ANÁLISIS, EVALUACIÓN E IMPLEMENTACIÓN DE ALGORITMOS DE SEGMENTACIÓN SEMÁNTICA PARA SU APLICACIÓN EN VEHÍCULOS INTELIGENTES

Alejandro Barrera, Carlos Guindel, Fernando García, David Martín
Laboratorio de Sistemas Inteligentes (LSI), Universidad Carlos III de Madrid
alebarre@pa.uc3m.es, cguindel@ing.uc3m.es, fegarcia@ing.uc3m.es, dmgomez@ing.uc3m.es

Resumen

Los algoritmos de segmentación semántica, cuyo objetivo es asignar una etiqueta a cada píxel de la imagen, están adquiriendo una gran relevancia en los últimos años. Uno de sus principales ámbitos de aplicación son los sistemas embarcados en vehículos, donde pueden desempeñar distintas funciones para el entendimiento del entorno. Sin embargo, los particulares requisitos de este tipo de aplicaciones, impuestos por las limitadas capacidades de procesamiento disponibles y la complejidad de las escenas, requieren un análisis específico que vaya más allá de los parámetros clásicos. En este artículo se presenta un análisis detallado de varias arquitecturas contemporáneas para segmentación semántica en el contexto de su aplicación en vehículos, así como un estudio de su viabilidad en una plataforma real.

Palabras clave: Segmentación semántica; Deep learning; Vehículos inteligentes.

1. INTRODUCCIÓN

El transporte juega un papel fundamental en la economía y es imprescindible en cualquier sociedad y actividad, lo que explica el continuo crecimiento del parque móvil mundial. Sin embargo, esto último conlleva atascos más frecuentes, más contaminación y un mayor número de accidentes.

Los sistemas inteligentes de transporte (ITS) pretenden mitigar estos efectos negativos; especialmente, el referido a la accidentalidad. Sistemas avanzados de asistencia al conductor (ADAS) orientados a esta labor son, por ejemplo, los sistemas de alerta de abandono de carril (LDWS), que se fundamentan en la detección de líneas de carretera [20], o los sistemas de reconocimiento de señales de tráfico en tiempo real [4].

Estos sistemas, sin embargo, se enfocan en funcionalidades específicas y, en ocasiones, combinar distintos ADAS puede no resultar viable por su gran coste computacional. Esto lleva al estudio del reconocimiento de escenas completo, donde la segmentación semántica [26] está siendo de gran interés estos últimos años.

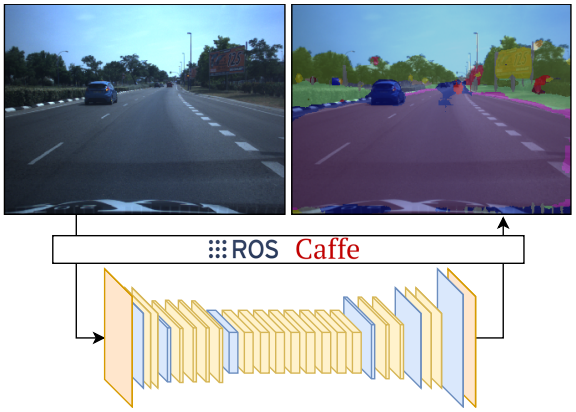


Figura 1: Representación de la propuesta experimental

La llegada de las técnicas basadas en "aprendizaje profundo" (*deep learning*), en concreto las redes neuronales convolucionales (CNN), ofrecen la posibilidad de obtener características apropiadas para el reconocimiento de patrones de manera autónoma. Dentro de estas CNN se encuentran redes capaces de clasificar una imagen dentro de un número muy amplio de escenarios posibles [11], u otras que permiten la detección de objetos [19]. Desplazándose hacia una clasificación más fina, puede obtenerse una segmentación semántica que permite obtener la predicción por píxel identificando y posicionando de forma precisa cada elemento.

Utilizando este tipo de técnicas, es posible unificar varias funcionalidades, mejorando así el coste computacional, la complejidad de las operaciones y, en muchos casos, la calidad de la respuesta [22].

Pese a que el problema de la segmentación semántica es, hoy día, objeto de una intensa actividad investigadora, los sistemas embarcados en vehículos tienen unos requisitos específicos que exceden los aspectos comúnmente tratados en la literatura de visión por computador. Por esta razón, en este trabajo se pretende ofrecer una visión exhaustiva de algunos de los algoritmos de segmentación semántica más relevantes en el contexto de su aplicación en vehículos inteligentes, así como un caso de uso particular en una plataforma de investigación propia. Las distintas técnicas se han im-

plementado dentro de un mismo *framework* (Fig. 1), permitiendo una comparativa consolidada de las mismas.

Las aportaciones de este trabajo, por tanto, son las siguientes:

- Descripción y análisis cuantitativo de métodos relevantes de segmentación semántica, orientado a su aplicación en sistemas embarcados, proporcionando una visión general que pueda ayudar a otros investigadores.
- Implementación de un sistema de segmentación semántica dentro de la arquitectura de un vehículo inteligente bajo ROS [18], para detección de escenas urbanas en tiempo real.

El resto del documento se estructura de la siguiente forma: en la Sección 2 se realiza un breve repaso de técnicas relacionadas, que es completado más tarde, en la Sección 3, con una descripción detallada de las distintas arquitecturas analizadas. En la Sección 4, se expone la implementación realizada en la plataforma experimental propia. Por su parte, en la Sección 5, se ofrecen resultados cuantitativos de las distintas alternativas, así como pruebas cualitativas del funcionamiento del sistema en la plataforma. Finalmente, en la Sección 6, se ofrecen conclusiones sobre el trabajo.

2. TRABAJOS RELACIONADOS

La semántica tal como se conoce hoy en día comienza de la mano de [13], cuyo hito fue sustituir las capas totalmente conectadas (*FC layers*) por sus análogas convolucionales; de esta forma se obtienen mapas de características en vez de puntuaciones. Por ello, en las llamadas "redes completamente convolucionales" (FCN) se toma ventaja de la extracción de características propia de las CNN como punto de partida para crear predicciones a nivel de píxel. Dada la salida de las CNN, esa respuesta se somete a un proceso de *upsampling* obteniendo un mapa de la escena completa. Estas arquitecturas son comúnmente llamadas *encoder-decoder* [23]. A pesar de las características de la red y su idea innovadora, existen desarrollos posteriores que han ido solventando pequeñas deficiencias en estos modelos.

La segmentación semántica necesita un entendimiento de la escena global; para ello, algunos modelos como [2] realizan un post-procesado para afinar las predicciones finales. Asimismo, el empleo de *dilated convolutions* permite ampliar el campo receptivo exponencialmente logrando la extracción de características en un ámbito mayor [17]; siendo las deconvoluciones una implementación de esta técnica.

La fusión de datos entre niveles de abstracción mayor y menor también permite utilizar un contexto global [13]. Destacan de igual forma las redes neuronales recurrentes, que permiten acceder a un mapa mayor empleando ventanas de distintos tamaños [25].

Centrándose en los ITS, se puede observar cómo la participación de la semántica va incrementando. Las arquitecturas para problemas de esta naturaleza suelen utilizar bases de datos públicas, tales como KITTI [1] y Cityscapes [3], que permiten replicar y comparar los resultados obtenidos por las distintas propuestas.

En entornos de percepción para vehículos inteligentes, [16] propone una arquitectura eficiente basada en la red *Inception-v1* con fusión de datos a un *decoder* que permite obtener tanto la segmentación semántica de la escena, como la topología de la vía. En [15], la detección real de la carretera a través de la red ResNet-101 se combina con un *decoder* similar al que existe en la FCN y técnicas de aumento de datos (*augmentation*), lo que da lugar a un sistema más robusto. Igualmente, en [12], gracias al entrenamiento con una vasta base de datos, se consiguen detecciones resistentes a cambios de temporal e iluminación para detección de líneas de carretera y carriles con la implementación de varios *decoders*.

3. ANÁLISIS DE ARQUITECTURAS

En esta sección se discutirán las cualidades de cada arquitectura utilizada en la posterior comparativa, además de los cambios realizados en cada una para utilizarlas bajo el mismo *framework* y la misma entrada. El presente trabajo se enfoca en cuatro arquitecturas cuya aparición ha tenido un impacto relevante: FCN [13], Bayesian Segnet [9], U-Net [23] y ERFNet [21].

FCN. Aunque en [13] se proponen muchos modelos, se ha escogido el llamado FCN-8s *all-at-once*. Esta arquitectura la forma un *encoder* proveniente de la red VGG-16 [27] al cual se le ha agregado posteriormente una etapa de *upsampling* no lineal basada en deconvoluciones y funciones de activación. Por otro lado, para refinar la precisión espacial se han agregado varios hilos (*skips*) que permiten realizar operaciones de adición entre capas del inicio y final de la arquitectura, respetando así la estructura global. En su versión *all-at-once* se realiza todo ello en un hilo permitiendo resultados similares y un tiempo menor.

Bayesian Segnet. De forma similar a la anterior, esta arquitectura completamente simétrica basa su *encoder* en la distribución de la VGG-

16 y de forma inversa en su *decoder* insertando deconvoluciones. Lo interesante es que empleando capas *dropout* en las capas intermedias, cuya misión normalmente recae en evitar el *overfitting* (entre otros problemas) [7], consiguen una aproximación al modelo Gausiano de [6]. Este modelo indica las zonas donde existe mayor incertidumbre (intersecciones entre objetos u oclusiones).

U-Net. Red que parte de una arquitectura similar a la de VGG-13 como *encoder*. Tiene la particularidad de propagar la información del contexto extraída de determinadas capas iniciales sus análogas en el *decoder* mediante la concatenación de características. Se ha adaptado su entrada puesto que el objetivo de la misma estaba orientado a imágenes provenientes de un microscopio en un único canal; además, se ha incluido un *padding* igual a la unidad para solventar la pérdida de resolución.

ERFNet. Arquitectura basada en la red ENet [17]. Propone un novedoso bloque residual llamado *Non-bottleneck-1D*, el cual proporciona la información de la capa previa, disminuye el coste computacional y permite conservar la precisión y capacidad de aprendizaje. A este bloque lo complementan varios bloques *downsampling* al comienzo para disminuir el espacio de características, y una etapa final de *upsampling*. Esta arquitectura ha sido portada desde un *framework* a otro cambiando la formulación de cada capa para adecuarse a la nueva plataforma.

4. PLATAFORMA DE DETECCIÓN

En esta sección se van a comentar las características de la plataforma experimental, así como el nodo de ROS empleado para la segmentación semántica de las imágenes tomadas por la misma.

4.1. CARACTERÍSTICAS DEL SISTEMA

Actualmente el número de arquitecturas enfocadas a entornos urbanos ha crecido enormemente. Sin embargo, se debe tener en cuenta que un sistema real tiene que lidiar con distintos problemas, entre ellos la comunicación.

De este modo, en el presente trabajo se propone una manera de conectar los sistemas de visión y obtener verdaderas predicciones en tiempo real de la escena completa. La plataforma experimental empleada es el vehículo IVVI 2.0 [14], que pertenece al Laboratorio de Sistemas Inteligentes (LSI) de la UC3M. Este vehículo posee una unidad de procesamiento dotada de una GPU NVIDIA Titan Xp de 12GB.

Para esta labor se ha utilizado la conexión *publisher/subscriber* de ROS, donde se recibe una imagen desde una de las cámaras del vehículo y, una vez la imagen es segmentada, se responde por otro *topic*.

Por otra parte, las implementaciones de los diferentes métodos difieren en cuanto al *software* utilizado. Con el objetivo de permitir una mejor comparación entre diferentes arquitecturas, se ha utilizado un único *framework* como es Caffe [8] con las tecnologías CUDA, cuDNN y cuBLAS para mejorar el rendimiento del mismo. Esto ha permitido la rápida adaptación de parámetros de unas arquitecturas a otras.

4.2. NODO DE SEGMENTACIÓN

La implementación de los algoritmos de segmentación semántica en un nodo C++ está basada en un paquete públicamente disponible[1]. Cuando el nodo se inicializa, en primer lugar se carga una de las arquitecturas mencionadas en la Sec. 3 en memoria, además de sus pesos pre-entrenados. Seguidamente se suscribe a un canal determinado y en el momento que el nodo recibe una imagen de entrada, continúa del siguiente modo:

1. Se obtienen las características de la red y se preprocesa la imagen para que utilice el formato requerido por la misma (canales, tamaño, precisión y ajusta la media a cero).
2. Se realiza la inferencia y toma los resultados de probabilidad por clase. El resultado final se obtiene con una operación *ArgMax*, que permite conocer la clase con mayor probabilidad en cada píxel. Por último, esa segmentación es publicada por el *topic* seleccionado.

5. RESULTADOS

En esta sección se va a hacer una exposición detallada de los diferentes experimentos llevados a cabo sobre diversas técnicas y parámetros, y posteriormente, un análisis de los resultados obtenidos.

5.1. MÉTRICAS

De forma que se puedan comparar las distintas técnicas y arquitecturas, en los posteriores apartados se han empleado una serie de métricas como las que aparecen en [13] para obtener la precisión total (*overall accuracy*), la probabilidad de obtener una predicción correcta (*mean recall*), proba-

[1] `https://github.com/tzutalin/ros_caffe`

bilidad de decidir correctamente (*mean accuracy*), la adaptación de la predicción al *ground truth* (*mean Intersection-over-Union*) y el IoU según su frecuencia de aparición (*frequency-weighted IoU*). Del mismo modo se han obtenido el tiempo de iteración y uso de memoria de cada red.

5.2. INFLUENCIA DE PARÁMETROS

Para probar cómo afectan distintos parámetros a los entrenamientos en el campo de la semántica y corroborar diversos datos de la bibliografía actual de forma experimental, se ha utilizado la arquitectura ERFNet. Esta ha sido escogida por su bajo consumo de recursos en fase de entrenamiento.

Además se han fijado otra serie de parámetros, la mayoría basados en los dispuestos en [21] como es el número de iteraciones $100k$, *learning rate* (lr) ajustado a $2 \cdot 10^{-4}$ disminuyéndolo con un factor de 0,5 cada $50k$ iteraciones, *weight decay* en $5 \cdot 10^{-4}$, momento a 0,9 y momento2 a 0,999 con el optimizador ADAM [10] y *batch size* a 1.

Por otro lado, la base de datos empleada es Cityscapes. Esta aporta un buen detalle de escena en entornos urbanos. Asimismo, el coste computacional generado por la misma se ha reducido escalando su resolución original de 2048x1024 a 512x256 para la consecución de las pruebas.

Utilizando un conjunto de parámetros fijo, se ha estudiado el efecto de distintas alternativas en el entrenamiento de la red, con el fin de mejorar la precisión de este tipo de algoritmos. A continuación se describen las alternativas contempladas, así como el impacto cuantitativo de las mismas en los resultados (Cuadro 1).

Equilibrado de clases. En Cityscapes se ha observado que no todas las clases están equilibradas, ya que existe una discrepancia significativa entre el número de instancias de algunas clases. Esto muchas veces facilita la omisión del aprendizaje de estas clases minoritarias. Para comprobar si es posible mejorar la precisión equilibrando dichas clases se han calculado pesos para cada clase siguiendo [5] y se ha utilizado la capa *InfogainLoss* de Caffe para su implementación.

Los resultados mejoran ligeramente la respuesta anterior. Asimismo se ha podido apreciar como el conjunto de precisiones por clase tiende a normalizarse.

Transferencia de aprendizaje. Este término corresponde a la posibilidad que ofrecen estas redes para utilizar pesos (características) preentrenados. Muchos autores utilizan esta técnica para utilizar grandes bases de datos muy diversas como ImageNet [24]. Estos *datasets* son realmente útiles porque utilizan una gran cantidad de imágenes que les permiten generalizar su respuesta y por tanto ofrecer unos pesos que pueden ser útiles en otro dominio.

Se aprecia como la influencia de utilizar un preentrenamiento sobre ImageNet previo al entrenamiento con las imágenes de Cityscapes resulta en una variación positiva de la respuesta.

Preprocesado y aumento de datos. La técnica de *data augmentation* consiste en aumentar la base de datos realizando cambios en las imágenes, ya sean morfológicos, variar los valores de cada canal, añadiendo ruido, etc. Esto permite que aumente la generalización debido a que nunca se introduce la misma entrada. Esta parte es especialmente importante para *datasets* con un número limitado de imágenes. En este trabajo se han empleado las siguientes técnicas:

- Se ha aplicado una variación de origen Gaussiano centrada en cero y desviación estándar 0.1 en los canales RGB [24]. Esto simula cambios de luminosidad de forma natural.
- Se ha aplicado un volteo horizontal en ciertas imágenes utilizando una selección aleatoria.
- En cuanto a transformaciones geométricas de la imagen, se ha optado por recortar la imagen (con un tamaño específico) simulando una traslación en las coordenadas X e Y, permitiendo conseguir así imágenes sin distorsión pero distintas a las anteriores.

Implementando estas técnicas, se adquieren mejoras ligeramente perceptibles en la precisión final, además de aportar robustez.

Resolución. El tamaño de entrada es un factor importante cuando se quiere realizar una segmentación. Una resolución mayor implica una menor variación entre la aparición de clases y un mayor número de muestras (píxeles) para aprendizaje. Por otro lado, también conlleva un mayor tiempo de cómputo y detecciones menos contextualizadas en las primeras etapas.

Al experimentar con dos tipos de resolución, la mitad y un cuarto de las dimensiones originales, la comparación entre ellas revela que una resolución mayor para este caso mejora en un 5,2 % los resultados obtenidos.

Número de clases. El incremento de clases se ha probado ser un factor perjudicial para las métricas finales. Esto es debido a que el sistema debe discernir entre más objetos que muchas veces son similares, lo que aumenta la incertidumbre. Además, el número de muestras por clase es menor.

En el presente trabajo se ha comprobado como utilizando 7 etiquetas más genéricas (*Category ID*) en lugar de 19 (*Train ID*) se alcanza un 57,62 % en la precisión media entre clases y de forma similar en las intersecciones entre las mismas. Por lo que si no se requiere una granularidad fina, es posible obtener un buen rendimiento reduciendo el conjunto de clases.

Cuadro 1: Resultados(%) para ERFNet utilizando distintos parámetros

Bal. class	Fine tun.	Aug.	Resol.	Mean IoU
				40,23
✓				40,42
✓	✓			43,51
✓		✓		42,14
✓			✓	45,43

5.3. COMPARATIVA DE ARQUITECTURAS

Implementando las diferentes arquitecturas mencionadas en la Sección 3, obtenemos la segmentación semántica de distintas escenas urbanas. Para apreciar mejor esto se va a exponer una relación de resultados valorados en función de tres características clave en vehículos inteligentes: precisión, coste computacional y velocidad.

Dichos resultados se han llevado a cabo recogiendo los parámetros que mejoraban los resultados de la relación anterior y ajustando las arquitecturas para el empleo de los mismos. Hay que tener presente que el principal objetivo no es mejorar los resultados de los que proviene cada arquitectura, sino ofrecer un enfoque experimental dentro de unos parámetros similares en una plataforma real. En el Cuadro 2 se tiene una relación de los resultados empleando las métricas de precisión citadas anteriormente para el conjunto de validación de Cityscapes.

Como se puede observar las arquitecturas obtienen respuestas muy similares. Esto, en cierto modo, se debe a que todas, salvo la ERFNet, comparten ciertas similitudes en su arquitectura. Se puede apreciar como la red FCN obtiene un mayor número de predicciones correctas para cada clase. Sin embargo, la red Bayesian Segnet en este caso es la que mejor IoU obtiene.

Consiguiendo una precisión media parecida en las distintas arquitecturas, la velocidad y coste computacional menores recaen directamente en la ERFNet (Cuadro 3), la cual al utilizar menos características que el resto y emplear el bloque *Non-bottleneck-1D* obtiene una precisión similar, además de ser la más rápida ($\approx$ 15 Hz) y eficiente en la presente comparativa.

5.4. INTEGRACIÓN EN PLATAFORMA FINAL

Tal como se describió en la Sección 4, se han implementado los algoritmos comparados en la arquitectura basada en ROS de la plataforma experimental IVVI 2.0. Las respuestas obtenidas (Fig. 2 y Fig. 3) permiten llegar a las siguientes conclusiones:

- El tamaño de la entrada debe coincidir con el utilizado en el entrenamiento para lograr la máxima precisión. Es un hecho que las redes tienden a aprenderse además de características, su posicionamiento. Estas respuestas por lo tanto se han escalado y recortado.
- El cambio de los parámetros de un sistema de visión a otro también afecta a los resultados. De este modo se concluye que además de aprender las características del entorno, estos sistemas quedan modelados durante el entrenamiento de acuerdo a los parámetros particulares de los sistemas de captura de imágenes.
- Se ha comprobado como los cambios de iluminación, reflejos, sombras, etc., influyen en gran medida en la respuesta. Esto demuestra la importancia de una gran base de datos de entrenamiento que disponga de suficientes ejemplos de este tipo de casuística.

6. CONCLUSIONES

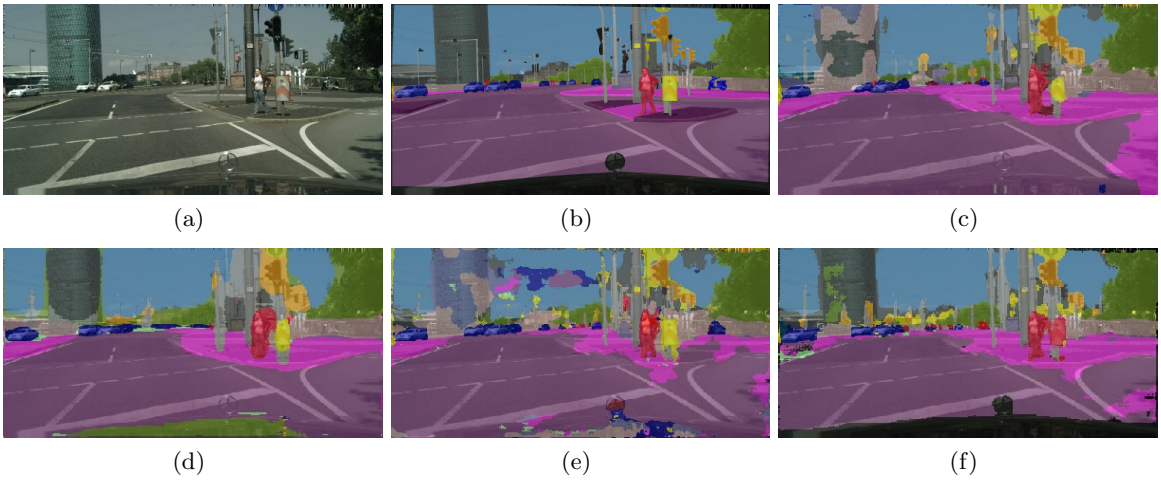
Se ha realizado una introducción a la segmentación semántica exponiendo diferentes arquitecturas empleadas en sistemas inteligentes de transporte. Se ha fijado una comparativa bajo una misma plataforma de algunos de los principales aspectos que influencian la precisión, velocidad y coste computacional. Asimismo, se ha propuesto un nodo de ROS que permite realizar estas operaciones en plataformas reales.

Con esto se espera aportar un punto de vista experimental a aquellos que se estén introduciendo en el campo de la segmentación semántica para problemas de visión por computador.

En un futuro dicha aplicación podría actualizarse con nuevas arquitecturas que aporten una mejora en las métricas existentes o encaminarse a su uso en segmentación por instancias o 3D combinando semántica con nubes de puntos obtenidas por LiDAR sin realizar grandes cambios.

Cuadro 2: Resultados(%) en Cityscapes (19 clases) para las distintas arquitecturas referenciadas en Caffe

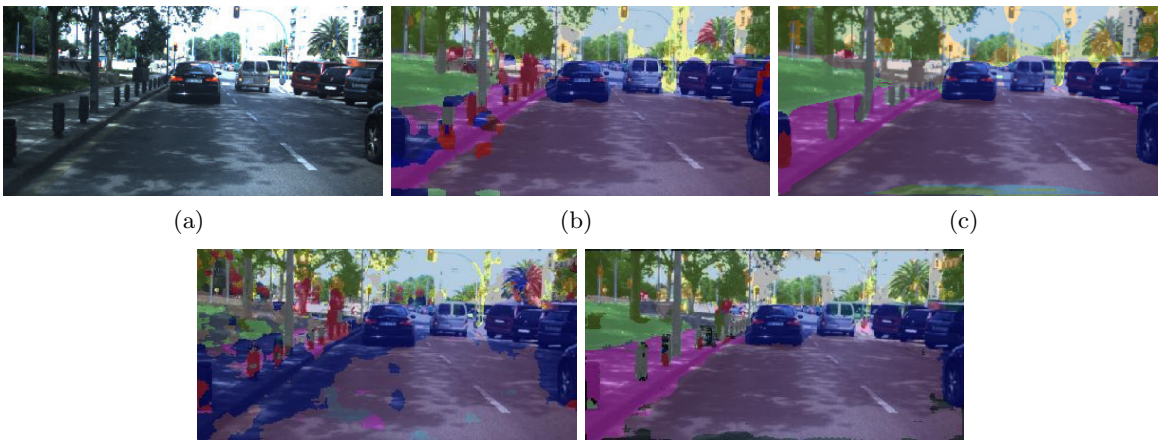
Nombre	Overall acc.	Mean recall	Mean acc.	Mean IoU	F.W.IoU
FCN	83,51	50,44	**70,57**	43,27	76,11
Bay. Segnet	84,00	**54,58**	68,84	**44,25**	76,77
U-Net	82,81	49,94	62,80	41,31	75,52
ERFNet	**85,14**	51,79	67,10	43,34	**78,21**



(a) (b) (c)

(d) (e) (f)

Figura 2: Segmentación con el conjunto de validación de Cityscapes [3] para distintas arquitecturas. A la izquierda se sitúa la entrada original (a) con el *ground truth* (b) y posteriormente las respuestas de FCN (c), Bayesian Segnet (d), U-Net (e) y ERFNet (f).



(a) (b) (c)

(d) (e)

Figura 3: Segmentación aportada por el nodo de detección en una escena de tráfico real obtenida con la plataforma experimental (a) para distintas arquitecturas: FCN (b), Bayesian Segnet (c), U-Net (d) y ERFNet (e).

Cuadro 3: Tiempo de iteración y coste computacional para diferentes arquitecturas

Nombre	Tiempo it.(ms)	Coste comp.(MB)
FCN	170	1335
Bay. Segnet	182	1431
U-Net	135	1095
ERFNet	**61**	**937**

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía, Industria y Competitividad del Gobierno de España a través de los proyectos TRA2015-63708-R y TRA2016-78886-C3-1-R, y por la Comunidad de Madrid a través del proyecto SEGVAUTO-TRIES (S2013/MIT-2713). Agradecemos el apoyo de "NVIDIA Corporation" con la donación de las GPU usadas en esta investigación.

English summary

ANALISYS, EVALUATION AND IMPLEMENTATION OF SEMANTIC SEGMENTATION ALGORITHMS FOR INTELLIGENT VEHICLES

Abstract

Semantic segmentation algorithms, whose goal is to assign a label to each pixel of the image, are acquiring a great relevance in the last years. One of its main areas of application is vehicular embedded, where they can be used to different functions aimed to understand the environment. However, the particular requirements of this type of applications, imposed by the high processing requirements and the complexity of the scenes, require a specific analysis that goes beyond the classical parameters. This article presents a detailed analysis of several contemporary architectures for semantic segmentation in the context of its application in vehicles, as well as a study of its viability in a real platform.

Keywords: Semantic segmentation; Deep learning; Intelligent vehicles.

Referencias

[1] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets deep learning for car instance segmentation in urban scenes," in *British Machine Vision Conference (BMVC)*, 2017.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] A. De La Escalera, J. M. Armingol, J. M. Pastor, and F. J. Rodríguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 2, pp. 57–68, 2004.

[5] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[6] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[9] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1965–1973.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[14] D. Martín, F. García, B. Musleh, D. Olmeda, G. A. Peláez, P. Marín, A. Ponz, C. H. Rodríguez Garavito, A. Al-Kaff, A. de la Escalera, and J. M. Armingol, "IVVI 2.0: An intelligent vehicle based on computational perception," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7927–7944, 2014.

[15] J. Muñoz-Bulnes, C. Fernández, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 366–371.

[16] M. Oeljeklaus, F. Hoffmann, and T. Bertram, "A combined recognition and segmentation model for urban traffic scene understanding," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[17] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[18] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[20] C. H. Rodríguez-Garavito, C. Guindel, and J. M. Armingol, "Sistema de asistencia a la conducción para detección y clasificación de carriles," in *XXXVI Jornadas de Automática*, Bilbao, Spain, 2015, pp. 26–31.

[21] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[22] E. Romera, L. M. Bergasa, and R. Arroyo, "Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of cnns?" *arXiv preprint arXiv:1607.00971*, 2016.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "Dag-recurrent neural networks for scene labeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3620–3629.

[26] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," *arXiv preprint arXiv:1707.02432*, 2017.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.