Semantic Segmentation on Resource Constrained Devices

Sachin Mehta

University of Washington, Seattle

In collaboration with Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi

Project page: https://sacmehta.github.io/ESPNet/

Problem Statement

- Limited computational resources
  - Only **256 CUDA cores** in comparison to standard GPU cards such as TitanX which has **3500+ cuda cores**
- CPU and GPU shares the RAM
- Limited Power (TX2 can run in two modes that has TDP requirement of 7.5V [Max-Q] and 15 V [Max-P])
  - Max-Q's performance is identical to TX1. GPU Clock @ 828 MHz
  - Max-P boosts the clock rates to the max. value. GPU clock @1300 MHz

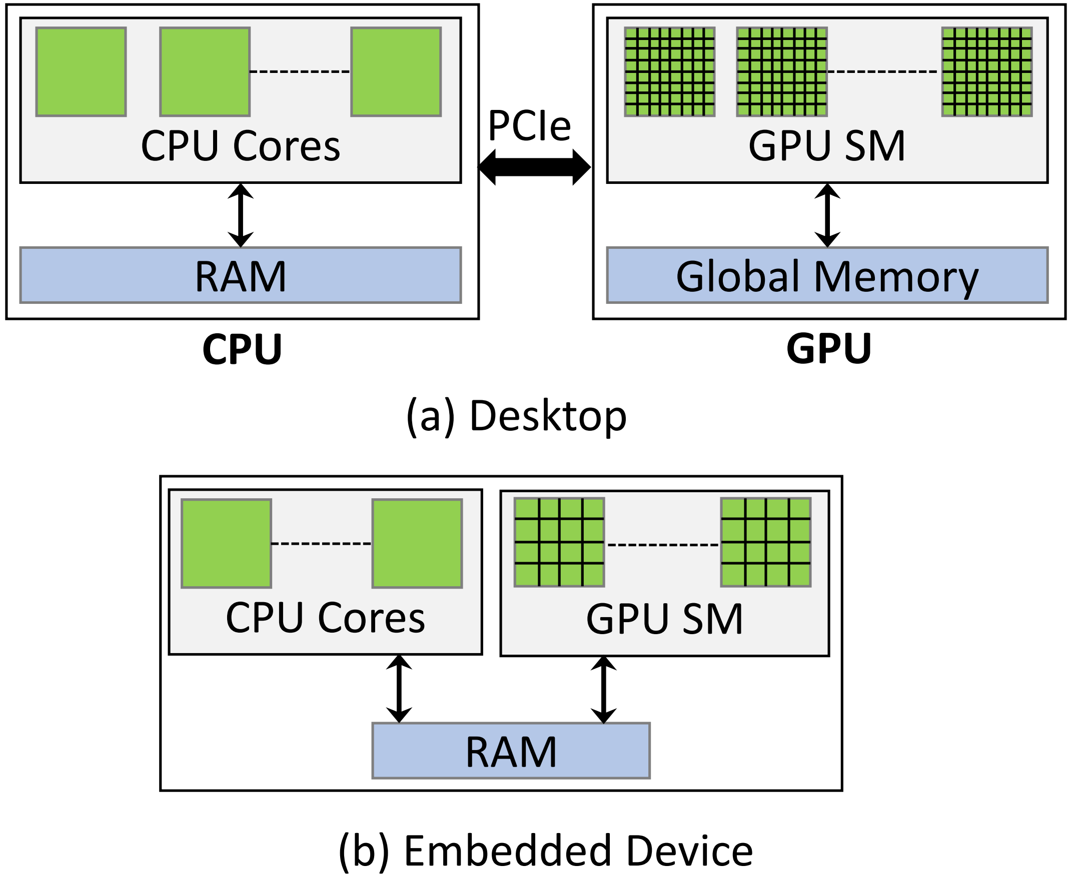
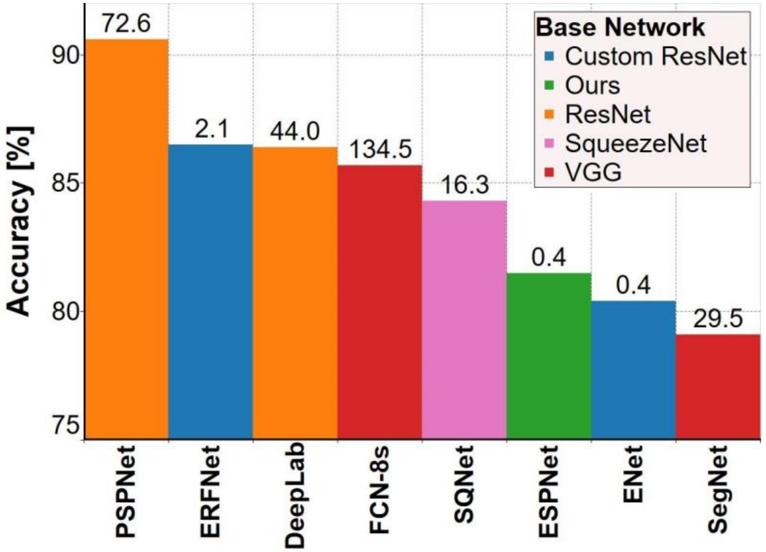
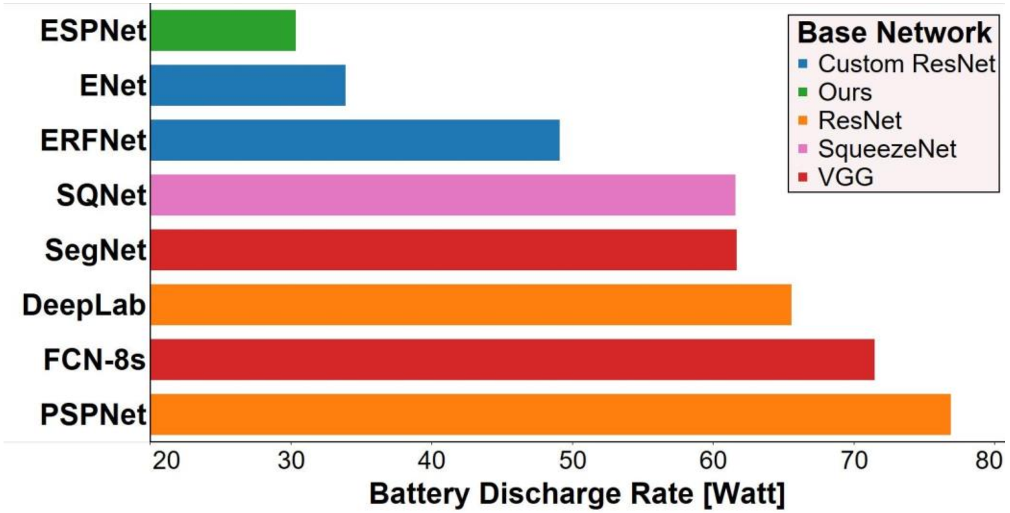


Figure: Hardware-level resource comparison on a desktop and embedded device

Problem Statement

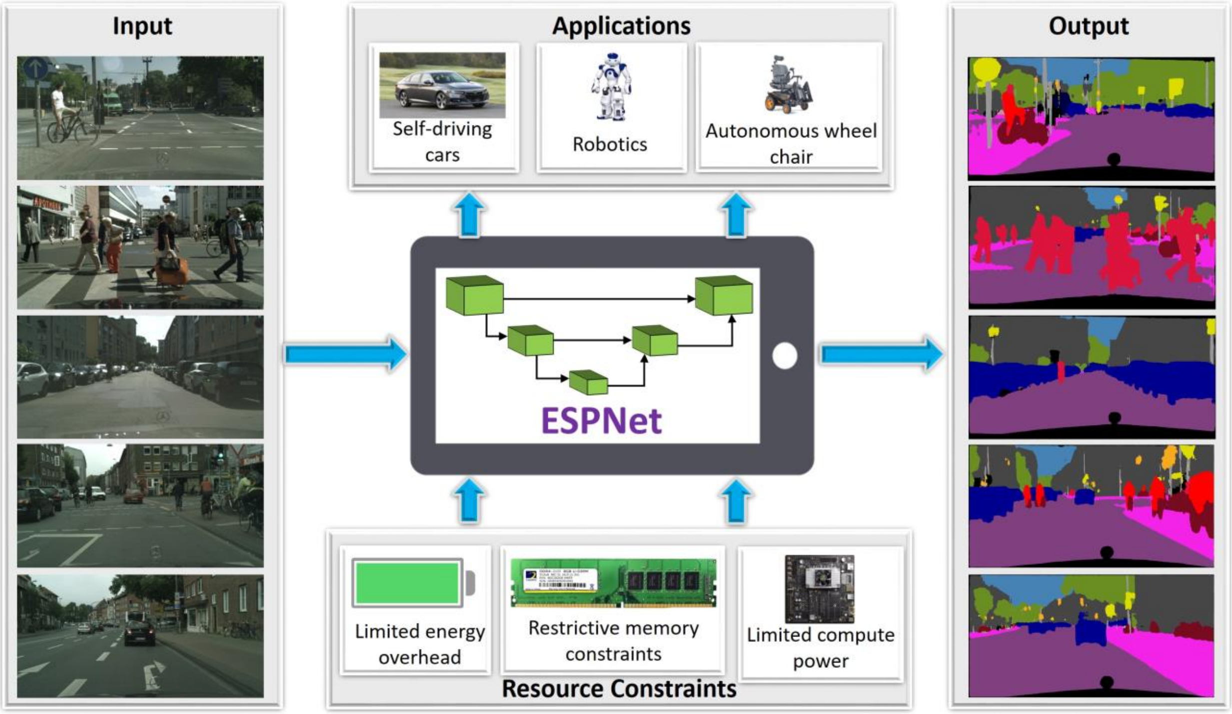
- Accurate segmentation networks are deep and learns more parameters. As a consequence, they are **slow and power hungry**.





