediante una superposición de colores que facilita la interpretación de los resultados.

0.5.3. Clasificación de prendas de vestir

Las regiones correspondientes a prendas de ropa, obtenidas a partir de la segmentación semántica, se recortan de la imagen original y se procesan individualmente mediante un modelo de clasificación de imágenes:

`wargoninnovation/wargon-clothing-classifier`

Este modelo predice la categoría de cada prenda y proporciona una probabilidad asociada que indica el grado de confianza de la predicción.

0.5.4. Clasificación de calzado

Las regiones identificadas como calzado se procesan utilizando un modelo específico basado en la arquitectura *SigLIP*:

`prithivMLmods/shoe-type-detection`

El sistema clasifica el tipo de calzado en una de las siguientes categorías:

- Cholas/Bailarinas
- Botas
- Mocasines
- Zuecos
- Deportivas

Como resultado, se obtiene la clase más probable junto con su correspondiente porcentaje de confianza.

0.5.5. Obtención de contexto

Ubicación geográfica

La ubicación del usuario se determina automáticamente mediante la API `ipinfo.io`, a partir de la dirección IP. Se obtienen datos de ciudad, región y coordenadas geográficas, los cuales se traducen al español para su presentación en la interfaz.

Información meteorológica

La información meteorológica se obtiene en tiempo real mediante la API *Open-Meteo*, utilizando las coordenadas geográficas previamente obtenidas. Los datos recuperados incluyen la temperatura actual, la velocidad del viento y el estado del clima.

Hora actual

La hora del sistema se utiliza como variable contextual adicional para mejorar la personalización de la recomendación final.

0.5.6. Sistema de recomendación

El sistema de recomendación se implementa como un servicio externo accesible mediante una API REST desarrollada con python nativo. A dicho servicio se le envían la ubicación del usuario, las prendas detectadas, la hora actual y la temperatura. Como respuesta, el sistema recibe una recomendación textual personalizada acorde al contexto climático y la vestimenta identificada.

0.5.7. Interfaz gráfica de usuario

La interacción con el sistema se realiza mediante una interfaz gráfica desarrollada con la librería Gradio. La interfaz permite la carga de imágenes, la visualización de la segmentación de prendas, una galería con las prendas detectadas, información meteorológica y la recomendación final de outfit.

0.5.8. Servidor de recomendación de vestimenta

El sistema incorpora un servidor de recomendación independiente que actúa como backend para la generación de sugerencias de outfit. Este servidor se implementa utilizando la librería estándar `http.server` de Python, evitando dependencias externas y facilitando su despliegue en entornos locales o controlados.

El servidor expone una API REST ligera que escucha en el puerto 8000 y gestiona las siguientes rutas:

- `/`: endpoint informativo que confirma el funcionamiento del servicio.
- `/health`: endpoint de monitorización que devuelve el estado del servidor y una marca temporal.

- `/recommend`: endpoint principal encargado de generar recomendaciones de vestimenta.

La comunicación entre el sistema de análisis visual y el servidor se realiza mediante peticiones HTTP POST en formato JSON.

0.5.9. Modelo de datos de entrada

Para estructurar la información recibida, el servidor define un objeto de tipo `ClothingRequest`, que encapsula las siguientes variables contextuales:

- Localización del usuario
- Prendas detectadas en la imagen
- Hora actual
- Temperatura

Este modelo de datos permite un manejo coherente y extensible de la información utilizada para la generación de recomendaciones.

0.5.10. Integración con modelos de lenguaje (Ollama)

La generación de recomendaciones se realiza mediante la integración con un modelo de lenguaje de gran escala (LLM) desplegado localmente a través de la plataforma Ollama. El servidor se comunica con Ollama utilizando una API compatible con el formato de *chat completions*.

El modelo utilizado es:

`deepseek-v3.1:671b-cloud`

La solicitud enviada al modelo incluye un mensaje de sistema que define el rol del asistente como experto en clima, hora y vestimenta, así como un mensaje de usuario que contiene el contexto completo de la situación del usuario.

El prompt está diseñado para que el modelo:

- Evalúe si la vestimenta es adecuada para el clima y la hora
- Indique si el usuario va demasiado abrigado o con poca ropa
- Sugiera posibles cambios sin describir explícitamente el outfit actual

0.5.11. Gestión de errores y tolerancia a fallos

El servidor implementa mecanismos de manejo de errores para garantizar la continuidad del servicio. En caso de que la comunicación con Ollama falle, el sistema genera una respuesta alternativa simulada basada en reglas simples y en la información contextual disponible.

