UNIVERSIDAD AUTÓNOMA DE MADRID

Grado en Informática y Matemáticas

TRABAJO FIN DE GRADO

Clustering de Zonas en Imágenes

Una aplicación a la detección de peatones

Autor: Jorge González Villacañas
Tutor: Eduardo Cermeño Mediavilla
Ponente: Juan Alberto Sigüenza Pizarro

junio 2019

Jorge González Villacañas
Clustering de Zonas en Imágenes

Jorge González Villacañas

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Resumen

La aparición de voluminosas bases de datos, y la mejor accesibilidad a hardware de alto rendimiento han propiciado un incremento en la popularidad de la visión artifical.

Dentro de este campo de la informática se encuentra la tarea de detección de objetos e identifiación de escenas. El presente trabajo trata sobre la detección de peatones que se encuetran cruzando la calzada en la vía pública. Se hace especial incapié en la aplicación de aprendizaje supervisado. En particular, ejemplificamos el uso de redes neuronales convolucionales para la segmentación semántica de imágenes.

Palabras clave

Visión artifical, segmentación semántica, aprendizaje supervisado, redes neuronales convolucionales

Abstract

The increasing availability of new data bases together with high performance hardware becoming more accesible has led to the rise in popularity of computer vision.

Object detection and scene identication are among the common topics in this area of computer science. This work deals with the task of pedestrian recognition across the street. It emphasises the practicality of supervised learning in computer vision. In particular, we employ convolutional neural networks for image semantic segmentation .

Keywords

Computer Vision, semantic segmentation, supervised learning, convolutional neural networks

ÍNDICE

Listas

Lista de tablas

1

Introducción

En esta sección se explica el objetivo y motivación del trabajo de fin de grado realizado. Además se describe la organización del documento.

1.1. Motivación

La comprensión de la escena es uno de los objetivos de la visión artificial. La capacidad de entender qué está transcurriendo en una secuencia de vídeo constituye la base de nuevas tecnologías como la conducción autónoma o la vídeo vigilancia inteligente, donde se precisa información de más alto nível ya sea para evitar accidentes o detectar infracciones.

Este trabajo fin de grado se centra en la detección de peatones en zonas potencialmente peligrosas. Los informes de la DGT indican que el número de peatones fallecidos en 2018 a causa de un atropello es de 135, 46 personas más que en 2017. Este trabajo se suma a la iniciativa de impulsar todavía más el uso de la visión artificial en los sistemas de seguridad ciudadana.

Es natural pensar que todo sistema de visión artificial acabe dotado de esta capacidad de adquirir información más conceptual, similar al ser humano. La facultad de relacionar figuras con conceptos abre un nuevo campo en el análisis inteligente de los vídeos. Estos avances, tomados con precaución y rigor, pueden servir para reducir el número de accidentes provocados en la vía pública. Lamentablemente en España a día de hoy se siguen superando los mil fallecimientos al año.

1.2. Objetivo

En este trabajo se propone un sistema para la **detección automática de peatones** sobre la calzada. Se elabora un experimento para medir la calidad de la propuesta.

Podemos dividir la estrategia en dos pasos:

- Segmentación semántica de las imágenes.
- Aplicación de reglas heurísticas para la identificación de peatones.

La **segmentación semántica** de las imágenes aporta información de alto nivel que puede ser empleada para detectar situaciones presentes en la escena. En nuestro caso, buscamos aquellas personas situadas sobre la vía urbana. Para realizar la segmentación proponemos el uso de una red neuronal. El modelo de red elegido se denomina Deeplab [1], creado por Google. La base de datos que se emplea es Cityscapes [2]. El primer hito consiste en entrenar sobre este conjunto de datos durante las épocas necesarias hasta alcanzar cierta condición de parada.

A continuación, aprovechando la información extraida en la fase previa, se proponen una serie de reglas que puedan clasificar la existencia de personas sobre la calzada. Valoramos varias alternativas y exponemos los resultados.

Otro foco del trabajo reside en el aspecto técnico. Existen diferentes alternativas tecnológicas para trabajar con redes neuronales. Parte del trabajo puede servir como ejemplo del uso de nuevas herramientas como el servicio Google Colab o la biblioteca de Python Keras. Ambas son abiertas y se presentan como buenos recursos para llevar a cabo proyectos de aprendizaje automático a nivel académico.

1.3. Organización

La memoria se divide en las siguientes secciones:

- **Estado del Arte**: Se explican las técnicas actuales más novedosas y efectivas para la detección de peatones. También se exponen métodos de segmentación semántica y detección de objetos en imágenes.
- **Método**: Explicación de la propuesta y de los conceptos que se utilizan.
- **Experimento** Por un lado se incluyen las herramientas y metodologías utilizadas durante la fase de desarollo. Luego se procede a detallar el proceso de entrenamiento de la red neuronal y se obtienen resultados de segmentación y detección de peatones.
- **Conclusiones** Valoración de los resultados y trabajos futuros.

2

Estado Del Arte

En esta sección se incluyen algunas de las técnicas más populares para la detección de personas en la escena de la calle. Se exponen métodos para la detección de objetos y para la segmentación semántica de imágenes ya que consideran el mismo problema de una manera más general.

2.1. Técnicas Para La Segmentación Semántica

La segmentación semántica de una imagen consiste en crear una división de la misma tal que las regiones obtenidas puedan entenderse como elementos de alguna categoría. Mientras que en la detección de objetos se obtienía una ubicación, la segmentación trabaja con más precisión. El resultado es una imagen donde cada píxel es etiquetado en función de la categoría a la que pertenece.

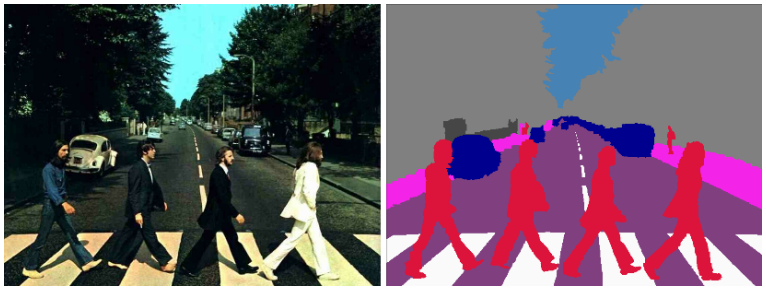


Figura 2.1: Ejemplo segmentacion semantica

Este trabajo se centra en la temática de la calle, donde se puede encontrar clases como vehículo, peatón, cielo, suelo, acera ... No se efectúa una distincción entre elementos de la misma clase. El problema de diferenciar miembros de una misma categoría se denomina **Segmentación de Instancia**.

El objetivo concreto de detección de peatones en zonas de interés puede beneficiarse enormente de estas técnicas. Ya que el resultado que proporcionan es una descripción exhaustiva de la escena presente. A contiuación se exponen algunas de las más comunes.

Redes Neuronales Totalmente Convolucionales: FCN

En [3] se plantea un modelo de red que sigue el esquema Encoder-Decoder [4], donde únicamente se usan capas de convolución. Se propone crear una nueva arquitectura basada en aquellas redes neuronales orientadas a la clasificación que están resultando más exitosas, AlexNet, VGG net o GoogLenet [5].

