

Universidad de Alcalá

Escuela Politécnica Superior

Máster Universitario en Ingeniería Industrial

Trabajo Fin de Máster

Diseño, implementación y evaluación de una estrategia de
detección de objetos abandonados en aplicaciones de
videovigilancia

ESCUELA POLITECNICA
SUPERIOR

Autor: Jesús Mudarra Luján

Tutor: Javier Macías Guarasa

Cotutora: Marta Marrón Romera

2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Máster Universitario en Ingeniería Industrial

Trabajo Fin de Máster

**Diseño, implementación y evaluación de una estrategia de
detección de objetos abandonados en aplicaciones de
videovigilancia**

Autor: Jesús Mudarra Luján

Tutor: Javier Macías Guarasa

Cotutora: Marta Marrón Romera

Tribunal:

Presidente: Name of the tribunal president

Vocal 1º: Name of the first vocal

Vocal 2º: Name of the second vocal

Fecha de depósito: 12 de marzo de 2021

A mi hermano Carlos...

“Empieza haciendo lo necesario, luego haz lo posible y de pronto empezarás a hacer lo imposible.”

Francisco de Asís

Falta añadir el tribunal en la página anterior y la fecha de depósito! cambiarlo en el archivo ./Config/myconfig.tex del repositorio plantilla

Agradecimientos

Este trabajo es el fruto de muchas horas de trabajo, tanto de los autores últimos de los ficheros de la distribución como de todos los que en mayor o menor medida han participado en él a lo largo de su proceso de gestación.

Mención especial merece Manuel Ocaña, el autor de la primera versión de las plantillas de proyectos fin de carrera y tesis doctorales usadas en el Departamento de Electrónica de la Universidad de Alcalá, con contribuciones de Jesús Nuevo, Pedro Revenga, Fernando Herránz y Noelia Hernández.

En la versión actual, la mayor parte de las definiciones de estilos de partida proceden de la tesis doctoral de Roberto Barra-Chicote, con lo que gracias muy especiales para él.

También damos las gracias a . . . que nos han proporcionado secciones completas y ejemplos puntuales de sus proyectos fin de carrera.

Finalmente, hay incontables contribuyentes a esta plantilla, la mayoría encontrados gracias a la magia del buscador de Google. Hemos intentado referenciar los más importantes en los fuentes de la plantilla, aunque seguro que hemos omitido alguno. Desde aquí les damos las gracias a todos ellos por compartir su saber con el mundo.

Resumen

Este Trabajo Fin de Máster tiene como objeto el estudio e implementación de algoritmos empleando redes neuronales convolucionales (*CNN*) con la finalidad de detectar objetos abandonados mediante el uso de aplicaciones de videovigilancia. Estas redes se tratan de algoritmos de aprendizaje supervisado especializados para trabajar con imágenes (poner explicación de DotCSV).

En el presente trabajo se ha realizado un estudio teórico de los distintos algoritmos de detección de objetos sobre determinadas bases de datos así como algoritmos de rastreo o *tracking* disponibles en el Estado del Arte. Para la detección de objetos se ha empleado YOLOv4 [1]. Posteriormente se ha desarrollado un algoritmo de DeepSORT [2] donde se ha filtrado que rastree únicamente a personas y objetos de interés: mochilas, maletas, bolsos y bolsas de mano. Se ha utilizado COCO [3] como conjunto de datos ya que se trata de un estándar de referencia muy utilizado en la evaluación del rendimiento de modelos de visión por computador.

Por último se ha implementado y evaluado un algoritmo que determine si un objeto ha sido abandonado o no. Para ello se deberán de tener en cuenta diferentes métricas como el tiempo que el objeto se encuentra estático en un determinado punto o la distancia a la que se encuentra respecto a la persona que lo portaba.

Palabras clave: Redes neuronales convolucionales, YOLOv4, DeepSORT, videovigilancia, visión por computador.

Abstract

The aim of this Master's Thesis is to study and implement algorithms using convolutional neural networks (*CNN*) in order to detect abandoned objects through the use of video surveillance applications. These networks are specialized supervised learning algorithms for working with images (put DotCSV explanation).

In the present document, has been carried out a theoretical study of the different object detection algorithms on certain databases, as well as tracking algorithms available in the State of the Art. YOLOv4 [1] has been used for object detection. Subsequently, a DeepSORT algorithm [2] has been developed where it has been filtered that only tracks people and objects of interest: backpacks, suitcases and handbags. COCO [3] has been used as the dataset as standard benchmark for evaluating the performance of computer vision models.

Finally, an algorithm has been implemented and evaluated to determine whether an object has been abandoned or not. To do this, different metrics must be taken into account, such as the time that the object is static at a certain point or the distance it is from the person who was carrying it.

Keywords: Convolutional Neural Networks, YOLOv4, DeepSORT, Video Surveillance, Computer Vision.

