

Práctica 2ª evaluación

1. Descripción del Proyecto

El proyecto consistirá en el desarrollo de un sistema distribuido usando Docker que pueda recibir imágenes, detectar rostros en ellas, determinar la edad de las personas en los rostros detectados y, si son menores de 18 años, pixelar sus rostros.

Para ello se van a crear 2 contenedores en docker que realizarán la tarea y usar 2 IAs

Este proyecto va a ser desarrollado por los alumnos en grupos de 3 a 4 alumnos. Cada grupo desarrollará el mismo proyecto pero de forma independiente de otros grupos.

Los integrantes de cada grupo son los siguientes:

1. Grupo
 - a. Julian Benito Sanchez
 - b. Javier Muelas
 - c. Joan Climent Sania
 - d. Raul Robla
2. Grupo
 - a. Ali Vafaev
 - b. Tomás Bastante
 - c. Jesus Domingo
3. Grupo
 - a. Moisés Guerola
 - b. Joaquim Ribera
 - c. Fernando Ezequiel Vera
4. Grupo
 - a. Adrián Ruiz
 - b. Juan Vicente Villanueva
 - c. Jesús Zambrano

2. Contenedores Docker

Los contenedores Docker son los siguientes:

- API Gateway Container: Su función es exponer publicamente el API para su uso por los clientes y enrutar las peticiones al "IA Container" que finalmente realizará la tarea.

Su tarea será la de aceptar una imagen, enviar la imagen al "IA Container" , y devolver la imagen que retorne el "IA Container"

- IA Container: Este contenedor no será accesible desde internet sino que lo podrá llamar únicamente el API Gateway Container.

Sus funciones serán las siguientes:

- Recoger la petición con la imagen enviada desde el “API Gateway Container”
- Llamar a la “IA Caras” que nos diga las regiones donde hay caras
- Recortar cada una de las caras de la imagen original ,creando una nueva imagen con cada cara.
- Llamar a la “IA Menor Edad” para que nos diga si cada cara es de un menor de edad.
- En caso de que cada cara sea un menor de edad , pixelar la correspondiente cara en la imagen original.
- Devolver la imagen con las caras de los menores pixeladas.
- Generar un fichero de log con:
 - Instante en el que se hizo la petición
 - Para cada cara:
 - Dónde está cada cara (coordenadas)
 - El resultado de cada cara: Si es menor de edad o no
 - El tiempo(ms) que se tarda en ejecutar la llamada a “IA Caras”
 - El tiempo(ms) que se tarda en ejecutar “IA Menor edad”
 - El tiempo(ms) total de la petición

Crear un “docker-compose” para definir y orquestar todos los servicios. Haciendo redes privadas para la comunicación interna excepto en el API Gateway.

3. IAs

Las IAs usadas son las siguientes:

- IA Caras: Esta IA es la encargada de detectar las caras en una foto. Para ello se usará una IA ya entrenada llamada Yolov8. Más información en: <https://github.com/ultralytics/ultralytics?tab=readme-ov-file>
- IA Menor Edad : Esta IA hay que crearla, y está especificado en el siguiente apartado.

4. Desarrollo “IA Menor Edad”

Para el desarrollo de esta IA encargada de ver si alguien es menor de edad según su cara en una foto; se va a tener que entrenar desde el principio usando los siguientes datos de entrenamiento:

<https://www.kaggle.com/datasets/frabbisw/facial-age>

Los datos de esta página no son según si es o no menor de edad sino que dicen la edad de una persona pero habrá que transformarlos para que simplemente clasifiquen la foto según sea o no menor de edad (menor de 18 años)

La IA tendrá la siguiente estructura:

- Emplee una imagen de entrada de 200x00.
- Esté formado por 4 bloques convolucionales. Cada bloque convolucional estará formado por las siguientes capas:
 - 2 filtros convolucionales con un kernel de 3x3, padding same y función de activación ReLU. El número de filtros se establecerá a 32, 64, 128 y 256 en el 1º, 2º, 3º y 4º bloque convolucional, respectivamente.
 - 1 capa de agrupación máxima de forma que se reduzca el tamaño de la imagen a la mitad.
- Convierta la salida del último bloque convolucional a un vector empleando una capa GlobalAveragePooling.
- Emplee una capa densa oculta de 32 neuronas y función de activación ReLU.
- Tenga una capa de salida considerando que estamos ante un problema de clasificación binaria.