Para pasar de la tarea de clasificación al formato de segmentación se elimina la última capa, donde se efectuaba la predicción. Se sustituya por otra que recupere las dimensiones de la imagen original. Es decir, la red adquiere la capaciad de aprender a pasar de una versión reducida de la imagen a su formato original. Donde cada pixel ahora codifica una categoría. Los pesos previamente adquiridos no son descartados sino que se emplean como punto de partida en el proceso de entrenamiento de la nueva red.

Para entrenar este tipo de redes hace falta un conjunto de imágenes junto con sus máscaras por categoría. Generar una base de datos resulta costoso. Algunas de las más populares centradas en la escena de la calle se llaman Camvid, Kitti, Cityscapes o Pascal VOC, y están totalmente disponibles.

Redes Neuronales Recurrentes: ReSeg

Las redes neuronales recurrentes [6] han sido utilizadas con gran exito para el modelado de secuencias de corto y largo tiempo. Son una opción para introducir información contextual a la hora de clasificar.

El trabajo propuesto en [7] integra esta estrategia junto con una arquitectura de red convolucional orientada a la segmentación de imágenes. Combinando de esta manera información local y contextual.

W-Net

W-Net [8] es un ejemplo de un tipo de arquitectura conocida como autoencoder [9], empleada para la segmentación semántica dentro de la categoría de apredizaje sin supervisar.

La técnica consiste en concatenar dos redes totalmente convolucionales, como FCN [3]. La primera red produce una segmentación de la imagen inicial y la siguiente trata de reconstruirla.

Sí que existe una fase de entrenamiento, pero no hace falta un conjunto previamente clasificado. Lo único necesario son imágenes y el apredizaje consiste en lograr obtener una buena reconstrucción de las mismas. Se trata de minimizar tanto el error en la reconstrucción como el error en la segmentación. Como no se cuenta con un conjunto de imágenes segmentadas, esto último no puede realizarse comparando con la realidad (sería el caso supervisado). Alternativamente se emplea un criterio de clustering [10].

2.2. Técnicas Para La Detección de Personas

En esta sección se trata el problema de detección de personas en imágenes o vídeo, sin imponer condiciones de ubicación. La tarea consiste en delimitar el area donde con cierta seguridad se sabe que hay una persona. Es un caso particular de la **detección de objetos**.

Figura 2.2: Ejemplo deteccion de objeto

Selective Search

En [11] se propone un método para la detección de objetos partiendo de una división inicial en regiones. Esta fase de clustering también comienza con una partición de la imágen, obtenida a partir de la búsqueda de posbibles bordes en las figuras. Las zonas finales se obtienen combinando las pequeñas regiones según su similitud en términos de color, textura y tamaño. Cada una de ellas ahora tiene una mayor confianza de albergar cierta clase de objeto.

Finalmente por cada una de las localizaciones obtenidas se pasa un modelo de clasificación. En particular, se propone utilizar una máquina de vector soporte. Esta estrategia se encuentra en la categoría de métodos basados en regiones (region-based).

YOLO: You Only Look Once

A diferencia de las estrategias de detección por regiones, donde se emplea un clasificador por cada ubicación predefinida, [12] clasifica la imagen una única vez. Utilizando una arquitectura de red neuronal convolucional, logran dividir la imagen en una malla y predecir una confianza de los tipos de objeto contenidos en cada una de sus celdas. No sólo aproximan una probabilidad sino también su ubicación en forma de recuadro dentro de la imagen completa.

La salida es una amplia colección de recuadros que tienen asociados una probabilidad de contener cierta clase de objeto. Eliminando aquellos con baja confianza se obtiene el resultado final.

Detección de Peatones Parcialmente Visibles

El trabajo realizado en [13] plantean un método para lidiar con la variabilidad en la apariencia de los peatones. Tanto la postura, el color de la ropa, la luminosidad o la obstrucción de algún objeto de por medio, dificultan la detección de personas cuando se buscan en su totalidad.

Proponen una estrategia divida en dos fases. Primero se emplea una red convolucional para extraer los recuadros susceptibles a contener peatones. Seguidamente se efectúa una alineación de los mismos en la dirección que mejor esclarezca la detección del peatón. Se emplea una combinación de clasificadores de partes del cuerpo (cabeza, tronco, piernas) para guiar en la búsqueda.

2.3. Técnicas Para La Detección de Peatones en la Calle

Se exponen diferentes trabajos cuyo objetivo es detectar peatones en localizaciones concretas. El problema no sólo es identificar a las persona presentes sino también comprender dónde están situados. Esto es de especial interés para sistemas de conducción autónoma o seguridad.

Detección de Peatones Sobre la Calle

En [14] plantean la detección de personas cruzando la calle combinando diferentes técnicas para la determinación del flujo óptico en la secuencia de imágenes.

Primero se efectúa una división en bloques por solapamiento. Después se clasifican como elementos en movimiento o como fondo estático. Para lograrlo se utiliza el modelo de mezcla de gaussianas, con el cual se mide la diferencia del mismo bloque en tiempos distintos. El bloque se clasifica como objeto en movimento cuando se supera un umbral previamente fijado. Finalmente se agrupa en cada en el grupo correspondiente para obtiener una división a nivel pixel.

Para la detección de peatones dentro del conjunto dinámico se mide el volumen de la región susceptible a contener una persona. Si el valor se encuentra dentro de un intervalo previamente establecido mediante observación, se considera humano. La identificación de la calle se efectúa por separado y combina técnicas de detección de bordes junto con métodos de detección de patrones de colores y texturas.

Detección de Peatones Sobre Pasos de Cebra

Este trabajo [15] tiene como objetivo la detección de infracciones de tráfico. En concreto, identican vehículos y peatones sobre pasos de zebra. En caso de coincidir ambos elementos, se estaría cometiendo una infracción.

Proponen una detecc ión por separado del paso de cebra, peatón y vehículo. El paso de cebra

no se identifica automáticamente, debe realizarse una delimitación previa del área que cubre. Para la localización de vehículos se emplean redes neuronales convolucionales. Por otro lado, los peatones presentes en la escena se extraen utilizando el método de mezcla gaussiana para la identificación de objetos en movimiento.

Frenado Automático de Vehículos

[16] Plantea un método para el frenado automático en vehículos ante la presencia de peatones en zonas potencialmente peligrosas. Combinan un escaneado laser con un sistema de visión artificial basado en el algoritmo de detección de peatones AdaBoost [17].

El algoritmo se centra en la detección de peatones que aparecen repentinamente en la escena. Consecuentemente su búsqueda se realiza sobre aquellas zonas clasificadas como críticas. Principalmente las esquinas de la calzada o zonas obstaculizadas, donde la existencia de personas es incierta.

Un peatón que se encuentre parcialmente visible produce una alerta interna. Se efetúa entonces un seguimiento hasta que posiblemente sea completamente visible. Una vez expuesto al laser se calcula la dirección de su movimiento. En caso de ir hacia el centro de la calle se alerta al conductor.

3

Método

A continuación exponemos nuestra propuesta para la detección de peatones que se encuentren en la vía urbana. Consiste en una identificación automática, no es necesario ningún tipo de configuración previa. Dada una imagen como entrada, el sistema devolverá o no una alarma en función de si detecta una persona situada sobre la calzada.

