MobileNetV2: Inverted Residuals and Linear Bottlenecks

Índice

Aprendizaje Profundo

Profesores

Miguel A. Gutierrez Naranjo
David Solis Martin

Alumno

Germán Lorenz Vieta

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Introducción

Sobre redes profundas

Desde LeNet-5 ‘98 a AlexNet ‘12 no hubo avances significativos por desconocimiento del potencial de las redes profundas

Desde AlexNet pasando por VGGNet, GoogLeNet/Inception, ResNet, etc se probaron múltiples arquitecturas de redes convolucionales y estrategias para aumentar la detección.

Cuanto más profundo es el modelo su complejidad computacional tiende a aumentar en millones de MAdds.

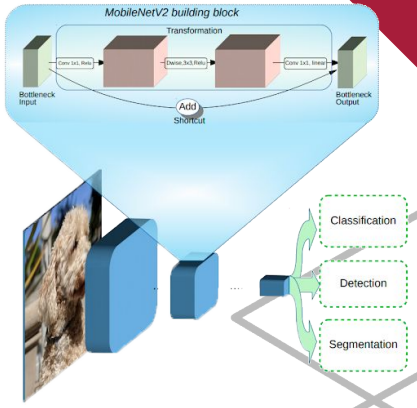
Cuanto más rápido es el modelo se tiende a perder capacidad de detección.

La conectividad y escasez están relacionadas con la rapidez.

Introducción

¿Qué busca MobileNetV2?

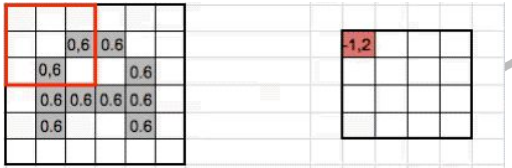
- Arquitectura para entornos móviles
- Optimizar el uso de recursos cuando estos son limitados
- Disminución de operaciones matemáticas complejas y uso de los distintos tipos de memoria de los sistemas
- Mantener la precisión
- Adaptarse a multiples propositos

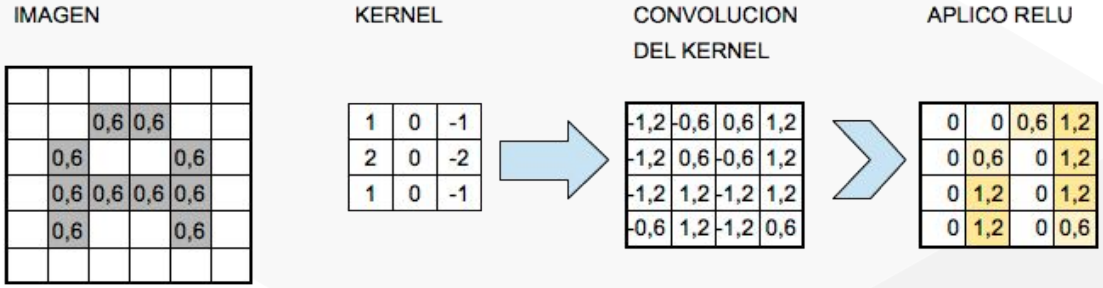


Intuiciones sobre el modelo

Convolución y la búsqueda de una estructura liviana

- En CNN una imagen es una matriz de números
- La convolución es aplicar kernels para obtener mapas de caracteristicas
- El mapa de caracteristicas se obtiene al aplicar una función de activación al filtrado



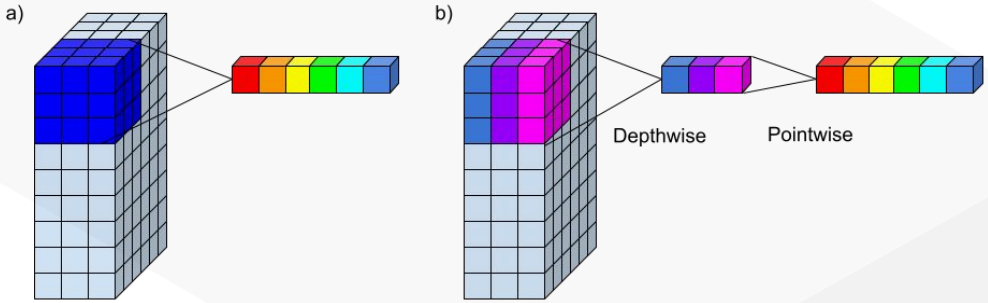


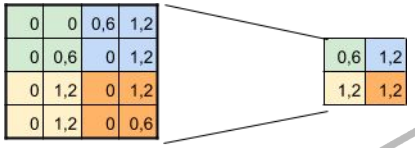
Escuela Técnica Superior de Ingeniería Informática