0.5.12. Formato de respuesta

La respuesta del servidor se devuelve en formato JSON e incluye los siguientes campos:

- Identificador único de la recomendación
- Marca temporal de generación
- Modelo utilizado
- Recomendación textual generada

Esta estructura facilita la integración con la interfaz gráfica y con otros posibles consumidores del servicio.

0.6. Fuentes y tecnologías utilizadas

Las principales tecnologías y herramientas empleadas en el desarrollo del proyecto son:

- Python
- PyTorch
- Hugging Face Transformers
- APIs REST
- Gradio

0.7. Conclusiones y propuestas de ampliación

Como conclusión del proyecto se ha pensado que podría ayudar a gente a decidir más rápido sus prendas y accesorios para salir sin preocuparse de ver noticias o si eres muy depistado. En cuanto a propuestas de ampliación sería mejorar el diagnostico cambiando el modelo de segmentación semántica por instancia,modulo de voz para personas no puedan ver el texto de la recomendación,mejoras de rendimiento del servidor con el LLM

0.8. Indicación de herramientas/tecnologías con las que les hubiera gustado contar

Me hubiera gustado tener un mejor equipo y prestaciones de GPU,CPU y RAM.

0.9. Diario de reuniones

Todos los días desde que se empezo la entrega de 10 a.m a 10.15 a.m

0.10. Creditos materiales no originales del grupo

En el desarrollo del presente proyecto se han utilizado diversos recursos, modelos y servicios externos que no han sido desarrollados por el grupo, y cuyo uso ha sido fundamental para la implementación del sistema. A continuación, se detallan dichos materiales junto con su finalidad dentro del proyecto.

0.10.1. Modelos de aprendizaje profundo

- **SegFormer B2 para segmentación de ropa:** Modelo de segmentación semántica entrenado para la detección de prendas de vestir a nivel de píxel. Fuente: [mattmdjaga/segformer_b2_clothes](https://huggingface.co/mattmdjaga/segformer_b2_clothes) (Hugging Face).
- **Modelo de clasificación de prendas de vestir:** Modelo de clasificación de imágenes utilizado para identificar el tipo de prenda a partir de regiones segmentadas. Fuente: [wargoninnovation/wargon-clothing-classifier](https://huggingface.co/wargoninnovation/wargon-clothing-classifier) (Hugging Face).
- **Modelo de clasificación de calzado basado en SigLIP:** Modelo especializado en la detección de tipos de calzado. Fuente: [prithivMLmods/shoe-type-detection](https://huggingface.co/prithivMLmods/shoe-type-detection) (Hugging Face).
- **Modelo de lenguaje de gran escala (LLM):** Modelo utilizado para la generación de recomendaciones de vestimenta en función del contexto climático y temporal. Fuente: deepseek-v3.1:671b-cloud, desplegado localmente mediante Ollama.

0.10.2. APIs y servicios externos

- **API de geolocalización por IP:** Servicio utilizado para obtener la ciudad, región y coordenadas geográficas del usuario a partir de su dirección IP. Fuente: ipinfo.io.
- **API meteorológica Open-Meteo:** Servicio empleado para la obtención de información meteorológica en tiempo real, incluyendo temperatura, viento y estado del clima. Fuente: open-meteo.com.
- **API de modelos de lenguaje (Ollama):** Plataforma utilizada para la ejecución local de modelos de lenguaje y la generación de texto mediante una API compatible con *chat completions*. Fuente: ollama.ai.

0.10.3. Librerías y frameworks de software

- **PyTorch:** Framework de aprendizaje profundo utilizado para la ejecución de los modelos de segmentación y clasificación.
- **Hugging Face Transformers:** Librería empleada para la carga y uso de modelos preentrenados de visión por computador y clasificación de imágenes.
- **Gradio:** Librería utilizada para el desarrollo de la interfaz gráfica de usuario interactiva.
- **Requests:** Librería empleada para la comunicación HTTP con APIs externas y servicios REST.
- **NumPy y PIL:** Librerías utilizadas para el procesamiento y manipulación de imágenes.

0.11. Bibliografía

- <https://huggingface.co/>
- <https://www.gradio.app/>