Los modelos previos incluidos en el estado del arte [refs] no cuentan con esta característica. Según entiendo, tras haber llevado acabo un estudio de diversos trabajos, no existe ninguna propuesta para efectuar una detección totalmente automática. Por esta razón, consideramos que hay cierto interés en la opción que presentamos.

Esta sección comienza con una descripción a alto nivel del método planteado. Las siguientes subsecciones se emplean para esclarecer aquellas técnicas y definiciones que intervienen en el proceso. Se incluye ordenadamente:

- Visión General
- Segmentación Semántica
- Redes Neuronales Convolucionales

3.1. Visión General

En formato de caja negra, el sistema debe tomar una imagen y devolver una clasificación binaria en función de si se visualiza una zona de calzada y un peatón sobre ella.

La primera fase consiste en obtener de la imagen información de más alto nivel para facilitar la comprensión de la escena visualizada. Se efectúa un proceso de **segmentación semántica** 3.1 de la imagen en zonas y objetos. En la vía urbana suelen encontrarse categorías como calle, acera, edificio, coche, persona o vegetación entre otras. Para ello proponemos el uso de **redes neuronales convolucionales**. Su exitosa aplicación en diversos problemas de la visión artifical ha propiciado la aparición de técnicas similares para la segmentación en imágenes [5]. Además otorgan flexibilidad en el formato de imagen entrante y han demostrado ser eficaces lidiando con elementos a diferentes

escalas.

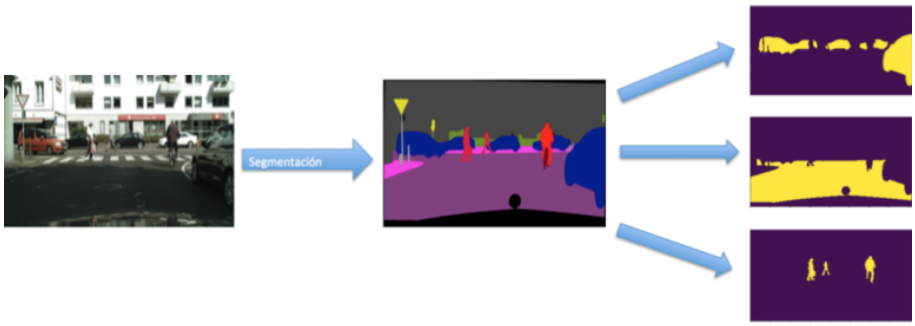


Figura 3.1: Segmentación de la imagen inicial.

La segmentación implica la separación en zonas de cada categoría semántica obtenida. La siguiente fase consiste en combinar esta información e identificar patrones para la detección de peatones sobre la calzada 3.2. Se plantean diversas reglas heurísticas que toman como entrada la imagen segmentada y devuelven una clasificación binaria. Siendo el caso positivo encontrar un peatón en la calle y el caso negativo no detectar ninguno.

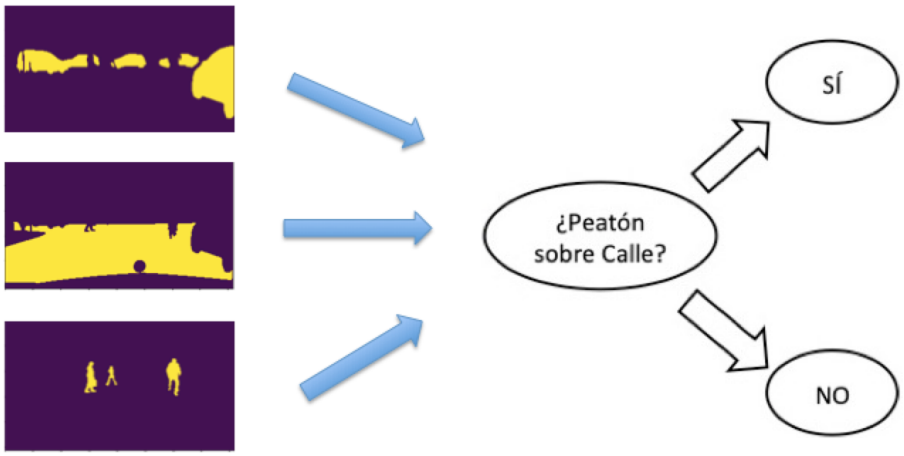


Figura 3.2: Esquema aplicación de regla para la detección de peatones a partir de las máscaras de segmentación.

En el trabajo exploramos dos opciones como regla:

1. Solapado Persona sobre Carretera: Una primera estrategia consiste en buscar aquellas zonas donde solapen las clases persona y calle. Para lograrlo primero se deben extraer por separado ambas máscaras. Para la calzada se eliminan los puntos considerados como ruido y se efectúa la **envoltura convexa** del conjunto, como forma heurística de definir el área total de calle, posiblemente incompleta tras la segmentación. Se eliminan todos los puntos que solapen con clases colindates: acera, vegetación y terreno. A continuación se intersecta con la máscara de la clase persona. Si en la salida existe cierto número de píxeles se considera que efectivamente hay un peatón sobre la calle.

2. Búsqueda de Fronteras: Intuitivamente una heurística sería buscar únicamente los 'pies' de los

peatones. Teniendo en cuenta que una persona puede encotrarse sobre la acera, pero dependiendo de la perspectiva podría visualizarse parte de su cuerpo en la calzada.

Para ello efectuamos una búsqueda de la frontera entre la clase persona y calle intentando extraer únicamente aquellas que dejen en la parte de arriba píxeles persona y debajo píxeles calle.

Empleamos un filtro de tamaño 3x3 a lo largo de toda la imagen:

$$filtro = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

3.2. Segmentación Semántica

Efectuar una segmentación semántica sobre una imagen consiste en asignar a cada pixel una etiqueta como 'persona' o 'coche'. Se consigue crear una división de la misma tal que las regiones obtenidas puedan entenderse como elementos de alguna categoría. Este trabajo se centra en la temática de la calle, donde se puede encontrar clases como vehículo, peatón, cielo, suelo, acera . . . No se efectúa una distincción entre elementos de la misma clase. El problema de diferenciar miembros de una misma categoría se denomina **Segmentación de Instancia**.

Los métodos predecesores de clasificación de imágenes y localización de objetos proporcionan una primera aproximación a la información de alto nivel que puede comprenderse en una imagen. La segmentación semántica es el paso natural en la progresión de la visión artificial. El objetivo es conseguir mayor precisión en la predicción de ubicaciones. La naturaleza de este trabajo requiere de tal nivel. Cuanto mayor precisión en la localización de un peatón y mejor delimitado quede el area de la calle, más fácil será identificar la situación en escenas delicadas: lejanas, obstaculizadas, bulliciosas. . .

3.3. Redes Neuronales Convolucionales

La red neuronal artificial más tradicional consiste en un conjunto de nodos y conexiones estructurado de tal manera que evoca a una red neuronal natural. La división en capas relaciona los nodos neuronas en forma de combinación lineal:

$Y_j = g(b_j + \sum_i k_{i,j} * X_i)$

Donde b_j se denomina **valor bias** y g es una **función diferenciable** a modo de activación de la neurona Y_j. Por ejemlo una sigmoidal, cuya salida se encuentra en el intervalo $[0, 1]$.