Problem Statement

- Accurate segmentation networks are deep and learns more parameters. As a consequence, they are **slow and power hungry**.
- Deep networks cannot be used in embedded devices because of hardware constraints
  - Limited computational resources
  - Limited energy overhead
  - Restrictive memory constraints



Agenda

- What is semantic segmentation?
- CNN basics
- Overview of SOTA efficient networks
- ESPNet
- Results

What is Semantic Segmentation?

Input: RGB Image

Output: A segmentation Mask

Overview

- A standard CNN architecture stacks
  - Convolutional layers
  - Pooling layers
  - Activation and Batch normalization layers (see [r1])
  - Linear (Fully connected) layers

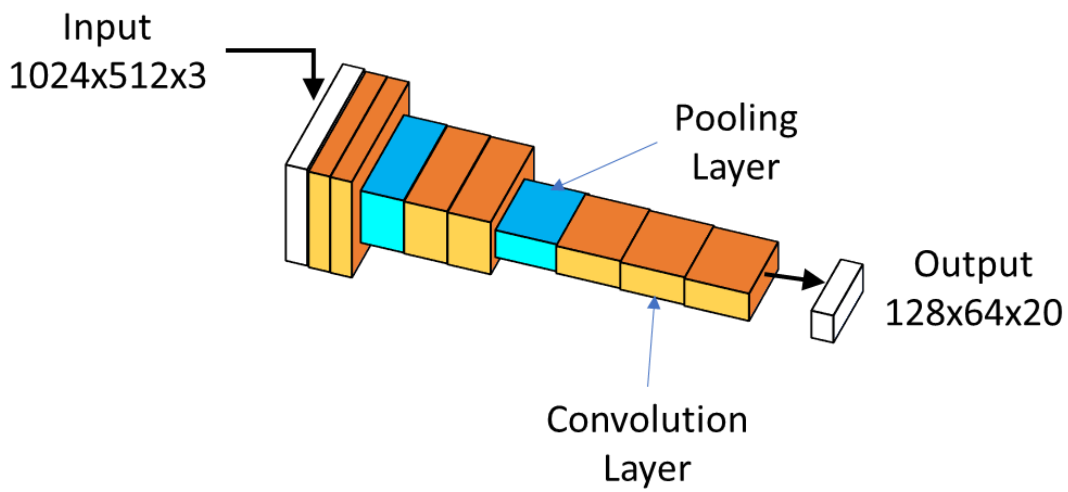


Figure: Example of Stacking layers in CNN network

Source:
[r1] Xu, Bing, et al. "Empirical evaluation of rectified activations in convolutional network." *arXiv preprint arXiv:1505.00853* (2015).

Overview: Convolution

- A convolution layer compute the output of neurons that are connected to local regions in the input.
- For a CNN processing RGB images, a convolutional kernel is usually a 3-dimensional ($M \times n \times n$) that is applied over M channels to produce the output feature map.

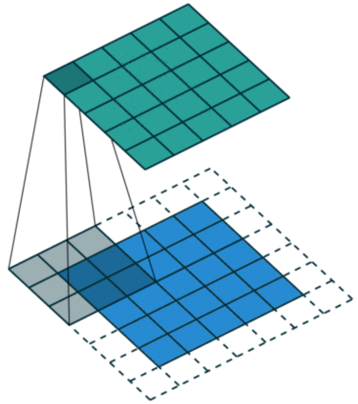


Figure: An example of 3x3 convolutional kernel processing an input of size 5x5

Source: http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

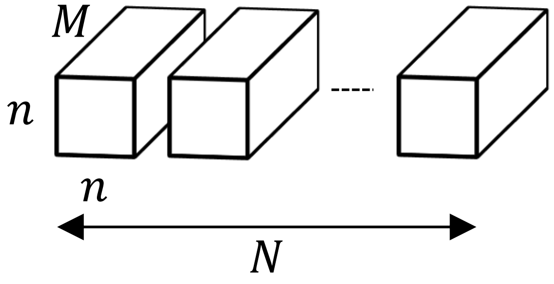


Figure: A convolutional kernel visualization

Pooling

- Pooling operations help the CNN network to learn scale-invariant representations.
- Common pooling operations are:
  - Max. Pooling
  - Average Pooling
  - Strided convolution

Pooling: Max Pooling

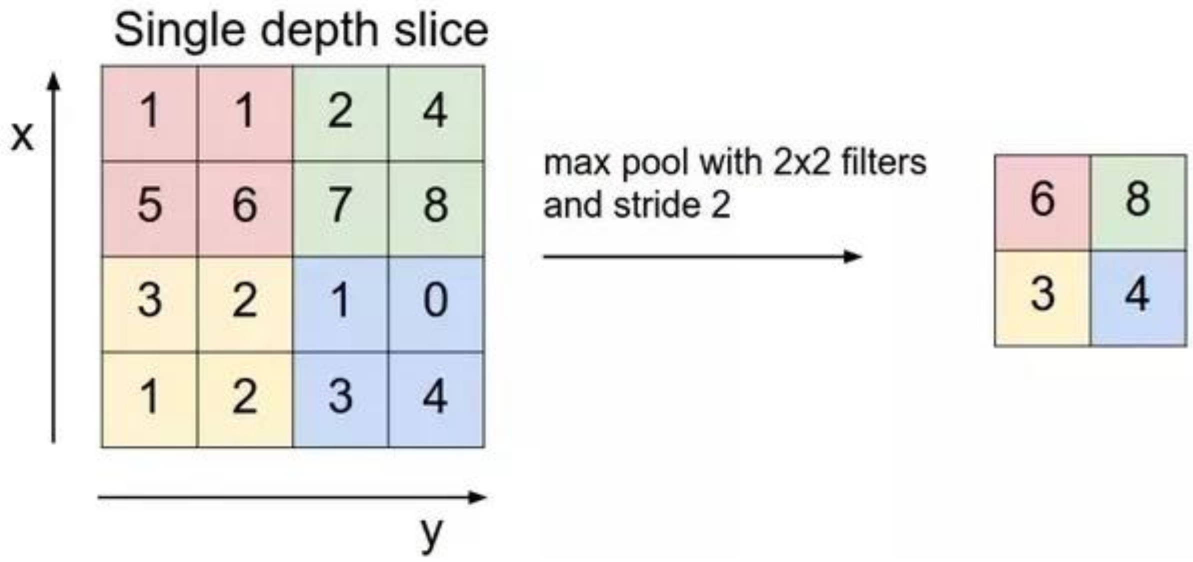


Figure: Max pooling example

Note: Average pooling layer is the same as Max pooling layer except that the kernel is performing a averaging function instead of maximum.

Source: http://cs231n.stanford.edu/

Pooling: Strided Convolution

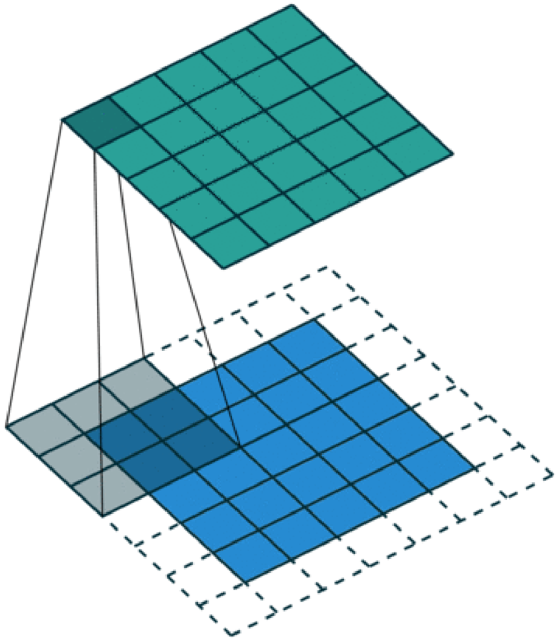


Figure: 3x3 convolution with a stride of 1

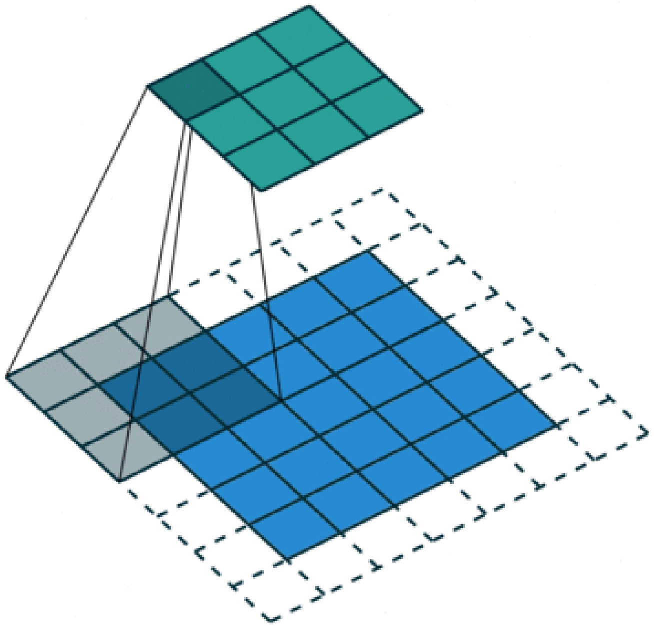


Figure: 3x3 convolution with a stride of 2

Source: http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Efficient Networks

MobileNet

- Uses depth-wise separable convolution
  - First compute kernel per input channel
  - Apply point-wise convolution to increase the number of channels.

