XLIV Jornadas de Automática 2023

https://jautomatica.es

Segmentación semántica de imágenes para navegación de vehículos autónomos en entornos estructurados

Tornero, P.[a,*], Yagüe, D.[a], Armingol, J. M.[a], de la Escalera, A.[a]

[a]*Laboratorio de Sistemas Inteligentes, Universidad Carlos III de Madrid Avda de la Universidad, 30, 28911, Leganés, Madrid, España*

To cite this article: Tornero, P., Yagüe, D., Armingol, J.M., de la Escalera, A. 2023. Semantic image segmentation for autonomous vehicle navigation in structured environments.
XLIV Jornadas de Automática, 891-896. https://doi.org/10.17979/spudc.9788497498609.891

Resumen

El avance e inclusión de la Inteligencia Artificial en la sociedad ha permitido que tareas como la visión por computador puedan mejorar notablemente su desempeño y adaptabilidad a tareas de todo tipo. Cada vez más se encuentran soluciones basadas en IA que automatizan trabajos antes realizados por humanos. Sin embargo, labores sencillas para una persona como la comprensión del entorno que le rodea, identificando sus elementos, siguen siendo desafiantes para las computadoras, pero lograrlo tendría múltiples beneficios. Dado este planteamiento, el objetivo de este trabajo es el estudio del estado del arte actual sobre los métodos de segmentación semántica y desarrollar un sistema de visión artificial, que haga uso de los métodos analizados, para la navegación de un vehículo autónomo en un campus universitario. El sistema permitirá al vehículo conocer su entorno, planificar su movimiento y prever peligros, mejorando la seguridad y objetividad de la conducción. Además, se ha trabajado con una base de datos de imágenes en alta resolución del campus para probar y validar el sistema, logrando una mejor integración con el entorno real de operación.

Palabras clave: Inteligencia Artificial, Segmentación semántica, Deep learning, Visión por Computador, Vehículo Autónomo.

Semantic image segmentation for autonomous vehicle navigation in structured environments

Abstract

The development and inclusion of AI in society has allowed tasks such as computer vision to significantly improve their performance and adaptability to tasks of all kinds. Increasingly, AI-based solutions are being found that automate jobs previously performed by humans. However, simple tasks for a person such as understanding the surrounding environment, identifying its elements, remain challenging for computers, but achieving this would have multiple benefits. Given this approach, the objective of this work is to study the current state of the art on semantic segmentation methods and to develop a computer vision system, which makes use of the analyzed methods, for the navigation of an autonomous vehicle on a university campus. The system will allow the vehicle to know its environment, plan its movement and foresee dangers, improving the safety and objectivity of driving. In addition, we have worked with a database of high-resolution images of the campus to test and validate the system, achieving a better integration with the real operating environment.

Keywords: AI, Semantic segmentation, Deep learning, Computer vision, Autonomous vehicle.

1. Introducción

En los últimos años, el campo de la visión por computador se ha ido popularizando a un ritmo desenfrenado, no solo aportando nuevas y mejores técnicas y soluciones, sino aumentando la usabilidad de sus herramientas en aplicaciones reales. El volumen de mercado ha alcanzado un valor de 12.178 millones de dólares en 2021 y se espera que pueda alcanzar los 205.104 millones de dólares en 2030, creciendo con un CAGR (Tasa de crecimiento anual compuesto) del 37,05 % desde 2023 hasta 2030 (Verified Market Research, 2023).

*Autor para correspondencia: ptornero@pa.uc3m.es

En la tarea que concierne a este trabajo, correspondiente a la segmentación semántica de imágenes, la aparición de las técnicas de aprendizaje profundo produjo un gran cambio en el sector, ya que este problema era complicado de solucionar con las técnicas tradicionales de visión por computador. Las nuevas soluciones aportaron, además de una mayor facilidad de implementación, una notable mejoría en el nivel de acierto y velocidad de los algoritmos para realizar la tarea.

Una de las aplicaciones capaz de aprovechar estas mejoras es la de los vehículos autónomos. De forma análoga a la visión por computador, la industria de los vehículos autónomos está en plena expansión, y en ésta se trata de lograr que un vehículo pueda alcanzar un grado de independencia del operador humano cada vez más alto. Algunos ejemplos de los últimos vehículos con estas características presentados son el Model S de Tesla, el Proyecto Waymo de Google y el Elaina Concept de Audi (Terol, 2021).