$sigmoid(x) = \frac{1}{1+e^{-x}}$

La arquitectura de una red neuronal es suficientemente flexible como para entender una entrada

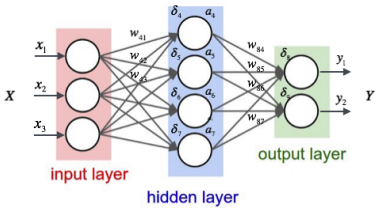


Figura 3.3: Estructura de una red neuronal convencional

en formato de imagen. Es decir, en lugar de tener una lista de valores entrantes, se recibe una matriz de píxeles.

La **red neuronal convolucional** extiende la red clásica en el tipo de conexiones que pueden existir entre capas. A continuación se introducen sus caraterísticas principales.

Con el objetivo de ser lo más claros posibles vamos a tratar el caso bidimensional de los datos. Es decir, entender cada capa de neuronas como una matriz NxM de números, igual que la codificación de una imagen en escala de grises. Cada neurona corresponderá a un píxel en la imágen. Esto puede generalizarse al caso de imágenes a color.

3.3.1. Convolución

Intuitivamente la convolución consiste en recorrer una matriz PxQ, también denominada filtro o kernel, a lo largo de la matriz de neuronas. De manera que por cada hueco que va cubriendo el filtro se efectúa la combinación lineal de las neuronas con los pesos del filtro.

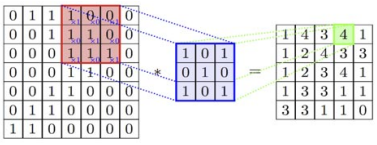


Figura 3.4: Ejemplo de la operación de convolución sobre una imagen

De la misma manera que en una red clásica, al valor resultante se le suma un valor bias y se le aplica una función de activación del tipo sigmoidal. Se puede expresar de manera sintética el valor de una neurona en la capa siguiente:

$$x_{i',j'} = g(b + \sum_{i,j} K_{i,j} * X_{i',j',i,j})$$

Donde, $K_{i,j}$ representa la matriz del filtro. $X_{i',j',i,j}$ el área de la matriz inicial sobre la que se efectúa la convolución b el valor bias.

La función g es una función no lineal. Abstrae el concepto de activación de las neuronas biológicas.

La novedad reside en que se aplica el mismo filtro todo el recorrido. Esto otorga cierto grado de

invarianza respecto a traslaciones y cambios de escala de de los elementos presentes en la imagen. Por ello se convierte en una popular herramienta en los sistemas de detección de objetos y clasificación de imágenes.

3.3.2. Pooling

Para reducir el número de parametros de la red se utiliza esta técnica. Al igual que en la convolución, se hace pasar un filtro a modo de 'ventana deslizante' a lo largo de la imagen. Por cada hueco que cubre se efectúa una operación que resuma el contenido en cierta manera. Por ejemplo, tomar el máximo o la media aritmética.

Es importante destacar que no hay ningún paramétro que aprender. Es una operación determinista, siempre se realiza de la misma manera.

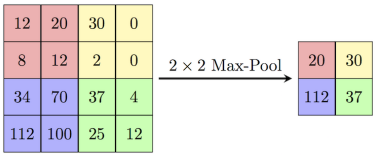


Figura 3.5: Ejemplo de la operación de Pooling tomando el máximo

3.3.3. Arquitecturas Para La Segmentación Semántica

La segmentación semántica de imágenes consiste en asignar a cada píxel una etiqueta que le identifique dentro de un cojunto sémantico. Al igual que la entrada de una red neuronal puede generalizarse al caso de imágenes, la salida también. Concretamente, la salida de una red de este tipo tiene que ser del mismo formato que su entrada. El resultado tiene que ser la codificación en etiquétas de la imagen inicial.

La arquitectura de red necesaria para lograr esta flexibilidad en la entrada y la salida extiende la clásica red de clasificación binaria. En este trabajo estudiamos y usamos la arquitectura conocida como Encoder-Decoder. Inicialmente propuesta por Google [4], motivada por la investigación en la traducción de textos mediante redes neuronales, logra asociar secuencias con estructuras genéricas. Tras su publicación no se tardó en aplicar al campo de la visión artificial. Consecuencia directa fueron las primeras redes de este tipo [3] [18] orientadas a la segmentación semántica o de instancia, donde la sálida a predecir son máscaras.

Hoy en día son un opción muy común en las redes orientadas a la segmentación.

Encoder - Decoder

Esta arquitectura plantea dividir la red en dos partes. Primero se cuenta con las capas del Codificador (Encoder). Su composición a penas difiere de cualquier red convolucional de clasifición. Es una práctica común tomar aquellas arquitecturas que han obtenido buenos resultados en las tareas detección o localización de objetos.

Su tarea es crear una representación de la imágen de menor tamaño perdiendo el mínimo de información. Para ello se concatenan los filtros de convolución, capas de pooling y capas directamente conectadas.

Seguidamente se enlaza con el decodificador, cuyo objetivo es devolver la versión reducida de la imagen a sus dimensiones originales. Métodos clásicos para aumentar la resolución en una imagen:

- Interpolación Bilineal
- Interpolación Bicúbica
- K-Vecinos Próximos

Esta fase también puede entrenarse. Tras el aumento de la imagen reducida se puede continuar con una capa de convolución. De esta manera, en lugar de efectuarse un aumento de resolución determinista se adapta al problema al que se está exponiendo durante el aprendizaje.

4

EXPERIMENTO

En esta sección se detalla el experimento realizado. Entrenamos una arquitectura de red neuronal convolucional DeepLabV3+ [1] para la tarea de segmentación semántica de imágenes cuya escena se centra en la vía urbana. El conjunto de datos empleado es CityScapes [2]. Seguidamente aplicamos el resultado obtenido a la tarea de detección de peatones cruzando la calle.

Dividimos en tres apartados ordenadamente:

- Metología Experimental
- Descripción del entrenamiento de la red
- Resultados de la segmentación
- Detección de peatones cruzando la calle

4.1. Metodología Experimental

A continuación explicamos qué implementación se ha usado para probar el método propuesto. Mencionamos qué tipo de modelo de red neuronal se emplea, el conjunto de datos y las herramientas software utilizadas.

4.1.1. Red Neuronal 'DeepLab'

La red neuronal DeepLab [1] es un modelo de red neuronal para la segmentación semántica de imágenes ideado por Google. Se enmarca dentro de la familia de redes convolucionales. El objetivo de esta subsección es destacar aquellas características que lo diferencian:

- Arquitectura Encoder-Decoder [4]
- Piramides espaciales de pooling [19]
- Convoluciones Dilatadas [1]

Encoder-Decoder: La red DeepLab tiene una estructura conocida como Codificador-Decodificador. Se identifican dos partes principales. Un primer módulo de codificación que gradualmente reduce la

representación de la imagen capturando información de más alto nivel semántico. Seguidamente se enlaza con la parte de decodificación, cuyo objetivo es recuperar la información espacial devolviendo la salida a las dimensiones originales.

Piramides espaciales de pooling Una característica deseable en un modelo de segmentación es la robustez frente a cambios en el tamaño de los objetos que se visualizan. Deeplab proporciona una solución incorporando las pirámides espaciales de pooling. Consiste en aplicar un mismo tipo de filtro a diferentes escalas de la 'imagen' y fusionando sus salidas en una nueva capa.

Convoluciones Dilatadas: Se trata de una generalización de la operación clásica de convolución. Aportan un parámetro extra que controla el rango de visión de la convolución. Se especica el número de ceros entre las casillas de la matriz del kernel, consecuentemente ampliando su tamaño y manteniendo el mismo número de variables en el aprendizaje. La localización i en la salida de esta operación se expresa de la siguiente manera:

$$y[i] = \sum_k x[i + rk]w[k]$$

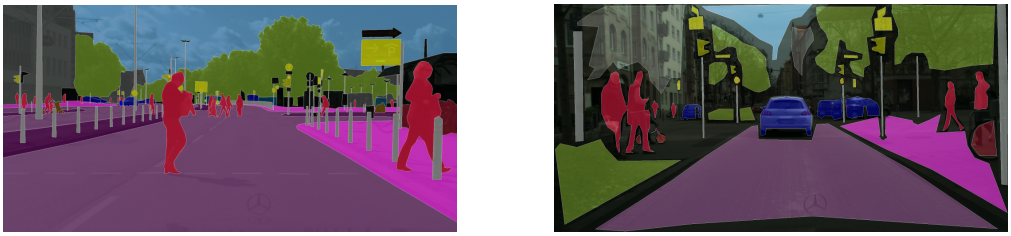
Donde, y, x son la capa de salida y entrada respectivamente, w la matriz del filtro de convolución, y r mide la dilatación.

4.1.2. Base de Datos CityScapes

Publicado en abril de 2016 la base de datos Cityscapes [2] contiene **imágenes de la escena urbana** de 50 ciudades distintas durante el día, obtenidas desde la persepectiva de un vehículo en circulación. Se guardan en un tamaño de 2048 x 1024 y capturan diferentes épocas del año (primavera, verano, otoño) bajo unas buenas condiciones climáticas(sol y nubes). Favoreciendo la claridad. Se identifican 34 clases y se proveen anotaciones tanto a nivel semántico como de instacia. Además este dataset proporciona anotaciones de dos tipos atendiendo a su precisión respecto a las imágenes originales. Por un lado hay anotaciones finas, de alta precisión a nivel píxel, y por otro lado gruesas, de más baja precisión, donde se intenanta sobreponer polígonos sobre las figuras que se anotan. Debido a ello se cuenta con 5000 imágenes anotadas de manera fina y 20000 imagenes con anotaciones gruesas.

Las clases que se identifican en el conjunto de datos:

- Plano: carretera, acerca, parking, rail track
- Humano: persona, ciclista
- Vehículo: coche, camión, autobús, on rails, moto, bicicleta, caravana, trailer
- Construcción: edificio, pared, valla, guardarrail, puente, túnel
- Objeto: Poste, grupo de postes, semáforo, señal de tráfico
- Naturaleza: vegetación, terreno,cielo
- Sin definir: estático, dinámico, suelo, sin etiquetar, fuera del área de interés, barrera de rectificación, coche



(a) Anotaciones de alta calidad

(b) Anotaciones de baja calidad

Figura 4.1: Ejemplo del tipo de anotaciones en la base de datos Cityscapes

principal (desde donde se toman las fotos).

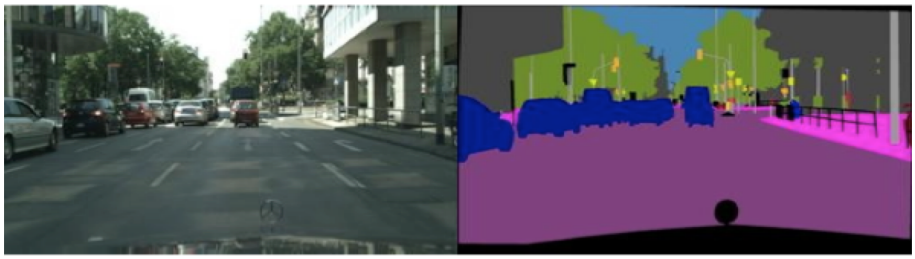


Figura 4.2: Ejemplo de peatones cruzando la calle

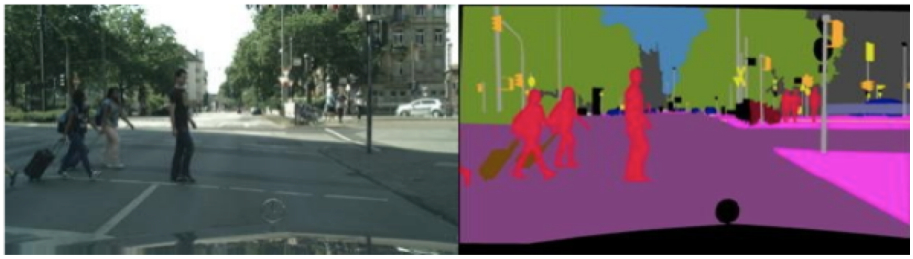


Figura 4.3: Ejemplo de calle con tráfico

Para nuestro trabajo vamos a emplear el conjunto de 5000 imágenes con anotaciones de alta calidad. Se efectúa una división del número de pares imagen/anotación en **3 subconjuntos** para distinatas tareas:

- Entrenamiento 2975
- Validación 500
- Test 1525

Esta división no se ha efectuado de manera aleatoria, pero de forma que cada conjunto pueda representar la variabilidad de los diferentes escenarios de la calle. En [2] se realiza un análisis estadístico donde se mide la complejidad de las escenas tomando como referencia el número de vehículos y peatones:

Personas $[10^3]$	Vehículos $[10^3]$	Personas/Imagen	Vehículos/Imagen
24.4	41.0	7.0	11.8

Tabla 4.1: Valor absoluto y medio del número de peatones y vehículos presentes en las particiones de entrenamiento y validación conjuntamente

Extensión de la Base de Datos

Para la tarea de detección de peatones hemos tenido que **etiquetar manualmente** el conjunto de entrenamiento y validación. La extensión ha consistido en realizar un fichero donde se recoja por cada imagen la presencia o no de un peatón sobre la calzada. Empleamos el conjunto de validación como conjunto de prueba de las reglas de detección propuestas más adelante.

	Calzada con Personas	Calzada sin Peronas
Acera con Personas	117	218
Acera sin Personas	30	135

Tabla 4.2: Número de imágenes de validación en la base de datos Cityscapes que contienen o no personas sobre la calzada o sobre la acera.

El conjunto de entrenamiento también se ha etiquetado para tener una idea del número de positivos y negativos que visualiza la red neuronal durante el aprendizaje:

	Calzada con Personas	Calzada sin Peronas
Acera con Personas	373	1156
Acera sin Personas	242	1204

Tabla 4.3: Número de imágenes de entrenamiento en la base de datos Cityscapes que contienen o no personas sobre la calzada o sobre la acera.

4.1.3. Google Colab/GPU

Google Colab es un servicio ofecido de manera gratuita por la empresa Google. Es una herramienta pensada para la eduación y la exploración del aprendizaje automático.

Consiste en un entorno de Jupyter Notebook, editor e interprete del lenguaje Python, que no requiere configuración y se ejecuta completamente en la nube. Se accede desde el navegador y se puede tanto escribir y ejecutar código como descargar o subir archivos.

Detrás de los entornos virtuales hay tres posibles configuraciones del hardware: CPU, GPU y TPU. Esto convierte a Google Colab en un gran aliado para los estudiantes ya que pone a su disposición un hardware muy potente, pero altamente costoso.

Para nuestro trabajo utilizaremos la Unidad de Procesamiento Gráfico (GPU). El modelo disponible

es la Nvidia Tesla K80 GPU. El entorno virtual que elegimos es Python 3. El cual viene provisto de todas las librerias comunes para el Aprendizaje Automático. En caso de necesitar paquetes no instaldos, pueden descargarse con el gestor de paquetes de Python de la misma forma que en un entorno local. La RAM total disponible oscila los 12 Gbits, y el almacenamiento en disco duro 350 Gbits.

Para el acceso a ficheros externos, Colab ofrece la posibilidad de cargarlos directamente. Por otro lado, existe la opción de montar Google Drive dentro del sistema de archivos del entorno. Esta combinación de Google Colab con Google Drive permite cómodamente llevar a cabo pequeños y medianos proyectos de aprendizaje automático.

Las principales limitaciones a tener en cuenta usando Colab: 12 horas máximo de ejecución seguida en GPU. Tampoco se garantiza la continuidad dentro de esas 12 horas. El número de unidades gráficas es obviamente limitado y su disponibilidad no siempre está garantizada.

4.1.4. Python Keras

El lenguaje de programación elegido es Python en su versión 3. Y las bibliotecas necesarias para efectuar el trabajo:

- Matplotlib
- Numpy
- OpenCV
- Pandas
- Keras
- Tensorflow

Tensorflow es una libreria para la programación y modelado de redes neuronales. Proporciona potentes herramientas tanto para construir modelos como para gestionar su entrenamiento. Está diseñada para ser compatible con CPU y GPU como hardware de entrenamiento.

Keras es una interfaz de alto nivel para Tensorflow. Esconde los detalles más técnicos dejando únicamente a merced del programador el diseño de las capas de la red y los parametros de entrenamiento: modo de optimización, función de pérdida, tasa de aprendizaje ...

4.2. Descripción del Experimento

Vamos a detallar los pasos llevados a cabo durante el entrenamiento del modelo Deeplab. El objetivo de esta fase es refinar lo máximo posible los parámetros de la red sobre el conjunto de imágenes dado. Comenzando desde el procesado de los datos de entrada hasta alcanzar la condición de parada del aprendizaje.

Dividimos la explicación en los siguientes subapartados:

- Conceptos Previos
- Carga del Modelo y Procesado de las Imágenes
- Fine Tuning
- Entrenamiento y Validación

4.2.1. Conceptos previos

Incluímos aquellos conceptos que se consideran más relevantes para el trabajo.

Precisión en la Segmentación de una Imagen

Al segmentar una imagen la comprobación natural es ver si las clases otorgadas a los píxeles coinciden con la realidad. Partiendo de un conociento exacto píxel a píxel (denominado ground truth) de la imagen real, puede efectuarse una medición de la precisión de las siguientes maneras:

Notación: sea k el número de clases, denotamos p_{ij} al número de píxeles de la clase i etiquetados como clase j. Es decir, p_{ii} denota el número de verdaderos positivos para la clase i.

- **Precisión píxel**: simplemente se basa en dividir el número de píxeles clasificados correctamente entre el total.
- **Media de precisión por clase**: se el calcula el ratio de aciertos, como en el caso anterior, por cada clase separadamente. Luego se obtiene la media de todas ellas. De manera esquemática:
  $MediaPorClase = \frac{1}{k}\sum_{i=1}^{k}\frac{p_{ii}}{\sum_{j=1}^{k}p_{ij}}$
- **Similitud Jaccard ò IoU**: es la métrica estandar para la segmentacion semántica basada en algoritmos de aprendizaje supervisado. Calcula el ratio entre la intersección y la unión de dos conjuntos. En nuestro caso se trata del ground truth y la segmentación efectuada.
  $Jaccard = \frac{1}{k}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=1}^{k}p_{ij}+\sum_{j=1}^{k}p_{ji}-p_{ii}}$

Función de Pérdida

Vamos a definir un concepto importante para entender como aprende una red neuronal a partir de un conjunto de datos.

Entrenar una red neuronal consiste en llevar a cabo un proceso de optimización. El objetivo es **minimizar una función de péridida**, la cual sirve como cálculo del error que se está cometiendo al predecir con el modelo. Cada capa de la red consta de una serie pesos, los cuales se van ajustando para que el valor de la función vaya disminuyendo.

Para actualizar la red con cada paso se emplea la técnica de **descenso por gradiente**. Es un método para alcanzar el mínimo local en funciones diferenciables. Para una red neuronal, la condición de diferenciabilidad debe cumplirse en todas las funciones de activación [ver 3.3] en cada capa.

Dado un problema de clasificación supervisada, definir una función de pérdida puede hacerse de

varias maneras. Una estrategia frecuentemente utilizada es el método de **máxima verosimilitud**. Consiste en dado un modelo estadístico paramétrico y un conjunto datos, en nuestro caso la red neuronal y los pares imágenes/etiquetas, obtener los valores de los parámetros que maximicen la probabilidad de que el modelo prediga los datos observados. Traducir la maximización a minimización se puede efectuar tomando el logaritmo y cambiando a signo negativo.

Envoltura Convexa

La envoltura convexa de un conjunto de puntos en un plano consiste en el conjunto convexo más pequeño que contiene a los puntos. Un conjunto convexo es aquel tal que para cada par de puntos contenidos, el segmento que los une también lo está.

4.2.2. Carga del Modelo y Procesado de los Datos

Originalmente las imágenes y las anotaciones son 2048 de ancho por 1024 de alto. Para reducir el coste de memoria durante el entrenamiento se reduce su tamaño a la mitad. Manteniendo la escala y sin deformar la imagen.

Las imágenes están a color. Al cargar en memoria se utiliza el formato RGB, donde cada uno de los tres canales corresponde a un valor en el intervalo $[0, 255]$. Como forma de **normalización** se reescalan estos valores al intervalo $[-1, 1]$. Se obtiene diviendo entre $127{,}5$ y restando 1.

También deben procesarse las máscaras. Originalmente se encuentran codificadas como imágenes a escala de gris. A cada píxel le corresponde una etiqueta, número entero entre el 0 y 34. Además de reducir su tamaño a la mitad, para coincidr con las imágenes, tienen que pasarse al formato **One-Hot-Encoding**. En consecuencia, cada máscara será interpretada como una matriz tridimensional. Se puede imaginar sencillamente que a cada punto bidimensional de la imágen le corresponde una lista ordenda toda a ceros salvo la casilla cuyo índice corresponda con la clase a la que pertence, que estará con el valor 1. Es decir, tienen un formato de 1024 x 512 x 34.

Una vez finalizado, se tiene en paralelo el conjunto de entrenamiento de imágenes que alimentar a la red y el conjunto de entrenamiento de anotaciones (máscaras) que sirven como etiqueta para el aprendizaje supervisado.