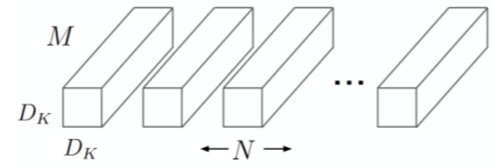


Figure: A standard convolution kernel

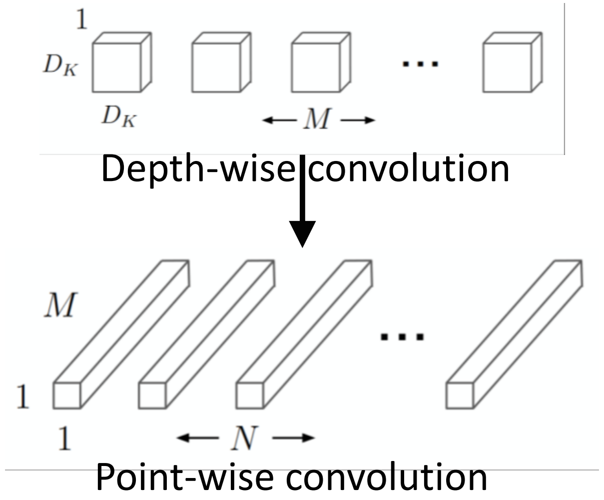


Figure: Depth-wise separable convolution kernel

MobileNet

- Uses depth-wise separable convolution
  - First compute kernel per input channel
  - Apply point-wise convolution to increase the number of channels.

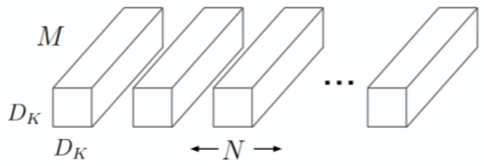


Figure: A standard convolution kernel

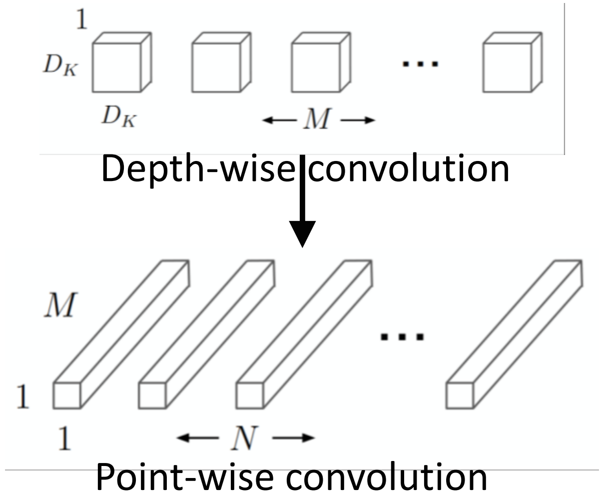


Figure: Depth-wise separable convolution kernel

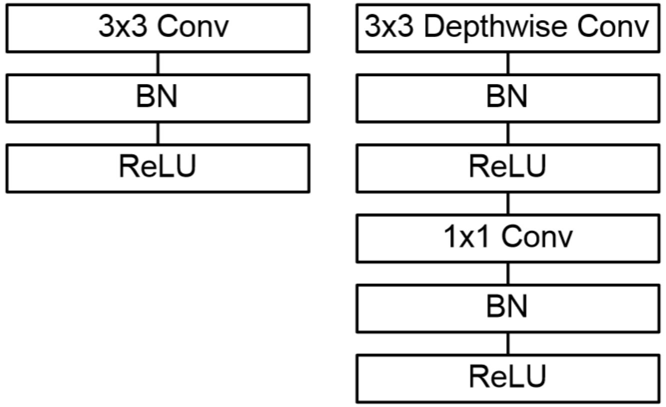


Figure: Block-wise representation

Source:
Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

ShuffleNet

- ShuffleNet uses the similar block structure as ResNet, but with following modifications:
  - 1x1 point-wise convolutions are replaced with grouped convolution
  - 3x3 standard convolutions are replaced with the depth-wise convolution

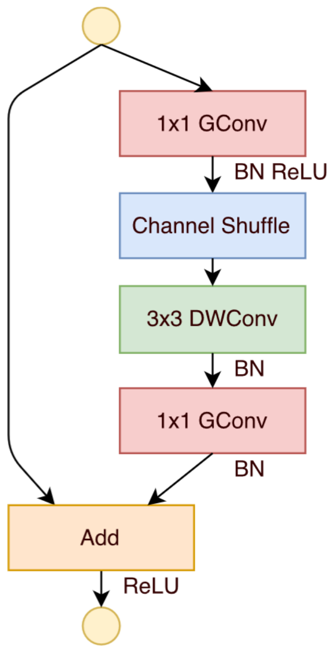


Figure: ShuffleNet block

Source:
Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." arXiv preprint arXiv:1707.01083 (2017).

ShuffleNet

- ShuffleNet uses the similar block structure as ResNet, but with following modifications:
  - 1x1 point-wise convolutions are replaced with grouped convolution
  - 3x3 standard convolutions are replaced with the depth-wise convolution

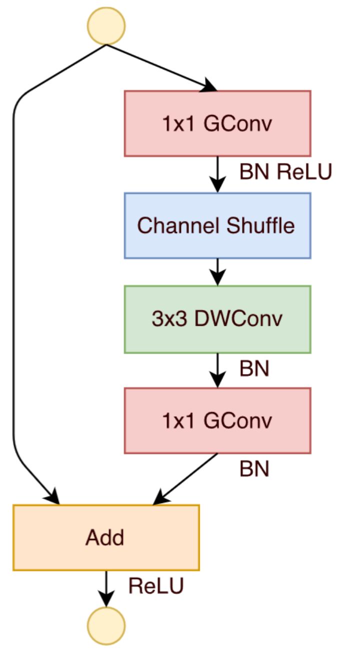


Figure: ShuffleNet block

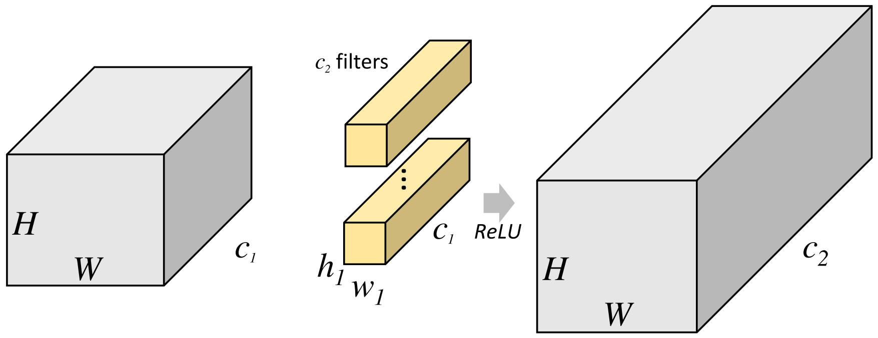


Figure: Standard convolution

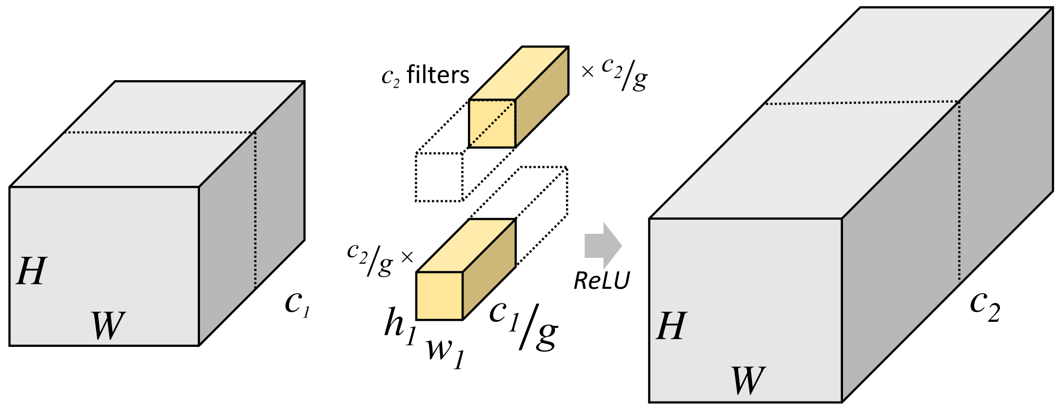


Figure: Grouped convolution

Source: https://blog.yani.io/filter-group-tutorial/

ESPNet

ESP Block

- ESP is the basic building block of ESPNet
- Standard convolution is replaced by
  - Point-wise convolution
  - Spatial pyramid of dilated convolution