En este punto es donde toma un alto grado de importancia la segmentación semántica, porque en los vehículos autónomos se requiere tener un conocimiento del entorno en el que se están desplazando, para de esta manera, poder adaptar su trayectoria, detenerse en caso de peligro o predecir diferentes situaciones de riesgo. Esta es una tarea que para las computadoras se torna en una mucho más compleja que para los seres humanos en la mayoría de casos. No obstante, un desarrollo positivo aportaría un gran número de beneficios para la conducción, como un mayor nivel de seguridad o invariabilidad ante factores cambiantes de los seres humanos, como el estado de ánimo o embriaguez, entre otros.

Pese a que en los últimos años se han desarrollado diferentes aproximaciones para la segmentación semántica en vehículos autónomos, todas estas están orientadas a la operación en entornos de carretera, ya sean urbanos o no. En este trabajo se realiza la adaptación de una técnica de segmentación semántica orientada a vehículos autónomos en el entorno de un campus de universidad, donde el entorno presenta determinadas variaciones respecto al de un contexto común de carretera, que deben ser tenidas en cuenta a la hora de desarrollar el sistema.

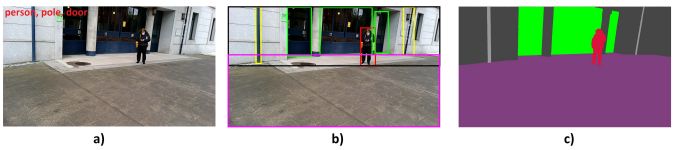

Figura 1: Comparación entre determinadas técnicas de visión por computador. a) reconocimiento, b) detección y reconocimiento, c) segmentación semántica.

En cuanto al porqué se ha escogido segmentación semántica frente a otras técnicas de visión, como podrían ser la detección o reconocimiento de objetos, se encuentra el hecho de que mediante la segmentación se tiene información a nivel de píxel, es decir, mucho más precisa que la información englobada en una *bounding box* que puede contener multitud de píxeles de otra categoría de objetos, o simplemente haber partes de la imagen englobadas en varios *bounding boxes* simultáneamente, siendo un ejemplo práctico el de la Figura 1. Sin embargo, esto tiene un inconveniente, y es que la complejidad es mayor, por lo que requiere un mayor tiempo de procesamiento, siendo este un factor a tener en cuenta para que no suponga una limitación al funcionamiento del sistema.

Para desarrollar este trabajo, se ha llevado a cabo un estudio de los últimos proyectos y mecanismos desarrollados en la segmentación semántica orientada a vehículos autónomos, con el objetivo de dar con la más adecuada de las implementaciones para los requerimientos que se han definido. Estas especificaciones son las de tener un nivel de acierto suficiente que vaya en equilibrio con la velocidad de inferencia, que es un factor vital, teniendo como objetivo que el sistema pueda funcionar en una frecuencia de, al menos, entre 5 y 10 Hz, para que la información sobre el entorno se renueve continuamente y se puedan detectar los peligros y solucionarlos antes de que ocurran. Este rango de valores ha sido calculado a partir de las especificaciones del vehículo autónomo en el que se van a llevar a cabo los experimentos, el iCab (Marin Plaza et al., 2021), el cual no supera la velocidad de 20 km/h, desarrollado por el Laboratorio de Sistemas Inteligentes (LSI) de la Universidad Carlos III de Madrid. Una vez se haya seleccionado el método a utilizar, se confecciona un *dataset* personalizado con imágenes del campus de la Universidad Carlos III de Madrid y se realizan pruebas con diferentes parámetros de entrenamiento para obtener, de esta manera, un modelo lo más cercano posible al ideal para este trabajo.

2. Estado del arte

Hasta la llegada de las soluciones que hacen uso de técnicas de aprendizaje profundo, la tarea de la segmentación semántica se realizaba mediante el uso de un clasificador que podía ser por ejemplo, *random forest* (Shotton et al., 2008), el cual tomaba las características obtenidas en una pequeña región rectangular de la imagen y otorgaba una etiqueta a toda esta región, aunque la salida obtenida poseía una gran cantidad de ruido.

