Recibido 12/11/2019 *ReCIBE, Año 8 No. 2, Noviembre 2019*

Aceptado 13/11/2019

ANÁLISIS DEL DESEMPEÑO DE REDES NEURONALES PROFUNDAS PARA SEGMENTACIÓN SEMÁNTICA EN HARDWARE LIMITADO

Performance Analysis of Semantic Segmentation Deep Neural Networks for Limited Hardware

Oscar Alejandro Soto-Orozco[1]
Oscar.soto@ieee.org

Alma Delia Corral-Sáenz[1]
adcorral@itchihuahua.edu.mx

Claudia Elizabeth Rojo-González[1]
cerojo@itchihuahua.edu.mx

Juan Alberto Ramírez-Quintana[1]
jaramirez@itchihuahua.edu.mx

[1]División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/ I.T. Chihuahua, México.

1 **Resumen.** Segmentación semántica consiste en encontrar objetos previamente definidos en una imagen digital y se aplica en tecnologías como vehículos autónomos, interacción humano-maquina, realidad aumentada, robótica, etc. Los modelos más comunes para llevar a cabo esta forma de segmentación son las redes totalmente convolucionales, ya que reportan los mejores desempeños en la detección de objetos. Sin embargo, la mayor parte de estas redes tienen alto costo computacional y requieren de computadoras costosas, por lo que han surgido recientemente modelos basados en estas redes, pero con baja complejidad en cálculos para que las aplicaciones de segmentación semántica se puedan implementar desde sistemas embebidos. Por lo tanto, para contribuir con este esfuerzo, se presenta en este artículo un análisis detallado las redes Enet, Mobilenet v2, ERFNet y ESPNet v2, las cuales son redes populares en la literatura que se pueden correr desde un sistema embebido. Con base en los resultados, se concluye que los métodos que reemplazan la convolución regular por factorizaciones como la convolución separada en profundidad y convoluciones dilatadas con diversas ramas y el uso de otras estrategias como convoluciones saltadas e interpolaciones articuladas reducen el costo computacional comparando las métricas generadas por cada red como la huella de memoria, la precisión y el tiempo que tarda en segmentar una sola imagen.

2 **Palabras clave.** Aprendizaje profundo, Segmentación semántica, Redes neuronales convolucionales, Procesamiento de imágenes y video.

Abstract. Semantic segmentation consists in finding previously modified objects in a digital image and is applied in technologies such as autonomous vehicles, human-machine interaction, augmented reality, robotics, etc. The most common models to carry out this form of segmentation are totally convolutional networks, since they report the best performances in the detection of objects. However, most of these networks have high computational cost and are affected by expensive computers, so they have recently operated specific models in these networks, but with low complexity in analysis for semantic segmentation applications can be implemented from embedded systems. Therefore, to contribute to this effort, this article presents a detailed analysis of the Enet, Mobilenet v2, ERFNet and ESPNet v2 networks, which are popular networks in the literature that can run from an embedded system. Based on the results, it is concluded that the methods that replace the regular convolution by factorizations such as the deep convolution and dilated convolutions with various branches and the use of other strategies such as skipped convolutions and articulated interpolations reduce the computational cost by comparing the metrics generated for each red as the memory footprint, the precision and the time it takes to segment a single image.

Keywords. Deep learning, Semantic segmentation, Convolutional neural networks, Image and video processing

1. Introducción

El aprendizaje profundo con redes neuronales convolucionales (CNN por sus siglas en inglés) ha atraído la atención en los últimos años debido a que permite analizar y modelar características abstractas de forma automática, y genera mejores resultados que otro algoritmos supervisados y no supervisados en visión por computadora (Lockheed Martin, 2018).

Este tipo de redes son modelos de tipo *feedforward* (Shelhamer, Long, & Darrell, 2017) y tienen como entrada imágenes RGB de alta definición que se propagan hacia adelante a través diversas neuronas con pesos y parámetros de aprendizaje. La arquitectura de estas redes se basa en módulos para extraer características y capas para clasificación. Los módulos de características se componen al menos una de las siguientes capas:

Convolución. Neuronas que se componen de pesos o filtros que se convolucionan con la salida de las capas más bajas para obtener información abstracta de la imagen.
Agrupamiento. Reducen la dimensión de la salida de una capa convolucional partiéndola en ventanas y seleccionando solo un valor que represente los elementos de dicha ventana.
Normalización por lote. Se normalizan los resultados de la capa anterior para evitar valores muy grandes o fuera de rango e incrementar la estabilidad de la red (Ioffe & Szegedy, 2015).
Funciones de activación. Definen la respuesta de una neurona y se modelan como sigmoidales, de base radial o de umbral.

Conforme la información de la entrada se propaga por la CNN hacia adelante, los módulos para extraer características encuentran información de colores, bordes y texturas cada vez más significativa y detallada. Las capas de clasificación se componen de neuronas similares a las de cualquier red neuronal *feedforward* y categorizan la información de la imagen de entrada para asignar una categoría semántica a cada uno de los pixeles de la imagen.

En la actualidad, existen una gran cantidad de arquitecturas de CNNs aplicadas a segmentación semántica (SS) que asignan una etiqueta *l* de un conjunto comprendido por $\mathcal{L} = \{ l_1, l_2, \dots, l_n\}$ para cada elemento del conjunto de variables aleatorias de entrada $X = \{x_1, x_2, \dots, x_N\}$. Cada etiqueta *l* representa una clase u objeto con significado semántico diferente como avión, vehículo, señal de tráfico o fondo. Por otro lado, la variable *X* es una imagen de una secuencia de video. A menudo se etiquetan las clases con máscaras de color para visualizarlas en las imágenes, como se puede observar en la Figura 1.

Figura 1. Ejemplos de segmentación semántica con base de datos Cityscapes (Cordts et al., 2016)

Existen arquitecturas profundas de CNNs populares en el estado del arte debido al desempeño que obtuvieron en la base de datos de ImageNet Large Scale Visual Recognition Challenge (ILSVRC sus siglas en inglés), la cual es utilizada formalmente en la literatura para realizar pruebas, ya que tiene más de 190,000 imágenes con sus *groundtruths* para clasificar diversos objetos (Krizhevsky, Hinton, & Sustskenvet, 2012).

La primera de estas arquitecturas fue AlexNet (Krizhevsky et al., 2012), una red de 8 capas ocultas compuestas por 5 convolucionales y 3 totalmente conectadas que logró un error de precisión de 37.5% en el conjunto de prueba de la base de datos de ImageNet. Este error fue considerablemente menor que el reportado en los métodos del estado del arte en ese momento.

Otra red popular es la Visual Geometry Group (VGG) (Simonyan & Zisserman, 2014), propuesta por el departamento de ingeniería y ciencia de la universidad de Oxford y se conforma de 16 capas convolucionales y 3 capas totalmente conectadas. La principal diferencia de VGG sobre AlexNet es que, en vez de tener pocas capas con grandes campos receptivos, utiliza una pila de capas de convolución con pequeños campos receptivos en las primeras capas.

Este uso en las capas de convolución tuvo tal impacto, que en los años posteriores sería la base para GoogleNet, una red propuesta en (Szegedy et al., 2015) por investigadores de la universidad de Carolina del norte y la universidad de Michigan en colaboración con Google Inc. Esta red tiene un total de 22 capas compuestas por convoluciones con agrupamientos y capas totalmente conectada, y agrega un mecanismo de aprendizaje basado en una regla hebbiana.

Además, tiene un procesamiento de múltiples escalas para generar la capacidad de adquirir información en contexto del escenario. GoogleNet es muy profunda y computacionalmente costosa, ya que es necesario desarrollar una gran cantidad de capas y filtros que conllevan muchas operaciones de punto flotante por segundo para obtener resultados aceptables en la literatura científica. Este número de capas exige una enorme cantidad de recursos como núcleos especializados, memoria dedicada a video y altas velocidades de procesamiento.

Estas redes se han utilizado para proponer modelos de SS funcionando con buenos desempeños y de forma eficiente en computadoras como estaciones de trabajo o computadoras de alta capacidad de procesamiento.

Esto causa que el desarrollo tecnológico en las áreas como vehículos autónomos, interacción humano-maquina, realidad aumentada y robótica sea sumamente costosa (Lateef & Ruichek, 2019).

Por ello, en los últimos años se han generado líneas de investigación en CNNs que se centran en analizar arquitecturas de redes para optimizar el uso de los recursos de procesamiento y permitir la implementación de SS en sistemas embebidos o plataformas de hardware a baja potencia.

A partir de esta línea, se han generado modelos de SS que funcionan de forma exitosa en plataformas embebidas de hardware limitado y han permitido ahorrar costos de forma significativa en el desarrollo de tecnología. Por lo tanto, para contribuir en este esfuerzo de desarrollar esquemas de SS en hardware limitado, en este artículo se propone un análisis detallado del desempeño de los modelos más populares en la literatura para SS que tienen un número reducido de operaciones de punto flotante (FLOPS), bajo uso de memoria y bajo consumo de energía.

El objetivo de este análisis es estudiar las estrategias de diseño que permiten el desarrollo de redes eficientes, y proveer un punto de partida para desarrollar arquitecturas novedosas cada vez más eficientes y que permitan desarrollar tecnología a costos significativamente bajo.

El resto del artículo presenta de la siguiente forma: la sección 2 describe diversas arquitecturas de CNNs para SS que son eficientes en el uso de recursos computacionales. La sección 3 analiza los diversos parámetros que ayudan a colocar a estas arquitecturas como modelos eficientes en recursos. La sección 4 presenta los resultados del análisis realizado a estos modelos y la sección 5 presenta las conclusiones.

2. Redes totalmente convolucionales para segmentación semántica

El éxito en desempeño que han tenido las CNNs en SS se debe a que estas encuentran características abstractas que definen a cada clase semántica. Por esta razón, en la actualidad se han generado múltiples arquitecturas de CNNs para de SS, entre las cuales, las más populares son las redes totalmente convolucionales (FCN por sus siglas en inglés)(Sevak, Kapadia, Chavda, Karungan, & Sujatha, 2017).

La primera FCN fue propuesta por Jonatrhan Long y Evan Shelhamer de la Universidad de Berkeley (Shelhamer, Long, & Darrell, 2017), y se basa en tomar ventaja de las arquitecturas existentes de CNNs de clasificación de imágenes para aprender características de diferente jerarquía. Para ello, la FCN tiene cinco bloques, como se ve en la Figura 2, donde el primer bloque es la imagen de entrada. El segundo bloque es el codificador y se compone de un conjunto de capas convolucionales que comprimen la información de la imagen de entrada *X* para extraer características.

El tercer bloque es una imagen comprimida *F(X)* que contiene el mapa de características y es la entrada para el cuarto bloque, el cual se le conoce como decodificador. Este bloque se basa en un conjunto de capas de desconvolución que descomprimen los mapas de características para obtener una imagen que contiene la segmentación de los objetos contenidos en la imagen de entrada *X'*. Las capas de este último bloque se componen de des convoluciones para obtener como salida mapas de características descomprimidos. La última capa del decodificador obtiene la combinación de todos los mapas de características generados por las capas anteriores para ser interpolados con la convolución transpuesta descrita por (Taylor, 2010), que toma la imagen de salida de la capa anterior Y^{i-1}, compuesto por k_0 canales $Y_1^i, \dots, Y_{k_0}^i$,lo siguiente:

$$\sum_{k=1}^{k_1} Y_k^{i-1} \oplus f_{k,c} = y_c^i \qquad (1)$$

Donde f son los filtros, y la capa genera una salida Y_c^i donde $c = 1, \dots, k_1$. Es decir, la FCN es una CNN que tienen un bloque de decodificación en vez de capas de clasificación. De esta manera se produce una imagen reconstruida $X' = (G^o F)(X)$ que tiene la misma resolución de la imagen de entrada y contiene los resultados de la segmentación, donde cada elemento de *X'* tiene asignada una categoría semántica que indica a que clase pertenece cada pixel de la entrada.

Al modelo de la FCN también se le conoce como red autocodificadas.

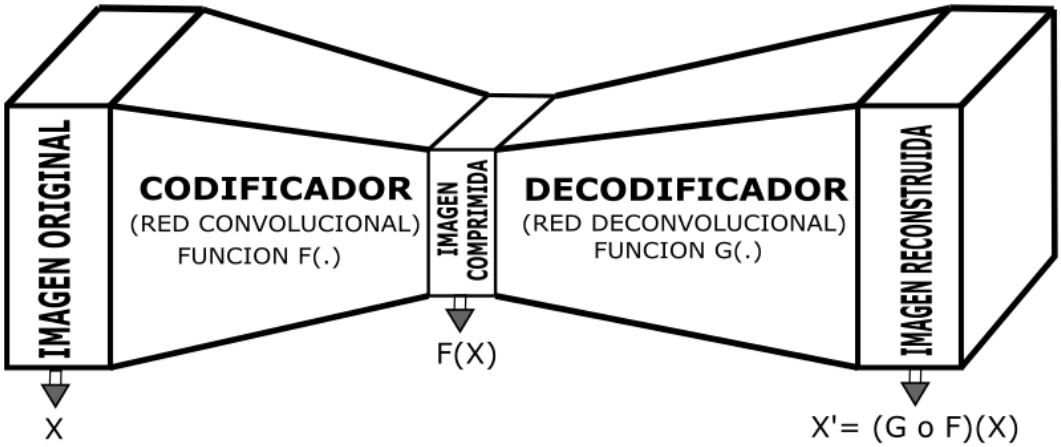

Figura 2. Estructura básica de red convolucional auto codificada.

En otras palabras, la FCN toma la imagen original y la comprime en mapas de características que contienen la información semántica de los objetos, pero con resolución espacial y de color reducidas.

Conforme la red se torna más profunda, se comprimen más los mapas de características, y cuando se propaga estos mapas por todo el codificador, se descomprimen en resolución espacial en el decodificador hasta llegar al tamaño de *X*. (Neapolitan & Neapolitan, 2018).

Esta FCN mostró un mejor desempeño para SS que los métodos estadísticos de aprendizaje automático ya que las capas convolucionales extraen características abstractas al igual que las capas de las CNNs mencionadas en la introducción y las capas de deconvolucion densifican las activaciones obtenidas por el codificador generando una activación ampliada, pero con una mayor densidad de pixeles.

3. Evaluación de modelos eficientes

A pesar de la potencia, buena precisión y flexibilidad del modelo FCN, este tiene diversos aspectos que dificultan su aplicación ya que no considera la variación espacial que pueda existir en los objetos, ni la relación que mantienen las diferentes clases en el escenario y no siempre se puede ejecutar en tiempo real con imágenes de alta resolución en sistemas embebidos.

A pesar de ello, en la literatura se reportan cuatro modelos para SS inspirados en la FCN, cuya arquitectura en el codificador y el decodificador les permiten desempeños aceptables en precisión y uso de recursos.

Este desempeño se puede describir en las siguientes métricas:

Tiempo de inferencia: es el total de operaciones de punto flotante por segundo (FLOPS) necesarias para ejecutar el modelo. Los FLOPS son proporcionales al tiempo en que se propaga la entrada por toda la red y a las especificaciones del hardware en el que se trabaje. Para el caso de este análisis, el tiempo de inferencia se obtuvo mediante pruebas realizadas en una laptop con procesador Intel Core I7 de 8va generación con una GPU GTX 1060 con 6 Gb de memoria dedicada a video.

Huella de memoria: se refiere a la cantidad de memoria requerida para procesar la FCN y obtener la máscara de SS.

Precisión: consiste en el porcentaje de similitud que tiene el resultado de segmentación obtenido por el modelo con un objetivo de segmentación o *groundtruth*. Generalmente, esta métrica es la más utilizada en las bases de datos de SS y se define por la intersección sobre la unión IoU.

Los modelos seleccionados poseen un desempeño equilibrado en estas 3 métricas ya que mantienen una precisión apropiada, una huella de memoria baja y un tiempo de ejecución en tiempo real en sistemas que utilicen procesadores ARM embebidos o computadoras con fuentes de poder de menos de 100 W.

Este tipo de hardware limitado es de bajos recursos computacionales y de acuerdo con un estudio de tecnología, puede generar costos significativamente más bajos en el desarrollo de aplicaciones de SS. De esta manera, fueron seleccionados para este análisis los modelos de Enet (Paszke, Chaurasia, Kim, & Culurciello, 2016), Mobilenet v2(Sandler *et al.*, 2017), ESPNet v2(Mehta, Rastegari, Caspi, Shapiro, & Hajishirzi, 2018), ERFNet (Romera, Álvarez, Bergasa, & Arroyo, 2018) y FastFCN (H. Wu, 2019). A continuación, se describen cada una de estas redes.

En (Paszke et al., 2016), se propone una FCN denominada Red eficiente (Enet por sus siglas en ingles), la cual tiene una arquitectura autocodificada de 23 capas. El módulo codificador se compone de 20 capas de conexiones residuales y convoluciones factorizadas que aumentan la eficiencia, mientras se mantiene un desempeño aceptable.

El módulo decodificador se compone de 3 capas deconvolucionales que aumentan la resolución de los mapas de características obtenidos por la sección codificadora. ENet logró una velocidad de procesamiento de 83 cuadros por segundo en una GPU NVIDIA Titan y 7 cuadros por segundo en una JetsonTX1.

Este modelo fue retomado por (Sandler et al., 2018), y se propuso una versión llamada Mobilenet v2, de arquitectura codificador-decodificador con módulos de pirámide de convoluciones dilatadas (ASPP), en donde se obtiene información de contexto (Mehta, Rastegari, Shapiro, & Hajishirzi, 2018) a diferentes niveles de dilatación.

También se utilizan convoluciones de 1x1 para agrupar características de la imagen y un *stride o* paso de salida que controle la resolución del mapa de segmentación. De esta manera, la extracción de características que proporciona Mobilenet v2 propicia un tiempo de inferencia y huella de memoria relativamente bajos en comparación a los de otros modelos, mientras la precisión del sistema permanece aceptable.

Esto es gracias a que la red está compuesta principalmente por módulos de convolución separables basadas en cuellos de botella, el cual consiste en generar una convolución 1x1 para aumentar la dimensión del canal, luego se hace una convolución separada en profundidad, realizando un filtrado ligero mediante la aplicación de un único filtro convolucional por canal de entrada para reemplazar las convoluciones regulares, en esencia ambas convoluciones, la regular y la separada por profundidad funcionan de la misma manera pero reduciendo significativamente la complejidad computacional.

Finalmente, se realiza otra convolución de 1x1 para reducir la dimensión del canal construyendo nuevas características al calcular combinaciones lineales de los canales de entrada. En la Figura 3. se aprecia este modelo de cuello de botella. La arquitectura de Mobilenet v2 contiene inicialmente una capa convolucional con 32 filtros, seguidos de 19 capas de cuello de botella previamente descritos.

Para evitar la no linealidad en la salida de los filtros, se usó la rectificación lineal ReLU6 (He et al., 2018). Con base en el estándar de las redes actuales, Mobilenet utiliza filtros de 3x3, un *dropout* (operación donde las neuronas generan un aprendizaje parcial de la red pare evitar el sobre entrenamiento) y normalización de lote durante el entrenamiento.

La experimentación que se tuvo con este modelo mostró que para redes menos profundas genera resultados más precisos, mientras que las redes de mayor profundidad tienen un rendimiento ligeramente más preciso al segmentar con filtros más profundos.

En (Romera et al., 2018), se propone la red factorizada residual eficiente (ERFNet por sus siglas en ingles). El elemento central de esta arquitectura es el diseño de una capa novedosa que aprovecha convoluciones con los núcleos de una dimensión y las conexiones saltadas mostradas en la Figura 4. donde la entrada se suma con el resultado de las capas convolucionales.

Mientras que las conexiones de salto permiten que las convoluciones aprendan funciones residuales que facilitan el entrenamiento, las convoluciones factorizadas permiten una reducción significativa de los costos computacionales, pero manteniendo una precisión similar a las redes con convoluciones de dos dimensiones (Asadi, Chen, Han, Wu, & Lobaton, 2019).

El bloque propuesto se apila secuencialmente para construir la arquitectura de la forma autocodificador, que produce la segmentación semántica de extremo a extremo, como se muestra en la Figura 5. en una visualización de la arquitectura de ERFNet.

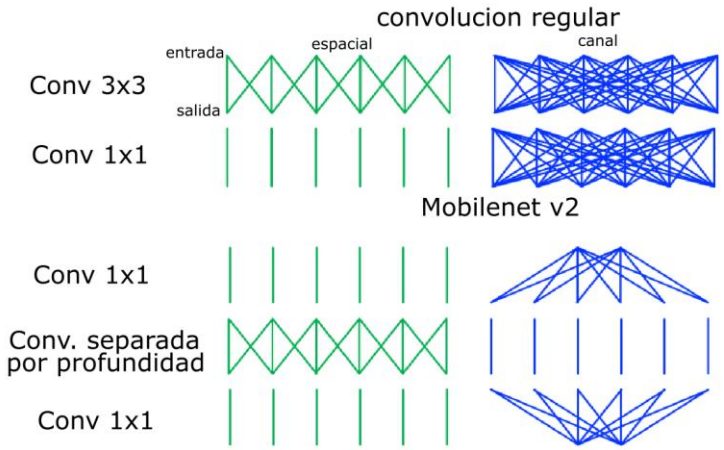

Figura 3**.** Diferencias entre las convoluciones de 1x1 y 3x3 regulares y el cambio de canal en arquitectura Mobilenet v2 de forma de cuello de botella

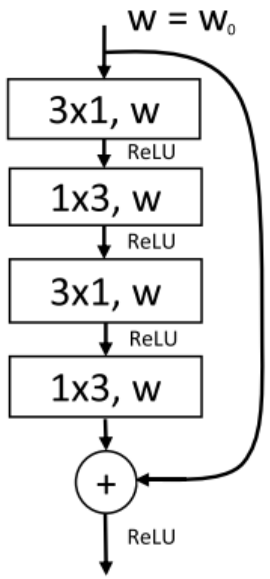

Figura 4. Representación de capas residuales, w representa el número de mapas de características que entran a cada capa, reducidos internamente en 4 en el diseño de cuello de botella.

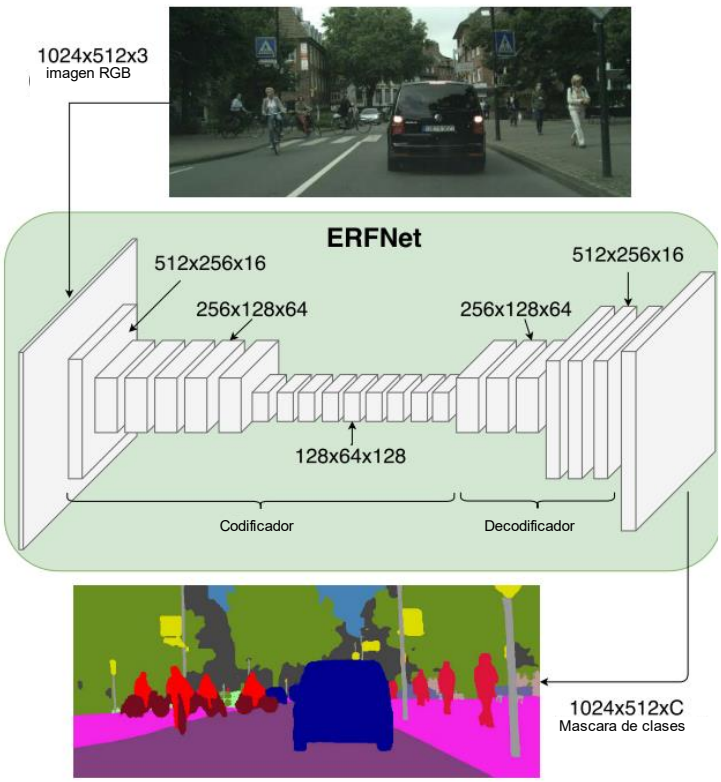

Figura 5**.** Arquitectura de ERFNet, con una imagen de alta resolución como entrada generando una máscara de segmentación de la misma resolución (Romera et al., 2018).

Por otro lado, en (Mehta, Rastegari, Shapiro, et al., 2018) se presentó una segunda versión de la red de pirámide espacial eficiente de convoluciones dilatadas (ESPNet por sus siglas en ingles), una red eficiente en términos de procesamiento, memoria y potencia. Esta arquitectura se muestra en la Figura 6. el bloque principal de construcción es un bloque convolucional nuevo llamado pirámide espacial eficiente (ESP por sus siglas en inglés) y se basa en descomponer una convolución estándar en dos pasos:

1. *Convolución puntual.* Consiste en el uso de un kernel de 1x1 para proyectar mapas de características de alta dimensión en un espacio de baja dimensión.
2. *Pirámide de convolución dilatada espacial.* Es un mecanismo similar al de ASPP que aprovecha las convoluciones atrous mostradas en la Figura 7. Pero en este caso, es menos costoso computacionalmente debido al acomodo en forma de pirámide, ya que el módulo ASPP utiliza una serie de convoluciones dilatadas sin orden jerárquico y tiene la capacidad de usar diferentes niveles espaciales para aprender representaciones.

Esta ayuda a decodificar los mapas de características descomprimiéndolos para aprender diferentes representaciones de los campos semánticos a grandes campos receptivos (cantidad de información visual que puede adquirir cada neurona o kernel) efectivos, usando *N* kernels de una dimensión *n x n* de convoluciones dilatadas de forma simultánea, cada uno con una relación de $2^{N-1}, N = \{1, \dots, N\}$ que genera el total de parámetros *p*.

Esta factorización reduce drásticamente el número de parámetros y memoria requeridos por el módulo, mientras se mantiene un gran campo receptivo, teniendo un total de parámetros:

$$p = [(n-1)2^{N-1} + 1]^2 \quad (2)$$

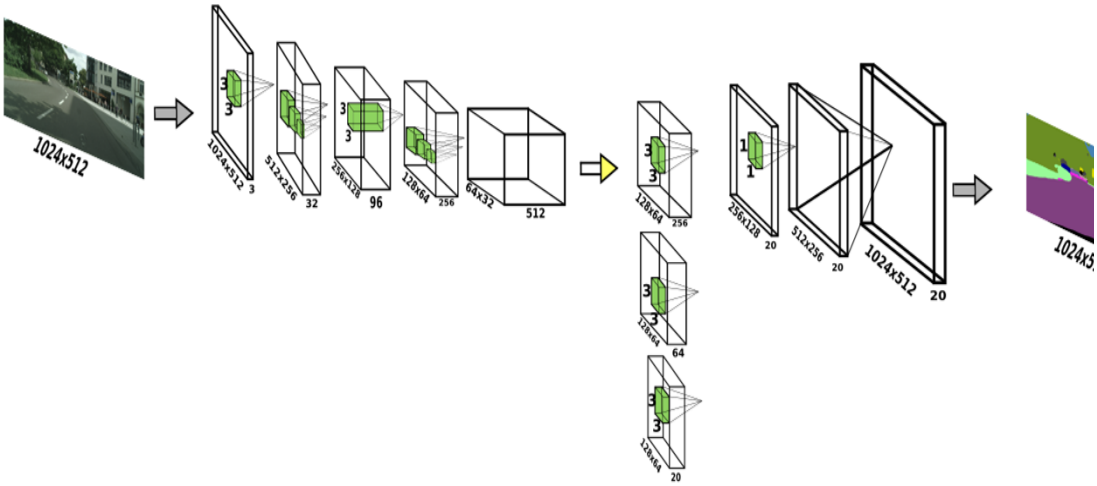

Figura 6. Estructura general de arquitectura ESPNet V2

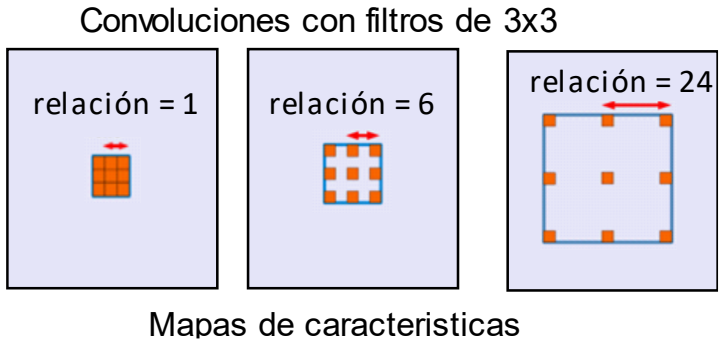

Figura 7. Convoluciones atrous con kernel de 3x3 de diferentes relaciones, la convolución regular corresponde al recuadro de la izquierda (Chen, Papandreou, Schroff, & Hartwig, 2017).

Esta operación de pirámide convolucional es llamada pirámide espacial de convoluciones dilatadas, porque cada kernel aprende pesos con diferentes campos receptivos y se asemeja a una pirámide espacial. Además, estos módulos ESP tienen un divisor de ancho *N* (total de filtros utilizados por capa) para reducir el costo computacional. Este parámetro tiene como propósito:

1. Reducir la dimensionalidad de las capas de características de manera uniforme en cada módulo ESP de la red. Es decir, por cada *N* los módulos reducen de un espacio dimensional *M* a un espacio $\frac{R}{K}$ usando la convolución puntual.
2. Dividir los espacios de características de baja dimensión en *K* ramas paralelas.
3. Transformar cada rama procesa los mapas de características de forma simultánea usando *kernels* de convoluciones dilatadas de *n x n* a diferentes relaciones de dilatación definidas por:

$$\boldsymbol{2^{N-1}, N = \{1, \dots, N-1\}} \quad (3)$$

4. Unir la salida de los kernel convolucionales dilatados paralelos *K* se concatena para producir un mapa de características de salida de dimensión *N.*

Finalmente, (Wu, Zhang, Huang, Liang, & Yu, 2019) presentaron la red totalmente convolucional rápida (FastFCN por sus siglas en inglés), y se trata de una arquitectura para SS que parte del modelo FCN la cual se muestra en la Figura 8. Utiliza convoluciones dilatadas en su estructura principal para extraer mapas de características de alta resolución. Para evitar una complejidad computacional pesada y una huella de memoria grande, se propone un módulo para descomprimir los mapas de características llamado pirámide de muestreo ascendente conjunta *(JPU por sus siglas en ingles).* Este módulo tiene como objetivo para una imagen de entrada baja resolución y un *groundtruth* de alta resolución generar una salida de alta resolución mediante la transferencia de detalles y estructuras de la imagen de entrada. Generalmente, el *groundtruth* de baja resolución Z_1 es generada empleando la transformación F(X) en la imagen de baja resolución de entrada X_l, por ejemplo, $Z_l = f(X_l)$. Teniendo X_l y Z_l es necesario obtener la transformación $\hat{\mathrm{F}}(\mathrm{X})$ para aproximar F(X), donde la complejidad computacional de $\hat{\mathrm{F}}(\mathrm{X})$ es menor que F(X). Por ejemplo, si F(X) es un perceptrón multi capa (MLP), entonces $\hat{\mathrm{F}}(.)$ puede ser simplificada como una transformación lineal. La imagen *grountruth* de alta resolución y_h es obtenida aplicando la transformación $\hat{\mathrm{F}}(.)$ en la imagen de alta resolución de entrada X_h de la forma $Z_h = \hat{\mathrm{F}}(X_h)$ formalmente dadas X_l, Z_l y X_h el muestreo ascendente conjunto es definido como:

$$\boldsymbol{Z_h = \hat{\mathbf{f}}(x_h), Donde\ \hat{\mathbf{f}}(.) = \min_{h(.) \in \varphi} ||y_1 - h(x_1)||} \quad (4)$$

Donde φ es el conjunto de todas las funciones de transformación posibles, y ||.|| es una métrica de distancia. El resto de la pirámide está formada por convoluciones y convoluciones dilatadas como se muestra en la Figura 9.

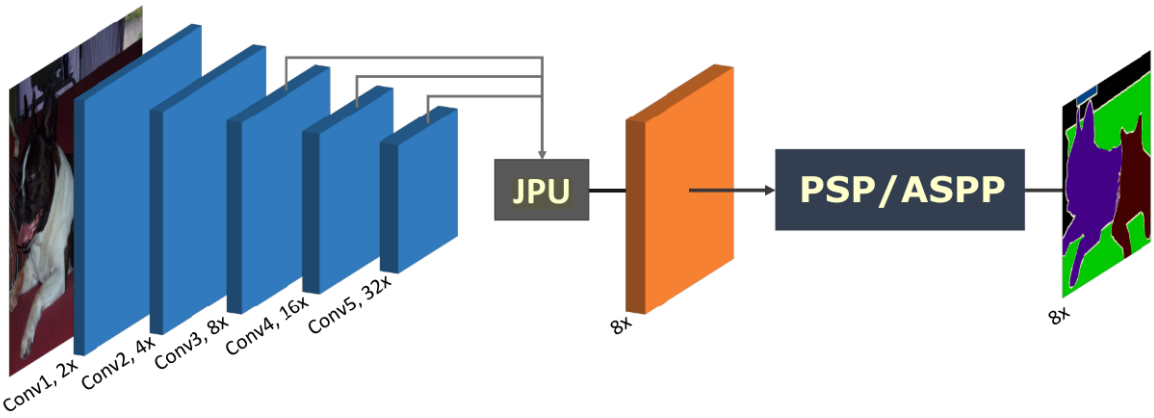

Figura 8. La arquitectura FastFCN que utiliza como base el modelo FCN, con un modelo novedoso llamado pirámide de interpolación articulada (JPU) que ayuda a interpolar y combinar las características obtenidas (Wu et al., 2019).

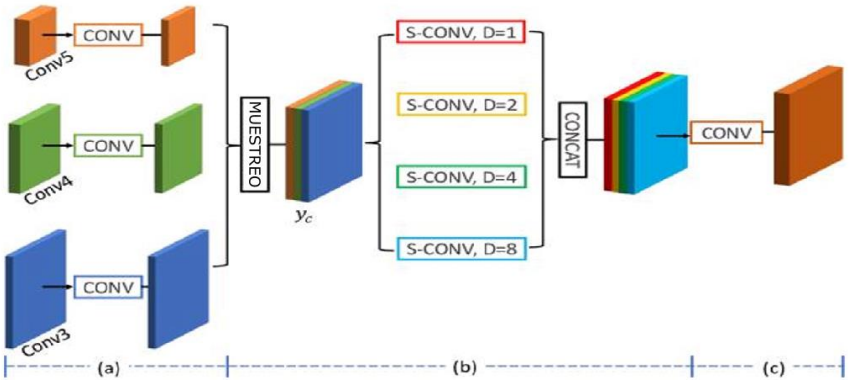

Figura 9. Modulo JPU visto de mejor perspectiva a colores.

4. Resultados

Las FCNs mencionados reportan buenos desempeños en la literatura con diversas base de datos como CamVid(Amoros et al., 2008), Mapillary Vistas (Neuhold, Ollmann, Bulo, & Kontschieder, 2017), Cityscapes (Cordts et al., 2016) y ADE20K (Zhou et al., 2017). CamVid fue la primera base de datos oficial para estudiar la SS y se compone de 700 imágenes grabadas desde un vehículo y tiene etiquetas clasificadas en 32 clases que representan objetos de un entorno urbano como edificios, caminos, vegetación, animales, luces de tráfico, ciclistas, trenes etc.

Mapillary Vistas tiene un total de 25,000 imágenes de alta resolución de diversas partes del mundo con etiquetas específicas para 37 clases. Cityscapes tiene un total de 5,000 *groundtruth* detallados de 50 ciudades diferentes y se generaron 30 clases diferentes divididas en 8 categorías que representan objetos urbanos. ADE20K es una de las bases de datos más extensa, ya que tiene un total de 20,000 imágenes de entrenamiento y 2,000 para validación. Cuenta con un total de 150 objetos etiquetados donde destacan entornos urbanos y diversos escenarios como hogares.

Entre los objetos contenidos en esta base de datos se encuentran camas, cuadros de pintura, muebles, artículos del hogar, etc.

Como se mencionó previamente, cada modelo fue implementado en diferentes plataformas de hardware y con diferentes bases de datos, por lo que no se puede hacer una evaluación objetiva de los modelos de su funcionamiento.

Por lo tanto, para realizar una evaluación del desempeño, en esta sección se presentan los resultados de un análisis de la precisión, tiempo de ejecución y huella de memoria de Enet, Mobilenet v2, ESPNet v2, ERFNet y FastFCN.

Esta evaluación se realizó con la base de datos de Cityscapes (Cordts et al., 2016), ya que es la más popular en la literatura y tiene un *brenchmark* (prueba de rendimiento o comparativa de un sistema) donde compiten diferentes modelos entrenados y probados con las diferentes imágenes de esta base de datos. La métrica utilizada es la PASCAL VOC *interection-over-union* que consiste en obtener la relación:

$$\boldsymbol{IoU} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{FN})} \quad \textbf{(4)}$$

donde TP, FP y FN son la cantidad de pixeles de verdadero positivo, falso positivo, y falso negativo respectivamente. La Tabla 1. muestra los resultados y comparación en tiempo de inferencia, promedio de precisión y huella de memoria en la base de datos Cityscapes. Respecto a los campos de la Tabla 1.

Precisión se tomó con la métrica IoU, el tiempo de inferencia se midió con la librería *time* de python y la huella de memoria se midió al propagar por la red una imagen de 3 canales a una resolución de 1024 por 512 pixeles, utilizando la herramienta *sumary* de pytorch. En este caso, únicamente FastFCN (Wu et al., 2019) fue probado en ADE20K y se obtuvo una precisión de 55.84% con un tiempo de inferencia de .025 segundos. FastFCN fue probado en esta base para otorgar otro punto de vista a estos modelos, ya que esta base de datos en específico no se centra en ambientes urbanos lo que aporta un mayor panorama del potencial que este tipo de redes eficientes tiene para otro tipo de aplicaciones.

Tabla 1**.** Comparación de modelos seleccionados en precisión y velocidad de inferencia con la base de datos Cityscapes con procesador Intel core i7 8750H con una GPU Gtx 1060 de 6Gb de memora para video.

	Precisión (IoU)	Tiempo de inferencia	Huella de memoria
Enet	0.513	0.05 s	6779.34 MB
Mobilenet v2	0.707	0.1 s	1616.50 MB
ERFNet	0.727	0.02 s	1917.87 MB
ESPNet v2	0.626	0.0089 s	2471.59 MB

El método con mayor precisión es ERFNet, ya que tiene un IoU más alto que el resto de los métodos. Esta precisión se debe a la profundidad que presenta en su codificador, ya que al generar un mayor número de mapas de características cada vez más profundas y toma más información contextual del escenario. Por otro lado, el método con menor precisión es Enet, y se debe principalmente a que la contribución de este se enfocaba en proponer las convoluciones dilatadas para reducir la complejidad computacional del modelo. Luego de Enet, se propusieron modelos con este tipo de convolución que mejoraban la precisión.

En cuanto tiempo de inferencia, ESPnet v2 tiene el menor tiempo de ejecución debido a la descomposición que presenta en convoluciones grupales, puntuales y dilatadas acomodadas en un esquema piramidal. Mobilenet v2 presenta el mayor tiempo en inferencia, lo cual se atribuye a la profundidad que presenta su arquitectura, además de no utilizar la operación de convolución dilatada.

En cuanto a huella de memoria, el modelo que presenta la memoria total más baja fue Mobilenet v2 gracias a la poca profundidad que mantiene el modelo. Por otro lado, el modelo con más huella de memoria es Enet debido principalmente a que utiliza un mayor número de convoluciones regulares con filtros de más de 3 pixeles de dimensión.

Un análisis cualitativo de los modelos seleccionados se muestra en la Figura 10. donde se ven reflejados las puntuaciones de precisión alcanzados por cada modelo, de izquierda a derecha se puede observar la imagen de entrada seguida de la imagen objetivo o *groundtruth*, posteriormente los resultados de Enet, ERFNet, ESPNet v2 y MobileNet v2 donde es posible destacar en cada imagen la forma en que se segmentan objetos distintos de interés, en la primera imagen observamos como MobileNet v2 tiene un resultado mas limpio aunque omite el fondo negro ubicado en la esquina superior izquierda que se encuentra en el *groundtruth*, por otro lado en la segunda imagen es ERFNet el que logra segmentar con una mayor

precisión el objeto que le cuesta mas al resto de modelos, que en este caso es el camión blanco que pasa por delante de la camara.

Por ultimo la tercera imagen muestra un paso peatonal donde encontramos distintos objetos destacando las personas y los vehículos del fondo para este caso ERFNet y ESPNet v2 presentan los resultados mas parecidos al *groundtruth* de esta imagen. Estos resultados muestran una correlación directa con las puntuaciones generales mostradas anteriormente, y se logra observar los objetos que resultan ser mas complicados para segmentar por cada modelo presentado en este análisis.

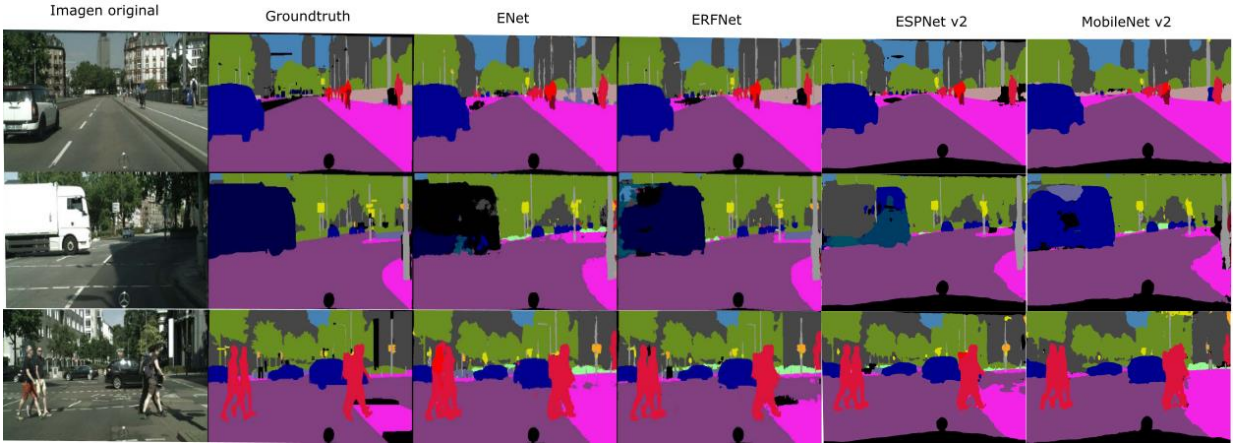

Figura 10. Análisis cualitativo de modelos comparando la imagen original, su *groundtruth* y cada una de las máscaras de segmentación generadas por cada modelo.

5. Conclusiones

Este artículo presenta un análisis del desempeño de las redes FCN eficientes enfocadas a segmentación semántica para hardware limitado. Las redes que fueron seleccionadas son Enet, Mobilnet v2, ERFNet, ESPNet v2 y FastFCN y las métricas utilizadas para el análisis son precisión, tiempo de inferencia y huella de memoria.

Estos modelos fueron evaluados con la base de datos Cityscapes y los resultados mostraron que los mejores métodos para SS fueron ERFNet gracias a su alta precisión obtenida validando el método, ESPNet v2 debido al poco tiempo que requiere para inferir el resultado de segmentación por cada imagen de entrada y Mobilenet v2 gracias a su reducida huella de memoria generado por los parámetros de la red.

Los buenos resultados en precisión de ERFNet se deben a que su modelo de baja profundidad y convoluciones con salto residual permiten una rápida convergencia en el entrenamiento, lo cual implica que la red siempre reduce la función de costo hasta encontrar los valores de los pesos de las neuronas que generan el error mínimo. Otro aspecto que ayuda a ERFNet es que el módulo JPU descomprime con una menor cantidad de FLOPS que las demás redes y logra generalizar los mapas de características obtenidos por las capas posteriores.

Por otro lado, ESPNet, generó una baja cantidad de FLOPS debido a la baja complejidad computacional obtenida por la estructura tipo ASPP. No obstante, este modelo no tuvo la menor huella de memoria ya que genera una gran cantidad de información con las convoluciones dilatadas en forma de pirámide.

De esta manera, se puede ver que diversas propuestas para la arquitectura de las redes influyen en la precisión o en el tiempo de inferencia. A futuro, los modelos de FCN eficientes se deben enfocar en mantener un equilibrio en el total de filtros utilizados en cada capa, haciendo una inspección del total de parámetros generados con herramientas que permitan el ambiente en el que se esté desarrollando el modelo.

Una estrategia clara presente en los trabajos analizados fue el evitar usar convoluciones completas ya que el total de parámetros generados por esta operación incrementa de forma exponencial, mientras que las ASPP y las convoluciones factorizadas grupales y puntuales reducen significativamente el total de parámetros generados.

El único problema con estas técnicas es la perdida de información generada por los huecos y los puntos de la imagen en donde no se convoluciona el filtro. Por lo tanto, es necesario incrementar y reestructurar los filtros dilatados de manera que se obtenga la información de contexto de cada objeto en su totalidad.

Referencias.

Amoros, P., Balsells, M. A., Buisan, M., Byrne, S., Fuentes-Pelaez, N., & Gabriel J. Brostow a, b,*, Julien Fauqueur a, R. C. a. (2008). Semantic object classes in video: A high-definition ground truth database. *Revista de Cercetare Si Interventie Sociala*, *42*(2), 120–144. https://doi.org/10.1016/j.patrec.2008.04.005

Asadi, K., Chen, P., Han, K., Wu, T., & Lobaton, E. (2019). *Real-time Scene Segmentation Using a Light Deep Neural Network Architecture for Autonomous Robot Navigation on Construction Sites*. Retrieved from http://arxiv.org/abs/1901.08630

Chen, L., Papandreou, G., Schroff, F., & Hartwig, A. (2017). *Rethinking Atrous Convolution for Semantic Image Segmentation*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … Schiele, B. (2016). *The Cityscapes Dataset for Semantic Urban Scene Understanding*. https://doi.org/10.1109/CVPR.2016.350

He, K., Zhang, X., Yang, H., Han, K., Zhu, D., Lun, P., & Zhao, Y. (2018). Delving Deep into Rectifiers: Surpassing Human-Level performance on imagenet classification. *Biochemical and Biophysical Research Communications*, *498*(1), 254–261. https://doi.org/10.1016/j.bbrc.2018.01.076

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. https://doi.org/10.1007/s13398-014-0173-7.2

Lateef, F., & Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, *338*, 321–348. https://doi.org/10.1016/j.neucom.2019.02.003

Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., & Hajishirzi, H. (2018). ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11214 LNCS*, 561–580. https://doi.org/10.1007/978-3-030-01249-6_34

Mehta, S., Rastegari, M., Shapiro, L., & Hajishirzi, H. (2018). *ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network*. Retrieved from http://arxiv.org/abs/1811.11431

Neapolitan, R. E., & Neapolitan, R. E. (2018). Neural Networks and Deep Learning. In *Artificial Intelligence*. https://doi.org/10.1201/b22400-15

Neuhold, G., Ollmann, T., Bulo, S. R., & Kontschieder, P. (2017). The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 5000–5009. https://doi.org/10.1109/ICCV.2017.534

Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). *ENet : A Deep Neural Network Architecture for Real-Time Semantic Segmentation*. 1–10.

Romera, E., Álvarez, J. M., Bergasa, L. M., & Arroyo, R. (2018). *ERFNet : Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation*. *19*(1), 263–272.

Sandler, M., Zhu, M., Zhmoginov, A., & Apr, C. V. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*.

Sevak, J. S., Kapadia, A. D., Chavda, J. B., Karungan, D., & Sujatha, N. (2017). Survey on Semantic Image Segmentation Techniques. *Proceedings of the International Conference on Intelligent Sustainable Systems*, *4*(Iciss), 306–313. Retrieved from www.jetir.org

Shelhamer, E., Long, J., & Darrell, T. (2017). *Fully Convolutional Networks for Semantic Segmentation*. *39*(4), 640–651.

Taylor, G. W. (2010). *Deconvolutional Networks slides*. 2528–2535. https://doi.org/10.1109/CVPR.2010.5539957

Wu, H., Zhang, J., Huang, K., Liang, K., & Yu, Y. (2019). *FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation*. (1). Retrieved from http://arxiv.org/abs/1903.11816

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene parsing through ADE20K dataset. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 5122–5130. https://doi.org/10.1109/CVPR.2017.544

Notas biográficas

Oscar Alejandro Soto Orozco. Obtuvo el grado de Ingeniero en Electrónica del Instituto Tecnológico Nacional campus Chihuahua 2018 y actualmente se encuentra estudiando para obtener el grado de Maestro en Ciencias en Ingeniería Electrónica del Instituto Tecnológico de Chihuahua, su investigación es en el área de procesamiento digital de señales e imágenes, enfocado a segmentación semántica en video con hardware limitado.

Alma Delia Corral Sáenz. Recibió el título de Ingeniera en Sistemas Computacionales en Hardware de la Universidad Autónoma de Chihuahua en 1999 y el de Maestra en Ciencias en Ingeniería Electrónica del Instituto Tecnológico de Chihuahua en 2003. Actualmente es profesora y coordinadora del Doctorado y la Maestría en Ciencias en Ingeniería Electrónica en el mismo Instituto, y participa en trabajos de investigación de las áreas de procesamiento de señales y visión por computadora.

Claudia Elizabeth Rojo-González. Recibió el título de Contador Público en 1995 y la Maestría en Administración en 2000, ambos títulos en la Universidad Autónoma de Chihuahua. Actualmente es profesora y coordinadora de la Maestría en Administración de Negocios en el Instituto Tecnológico de Chihuahua, y participa en trabajos de investigación auxiliando a profesores de ingeniería en el desarrollando de modelos financieros para desarrollo tecnológico.

Juan Alberto Ramírez Quintana. Recibió los grados de ingeniería (2004), maestría (2007) y doctorado (2014) en ingeniería electrónica del Instituto Tecnológico de Chihuahua, México. Actualmente trabaja como profesor-investigador en el Instituto Tecnológico de Chihuahua, cuenta con diversas publicaciones en revistas y congresos y dirige varias tesis a nivel licenciatura maestría y doctorado. Sus áreas de interés son visión por computadora, procesamiento de señales, aprendizaje automático, percepción visual y sistemas embebidos. El Dr. Ramírez es miembro del Sistema Nacional de Investigadores de México.