UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Aplicaciones de Visión Artificial en Dispositivos de Edge Computing Documentación Técnica

Presentado por Miriam Torres Calvo
en Universidad de Burgos — 9 de septiembre de 2022
Tutor: Bruno Baruque Zanón

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apéndice se va a mostrar la planificación del proyecto, la cuál es la base sobre la crea el proyecto *software*. Desde el punto de vista de la temporalidad y viabilidad. Siendo está una parte fundamental del proyecto, ya que permite visualizar el escenario en el que se desarrollará, de tal forma que podamos realizar una alineación estrategica de los elementos que deben de ser completados, con el objetivo de finalizarlo correctamente.

A.2. Planificación temporal

La planificación temporal se

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se recogen las necesidades funcionales que deberán de ser soportadas por el sistema que va a ser desarrollado. Con el objetivo de obtener una buena documentación, deben de identificarse y describirse los requesitos que tienen que ser satisfacidos por el sistema, pero sin entrar en su proceso de realización.

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se va a exponer cómo se han resuelto los objetivos anteriormente comentados. Así como la definición de datos que se utilizan en la aplicación, procedimientos ...

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este apéndice van a describirse de forma detallada la documentación técnica de programación. Se describirá la estructura de directorios que posee, la instalación y ejecución, así como las pruebas que se han llevado a cabo.

D.2. Estructura de directorios

- `/`: es la raíz del proyecto dónde se encuentran tanto el README, la licencia y las carpetas contenedoras del código, documentación y las pruebas previas.
- `/codigo`: es la carpeta que contiene todo el código funcional del proyecto.
- `/codigo/checkpoints`: es la carpeta contenedora de los modelos de detección en formato Tensorflow, Tensorflow Lite y Tensor-RT.
- `/codigo/checkpoints/custom-416`: carpeta que contiene el modelo de detección de las matrículas, que posee el tamaño 416.
- `/codigo/checkpoints/custom-416/saved_model.pb`: modelo en formato Tensorflow(.pb) de las matrículas.
- `/codigo/checkpoints/heads-416`: carpeta con el modelo de las cabezas en formato Tensorflow.
- `/codigo/checkpoints/heads-416/keras_metadata.pb`: punto de control del modelo de conversión a .pb.

- `/codigo/checkpoints/heads-416/saved_model.pb`: modelo detector de cabezas en formato Tensorflow(.pb)
- `/codigo/checkpoints/yolov4-416`: carpeta con el modelo oficial de YOLOv4 en formato Tensorflow.
- `/codigo/checkpoints/yolov4-416/keras_metadata.pb`: punto de control del modelo de conversión a .pb.
- `/codigo/checkpoints/yolov4-416/saved_model.pb`: modelo detector de YOLOv4 en formato Tensorflow(.pb)
- `/codigo/checkpoints/custom_tfl-416`: carpeta con el modelo de detección de las matrículas en formato Tensorflow previo a la conversióna TensorFlow Lite.
- `/codigo/checkpoints/custom_tfl-416/saved_model.pb`: modelo detector de matrículas en formato Tensorflow(.pb) preparado para su conversión a .tflite.
- `/codigo/checkpoints/custom_tflv2-416`: carpeta con el modelo de detección de las matrículas en formato Tensorflow previo a la conversióna TensorFlow Lite.
- `/codigo/checkpoints/custom_tflv2-416/saved_model.pb`: modelo detector de matrículas en formato Tensorflow(.pb) preparado para su conversión a .tflite.
- `/codigo/checkpoints/custom-416-int8.tflite`: modelo de detección de las matrículas en formato TensorFlow Lite(.tflite).
- `/codigo/checkpoints/custom-416v2.tflite.tflite`: modelo de detección de las matrículas en formato TensorFlow Lite(.tflite)
- `/codigo/checkpoints/models_trt.txt`: fichero con los enlaces de los modelos de TensorRT.
- `/codigo/core`: carpeta con los ficheros de configuración utilizados durante el proyecto.
- `/codigo/core/backbone.py`: fichero de Python que contine las funciones relacionadas con la red YOLOv4
- `/codigo/core/commom.py`: fichero de Python que contine la clase BatchNormalization, para los ajustes de la red YOLOv4
- `/codigo/core/config.py`: fichero de Python que contine permite la selección de los ficheros de etiquetas de cara al uso del modelo
- `/codigo/core/functions.py`: fichero de Python que contine las funciones utilizadas en a lo largo de la detección de los objetos.
- `/codigo/core/utils.py`: fichero de Python que contine las funciones relacionadas con la red YOLOv4.
- `/codigo/core/yolov4.py`: fichero de Python que retorna el modelo de YOLO correspondiente.

- `/codigo/data`: carpeta que contine la información necesaria para la detección.
- `/codigo/data/anchors`: carpeta que contiene los anchors de lass diferentes redes.
- `/codigo/data/anchors/basline_anchors.txt`: fichero de anchors.
- `/codigo/data/anchors/basline_tiny_anchors.txt`: fichero de anchors tiny.
- `/codigo/data/anchors/yolov3_anchors.txt`: fichero de anchors yolov3.
- `/codigo/data/anchors/yolov3_anchors.txt`: fichero de anchors yolov4.
- `/codigo/data/classes`: carpeta que contiene los fichero de etiquetas de los diferentes modelos.
- `/codigo/data/classes/coco.names`: fichero con las etiquetas del modelo oficial de YOLOV4.
- `/codigo/data/classes/custom.names`: fichero con las etiquetas del modelo de detección de matrículas.
- `/codigo/data/classes/heads.names`: fichero con las etiquetas del modelo de detección de las cabezas.
- `/codigo/data/classes/voc.names`: fichero de etiquetas del modelo de voc.
- `/codigo/data/classes/yymnist.names`: fichero de etiquetas del modelo de yymnist, detector de números.
- `/codigo/data/dataset`: carpeta que contiene los ficheros etiquetados a la hora de evaluar un modelo (ruta de la imagen posición detectada y valor de la clase).
- `/codigo/data/dataset/head.txt`: fichero de evaluacion del módelo de las cabezas.
- `/codigo/data/dataset/license_plate.txt`: fichero de evaluación del modelo de las matrículas.
- `/codigo/data/dataset/val2017.txt`: fichero de evaluación del modelo coco.
- `/codigo/data/images`: carpeta que contiene diferentes imagenes para su detección.
- `/codigo/data/video`: carpeta que contien diferentes vídeos para su detección/contabilización.
- `/codigo/deep_sort`: carpeta que contiene los diferentes ficheros en Python para su evaluacion con Object Tracking.
- `/codigo/deep_sort/detection.py`: fichero Python que contiene las funciones de detección para Obejct Tracking.

- `/codigo/deep_sort/iou_matching.py`: fichero Python que tiene las funciones de la maedida iou para Object Tracking.
- `/codigo/deep_sort/kalman_filter.py`: fichero Python que contiene el algoritmo del filtro de Kalman[1].
- `/codigo/deep_sort/linear_assignment.py`: Fichero Python que contiene las funciones relacionadas con la asignación linear.
- `/codigo/deep_sort/nn_matching.py`: Fichero Python con funciones de ajuste del algorimto de vecinos más cercanos[2].
- `/codigo/deep_sort/preprocessing.py`: Fichero Python con las funciones del preprocesado para Object Tracking.
- `/codigo/deep_sort/track.py`: fichero Python que contiene las funciones necearias para detectar los objetos y sus etiquetas correspondeitnes, con su respectivo número de identificación.
- `/codigo/deep_sort/tracker.py`: fichero Python que contiene las funciones necearias para detectar los objetos y sus etiquetas correspondeitnes, con su respectivo número de identificación.

D.3. Manual del programador

D.4. Compilación, instalación y ejecución del proyecto

D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice van a detallarse los requerimientos de la aplicación, su instalación y consejos de cara a usarlo de manera correcta.

E.2. Requisitos de usuarios

E.3. Instalación

E.4. Manual del usuario

Bibliografía

[1] ScienceDirect, "Kalman filter," 2019. [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/kalman-filter#:~:text=The%20Kalman%20Filter%20is%20an,the%20uncertainty%20of%20the%20estimates.

[2] IBM, "K-nearest neighbors algorithm." [Online]. Available: https://www.ibm.com/topics/knn