Para la estructura que acabamos de ver, **crea 12 modelos distintos** cambiando:

- la tasa de aprendizaje (debes probar con 3 distintas) ,
- tamaño de batch (debes probar con 2 distintos)
- Con y sin BatchNormalization

Entrena los 12 modelos de forma que ahorres en recursos computacionales.

Para cada modelo, muestra las siguientes gráficas:

- La pérdida en entrenamiento y validación en función de las épocas.
- La sensibilidad y la especificidad en entrenamiento y validación en función de las épocas.

Crea además estas 2 gráficas

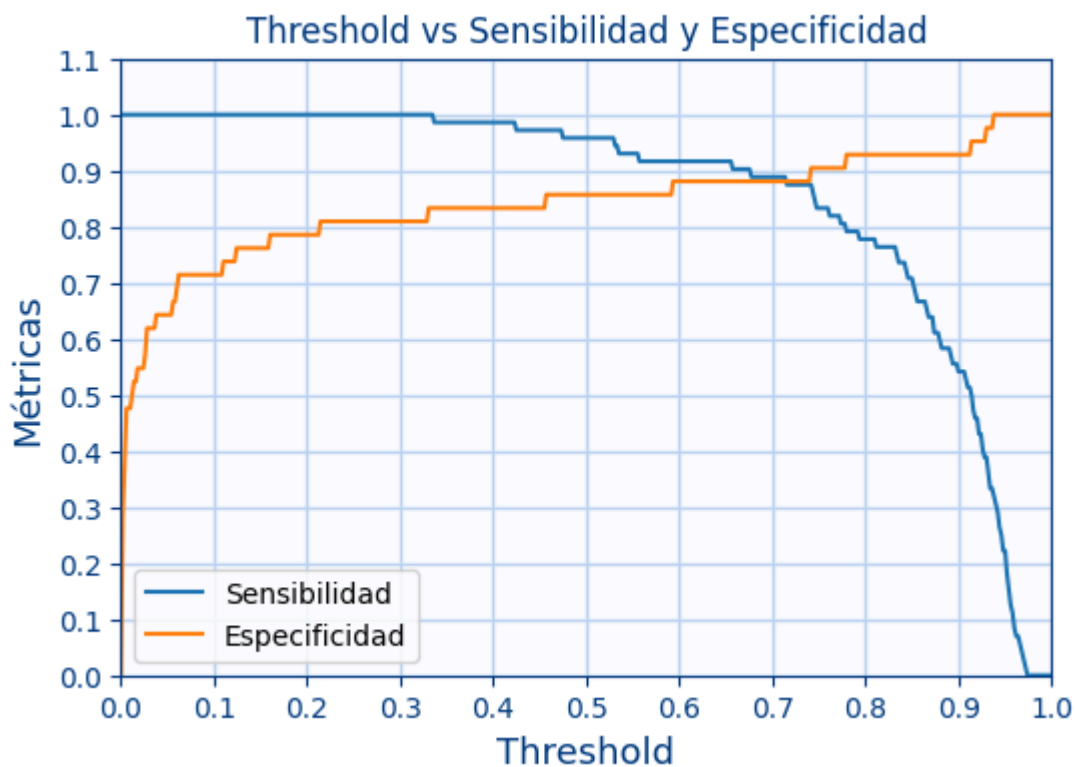
- En una gráfica , muestra para todos los modelos y la mejor época; la métrica de sensibilidad con el intervalo de confianza
- En una gráfica , muestra para todos los modelos y la mejor época; las métricas de especificidad con el intervalo de confianza

En función de todas las gráficas anteriores elige el modelo que vas a usar . Y justifica porqué has elegido ese modelo. **¡Es muy importante la justificación !!!**

Ahora entrena las épocas que sea necesario el modelo para conseguir el mejor rendimiento posible.