Índice general

Resumen	V
Abstract	VII
Índice general	IX
Índice de figuras	XI
Índice de cuadros	XIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura de la memoria	1
2. Estudio teórico	3
2.1. Introducción	3
2.2. Estado del arte	3
2.3. Técnicas utilizadas	3
2.3.1. YOLOv4 + Tensorflow 2	3
2.3.2. DeepSort	3
2.4. Conclusiones	3
3. Desarrollo algoritmo de objetos abandonados	5
3.1. Introducción	5
3.2. Desarrollo del algoritmo	5
3.3. Conclusiones	5
4. Resultados	7
4.1. Introducción	7
4.2. Entorno experimental	7
4.2.1. Bases de datos utilizadas	7
4.2.2. Métricas de calidad	7
4.2.3. Estrategia y metodología de experimentación	7
4.3. Resultados experimentales	7
4.4. Conclusiones	7

5. Conclusiones y líneas futuras	9
5.1. Conclusiones	9
5.2. Líneas futuras	9
Bibliografía	11
Apéndice A. Pliego de condiciones	13
A.1. Introducción	13
A.2. Requisitos de hardware	13
A.2.1. Requisitos mínimos	13
A.2.2. Requisitos de hardware recomendados	13
A.3. Condiciones hardware	13
A.4. Requisitos de software	14
A.4.1. Requisitos mínimos	14
A.4.2. Requisitos de software recomendados	14
A.5. Condiciones software	14
A.6. Condiciones generales	15
Apéndice B. Presupuesto	17
B.1. Equipo de trabajo	17
B.2. Timing	17
B.3. Presupuesto total	17
Apéndice C. Manual de usuario e instalación	19
C.1. Introducción	19
C.2. Manual	19
C.3. Ejemplos de inclusión de fragmentos de código fuente	19

Índice de figuras

2.1. Red neuronal ...	3
-----------------------	---

Índice de cuadros

3.1. Esto es una tabla ejemplo	5
--	---

Capítulo 1

Introducción

1.1. Motivación

Introducción del proyecto ...

1.2. Objetivos

1.3. Estructura de la memoria

Capítulo 2

Estudio teórico

Aquí se explicará el estado de arte actual ...

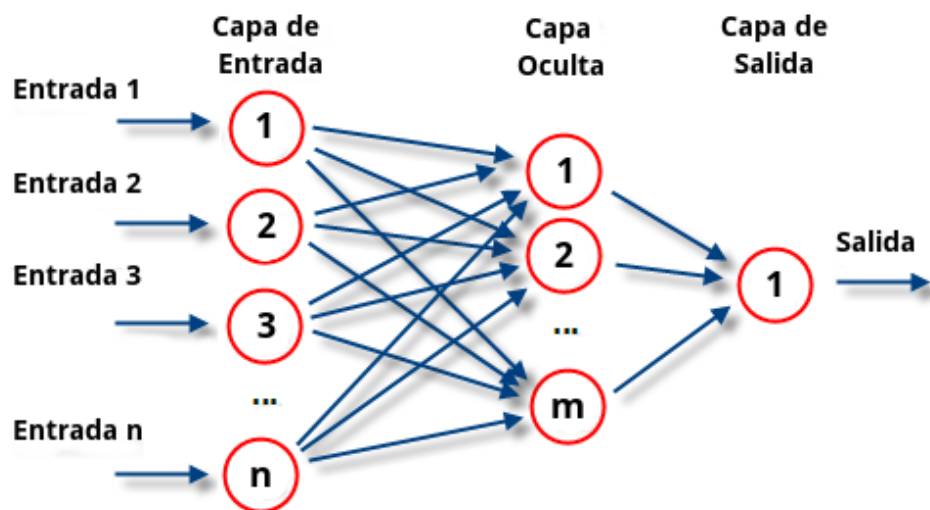


Figura 2.1: Red neuronal ...

La figura [2.1](#) muestra ...

2.1. Introducción

2.2. Estado del arte

2.3. Técnicas utilizadas

2.3.1. YOLOv4 + Tensorflow 2

2.3.2. DeepSort

2.4. Conclusiones

Capítulo 3

Desarrollo algoritmo de objetos abandonados

3.1. Introducción

3.2. Desarrollo del algoritmo

Esto	es	una	tabla	sencilla
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Cuadro 3.1: Esto es una tabla ejemplo

3.3. Conclusiones

Capítulo 4

Resultados

Aquí se mostrarán los resultados del proyecto ...
Como dijo [3] ...

4.1. Introducción

4.2. Entorno experimental

4.2.1. Bases de datos utilizadas

4.2.2. Métricas de calidad

4.2.3. Estrategia y metodología de experimentación

4.3. Resultados experimentales

4.4. Conclusiones

Capítulo 5

Conclusiones y líneas futuras

En esta sección se hablará sobre futuros proyectos derivados de éste ...

5.1. Conclusiones

5.2. Líneas futuras

Bibliografía

- [1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” 2020.
- [2] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
- [3] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” 2015.

Apéndice A

Pliego de condiciones

A.1. Introducción

En este apartado se evaluarán las condiciones para poner en marcha el software que se ha especificado en los apartados anteriores. Cabe resaltar el carácter de este proyecto, en el que se ha diseñado una colección de funciones que facilitan al programador la correcta adquisición de datos sonoros, y por lo tanto no aplican las condiciones técnicas o ambientales que pudieran afectarle, ya que las impone los requerimientos las aplicaciones que en un futuro se le quiera dar a este proyecto.