Posteriormente, tras la consecución de nuevos avances, se mejoró el rendimiento de los sistemas mediante el etiquetado de todo un segmento de la imagen en lugar de sólo una pequeña parte o el píxel central (Kontschieder et al., 2011) (Tighe and Lazebnik, 2010). Una gran diferencia es que, estos segmentos, se obtienen haciendo uso de un método que genera grupos de píxeles similares, según criterios de gradiente y color, de forma que tengan relación con los contornos de los elementos en la imagen.

Con la llegada del aprendizaje profundo se revolucionó toda la visión por computador, y tras el éxito en tareas de clasificación y detección de objetos, el uso de estas técnicas se convirtió en el nuevo camino a seguir en tareas de segmentación semántica. La investigación se centró en el desarrollo de métodos para reducir el tiempo de inferencia necesario, a la vez que no se vea afectado el nivel de acierto o que la reducción sea mínima. Algunos de los más destacados son los siguientes (Papadeas et al., 2021):

- **Uso de *decoders* de tamaño reducido**. La arquitectura *encoder-decoder* es una de las más utilizadas en segmentación semántica. Se propone simplificar el tamaño del *decoder* de forma que la computación de la información se realice de manera más sencilla. En la propuesta de (Paszke et al., 2016) se plantea que el único rol del *decoder* sea realizar un *upsampling* de la salida del *decoder* perfeccionando los detalles. En general, esta

propuesta resulta atractiva ya que la reducción del tamaño del *encoder* no suele afectar a la efectividad del sistema.

- ***Early Downsampling***. En el modelo de arquitectura de ENet (Paszke et al., 2016), se afirma que lo mejor para aligerar el proceso es realizar un *downsampling* de los datos de entrada para reducir el coste computacional. Es necesario prevenir la pérdida de datos como en la propuesta de SegNet (Badrinarayanan et al., 2017), donde se guardan los índices de los elementos seleccionados en las capas de *max-pooling* para una reconstrucción de los datos posterior usando menos memoria.

- ***Multi-branch network***. Se ha propuesto mediante una red de dos caminos alcanzar un balance entre el acierto y el tiempo de inferencia, donde uno captura y genera de forma más detallada características a una alta resolución, siendo más complejo y profundo, consistiendo en algunas convoluciones (Yu et al., 2020), mientras que el otro captura características semánticas de alto nivel a menor resolución, siendo un *encoder* sencillo. Además, de esta manera se guarda información parcial que se perdía en las operaciones de *downsampling*, combinando características de bajo y alto nivel, lo cual aumenta la efectividad. Este método es uno de los más comúnmente utilizados y su estructura se puede ver en la Figura 2.

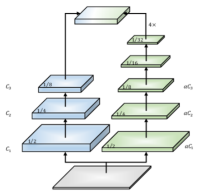

Figura 2: Ejemplo de arquitectura de una red del tipo multi-branch network.

Una vez mencionadas las herramientas desarrolladas para la mejora de la eficiencia de las técnicas de segmentación semántica, a continuación, se listan algunos ejemplos de los modelos que se han ido presentando en los últimos años gracias a los avances en la investigación:

- **SS** (Wu et al., 2017). SS es una *multi-branch network* que utiliza una de sus partes para recibir la imagen de entrada y otra para recibir la imagen de entrada pero con la mitad de resolución. SS introduce la dispersión espacial, con objetivo de reducir el coste computacional en un factor de 25, con conexiones en la propia columna (ese *branch*) y entre columnas (un *branch* con otro) eliminando unidades residuales.

- **ICNet** (Zhao et al., 2017). Propone un *framework* para guardar las operaciones en múltiples resoluciones e incluye una unidad de fusión de características en cascada. Procesa simultáneamente información semántica de *branches* de baja resolución junto con detalles de imágenes de alta resolución de una manera eficiente.

- **DFANet** (Li et al., 2019). Este modelo se centra en la agregación de *Deep Learning* haciendo uso de varios caminos de *encoders* interconectados y características codificadas del contexto en alto nivel.

Finalmente, una vez tratados los avances en cuanto a las técnicas, modelos y arquitecturas, se centra la atención en los *datasets* existentes y más utilizados como *benchmarks*. El primero de ellos es CamVid (Ronneberger et al., 2015), el cual cuenta con un total de 701 imágenes con una resolución de 960x720 píxeles, donde se pueden encontrar 32 categorías diferentes como pueden ser coche, carretera, viandante, etc. El segundo de ellos es Cityscapes (Cordts et al., 2016), el cual cuenta con un total de 5000 imágenes con una resolución de 2048x1024 píxeles, etiquetadas de manera detallada, que han sido tomadas en 50 ciudades diferentes de Alemania. En ambos *datasets* las instantáneas están captadas desde la parte frontal de un coche en entornos de carretera, principalmente en zona urbana.