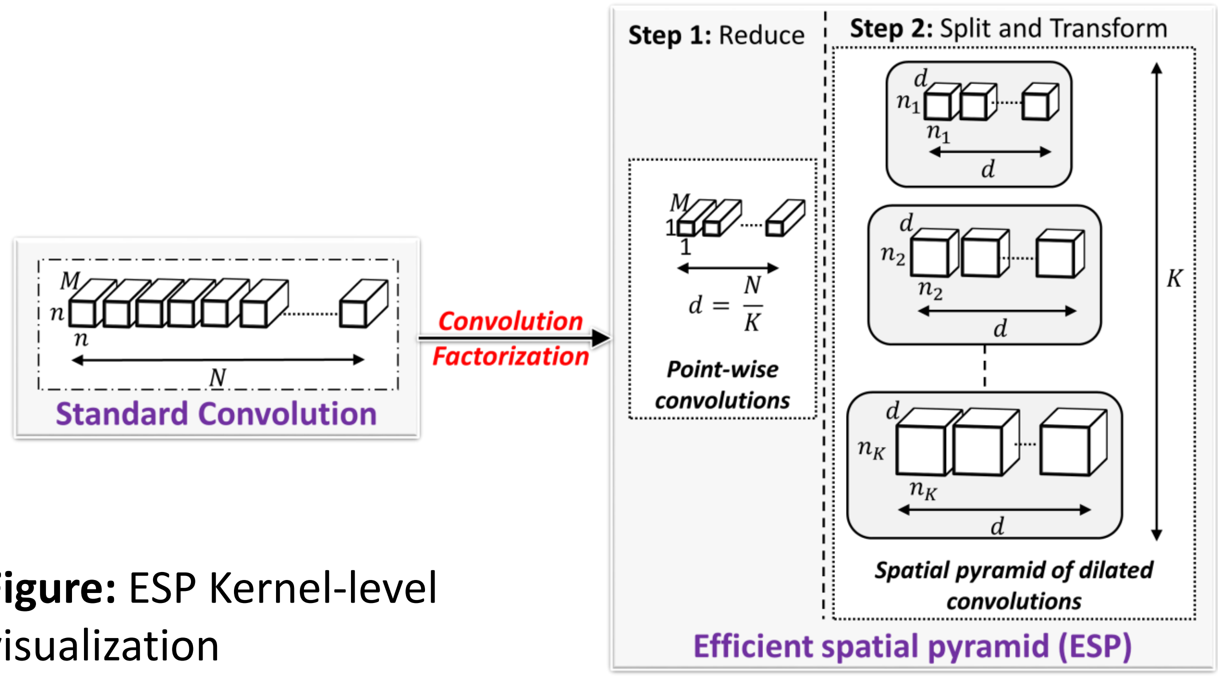


Figure: ESP Kernel-level visualization

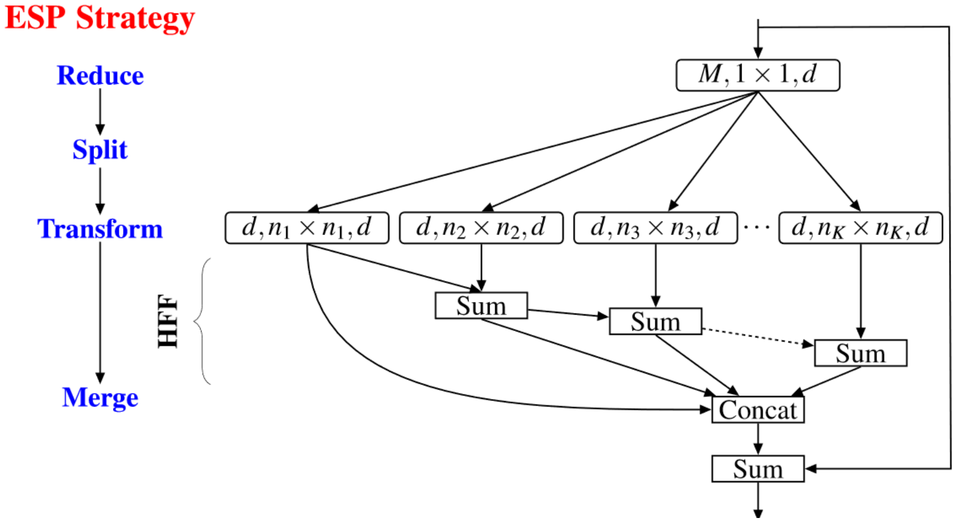


Figure: ESP block-level visualization

Dilated/Atrous Convolution

- Dilated convolutions are special form of standard convolution in which the effective receptive field is increased by inserting zeros (or holes) between each pixel in the convolutional kernel.

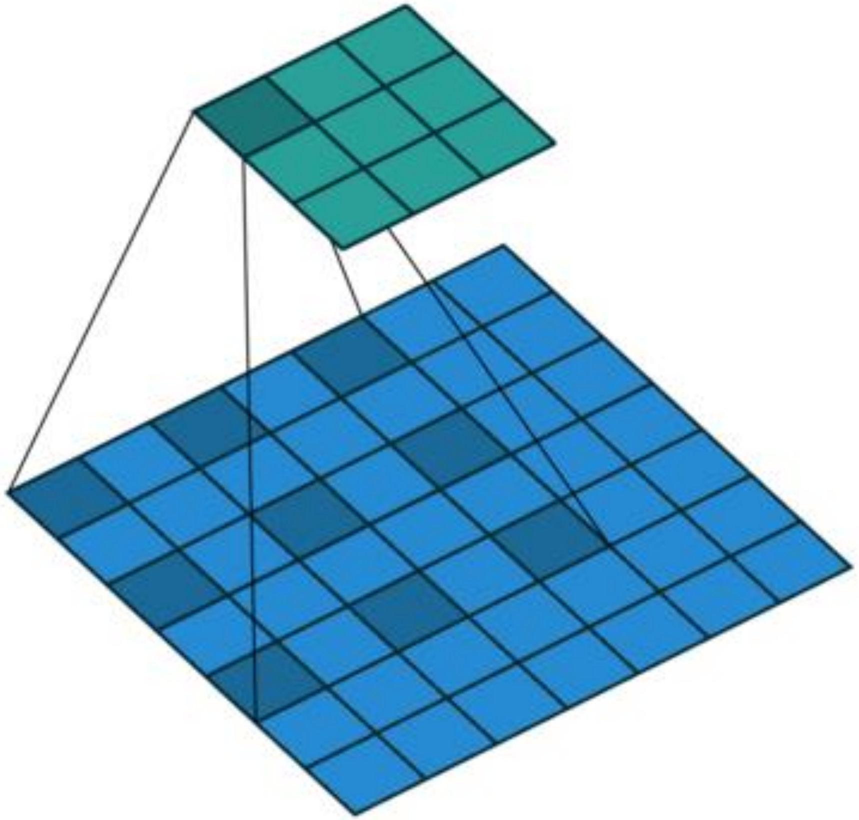


Figure: Dilated convoltuion

Source: http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Gridding problem with Dilated Convolutions

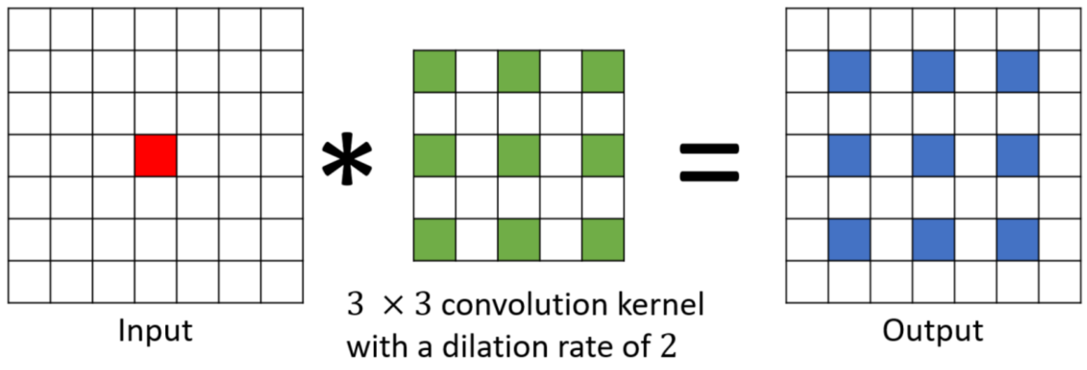
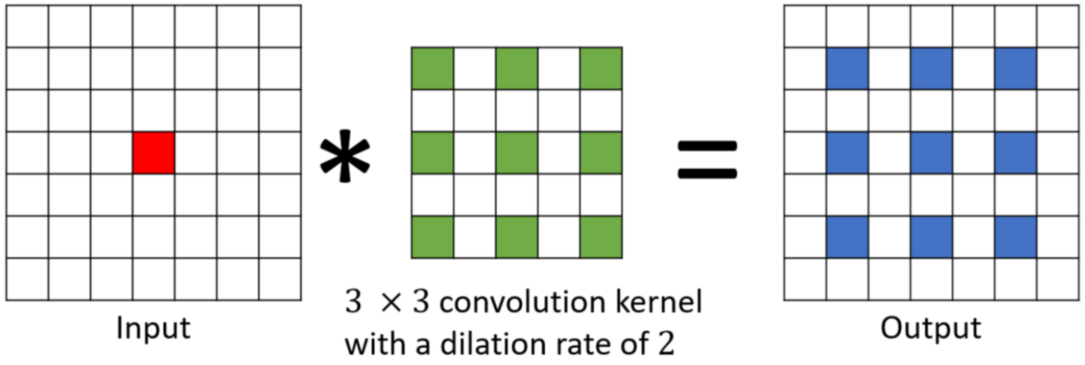


Figure: Gridding artifact in dilated convolution

Gridding problem with Dilated Convolutions



- Solution
  - Add convolution layers with lower dilation rate at the end of the network (see below links for more details)
  - Cons: Network parameter increases

Source:

- Yu, Fisher, Vladlen Koltun, and Thomas Funkhouser. "Dilated residual networks." *CVPR, 2017*.
- Wang, Panqu, et al. "Understanding convolution for semantic segmentation." *WACV,* 2018.

Hierarchical feature fusion for de-gridding

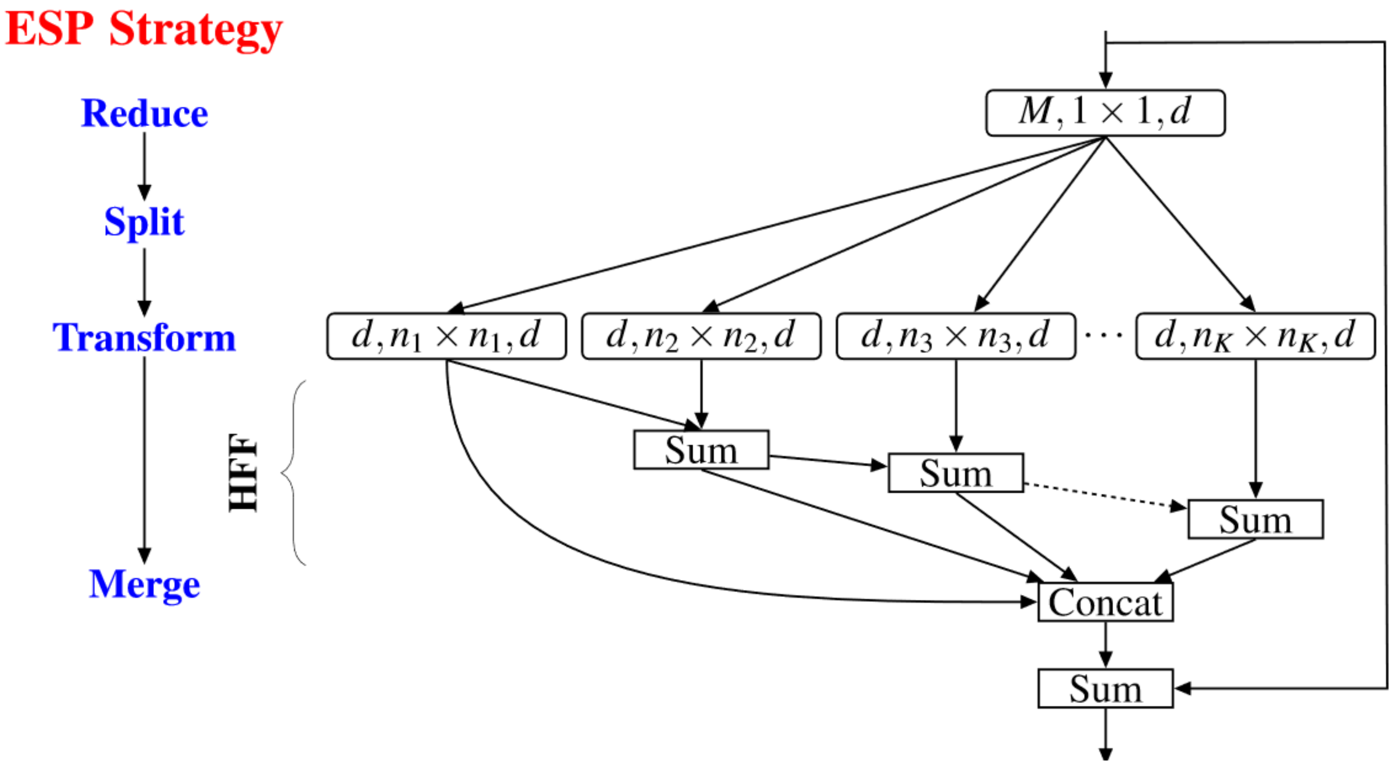


Figure: ESP Block with Hierarchical Feature Fusion (HFF)

Hierarchical feature fusion (HFF) for de-gridding

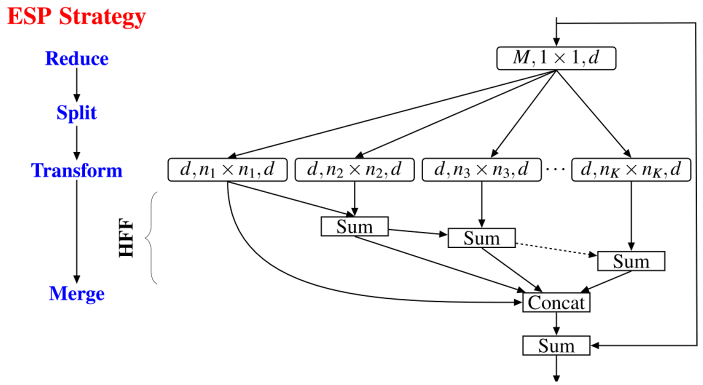


Figure: ESP Block with HFF

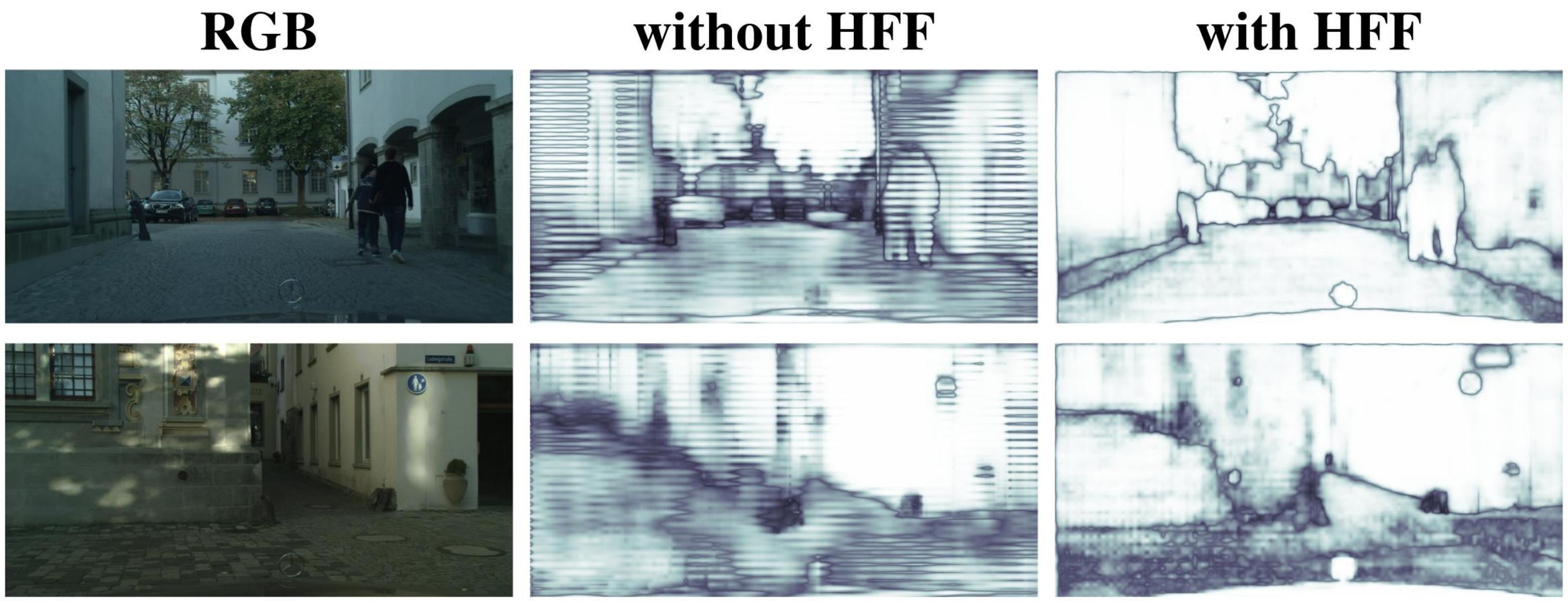


Figure: Feature map visualization with and without HFF

Input-reinforcement: An efficient way of improving the performance

- Information is lost due to filtering or convolution operations.
- Reinforce the input inside the network to learn better representations

	mIOU	Parameters
Without input reinforcement	0.40	0.186 M
With input reinforcement	0.42	0.187 M

* Results on the cityscape urban visual scene understanding dataset
* mIOU is mean intersection over union

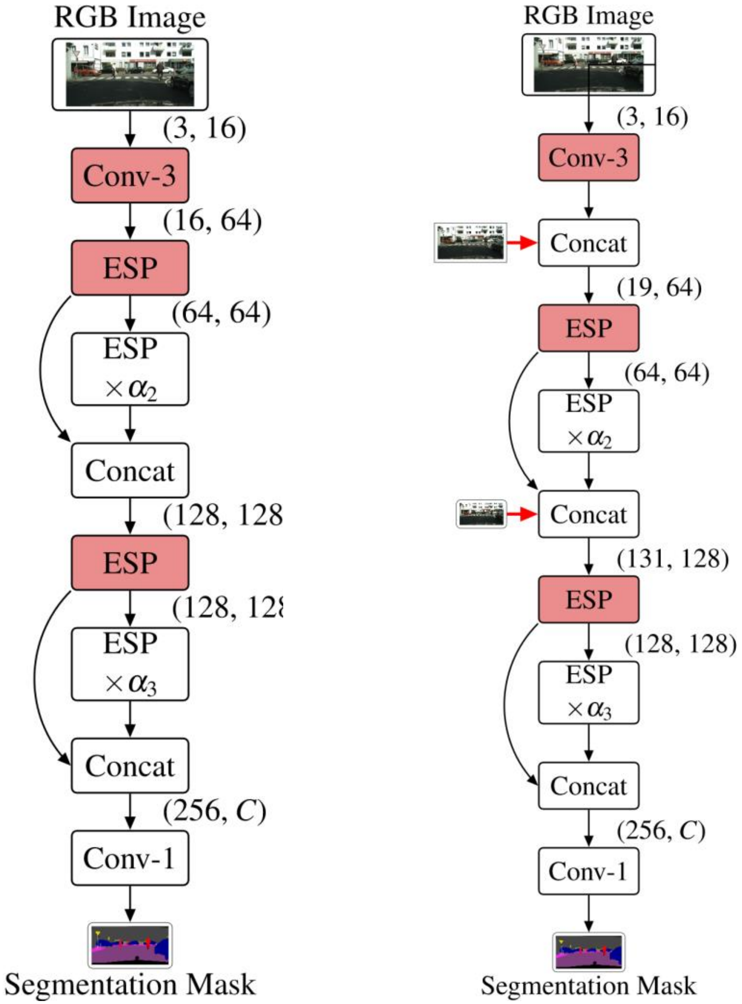


Figure: ESPNet without and with input reinforcement

ESPNet with a light-weight decoder

- Adding 20,000 more parameters improved the accuracy by 6%.

α_3	ESPNet-C (Fig. 4c) mIOU	# Params (in million)	Network size	ESPNet (Fig. 4d) mIOU	# Params (in million)	Network size
3	49.0	**0.187**	**0.75 MB**	56.3	**0.202**	**0.82 MB**
5	51.2	0.252	1.01 MB	57.9	0.267	1.07 MB
8	**53.3**	0.349	1.40 MB	**61.4**	0.364	1.46 MB

Figure: Comparison between ESPNet without and with light weight decoder on the Cityscape validation dataset

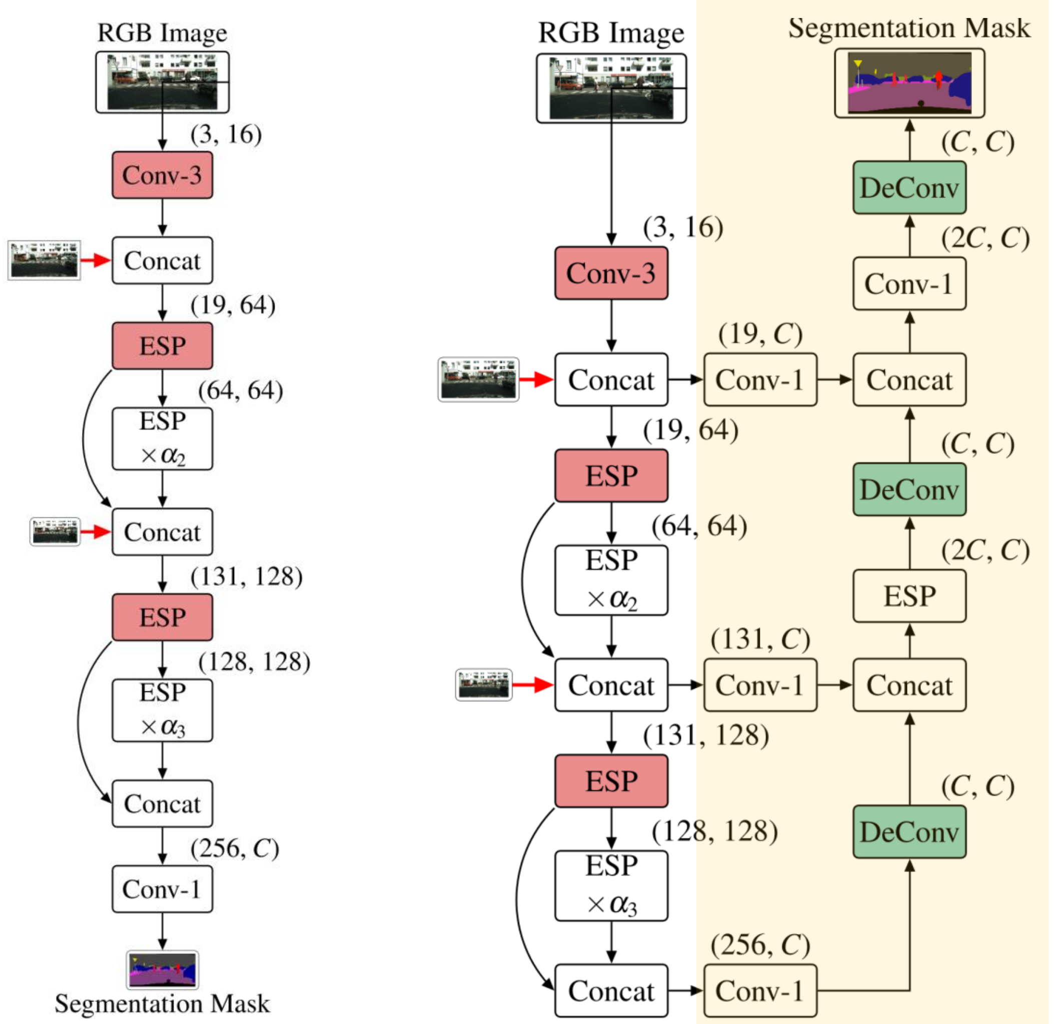
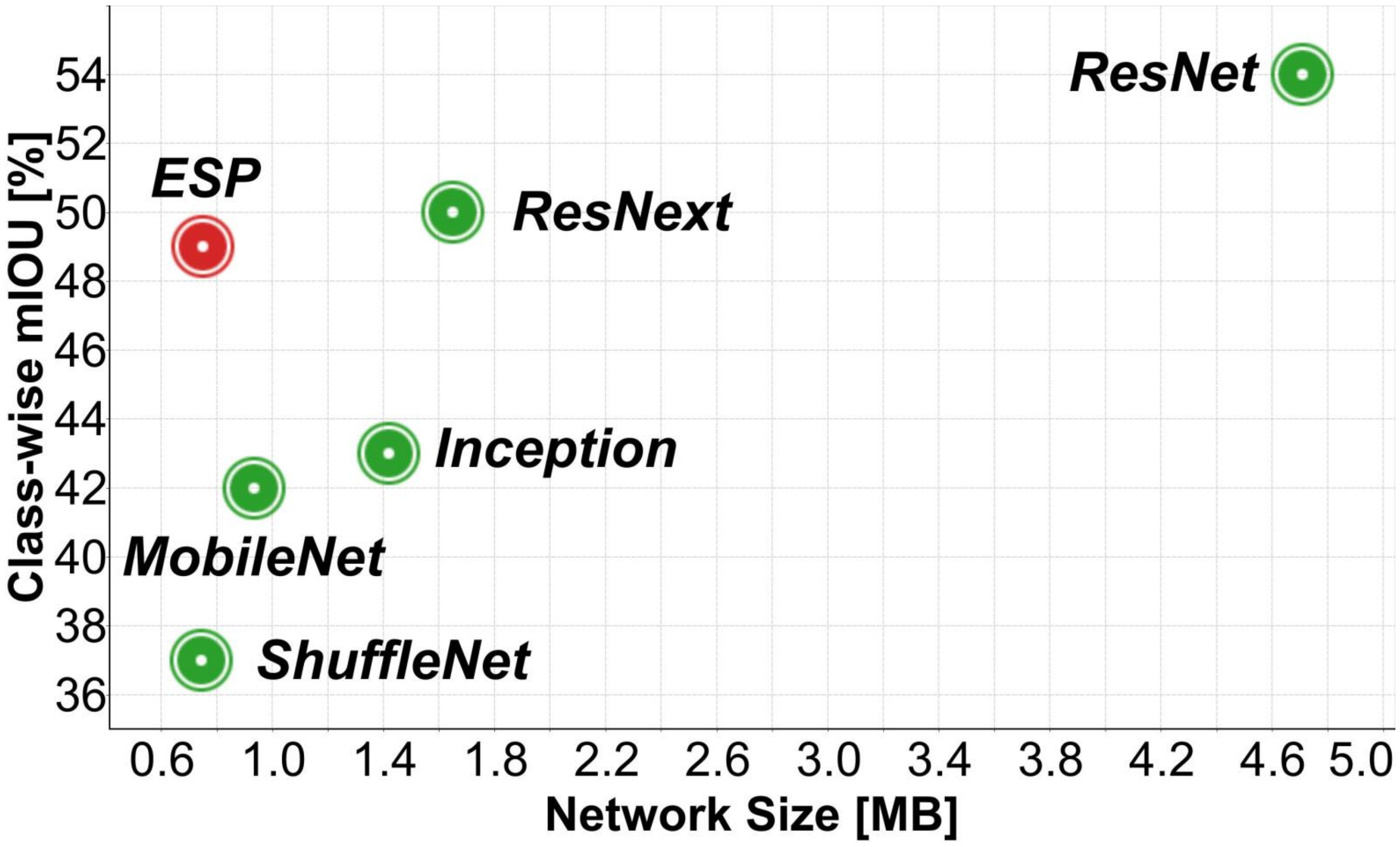


Figure: ESPNet without and with light weight decoder

Comparison with efficient networks

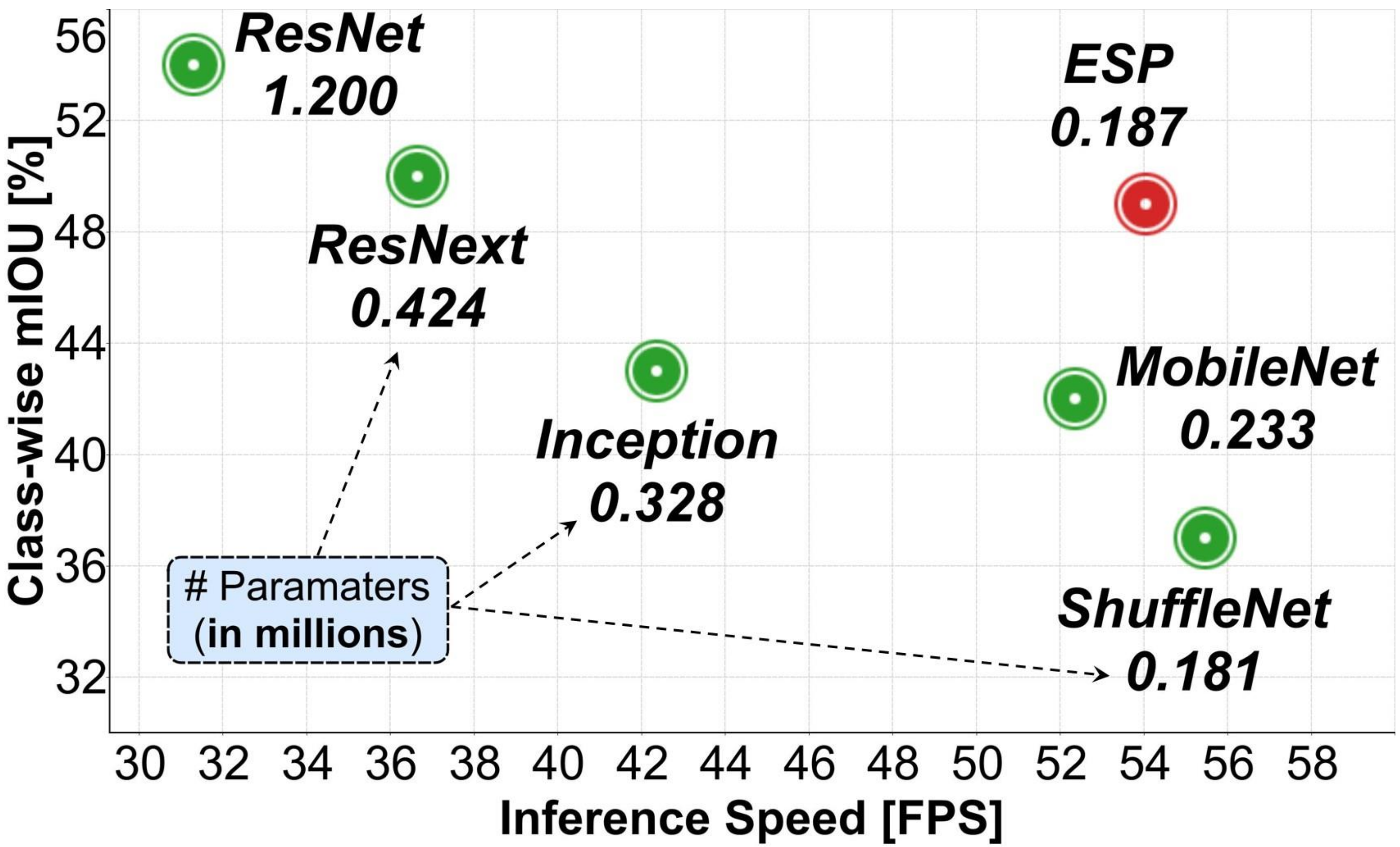
Network size vs Accuracy



Network size is the amount of space required to store the network parameters

Under similar constraints, ESPNet outperform MobileNet and ShuffleNet by about 6%.

