

Seedtag Codetest: Data Scientist

Agente Kallus,

si está leyendo este mensaje es que ha alcanzado el piso franco que El Imperio tiene instalado cerca de la base Rebelde en la ciudad de Norg Bral. Bien hecho, aunque su trabajo aquí no ha hecho más que empezar.

Como sabe, el planeta Mandalore es de una gran importancia estratégica para nosotros y tenemos que estar preparados ante cualquier movimiento de los comandos Rebeldes que se encuentran estacionados allí.

Para anticiparnos a ellos y poder defendernos eficazmente, estamos preparando una serie de incursiones, que no tendrán éxito si no inutilizamos sus sistemas de vigilancia y mejoramos los nuestros.

Detalles de la misión

Parte 1: Inhabilitar el sistema de vigilancia rebelde

Nuestro departamento de inteligencia ha sido capaz de interferir en la señal de vídeo del circuito de vigilancia de los rebeldes, de manera que podemos alterarla para hacer que el modelo que analiza la señal falle. Sin embargo, si la alteración es muy evidente, detectarán el problema fácilmente. Por ello nuestro objetivo es introducir una alteración en las imágenes que sea imperceptible a simple vista, pero que deje su sistema de vigilancia inoperativo.

Detalles:

- El script `src/model.py` descarga una réplica del modelo que analiza las imágenes en el sistema rebelde, lo guarda en la carpeta `assets` y ejecuta una clasificación de prueba.
- En la carpeta `assets` puedes encontrar una imagen de prueba. Hacer fallar el modelo en esa imagen es suficiente para dar esta parte por concluida.
- Idealmente, ninguno de los valores de los canales de RGB de cada píxel debe ser modificado en más de una unidad. Es decir, si un píxel cualquiera de la imagen original tiene valores (120, 90, 135), su valor en la versión modificada de la imagen podría ser (121, 89, 136).

Parte 2: Crear un sistema de vigilancia robusto

Para finalizar su misión nos gustaría poder dotar a nuestras instalaciones de un sistema de vigilancia preciso y robusto. Su misión será desarrollar un sistema de clasificación de imágenes con la mayor precisión posible. Adicionalmente, nos gustaría que el sistema fuese resistente a ataques como el que se detalla en la parte 1 de la misión.

Detalles:

- El objetivo es entrenar un sistema que pueda emitir una clasificación lo más precisa posible dentro de la taxonomía del dataset.
- En el script `src/dataset.py` encontrará varias maneras de obtener el dataset. Elija la que más se adapte a sus necesidades. Si se ejecuta el script, el dataset será descargado en su disco local.
- Como punto opcional, nos gustaría que el sistema fuese lo suficientemente robusto como para resistir el ataque que aplicó sobre las instalaciones rebeldes.

Resultado esperado.

- Esperamos encontrar los Notebooks o scripts que haya utilizado para la resolución del ejercicio, así como un documento (o el propio notebook) donde explique brevemente los pasos que ha seguido, por qué los siguió, y cualquier otra información analítica que considere relevante.
- Tenga en cuenta que aunque no consiga resolver completamente alguno de los puntos, cualquier análisis o fragmento de código en el que haya trabajado puede ser igualmente relevante.
- Recuerde que un correcto análisis del dataset y los resultados del modelo son esenciales para el éxito de la misión.
- Recuerde incluir cualquier instrucción que pueda ayudarnos a reproducir los resultados.
- Se incluye un fichero "requirements.txt" con las librerías mínimas para obtener los datos. Si requiere de otras herramientas, agradecemos que envíe este archivo (o equivalente para pipenv) actualizado para garantizar la reproducibilidad.

Recomendaciones.

- El tamaño de las imágenes que tomará el modelo como input en la parte 2 de la prueba es algo que dejamos a su elección. Sin embargo, tenga en cuenta que un input demasiado grande podría hacer que los entrenamientos fuesen excesivamente grandes.
- Consideramos muy valiosos los análisis del modelo, dataset, o todos aquellos que apoyen las decisiones que tome en el diseño del sistema.
- La prueba está pensada para ser resuelta en python. Recomendamos el uso de entornos virtuales, pero no es necesario.
- Si tiene dudas sobre cómo preparar su espacio de trabajo utilizando entornos virtuales, puede hacerlo ejecutando los siguientes comandos dentro de la carpeta de la prueba:

```
>> python3 -m venv venv
>> source venv/bin/activate
>> pip install -r requirements.txt
>> python src/model.py
>> python src/dataset.py
```

Instrucciones de entrega

Deberá generar un fichero zip con el nombre <username-en-github>_codetest4_seedtag.zip que contenga todos los scripts, notebooks o documentos que se deseen incluir, y enviarlo adjunto en un email a angelguirao@seedtag.com. En caso de que su sistema de correo no le permita enviar el fichero, puede renombrar la extensión a .a.

Suerte en su misión y que la fuerza le acompañe.