4.2.3. Fine Tuning

Antes de poder entrenar con las nuevas imágenes y máscaras se tiene que hacer compatible el formato de la red con la entrada y salida que demanda el problema. Acorde con el procesado previo, la entrada de la red debe sustituirse por otra que admita imágenes de dimensión 1024 x 512 x 3 . Para la salida usaremos el formato One-Hot-Encoding, es decir, que el formato de salida será 1024 x 512 x

34 donde 34 es el número de clases.

Mientras que la entrada puede adaptarse sin alterar ningún parametro debido a las características de la operación de convolución, la salida debe tratarse con más precaución.

La estrategia consiste en eliminar la última capa de neuronas con sus pesos y colocar una nueva salida acorde con el formato deseado. Como esta nueva 'cabeza' que se ha colocado a la red se ha inicializado de manera aleatoria podemos efecturar un primer entrenamiento de la red que únicamente afecte a los parámetros recién llegados. La razón subyacente es aprovechar que la red ha sido previamente entrenada sobre otro conjunto de datos y los pesos que se han aprendido pueden balancear los nuevos aleatorios.

En nuestro caso hemos optado por entrenar los pesos de la última capa durante 10 épocas (10 horas), sin atender a ninguna condición de parada concreta. Al terminar se obtiene una precisión píxel, número de píxeles clasificados correctamente entre el total, de 0.750.

4.2.4. Entrenamiento y Validación

Con el objetivo de refinar la red sobre el conjunto de datos Cityscapes vamos a entrenarla al completo. Actualizando ya todos los pesos en cada capa.

Para medir la calidad de la segmentación a medida que entrenamos contamos con el conjunto de validación, al margen del de entrenamiento. Consisten en 500 y 2975 imágenes/anotaciones respectivamente.

Los parámetros que se han empleado durante el entrenamiento son los siguientes:

- Tasa de entrenamiento: originalmente 0.00001, modificado a 0.00005 y 0.0001 para aligerar el proceso.
- Número de épocas: 102
- Tamaño del Batch: 1

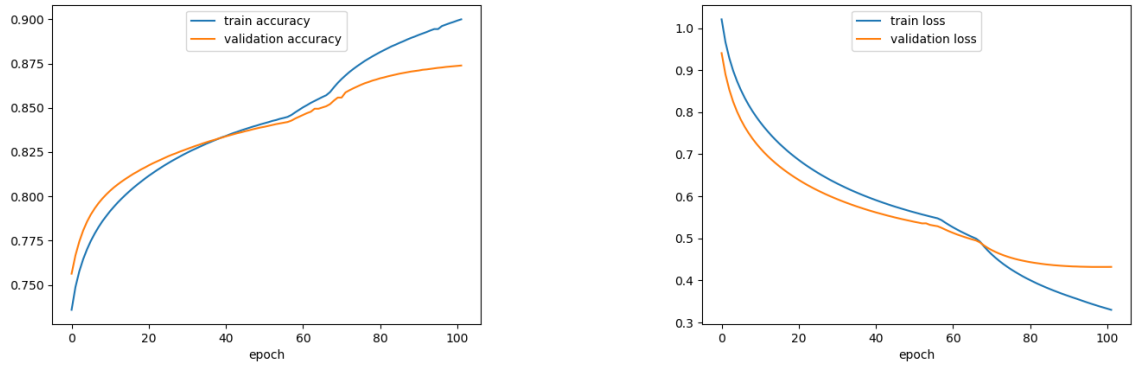
La función de péridida [ver 4.2.1] empleada es **la entropía categórica cruzada**:

$EC = -\sum_p \sum_{i=1}^{C} t_{i,p} log(f_{i,p}(x))$

Donde C es el número de categorías, p representa la variable píxel, $f_{i,p}(x)$ es la sálida de la red para el pixel p en la categoría i dada la entrada (imagen) x.

$t_{i,p}$ representa la etiqueta verdadera para el píxel p. Es decir, tiene un valor cero en todas las categorías salvo a la que pertenece el píxel, para la cual vale uno.

La métrica de precisión [ver 4.2.1] que se monitoriza es la **precisión píxel**, número de píxeles clasificados correctamente entre el total. Entrenando con la GPU de Google Colab se tarda una hora en completar cada época. En las siguientes gráficas se visualiza el progreso.



(a) Evolución de la precisión

(b) Evolución de la pérdida

Figura 4.4: Evolución de la precisión y pérdida sobre los conjuntos de entrenamiento y validación. La trayectoria azul y naranja son la progresión del entrenamiento y de la validación respectivamente

Mientras que las métricas siempre mejoran sobre el conjunto de entrenamiento, en el de validación llegamos a un punto de inflexión. En la **época** 92 se observa que la pérdida sobre el conjunto de validación ya no disminuye. Incluso comienza a aumentar ligeramente. También se estanca la mejora en la precisión. Se considera que estamos en el caso de **overfitting** sobre el conjunto de entrenamiento. El módelo empieza a ser menos efectivo a la hora de generalizar el aprendizaje. En consecuencia, se considera esta situción motivo de parada del entrenamiento.

	Pérdida	Precisión
Entrenamiento	0.3301	0.9
Validación	0.4321	0.8728

Tabla 4.4: Valores de la precisión y pérdida al final del entrenamiento.

4.3. Resultados de la Segmentación

El conjunto de datos sobre el que se evalúa el modelo Deeplab consiste en **un total de 1525 imágenes** al margen de los conjuntos de entrenamiento y validación. Las imágenes siguen el mismo estilo que el resto de la base de datos. Son fotografías tomadas desde el un automóvil durante trayectos a lo largo de diferentes ciudades.

Conjunto de test de Cityscapes

El conjunto de datos Cityscapes constituye un punto de referencia para los algoritmos de segmentación semántica. Existe la posibilidad de subir las predicciones a la web de Cityscapes para ser evaluadas. De esta manera se conforma una competición abierta, cuyos resultados se pueden visualizar

en la misma página.

Para realizar una entrega de las predicciones primero deben segmenatarse todas las imágenes del conjunto test. Los requisitos de formato están especificados:

- Único archivo zip
- Máximo 100 MB
- Las imágenes segmentadas con el mismo nombre que su versión original
- Las imágenes segmentadas deben tener ser **2048 x 1024**

Debido a limitaciones de memoria ofrecida por la plataforma Google Colab, se procedió a dividir a la mitad el formato de las imágenes. En consecuencia, para el último requisito hemos tenido que aplicar un reescalado de las predicciones obtenidas. A pesar de que se respeta el ratio entre el eje vertical y horizontal, el resultado de la transformación no será el igual de preciso. En nuestro caso, hemos utilizado la **interpolación por el vecino más cercano**.

Una vez cumplidos y enviado, se recibe un link al resumen con los valores obtenidos.

Intersección sobre Unión

La métrica de evaluación que se sigue en la competición de Cityscapes es la similitud Jaccard o intersección sobre unión [ver 4.2.1]. Los resultados se expresan en términos de las clases.

Clase	**IoU**
Carretera	94.9304
Acera	66.8439
Edificio	84.5573
Pared	27.4953
Valla	25.5778
Poste	27.4209
Semáforo	27.4209
Señal de tráfico	42.5245
Vegetación	86.8174
Terreno	57.1779
Cielo	89.2198
Persona	64.196
Ciclista	32.6026
Coche	88.2478
Camión	28.2226
Autobús	39.1838
Tren	37.923
Motocicleta	39.0993
Bicicleta	53.4582
Media	53.1844

Tabla 4.5: Índice Jaccard (IoU) por cada clase

Si en su lugar expresamos los resultados en función de categorías más generales.

Categoría	**IoU**
Superficie plana	95.8787
Naturaleza	86.019
Objeto	33.2244
Cielo	89.2198
Construcción	84.7612
Humano	66.5439
Vehículo	87.5307
Media	77.5968

Tabla 4.6: Índice Jaccard (IoU) por cada categoría

4.4. Detección de Peatones Cruzando la Calle

El experimento de detección de peatones consistió en la segmentación semántica de las 500 imágenes del conjunto de validación de Cityscapes mediante el modelo entrenado en la fase anterior. Sobre la salida se aplicaron las reglas heurísticas mencionadas en 3.1.

Resumidamente los datos verdaderos con los que trabajamos son los siguientes:

	Calzada con Personas	Calzada sin Peronas
Acera con Personas	117	218
Acera sin Personas	30	135

Tabla 4.7: Número de imágenes de validación en la base de datos Cityscapes que contienen o no personas sobre la calzada o sobre la acera.

Es decir, existen 147 imágenes donde se debe detectar peatón sobre calzada y 353 imágenes donde no.

Resultados

A continuación se muestran los resultados obtenidos en forma de matriz de confusión. Consideramos:

- VP = Verdadero Positivo, existe un peatón cruzando la calle y es detectado.
- FP = Falso Positivo, no existe un peatón cruzando la calle y se detecta una presencia de peatón.
- VN = Verdadero Negativo, no existe un peatón cruzando la calle y no se detecta nada.
- FN = Falso Negativo, existe un peatón cruzando la calle y no se detecta nada.

	Persona en Vía	Ninguna Persona en Vía
Detección	VP	FP
No Detección	FN	VN

Tabla 4.8: Matriz de confusión de ejemplo

Además se incluyen las métricas de Precisión, Recall y Valor-F.

- $Precision = \frac{VP}{VP+FP}$
- $Recall = \frac{VP}{VP+FN}$
- $ValorF = 2 * \frac{Precision*Recall}{Precison+Recall}$

Solapado Persona sobre Carretera

Se observa si existe solapamiento entre las clases persona y carretera tras su segmentación.

	Persona en Vía	Ninguna Persona en Vía
Detección	113	81
No Detección	34	272

Tabla 4.9: Matriz de confusión para la regla de solapado entre clases

- $Precision = 0{,}58$
- $Recall = 0{,}77$
- $ValorF = 0{,}66$

Búsqueda de Frontera

Búsqueda de la frontera entre la clase persona y calle que dejen en la parte de arriba píxeles persona y debajo píxeles calle.

	Persona en Vía	Ninguna Persona en Vía
Detección	88	44
No Detección	59	309

Tabla 4.10: Matriz de confusión para la regla de búsqueda de frontera

- $Precision = 0{,}66$
- $Recall = 0{,}60$
- $ValorF = 0{,}63$

5

Análisis de Resultados y Conclusiones

En este trabajo hemos presentado un método para la detección totalmente automática de peatones sobre la calzada. Además hemos extendido la base de datos Cityscapes [2] con información adicional.

Consideramos que el enfoque propuesto de combinar una red neuronal convolucional con reglas deterministas puede tener interés en determinados problemas. La flexibilidad en el formato de imagen que puede entender una red de este tipo hace que sea un método fácil de integrar en otros sistemas. Además, a pesar del alto coste computacional que requiere el entrenamiento, una vez acabado las predicciones se realizan con relativa rapidez.

Los resultados obtenidos en la fase de segmentación se observa un desiquilibrio a favor de los objetos de mayor tamaño. Mientras que elementos grandes como por ejemplo calle, edificio o vehículo, son segmentados con alta precisión, los de menor tamaño obtienen unos resultados muy bajos. Esto dificulta que el modelo entrenado sirva para tareas de reconocimiento de escenas más exigentes.

Teniendo en cuenta que el modelo líder en la competición de segmentación de Cityscapes obtiene los siguientes resultados:

- Media del índice Jaccard por clase: $83{,}6$
- Media del índice Jaccard por categoría: $64{,}7$

Consideramos que nuestros resultados 4.3 obtenidos admiten amplia mejora y debe profundizarse en la exploración de diferentes reglas heurísticas para la clasificación. Por ello planteamos posibles trabajos futuros.

Trabajos Futuros

Uno de los principales problemas a solucionar que se han encontrado es la visualiazación de peatones lejanos. Cuanto más cercana es la escena mejores predicciones se realizan, como cabe esperar. Una opción a valorar consistiría en acotar la distancia a la que se detectan las personas cruzando la calzada. Sería un sistema con menos alcance, pero más robusto. Esta solución puede ser de gran interés en situaciones donde los falsos positivos o negativos, una falsa alarma, provoquen una situación de riesgo.

Por otro lado, creemos que la aplicación de reglas de post procesado de los datos al salir de la red puede influir en una mejora general del sistema. Esto no ha sido estudiado y debería incluirse en caso de revisión del método.

Teniendo en cuenta la complejidad en la tarea de segmentación, es razonable pensar que ampliar la fase de entramiento de la red neuronal repercuta positivamente en la calidad de las predicciones. Existen más bases de datos para la segmentación de imagenes de la vía pública. Fusionar de alguna manera el concimiento adquirido hasta este punto con un nuevo proceso de aprendizaje sobre un nuevo conjunto de datos, podría conllevar una mejor segmentación, más robusta frente a la variabilidad de escenas y como consecuencia mejorar el sistema de detección de peatones.

Bibliografía

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. apr 2016.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. nov 2014.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. sep 2014.

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. apr 2017.

[6] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. may 2015.

[7] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. nov 2015.

[8] Xide Xia and Brian Kulis. W-Net: A Deep Model for Fully Unsupervised Image Segmentation. nov 2017.

[9] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent Advances in Autoencoder-Based Representation Learning. dec 2018.

[10] Dibya Jyoti Bora and Anil Kumar Gupta. Clustering Approach Towards Image Segmentation: An Analytical Study. jul 2014.

[11] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, sep 2013.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. jun 2015.

[13] Inyong Yun, Cheolkon Jung, Xinran Wang, Alfred O Hero, and Joongkyu Kim. Part-Level Convolutional Neural Networks for Pedestrian Detection Using Saliency and Boundary Box Alignment. oct 2018.

[14] Samir Ibadov, Ragim Ibadov, Boris Kalmukov, and Vladimir Krutov. Algorithm for detecting violations of traffic rules based on computer vision approaches. *MATEC Web of Conferences*, 132:05005, oct 2017.

[15] Joko Hariyono and Kang-Hyun Jo. Detection of pedestrian crossing road. 09 2015.

[16] Alberto Broggi, Pietro Cerri, Stefano Ghidoni, Paolo Grisleri, and Ho Gi Jung. A new approach to urban pedestrian detection for automatic braking. *IEEE Transactions on Intelligent Transportation Systems*, 10:594–605, 2009.

[17] Ying CAO, Qi-Guang MIAO, Jia-Chen LIU, and Lin GAO. Advance and prospects of adaboost algorithm. *Acta Automatica Sinica*, 39(6):745 – 758, 2013.

[18] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. nov 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. jun 2014.