Inference Speed vs Accuracy



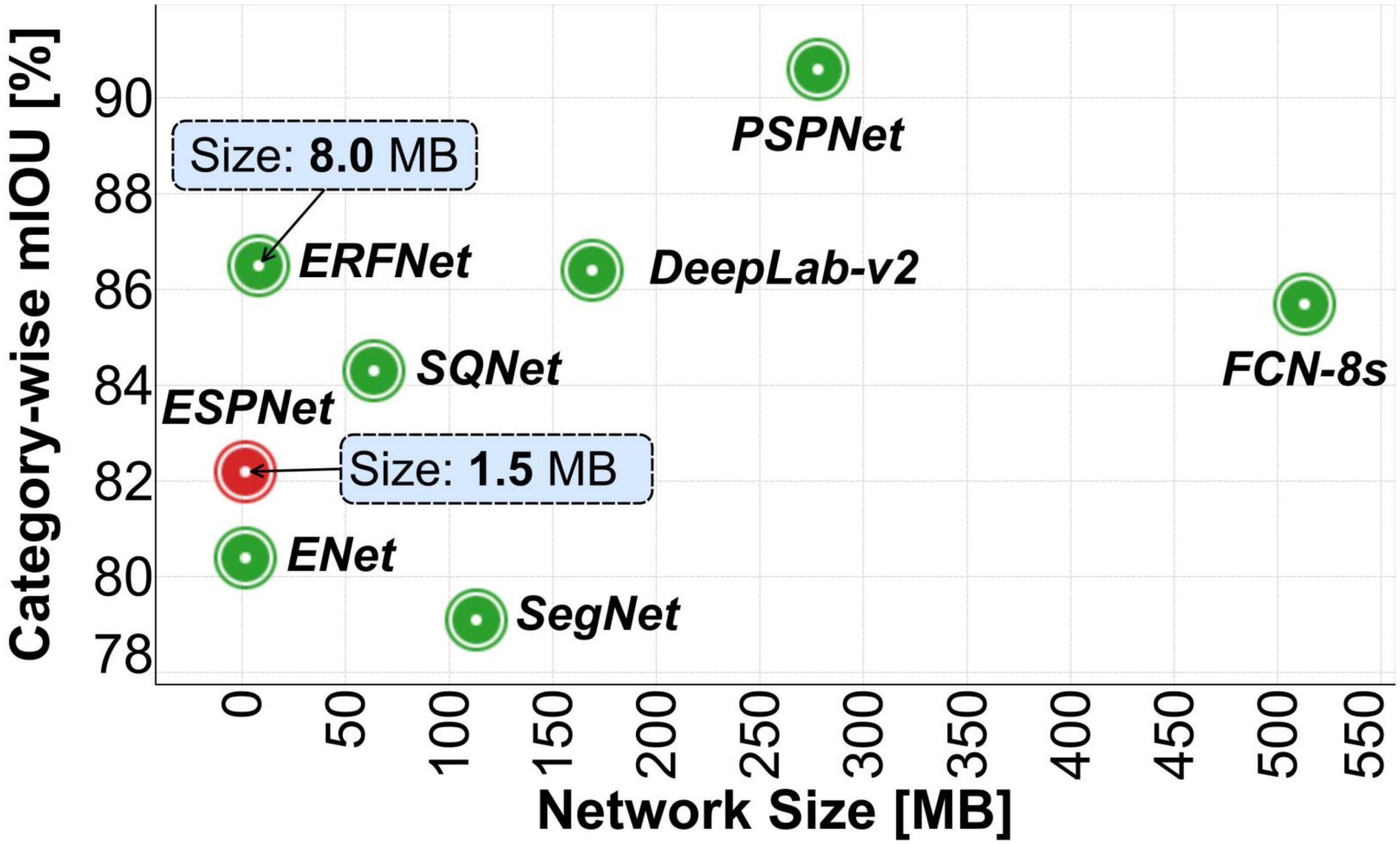
Inference speed is measured in terms of frames processed per second.

Device - Laptop
CUDA Cores – 640

Under similar constraints, ESPNet outperform MobileNet and ShuffleNet by about 6%.

Comparison with state-of-the-art networks

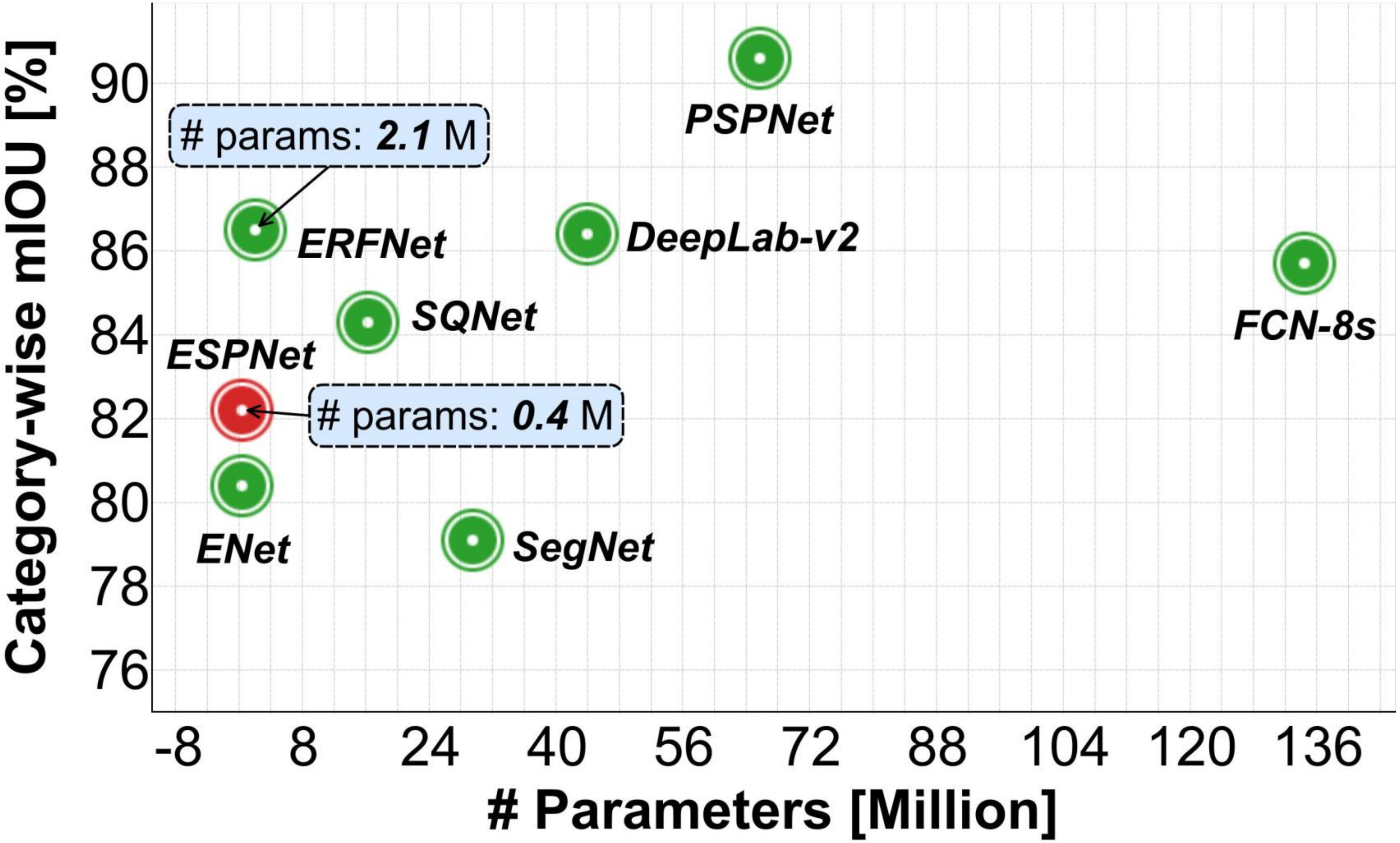
Accuracy vs Network size



Network size is the amount of space required to store the network parameters

ESPNet is **small in size** and well suited for edge devices.

Accuracy vs Network parameters



ESPNet learns **fewer parameters** while delivering competitive accuracy.

Power Consumption vs Inference Speed

ESPNet is **fast** and **consumes less power** while having a good segmentation accuracy.