Solamente afectan las condiciones de configuración hardware o software donde se quiera aplicar el programa informático.

A.2. Requisitos de hardware

A.2.1. Requisitos mínimos

- Utilización de un PC de 32 bits de escritorio con tarjeta de sonido.
- Un mínimo de 384 MB de memoria RAM.
- Al menos 100 MB de memoria libre en disco duro.

A.2.2. Requisitos de hardware recomendados

Estos requisitos son necesarios para implementar algoritmos de localización basados en onda sonora.

- CPU de 64 bits con 4 Cores o más.
- Sistema de adquisición con 8 canales o más.
- Utilización de al menos 4 Gb de memoria RAM.

A.3. Condiciones hardware

El sistema de adquisición que se propone hace uso de 4 hilos independientes en la adquisición en tiempo real. Es por esta razón que se recomienda sistemas multiprocesador, que permitan realizar las tareas de cada hilo de manera independiente.

Se precisa de un sistema de adquisición de audio profesional para la adquisición del audio proveniente de cada micrófono para ejecutar el algoritmo de localización. Además por esta razón y porque se van a generar gran cantidad de datos a procesar en la memoria RAM del equipo, es necesaria la utilización de un volumen importante de memoria RAM, se recomienda que la cifra de partida sean 4 Gb que es la cifra máxima que un sistema operativo de 32 bits basado en Linux puede direccionar.

Se recomienda tener 100 MB de disco duro libre para poder hacer grabaciones de corta duración. Es altamente recomendable disponer de 10 GB libres si se van a realizar grabaciones multicanal y de larga duración.

A.4. Requisitos de software

A.4.1. Requisitos mínimos

- Utilización de un sistema operativo Ubuntu 12.04.
- Librería Rtaudio.
- Librería SNDFile.
- Para el desarrollo del algoritmo de localización se deben utilizar las librerías propias del grupo GEINTRA.

A.4.2. Requisitos de software recomendados

- Utilización de un sistema operativo de 64 bits, Ubuntu 12.04 o superior.

A.5. Condiciones software

En este apartado software se recomienda utilizar el sistema operativo Ubuntu 12.04 al ser LTS, y proporcionar la compatibilidad con las nuevas librerías Qt para realizar profiling mediante KCachegrind.

En el caso de disponer un sistema hardware con una memoria RAM superior a 4 GB es necesario utilizar, una versión de Ubuntu de 64 bits para poder direccionarla. Es muy recomendable esta opción para poder adquirir y procesar sonido multicanal.

A.6. Condiciones generales

La utilización de esta librería supone la posibilidad de ejecutar cualquier programa de procesamiento sobre ella que tenga en cuenta las siguientes condiciones:

- El ancho de banda de las señales acústicas está condicionado por el rango que permita la tarjeta de adquisición a utilizar, por lo general cumple aproximadamente la del espectro del oído humano, de 20 a 20000 Hz.
- El número de canales a utilizar lo limita la tarjeta de adquisición, la librería está preparada para adquirir, sea cual sea el rango multicanal.
- Ante cualquier uso de esta librería deberá tenerse en cuenta el reconocimiento (BY), que no es comercial (NC), y que las obras derivadas se compartan de igual manera (SA).

Apéndice B

Presupuesto

B.1. Equipo de trabajo

Para la realización del proyecto se va a necesitar un Ingeniero de Telecomunicaciones.

B.2. Timing

Las fases necesarias para la realización del desarrollo son las siguientes.

- Configuración del hardware (0,5 meses)
 - Instalación de sistema operativo, librerías, IDE. (0,25 meses)
 - Configuración y puesta en marcha. (0,25 meses)
- Diseño e implementación de los módulos software necesarios (6 meses):
 - Diseño e implementación de librerías de soporte para la reproducción y adquisición de audio multicanal temporizado (tiempos fijos) (2 mes).
 - Diseño e implementación de librerías de soporte para la adquisición, reproducción y procesamiento de audio multicanal continuo (tiempo indefinido) (4 mes).
- Evaluación del sistema de procesamiento de audio multicanal (1,5 meses):
 - Adaptación del algoritmo SRP. (0,5 meses)
 - Evaluación del sistema de adquisición sobre la librería de tiempo real (1 mes)
- Documentación

Por supuesto, las fases de diseño, desarrollo, pruebas y documentación son cíclicas y abarcan todo el periodo de la vida del proyecto.

B.3. Presupuesto total

Por todo lo anterior, supone una duración de 8 meses naturales con un coste de 12.000 €(sin IVA).

Apéndice C

Manual de usuario e instalación

C.1. Introducción

Blah, blah, blah...

C.2. Manual

Pues eso.

C.3. Ejemplos de inclusión de fragmentos de código fuente

Ahora compila usando gcc:

1. bla bla:

```
$ python object_tracker.py —video ./data/video/video1.avi
```

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá