

DOCUMENTACIÓN PRUEBA TÉCNICA DESARROLLADOS DE ANALITICA JUNIOR

- Introducción y Descripción del Proyecto

Proyecto de segmentación semántica de un video, donde el cual se debe realizar segmentación semántica de al menos tres tipos de objetos en una secuencia de video, para ello se recomienda utilizar un video como el presentado en la documentación de la prueba técnica. Este video debe poder introducirlo a una red neuronal entrenada de tal forma que en el video de salida muestre con claridad la segmentación realizada de los tres tipos de objetos a identificar.

- Requisitos

Requisitos del sistema: El script desarrollado necesita del sistema tener instalado Python 3.9 en adelante y contar con las librerías o dependencias instaladas en el entorno de este (las librerías se enumeran en el siguiente ítem), este script se desarrolló para correr en un ambiente de sistema normal para no tener inconvenientes al momento de ejecutarlo.

Dependencias: Las librerías o dependencias usadas en este proyecto son las siguientes,

- a. Cv2 o OpenCV
- b. Torch de PyTorch
- c. Torchvision
- d. Numpy

- Instrucciones de Ejecución

Instalación del entorno de Python y Dependencias:

Primero realizamos la instalación de las dependencias en el sistema de Python 3.9 (video de referencia [Bing Videos](#)), luego de esto realizamos la instalación de las dependencias.

Las dependencias como Numpy, torch, torchvision y opencv deben instalarse con el siguiente comando:

i. `pip install numpy torch torchvision opencv-python`

si estos ya están instalados en el entorno verificar que estén en la última versión con el siguiente comando

ii. `pip install numpy torch torchvision --upgrade`

Ejecución:

Primero tomamos el video y lo ubicamos en la carpeta de videos que está en el proyecto al momento de descargarlo, este video debe estar en formato mp4 y lo llamamos video_prueba.mp4, ya con el video en la ubicación podemos realizar la ejecución del script, cabe aclarar que en el proyecto se tiene un video en este formato, en el cual, se ha realizado la segmentación semántica.

Ahora, para ejecutar el script podemos realizarlo de dos maneras. La primera podemos abrir el script en un IDLE o en Spyder de Anaconda o Visual Studio Code, al tener abierto el script realizamos a dar clic en la opción de ejecutar el script o correr código. La manera que recomiendo es abrir la carpeta raíz del proyecto y abrir una terminal en la consola, ya ubicado en la consola en esta carpeta ejecutar el siguiente comando

`Python Solution.py`

Si estas usando Python 3 o este comando no realiza la ejecución del script puedes usar `python3` en lugar de `Python`

Esto ejecutará el script y deberías ver la salida en la consola. Recordar que este script intentará abrir una ventana de visualización, por lo que es posible que necesites ejecutarlo en un entorno donde se admita interfaz gráfica (por ejemplo, fuera de Jupyter Notebook y en un entorno de línea de comandos o en un entorno de desarrollo como Spyder).

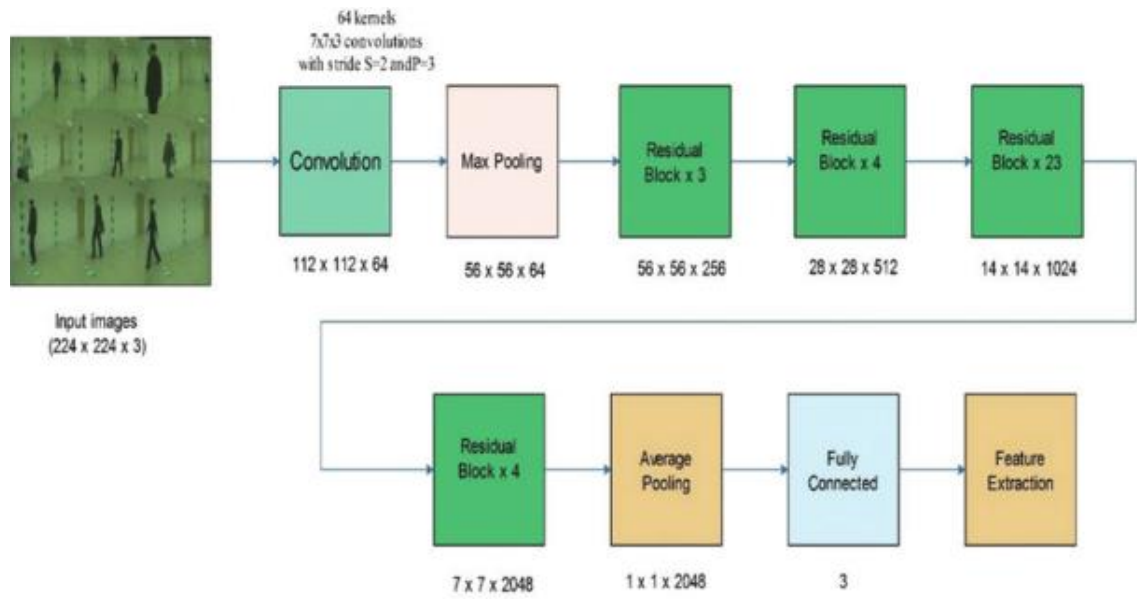
- Estructura del Proyecto

Prueba Desarrollador Analítica Junior

```
|
|--- videos/                                # Carpeta para almacenar datos y/o videos a
segmentar
|      |-- video_prueba.mp4
|
|--- docs/                                  # Carpeta de la documentación del proyecto
|
|-- NZ_INF_DOCX_PRUEBA_PRACTICANTE_ING_ANALYTICS_V2
|--- Solution.py                            # Script principal
|--- solutionOne.py                        # Script secundario realizado con tensorflow
|--- README.md
```

- Modelo de Aprendizaje Profundo

En este proyecto, se ha utilizado el modelo DeepLabv3 con la arquitectura ResNet101 pre entrenada. La arquitectura de DeepLabv3 está diseñada para la segmentación semántica y utiliza una estructura de codificación-decodificación para lograr la segmentación precisa de objetos en imágenes.

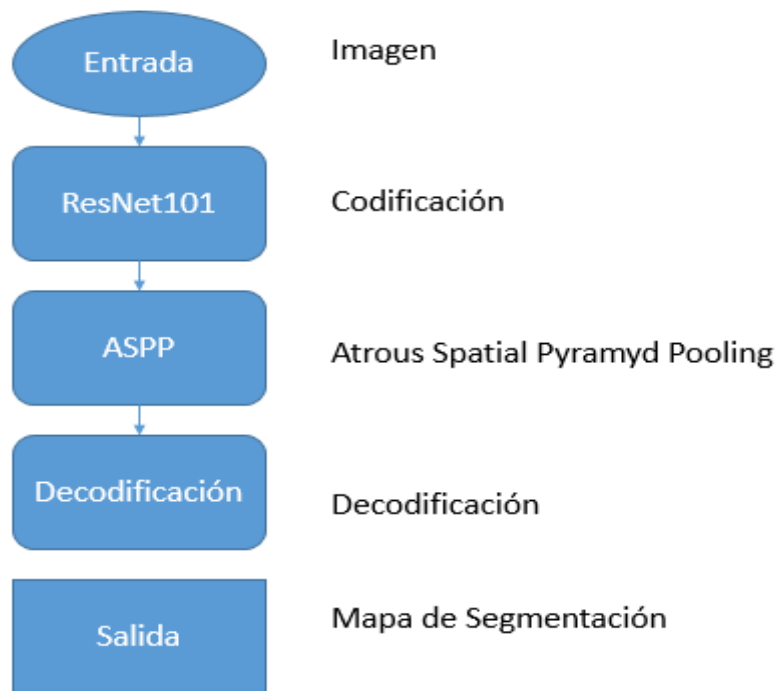


La arquitectura básica de DeepLabv3 incluye:

1. Codificación (Backbone): Utiliza una red neuronal profunda como ResNet101 para aprender características de nivel superior de la imagen de entrada.
2. Atrous Spatial Pyramid Pooling (ASPP): Un modulo que utiliza convoluciones atrous (o dilatadas) con diferentes tasas de dilatación para capturar información contextual a múltiples escalas espaciales.
3. Decodificación: Combina la información contextual de ASPP con características de bajo nivel para obtener segmentaciones precisas. Utiliza bilineales y atrous para mejorar la precisión espacial.

El modelo ResNet101 utilizado como backbone tiene 101 capas y ha sido pre entrenado en un conjunto de datos masivo para tareas de clasificación de imágenes.

Aquí hay una representación simplificada de la arquitectura del modelo DeepLabv3 con ResNet101:



- Contacto e Información del Autor

Nombre: Giovany Andres Angel Avila

Número de teléfono: 3003204872

Correo electrónico: neo_angel64@hotmail.com