Figura 3: Ejemplo de las imágenes contenidas en el dataset Cityscapes.

3. Modelo seleccionado

Tras el estudio desarrollado en el apartado anterior, se llegó a la conclusión de que el método adecuado era el de usar la red DDRNet (Hong et al., 2021). En la Figura 4 se pueden visualizar métricas de acierto y eficiencia de diferentes modelos, donde el que se ha escogido es DDRNet-23-slim, que es una versión más ligera de DDRNet, la cual sacrificando en torno a un 2 % de mIoU (mean Intersection over Union) logra tener una velocidad mucho mayor que su versión más pesada. Además de ser la más rápida de la lista, ocupa la cuarta posición en cuanto a nivel de acierto, lo cual cumple con el balance que se buscaba entre acierto y velocidad.

Networks (Backbone)	Cityscapes			CamVid		
	mIoU (%)	FPS	Hardware	mIoU (%)	FPS	Hardware
SQ (SqueezeNet)	**84.3**	16.7	Jetson TX1	-	-	-
DDRNet-23	79.4	38.5	GTX 2080Ti	**79.9**	94	GTX 2080Ti
HyperSeg-S (EfficientNet-B1)	78.1	16.1	GTX 1080TI	78.4	38.0	GTX 1080TI
DDRNet-23-slim	77.4	108.8	GTX 2080Ti	78.0	230	GTX 2080Ti
LinkNet (ResNet18)	76.4	18.7	Titan X	-	-	-
U-HarDNet-70 (DenseNet)	75.9	53	GTX 1080Ti	67.7	149.3	Titan V
HyperSeg-M (EfficientNet-B1)	75.8	36.9	GTX 1080Ti	-	-	-
SwiftNetRN-18 (ResNet-18, MobileNet V2)	75.5	39.9	GTX 1080Ti	73.86	43.3	GTX 1080Ti
TD4-BISE18 (TDNet)	74.9	47.6	Titan Xp	74.8	59.2	Titan Xp
ShelfNet18 (ResNet, Xception, DenseNet)	74.8	59.2	GTX 1080Ti	-	-	-
BiSeNet (ResNet18)	74.7	65.5	Titan Xp	68.7	-	Titan Xp
SegBlocks-RN18 (t = 0.4)	73.8	48.6	GTX 1080Ti	-	-	-
SS (ResNet18)	72.9	14.7	GTX 980	-	-	-

Figura 4: Comparativa de diferentes modelos de aprendizaje profundo en cuanto a acierto, medido en mIoU (mean Intersection over Union) y tiempo de inferencia, medido en FPS (frames por segundo procesados). Las métricas han sido probadas en los datasets Cityscapes y CamVid.

DDRNet consiste en una arquitectura formada por dos caminos diferentes, donde se van dando fusiones de características entre ambos en algunos puntos, hasta llegar al resultado final uniendo las características de distintos niveles conseguida en cada camino (Figura 5).

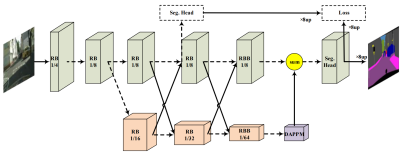

Figura 5: Arquitectura de DDRNet.

Para la fusión de características se hace uso de la llamada por los autores (Hong et al., 2021) "fusión bilateral", la cual combina la información del camino de baja resolución y el de alta resolución. Además, en la parte final de la red se incluye un extractor de información contextual llamado *Deep Aggregation Pyramid Pooling Module* (DAPPM), para ampliar el campo receptivo fusionando contexto multi escala (de diferentes escalas de la misma imagen) basados en mapas de características de baja resolución.

4. Desarrollo del sistema

Una vez se tiene ya el modelo de red que se quiere utilizar para realizar el trabajo, es necesario confeccionar un *dataset* a medida para la tarea que se quiere llevar a cabo. Esto es necesario, ya que, como se ha mencionado anteriormente, el entorno en el que se va a desarrollar el trabajo dista del reflejado en los *datasets* existentes, cambiando los elementos que son más comunes de encontrar, como personas por el camino de forma común, más vegetación, y más variedad de terrenos concebidos como transitables.