Intuiciones sobre el modelo

Convolución y la búsqueda de una estructura liviana

- Para reducir dimensionalidad se usan distintos tipos de subsampling
- En MobileNetV1 se reemplaza esta estrategia por un bloque Depthwise Separable Convolution y Pointwise Convolution





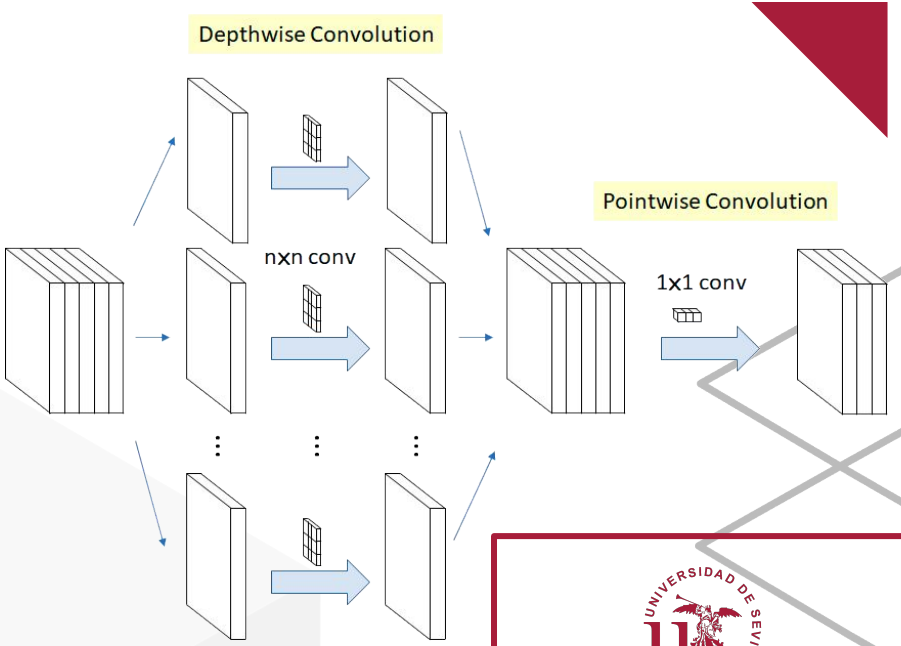
SUBSAMPLING:
Aplico Max-Pooling de 2x2
y reduzco mi salida a la mitad

Escuela Técnica Superior de Ingeniería Informática

Intuiciones sobre el modelo

Depthwise Separable Convolution y Pointwise Convolution

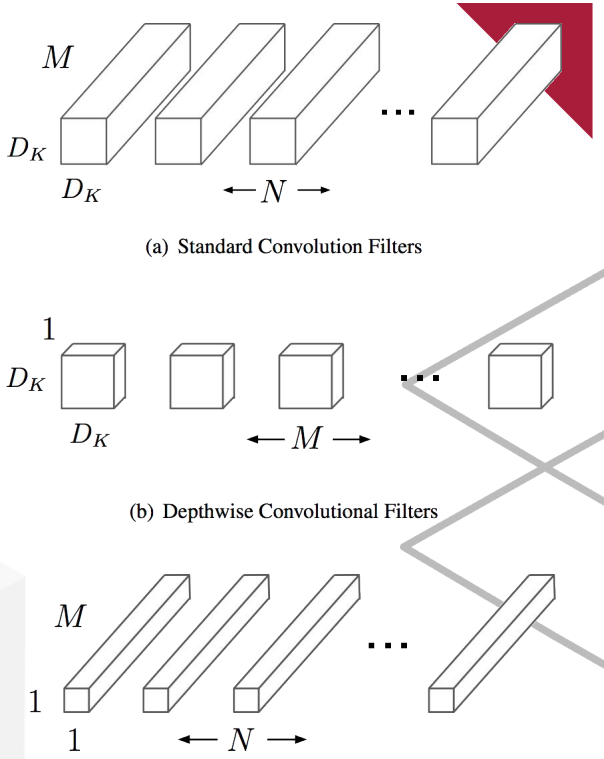
- Deep-Wise Separable Convolution realiza un filtrado ligero mediante convoluciones por canal de entrada
- Pointwise Convolution realiza mediante combinaciones lineales de los canales de entrada distintas funciones