Muestra las siguientes gráficas para el modelo

- Muestra una gráfica con la pérdida en entrenamiento y validación en función de las épocas.
- Muestra una gráfica con la sensibilidad y la especificidad en entrenamiento y validación en función de las épocas
- Muestra una gráfica en la que se muestre la sensibilidad y especificidad según el umbral. La gráfica es similar a la siguiente



En base a esta gráfica y según el problema que se está intentando solucionar (el pixelado de caras de menores) , elige el umbral más adecuado. **!!Y justifica tu respuesta!!!!**

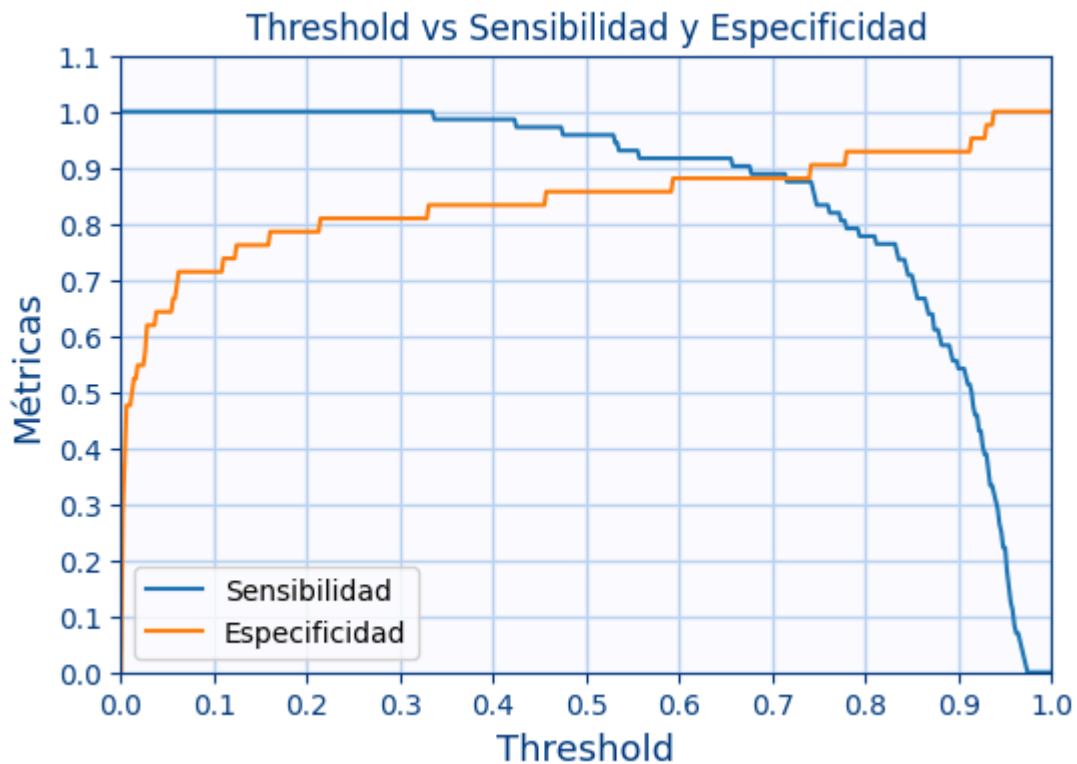
Ahora vas a crear de nuevo el modelo empleando transferencia de conocimiento. Para ello emplea el modelo preentrenado MobileNetV2. Para realizar transferencia de conocimiento tienes que realizar las fases tanto de extracción de características como de ajuste fino.

Ahora entrena las épocas que sea necesario el modelo para conseguir el mejor rendimiento posible.