En la Figura 6 se pueden ver ejemplos de las imágenes contenidas en la base de datos. Esta cuenta con un total de 550 instantáneas con una resolución de 1920x1080 píxeles, las cuales están tomadas en el campus de la Universidad Carlos III de Madrid, mediante una cámara OAK-1 y desde el vehículo autónomo iCab (Marin Plaza et al., 2021) automatizado por el LSI. Para el proceso de captura de las mismas se han grabado diferentes vídeos y se han guardado instantáneas cada segundo y medio de tiempo. Con el objetivo de poder describir el entorno de operación se definieron 16 clases diferentes a las cuales puede pertenecer cada elemento, listadas en la Tabla 2.

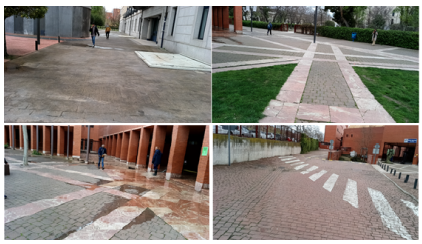

Figura 6: Ejemplo de los diferentes escenarios que se pueden encontrar en el campus.

Para el etiquetado de las fotos se hizo uso de la herramienta que provee Cityscapes (Cordts, 2022), la cual permite definir polígonos mediante diferentes puntos a mano y asignarles una etiqueta, que más tarde se pasa al formato requerido por el modelo para su entrenamiento. Cabe destacar que para un correcto etiquetado se han seguido unas buenas prácticas, entre las que se encuentra que es imprescindible definir lo mejor posible las fronteras entre cada elemento sin dejar puntos intermedios sin asignar, para lograr un correcto aprendizaje y testeo. Además, se ha etiquetado todo aquello que es distinguible para un operador humano, dejando, por ejemplo, elementos poco distinguibles a los lejos como parte del fondo.

Una vez finalizado el *dataset* y debido al relativamente bajo número de imágenes que lo componen se propuso aplicar la técnica de *data augmentation*, la cual es comúnmente utilizada para aumentar el número de ejemplos de una base de datos, tal y como se hace en (Sellat et al., 2022). Este método consiste en que, con el fin de multiplicar el número de imágenes rápidamente y aportar variedad, se le aplican diferentes operaciones a las fotos originales, como rotaciones, cambios de iluminación, traslaciones, etc. En este caso, se debe tener el cuenta que las nuevas imágenes deben ser representativas del entorno operacional, siendo un ejemplo de transformación no válida una rotación excesiva o un volteo sobre el eje horizontal, ya que el cielo quedaría en la parte de abajo y eso es una situación que nunca debería darse porque el entrenamiento no estaría aprendiendo de casos reales.

Las operaciones que se han seleccionado para el aumento de datos son: volteo sobre el eje vertical, aumento y disminución de la iluminación, aumento y disminución de la saturación, giro de 6 y -6 grados, zoom hacia la parte superior izquierda y zoom hacia la parte inferior derecha. De esta manera el número total de muestras del *dataset* aumentado es de 5500 imágenes.

5. Pruebas y resultados

En este apartado del documento se detallan las pruebas que se han llevado a cabo con el fin de obtener el mejor modelo posible que se ajuste a los requerimientos. Los experimentos se han realizado usando una GPU Nvidia GeForce GTX 1660 Ti, y el conjunto de datos se ha distribuido de manera que el de entrenamiento cuenta con el 67 % de las imágenes, el de validación el 16,5 % y el de test otro 16,5 %, todas ellas repartidas aleatoriamente y a una resolución de 1280x720 píxeles.

En primer lugar, se estudian medidas cuantitativas del acierto junto a valoraciones cualitativas de la segmentación realizada, y en segundo lugar, se analiza el desempeño en cuanto al tiempo de inferencia ante diferentes resoluciones de la imagen, buscando el equilibrio entre velocidad y pérdida de información. Las medidas cualitativas que se usan son:

- ***Pixel Accuracy*** **(PA)**: ratio de píxeles correctamente clasificados frente al número total de los mismos.

- ***Mean Pixel Accuracy*** **(mPA)**: ratio de los píxeles correctos basado en cada clase, realizando la media entre el valor de todas las clases:

- ***Intersection over Union*** **(IoU)**: se define como la intersección del mapa de segmentación predicho y el *ground truth*, dividido entre el área de unión entre el mapa de segmentación predicha y el *ground truth*.

- ***Mean Intersection over Union*** **(mIoU)**: es la medida de acierto más utilizada en segmentación semántica. Se define como la media de IoU entre todas las clases.

En las pruebas realizadas se han usado tanto el *dataset* normal como el aumentado y se han mantenido los hiperparámetros de entrenamiento por defecto, a excepción de los que se especifican en la Tabla 1, para observar cómo repercute cada uno de ellos en los resultados y buscar las mejores métricas.

Antes de comentar las pruebas, cabe destacar que en este caso, el valor de PA es muy alto en todos los experimentos, y esto es debido a que, como se puede ver en la Tabla 2 el acierto es más alto en las categorías que ocupan muchos píxeles en la imagen, como *road* o *building*. Como esta métrica va en función de los píxeles acertados frente a los totales, el valor que toma PA es muy alto, aunque no necesariamente indica un buen

aprendizaje, tal y como se demuestra con su media mPA que es bastante más baja. Por esta razón las métricas que se tienen en cuenta son mIoU y secundariamente mPA.

Tabla 1: Parámetros de las diferentes pruebas que se estudian y sus correspondientes métricas. P significa la prueba correspondiente, Ep significa el número de épocas, Ori significa dataset original, Aum el dataset aumentado y LR significa valor de learning rate.

P	Ep	Dataset	LR	mIoU	PA	mPA
1	50	Ori	0,1	52,15 %	95,96 %	58,32 %
2	50	Ori	0,01	64,62 %	97,69 %	71,78 %
3	50	Aum	0,01	75,18 %	98,73 %	80,35 %
4	250	Ori	0,01	74,08 %	98,24 %	79,77 %
5	250	Aum	0,01	76,78 %	98,85 %	81,83 %
6	350	Aum	0,01	78,12 %	98,98 %	82,61 %

A continuación, se analizan los resultados obtenidos (Tabla 1). En cuanto a los de las Pruebas 1, 2 y 3 se puede notar una clara mejoría entre cada una de ellas pese a que en todas se ha utilizado el mismo número de épocas de entrenamiento. De la Prueba 1 a la 2 la disminución del *learning rate*, el cual en la Prueba 1 se había puesto más alto a propósito, provoca una clara mejora debida a que el modelo aprende de manera más lenta pero tiene una mayor probabilidad de converger a un buen aprendizaje. Entre la Prueba 2 y 3 se cambia el *dataset* del normal al aumentado y los resultados siguen mejorando notablemente, ya que, de esta manera, el modelo tiene más ejemplos de los que aprender y por ende mejora su desempeño, aunque hay que tener cuidado con que esto no desemboque en un sobreentrenamiento.

En cuanto a las Pruebas 4, 5 y 6 se puede ver que ante las mismas condiciones de entrenamiento, el *dataset* aumentado sigue ofreciendo mejores resultados (caso de comparar las Pruebas 4 y 5) y al aumentar el número de épocas en la Prueba 6 las métricas siguen mejorando. No se ha seguido aumentando el número de épocas de entrenamiento ya que, además de no querer caer en el problema del sobreentrenamiento, se ha considerado suficientemente bueno el valor de mIoU obtenido debido a que es muy similar al reportado por los autores de DDRNet.

Observando la Tabla 1 y la Tabla 2 se pueden notar ciertos datos curiosos. El primero de ellos es que, pese a que en la Prueba 4 se llevan a cabo 250 épocas de entrenamiento frente a las 50 de la Prueba 3, las métricas generales son mejores en la Prueba 3, lo que demuestra la importancia de usar un *dataset* aumentado. Otra curiosidad es que, pese a que las pruebas con el *dataset* aumentado devuelven mejores resultados generales, en los experimentos que hacen uso de la base de datos original, hay clases que tienen un valor de IoU superior, tales como *pole*, *sky* o *traffic sign*.

Esto último puede deberse a que mediante las transformaciones del aumento de datos se altere la naturaleza de algunas clases. Por ejemplo, mediante los giros de imagen los postes (pole) ya no están totalmente verticales. Otra posibilidad es que mediante el zoom algunos objetos pequeños situados en los laterales de la imagen desaparezcan, apareciendo en un porcentaje de imágenes menor, como podría ser el caso de *traffic sign*. Sin embargo, pese a que esto ocurra, no es algo preocupante ya que la disminución del acierto no es elevada.

Una vez vistas las métricas de efectividad, se muestran algunos ejemplos cualitativos de la segmentación que realiza el modelo ante las imágenes de test comparándolas con el *ground truth* (Figura 7). Se puede notar que estos concuerdan con las métricas obtenidas, siendo la Prueba 6 la mejor y subjetivamente suficiente para desempeñar la tarea correctamente.

Una vez realizadas las pruebas de efectividad se lleva a cabo una experimentación con diferentes resoluciones de la imagen de entrada (Tabla 3), con el objetivo de comprobar a cuántos FPS es capaz de funcionar y buscar una relación resolución-velocidad adecuada mientras no se pierda información importante. Todos los experimentos cumplen con los requisitos iniciales de poder trabajar en el rango mínimo de 5 a 10 Hz, sin embargo, con el objetivo de optimizar al máximo el sistema, se realiza una apreciación cualitativa para decidir cuál es la más viable (Figura 8).

Tabla 3: Resultados de tiempo obtenidos al realizar la segmentación a diferentes resoluciones

Resolution	Time ms	FPS
1280x720	54	18,51
960x540	40	25
768x432	36	27,77
640x360	35	28,57
512x288	31	32,26
320x180	30	33,33

Como se puede ver en la Figura 8 cuando la resolución es de 512x288 o menos, aparentemente, los detalles comienzan a perderse y con esto precisión en la segmentación. Debido a esto,

Tabla 2: Resultados del IoU de cada clase en cada prueba.

Prueba	Road	Building	Wall	Fence	Pole	Traffic sign	Vegetation	Terrain	Sky	Person	Car	Dog	Door	Static	Unlabeled	Bicycle
1	97,45	86,56	90,63	74,60	54,81	7,14	86,92	90,77	66,97	48,69	44,75	0,00	53,57	61,42	0,00	22,16
2	98,46	92,42	93,94	86,06	66,41	35,63	92,89	93,88	73,20	74,35	66,32	0,00	77,55	74,63	5,85	66,93
3	99,43	96,74	95,21	90,45	71,65	33,68	95,57	94,55	71,59	84,44	80,13	34,72	94,38	83,23	64,77	87,56
4	98,72	94,36	94,17	89,05	74,85	52,26	94,52	94,36	78,93	82,47	71,57	23,30	89,53	83,76	55,26	82,37
5	99,48	97,14	95,29	90,97	74,49	34,69	96,02	94,96	72,23	86,45	81,40	39,80	95,51	86,40	70,45	89,55
6	99,55	97,61	95,53	91,72	77,53	35,39	96,55	95,25	74,26	87,51	84,67	42,42	96,00	87,44	75,00	90,69

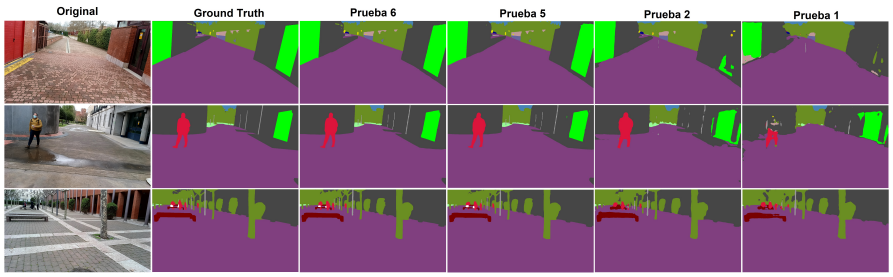

Figura 7: Resultados cualitativos de algunas pruebas respecto al *ground truth*.

lo lógico podría ser seleccionar 640x320 como resolución final, sin embargo, atendiendo a Tabla 3 la diferencia con la resolución de 768x432 es de apenas 1 FPS, por lo que se selecciona como resolución para el sistema la de 768x432 ya que se estima que por tan poca diferencia no merece la pena perder más detalle.