Intuiciones sobre el modelo

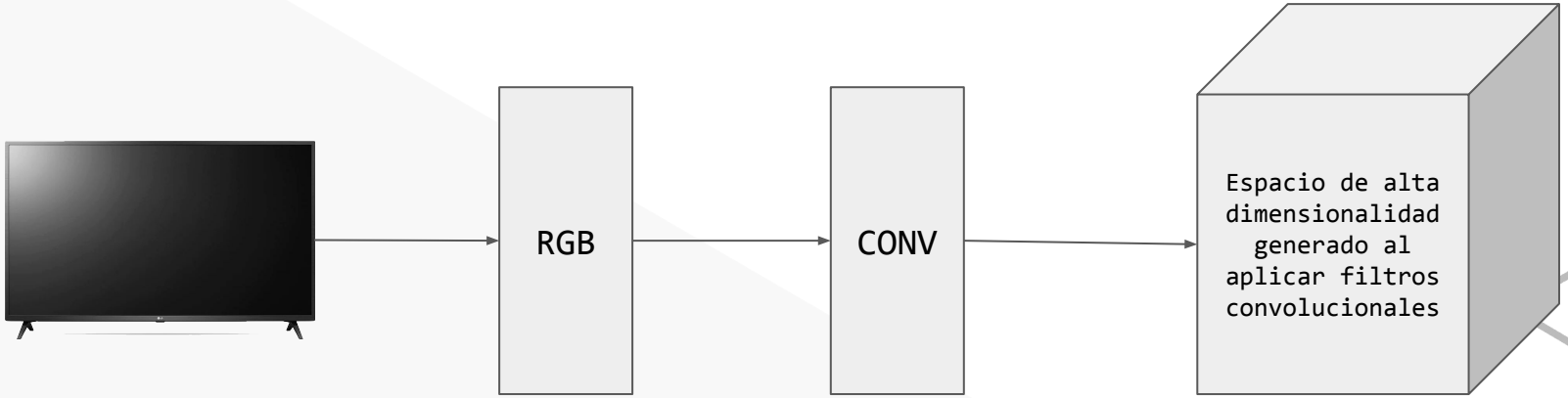
Costo computacional al separar procesos

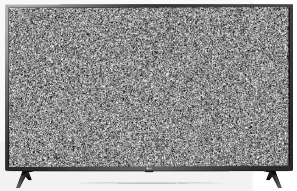
- Convolución Tradicional
  - Dimension de Input: h x w x d
  - Dimensión de Kernel: k x k x di x dj
  - Dimensión de Salida: h x w x di x dj
  - Costo: h * w * di * dj * k * k
- Depthwise Separable Convolution
  - Costo: h * w * di * (k^2 +dj) aproximadamente
- Según autor se reduce k^2 veces la complejidad mientras logra misma capacidad de detección



Bloque Inverted Residuals

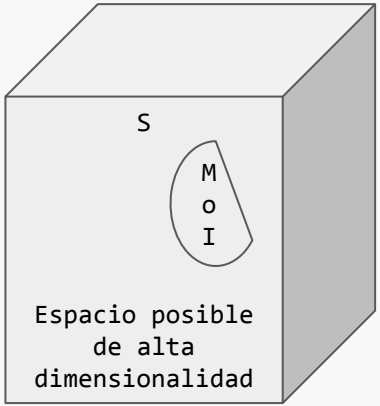
Principios conceptual del bloque Inverted Residual





Bloque Inverted Residuals

Principios conceptuales del bloque invertido



Si consideramos S el espacio posible

- El mundo real será un subespacio de S muy pequeño que consideraremos Manifold of Interest (MoI)
- Las imágenes entendibles serán aún un subespacio aún más pequeño
- Las no entendibles pueden ser otro subespacio de interés para el modelo

Entonces el autor considera que:

- Existe una “Variedad de Interés” con carga útil de información
- La dimensionalidad y la profundidad están relacionadas conceptualmente

Bloque Inverted Residuals

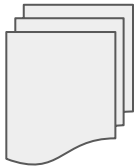
Intuición de la investigación dimensional por los autores

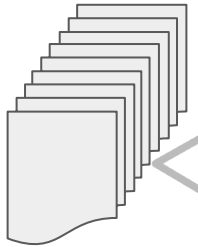
En MobileNetV1 el hiperparámetro α ∈ [0 , 1] y si por ejemplo vale 0.5 significa que la dimensionalidad de 128 se reduce a 64 lo que nos lleva a pensar de que a menor α menos MoI, entonces la "variedad de interes" capturara mejor las caracteristicas deseadas

PERO:

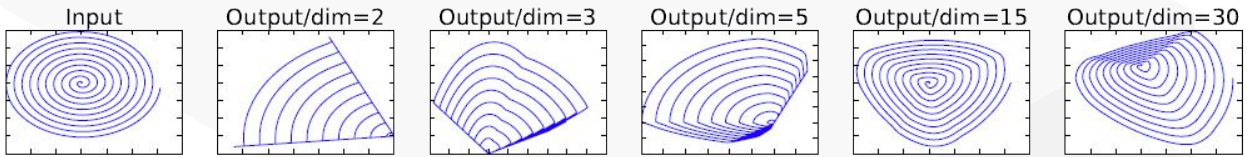
Si los canales disminuyen al aplicar ReLU existe alta probabilidad de pérdida de información

Si los canales aumentan al aplicar ReLU existe la probabilidad de perder menos información





Experimento de los autores al Transformar, aplicar ReLU y Invertir



Escuela Técnica Superior de Ingeniería Informática

Bloque Inverted Residuals

Conclusiones

- Aplicar ReLU en baja dimensión reduce la “Variedad de Caracteristicas”
- El MoI se logra codificando bajo bajas dimensiones obteniendo rapidez en el procesamiento por no usar S completo
- En base a esto se crea el **Bloque Inverted Residual con Lineal Bottleneck**

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s, and expansion factor t.

Bloque Lineal Bottleneck

Breve evolución del bloque Bottleneck

- El bloque Regular mezcla las capas
- El bloque Separable separa cada capa y añade el factor de expansión
- El bloque Separable con linear Bottleneck agregar el Pointwise para comprimir información
- El bloque Bottleneck con capa de expansión difiere en que su input es un Pointwise y su output también
    - PointWise es quien hace las combinaciones lineales de los canales de entrada

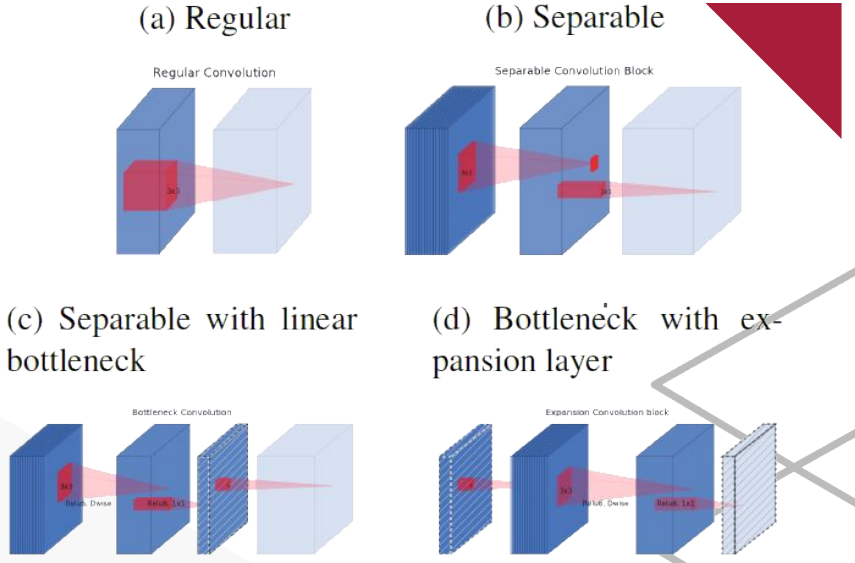
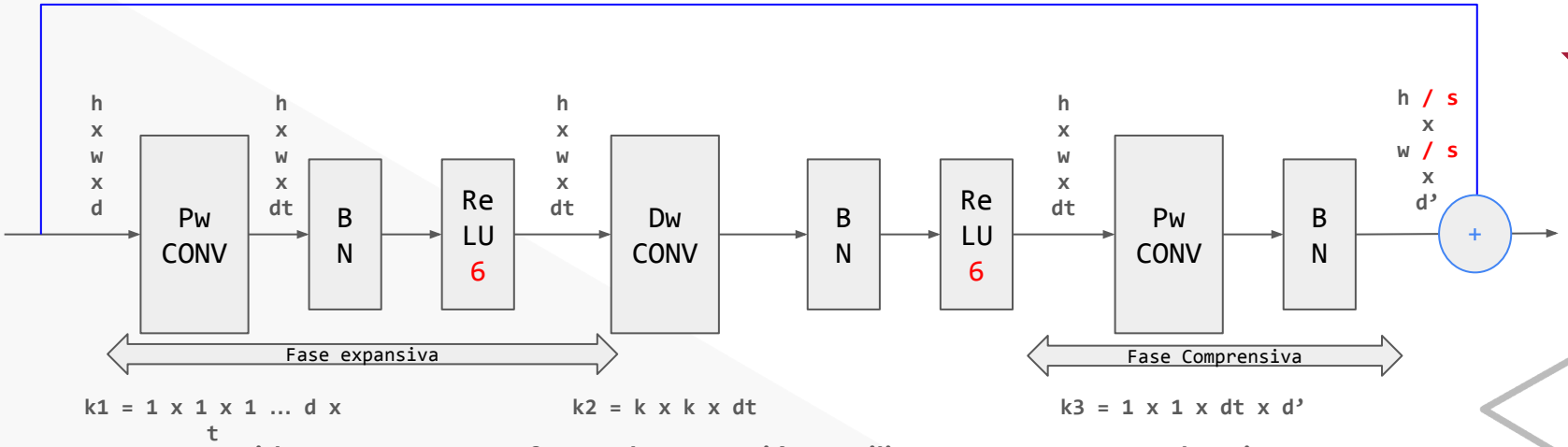


Figure 2: Evolution of separable convolution blocks. The diagonally hatched texture indicates layers that do not contain non-linearities. The last (lightly colored) layer indicates the beginning of the next block. Note: 2d and 2c are equivalent blocks when stacked. Best viewed in color.

Bloque Lineal Bottleneck

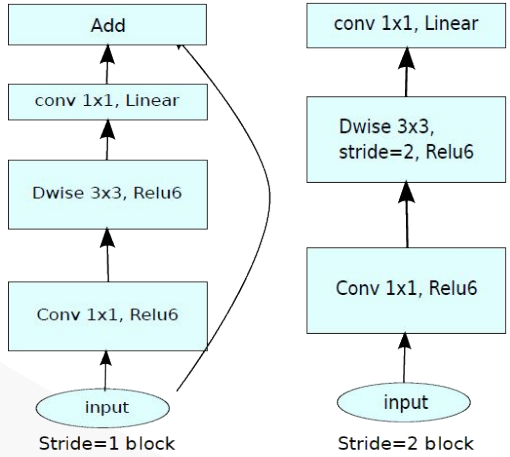


- Los autores consideran a t como un factor de expansión y utilizan t = 6 excepto en la primer capa
- ReLU6 es una implementación que evita la sobre dimensionalidad
- Modifican el comportamiento de Bottleneck para que tenga 2 comportamientos
- El kernel 3 controla la dimensionalidad de salida en el último bloque
- El bloque Inverted Residual se utiliza fusionando la salida cuando **d = d'**
- El bloque Inverted Residual se elimina cuando el Stride **s = 2**
- Se diferencia de MobileNetV1 ya que no pierde información en su salida

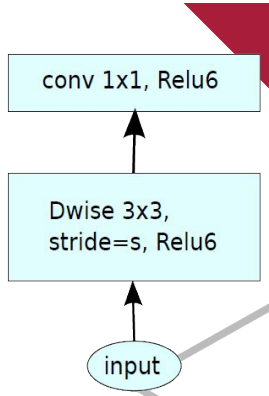
Arquitectura

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 2: MobileNetV2 : Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3×3 kernels. The expansion factor t is always applied to the input size as described in Table 1.



Mobilenet V2



(b) MobileNet[27]

Pruebas de rendimiento

Clasificación con ImageNet

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

Detección de Objetos con SSD en COCO

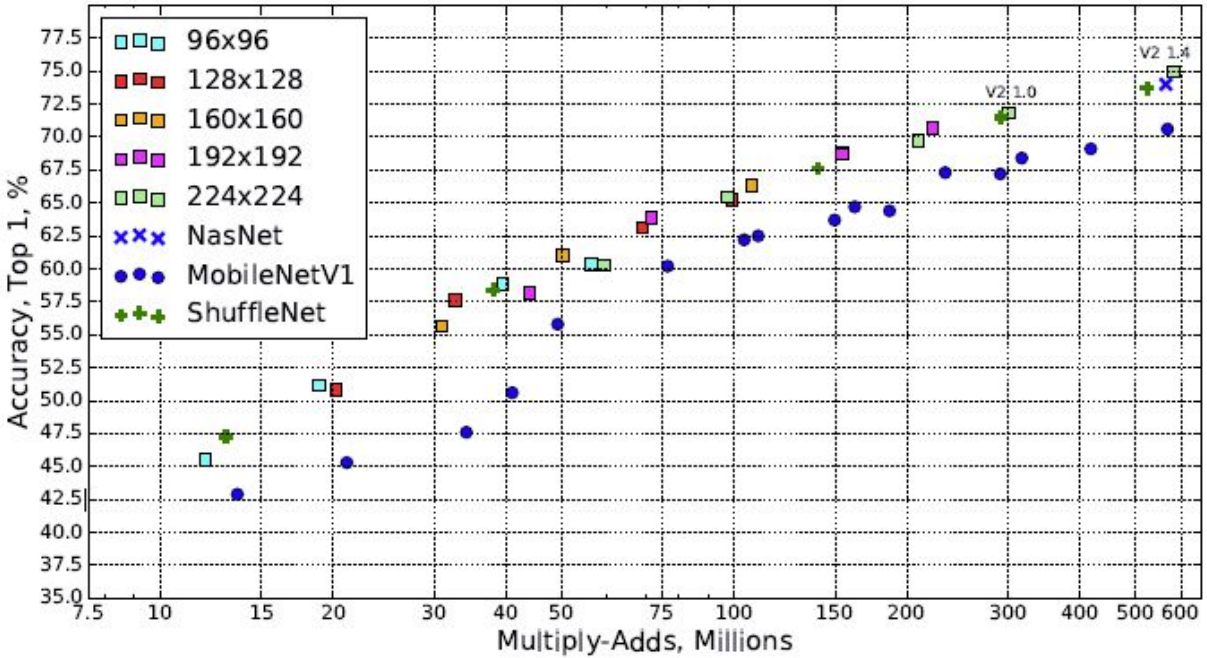
Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

Segmentación semántica con DeepLabv3 en PASCAL VOC 2012

Network	OS	ASPP	MF	mIOU	Params	MAdds
MNet V1	16	✓		75.29	11.15M	14.25B
	8	✓	✓	78.56	11.15M	941.9B
MNet V2*	16	✓		75.70	4.52M	5.8B
	8	✓	✓	78.42	4.52M	387B
MNet V2*	16			**75.32**	**2.11M**	**2.75B**
	8		✓	77.33	2.11M	152.6B
ResNet-101	16	✓		80.49	58.16M	81.0B
	8	✓	✓	82.70	58.16M	4870.6B

Pruebas de rendimiento

Curva de performance por MAdds entre MobileNetV1, MobileNetV2, ShuffleNet y NAS



Conclusiones

- En el artículo se describe una arquitectura de red simple que permite crear una familia de modelos para dispositivos móviles altamente eficiente.
- La arquitectura propuesta permite una inferencia muy eficiente en memoria confiando en operaciones estándar en todos los marcos neuronales.
- Para el conjunto ImageNet la arquitectura mejoró el estado de arte en cuanto a rendimiento.
- Para la detección de objetos la red supera a los mejores detectores en tiempo real en COCO tanto en precisión como en complejidad del modelo. La combinación con SSDLite brinda una capacidad superior a YOLOv2.
- Por el lado teórico el bloque convolucional propuesto tiene una propiedad única que separa la expresividad de la red de su capacidad