Muestra las siguientes gráficas para el modelo

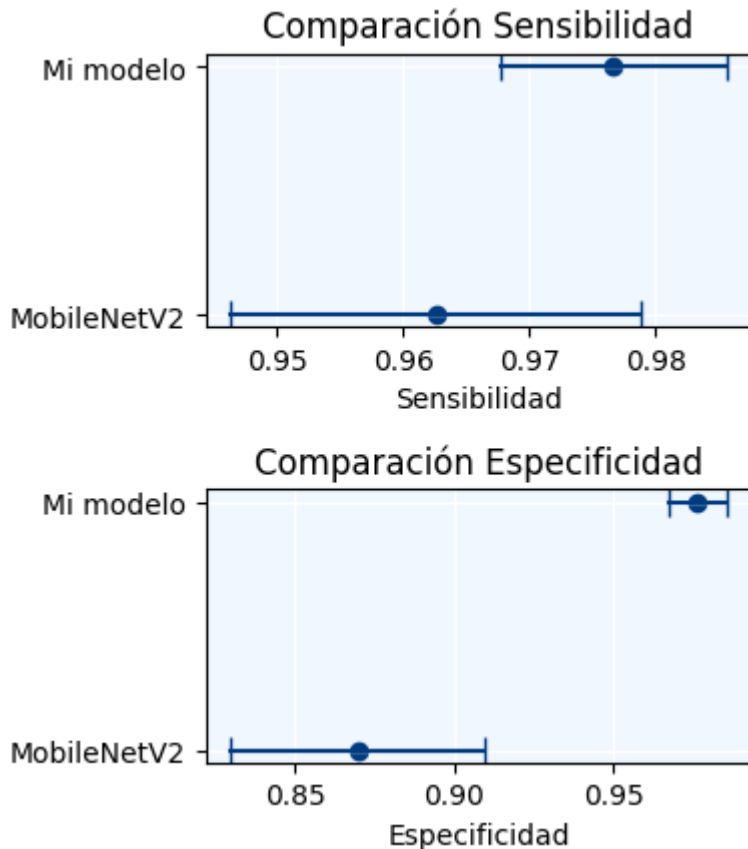
- Muestra una gráfica con la pérdida en entrenamiento y validación en función de las épocas.
- Muestra una gráfica con la sensibilidad y la especificidad en entrenamiento y validación en función de las épocas

- Muestra una gráfica en la que se muestre la sensibilidad y especificidad según el umbral. La gráfica es similar a la siguiente



En base a esta gráfica y según el problema que se está intentando solucionar (el pixelado de caras de menores) , elige el umbral más adecuado. **!!Y justifica tu respuesta!!!!**

Compara ahora los 2 modelos usando las métricas de sensibilidad y especificidad para el umbral elegido. Para ello crea una gráfica similar a la siguiente:



En función de todas las gráficas anteriores elige el modelo que vas a usar . Y justifica porqué has elegido ese modelo. **¡Es muy importante la justificación !!!**

Por último , muestra otra vez la gráfica pero solo para fotos de personas que tienen entre 16 y 17 años. ¿Qué conclusión sacas?

5. Entregables

En un ordenador de algún alumno se instalará todo el proyecto. El día de la presentación se borrarán todos los contenedores y se ejecutará “docker compose up -d” y ya deberá funcionar el proyecto.

En moodle se deberá entregar un zip con:

- El fichero “docker-compose.yaml “
- El código fuente usado en cada uno de los contenedores
- El fichero con la arquitectura de la “IA Menor Edad”
- El jupyter notebook con todos los pasos del desarrollo de la “IA Menor Edad”
- Cualquier otro fichero usado en la creación del proyecto. (Excepto los datos de entrenamiento de la IA)

La fecha de entrega será el 29 de mayo a las 4:00 de la madrugada.

6. Evaluación

El miércoles 29 de mayo se hará una presentación del proyecto por parte de todos los componentes de cada grupo.

En esa presentación se mostrarán varios ejemplos de funcionamiento del producto. Además deberán explicar en la pizarra como funciona en proyecto y explicando las líneas de código más importantes del mismo

Por último se mostrarán las gráficas del jupyter notebook y se justificará cada una de las decisiones que había que justificar.

El jueves 30 de mayo se podrá llamar a cualquier alumno para que haga de forma privada al profesor de nuevo la explicación del proyecto. El profesor podrá preguntar sobre cualquier aspecto del proyecto y el alumno deberá saber responder de forma ágil y clara a la pregunta.

7. Calificación

Cada proyecto será calificado con una nota grupal de todo el proyecto.

Si un alumno es requerido el jueves 30 de mayo para realizar la explicación del proyecto, la nota de la evaluación de ese alumno será la nota que obtenga ese día. Si el alumno no es requerido el 30 de mayo la nota de la evaluación será la nota grupal de todo el proyecto.

Tendrá la evaluación aprobada si la nota de la evaluación es mayor o igual a 5.

Por lo tanto, la realización de esta práctica es obligatoria para aprobar la 2º evaluación y no se realizará ningún examen durante la 2º evaluación

8. Consideraciones finales

A lo largo de los días hasta el día de la entrega se pueden hacer modificaciones de este documento para aclarar los puntos que no hayan estado claros.