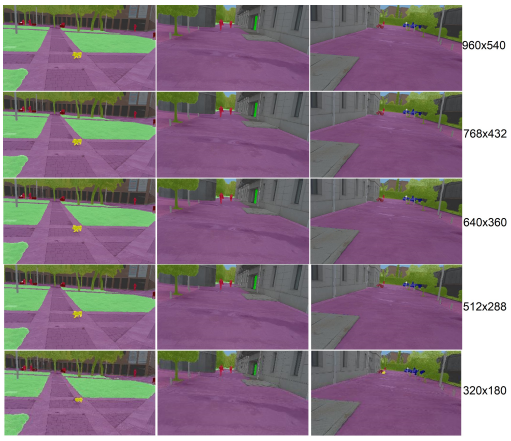

Figura 8: Ejemplo de segmentaciones a diferentes resoluciones

Agradecimientos

Subvención PID2019-104793RB-C31, PDC2021-121517-C31, PDC2022-133684-C31 and PID2021-124335OB-C21 financiados por MCIN/AEI/ 10.13039/501100011033 y por la Unión Europea "NextGenerationEU/PRTR".

6. Conclusión

Durante este trabajo se ha apreciado la complejidad que supone el uso de segmentación semántica frente a otras técnicas de visión por computador. Se ha podido apreciar la importancia del aumento de datos, ya que, sobre todo para bases de datos pequeñas, mejora el rendimiento del modelo al multiplicar el número de muestras. En cuanto a los resultados se ha logrado cumplir con los requerimientos iniciales, logrando un modelo entrenado que alcanza un 78,12 % de mIoU y es capaz de operar a 27 FPS con una resolución de 768x432 píxeles. Como trabajos futuros se plantea la transformación de estos datos a información útil y codificada para el vehículo autónomo, como crear un mapa de coste con el terreno transitable, o incluso, aumentar la base de datos con las mejores segmentaciones que haga durante sus trayectos.

Referencias

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (12), 2481–2495. DOI: 10.1109/TPAMI.2016.2644615

Cordts, M., 2022. The cityscapes dataset.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding.

Hong, Y., Pan, H., Sun, W., Jia, Y., 2021. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. DOI: 10.48550/ARXIV.2101.06085

Kontschieder, P., Bulò, S. R., Bischof, H., Pelillo, M., 2011. Structured class-labels in random forests for semantic image labelling, 2190–2197. DOI: 10.1109/ICCV.2011.6126496

Li, H., Xiong, P., Fan, H., Sun, J., 2019. Dfanet: Deep feature aggregation for real-time semantic segmentation. DOI: 10.48550/ARXIV.1904.02216

Marin Plaza, P., Yagüe Cuevas, D., Royo, F., de Miguel, M. A., Moreno, F. M., Ruiz de la Cuadra, A., Viadero Monasterio, F., Garcia, J., San Roman, J. L., Armingol, J. M., 2021. Project ares: Driverless transportation system. challenges and approaches in an unstructured road. Electronics 10 (15). DOI: 10.3390/electronics10151753

Papadeas, I., Tsochatzidis, L., Amanatiadis, A., Pratikakis, I., 2021. Real-time semantic image segmentation with deep learning for autonomous driving: A survey. Applied Sciences 11 (19). DOI: 10.3390/app11198802

Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., 2016. Enet: A deep neural network architecture for real-time semantic segmentation. DOI: 10.48550/ARXIV.1606.02147

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. DOI: 10.48550/ARXIV.1505.04597

Sellat, Q., Bisoy, S., Priyadarshini, R., Vidyarthi, A., Kautish, S., Barik, R. K., Oliva, D., 2022. Intelligent semantic segmentation for self-driving vehicles using deep learning. Computational Intelligence and Neuroscience 35 (8), 1915–1929. DOI: 10.1155/2022/6390260

Shotton, J., Johnson, M., Cipolla, R., 2008. Semantic texton forests for image categorization and segmentation, 1–8. DOI: 10.1109/CVPR.2008.4587503

Terol, M., 2021. Vehículos autónomos: un siglo de evolución marcado por la innovación tecnológica.

Tighe, J., Lazebnik, S., 2010. Superparsing: Scalable nonparametric image parsing with superpixels, 352–365. DOI: 10.1007/978-3-642-15555-$0_2$6

Verified Market Research, 2023. Computer Vision Market Size and Forecast.

Wu, Z., Shen, C., Hengel, A. v. d., 2017. Real-time semantic image segmentation via spatial sparsity. DOI: 10.48550/ARXIV.1712.00213

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N., 2020. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. DOI: 10.48550/ARXIV.2004.02147

Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J., 2017. Icnet for real-time semantic segmentation on high-resolution images. DOI: 10.48550/ARXIV.1704.08545