

Manual de funcionamiento – Maqueta Adelte

Reconocimiento de puertas de avión a través de machine learning.

Funcionamiento rapido:

1-> Nos conectamos por TeamViewer a la raspberry pi (adelterpi).

2-> Doble clic en ADELTE (esta en el escritorio dentro de la carpeta backend-websocket) , sale una ventana , clicar en ejecutar en terminal. Se abre un terminal, no cerrar ni tocar nada dejarlo abierto.

3-> esperar a que cargue y empiecen a salir las letras.

4-> abrir el navegador y clicar en la barra de favoritos Adelte-Client.

5-> pulsar en el boton verde INICIAR servidores y esperar a que salga el mensaje en la barra se ponga Servidor AWS: Iniciado, mostrara un mensaje (en color verde) , si no aparecen las letras volver a hacer el paso 2. (paciencia se tienen que cargar varias librerias de datos)

6 -> para arrancar pulsar el boton start door recognition, debe iniciarse la grabacion y el reconocimiento. Si no reconoce la puerta , bajar el brillo de la habitacion o cerrar una cortina para que no desenfoque la camara.

7-> para apagar pulsar el boton detener servidores. Y esperar a que salga el mensaje de servidores detenidos. Se puede cerrar la web si se queda Deteniendo..... (color amarillo)

Si se atasca en algun punto se puede actualizar la pagina para que funcione.

Empezara a funcionar la aplicación.

SOLUCIONAR POSIBLES ERRORES

La cuenta de AWS es compartida, la otra maqueta usa instancias en otra region, para que funcione tienen que estar en norte de virginia (us-east-1).

Puede ser que con el doble click falle la ejecucion del servidor. En ese caso estos comandos en terminal:

```
cd Documents/backend-websocket
```

```
conda activate backend-websocket
```

```
python app.py
```

IMPORTANTE NO MODIFICAR NINGUNA CONFIGURACION DE VIDEO DE LA IPCAM

IP de la cámara: 192.168.100.71

IP de la Raspberry PI: 192.168.233.7

PARA ACCEDER AL CLIENTE PONEMOS LA IP DE LA RASPBERRY EN EL NAVEGADOR

Desde aquí, iniciamos el servidor AWS con el botón y esperamos respuesta, una vez este activo ya emitirá imágenes cuando le demos a start.

Dentro de la raspberry se ejecutan 3 programas

- ➔ API en python (procesa y envía imágenes al servidor AWS, envía peticiones al servidor de procesamiento y llama a la IPCAM para tomar capturas).

Los archivos están en: Ibermatica/Documentos/backend-websocket

Se ejecuta en la ip: 127.0.0.1:8080/

- ➔ Servidor Web To RTSP que procesa el video para tenerlo en HLSII y poder streamearlo hacia la web cliente.

Esta ya configurado y se lanza a través de Supervisor, es un contenedor de docker no hay que tocarlo ni hacer nada.

Localhost puerto 8083. Esta guardado en favoritos.

- ➔ Servidor Web Apache2 donde se aloja el cliente web.

Ubicado en www/html/ servidor típico de apache2, se accede poniendo la ip de la raspberry desde cualquier equipo se puede acceder al cliente. El cliente es una web HTML5 con JS y CSS convencional, contiene un script de PHP para subir las imágenes y depende de varias librerías que hay que tener descargadas (Websockets).

Procesamiento de la imagen con el modelo Mask-RCNN

Las imágenes se procesan en las maquinas AWS EC2 que están contratadas, son servidores amazon linux con las librerías de CUDA preinstaladas y los modelos de deep learning cargados en dos servidores, están programados para que se carguen automáticamente las APIS y funcionen al arrancar.

[https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;\\$case=tags:true%5C,client:false;\\$regex=tags:false%5C,client:false;sort=tag:Name](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false;sort=tag:Name)

hay 2, una para los aviones reales y otra para los aviones de lego. Cada una se dedica a procesar un tipo de imagen.

Credenciales:

mail

showroomgestion0@gmail.com

password del mail

Showroom2022!

cuenta team viewer

showroomgestion0@gmail.com

contraseña

Sh0wr00m@2020!

Contraseña raspberry:

okoberry

Credenciales AWS

Id cuenta: 965946277713

Nombre de usuario: showroom.admin

Contraseña: G77d0l49OyeyoTgLMz3Z*