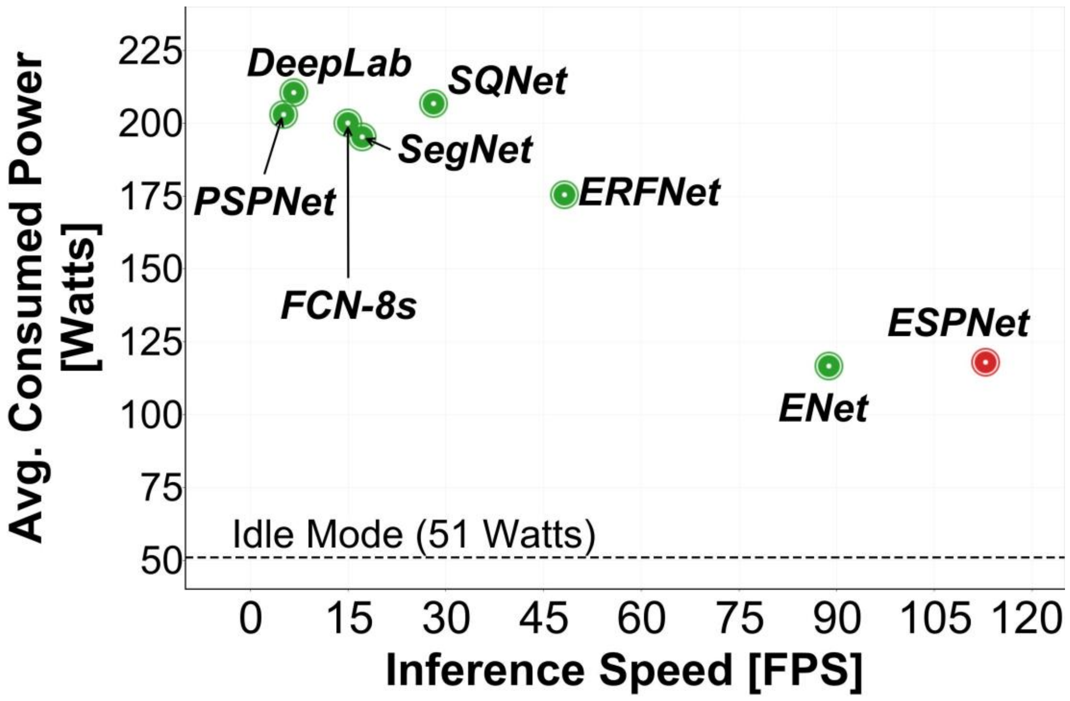


Figure: Standard GPU (NVIDIA-TitanX: 3,500+ CUDA Cores)

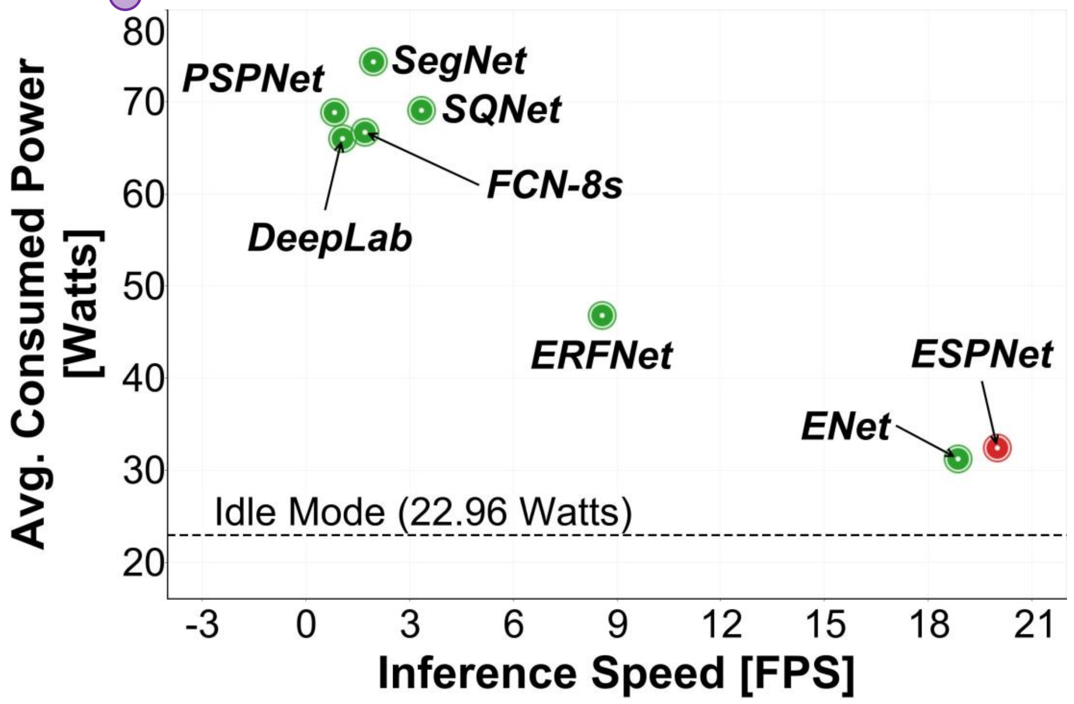


Figure: Mobile GPU (NVIDIA-Titan 960M: 640 CUDA Cores)

Inference Speed and Power Consumption on Embedded Device (NVIDIA TX2)

ESPNet processes a RGB image of size **1024x512** at a frame rate of **9 FPS**.

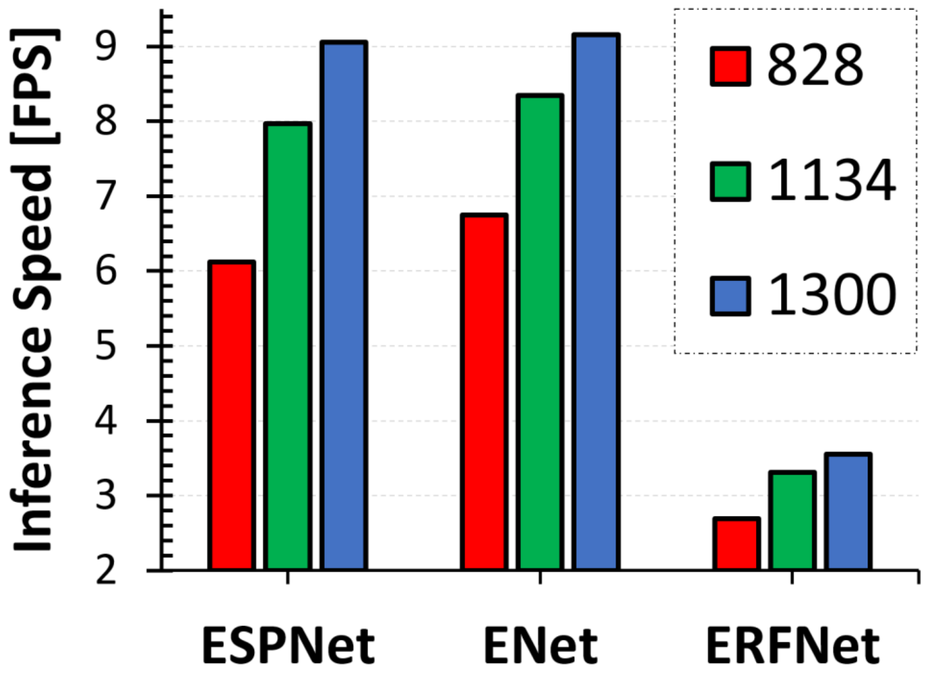


Figure: Inference speed at different GPU frequencies

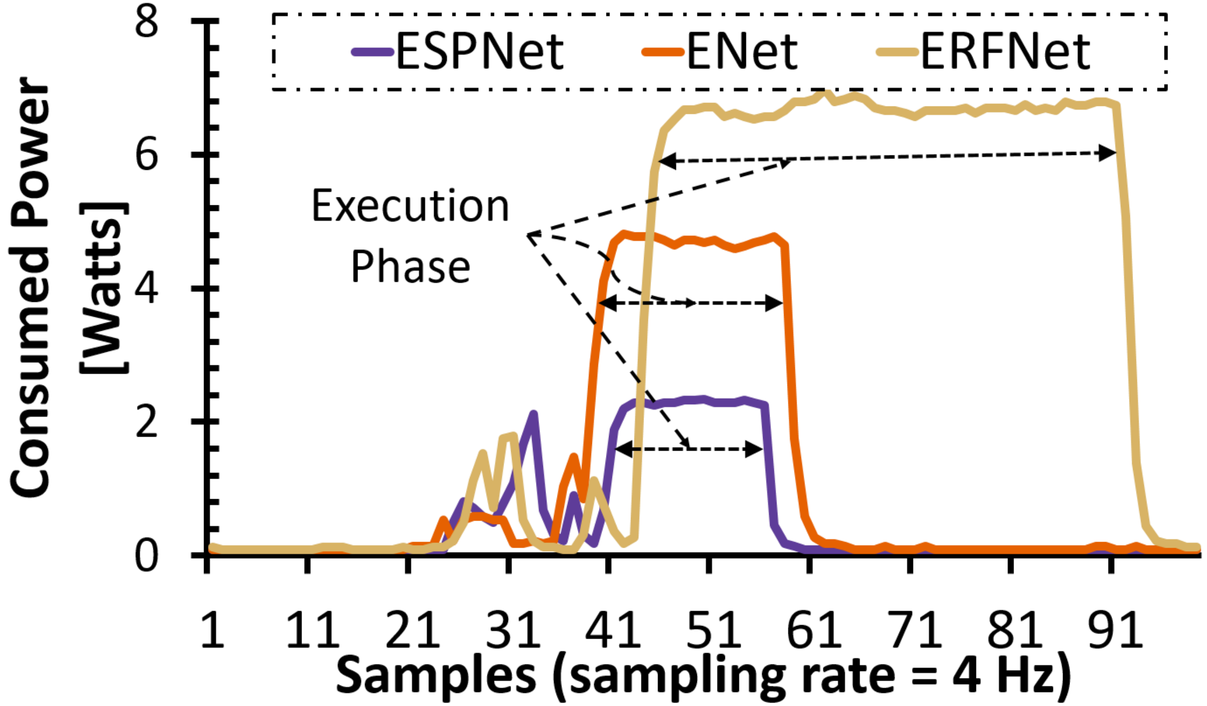
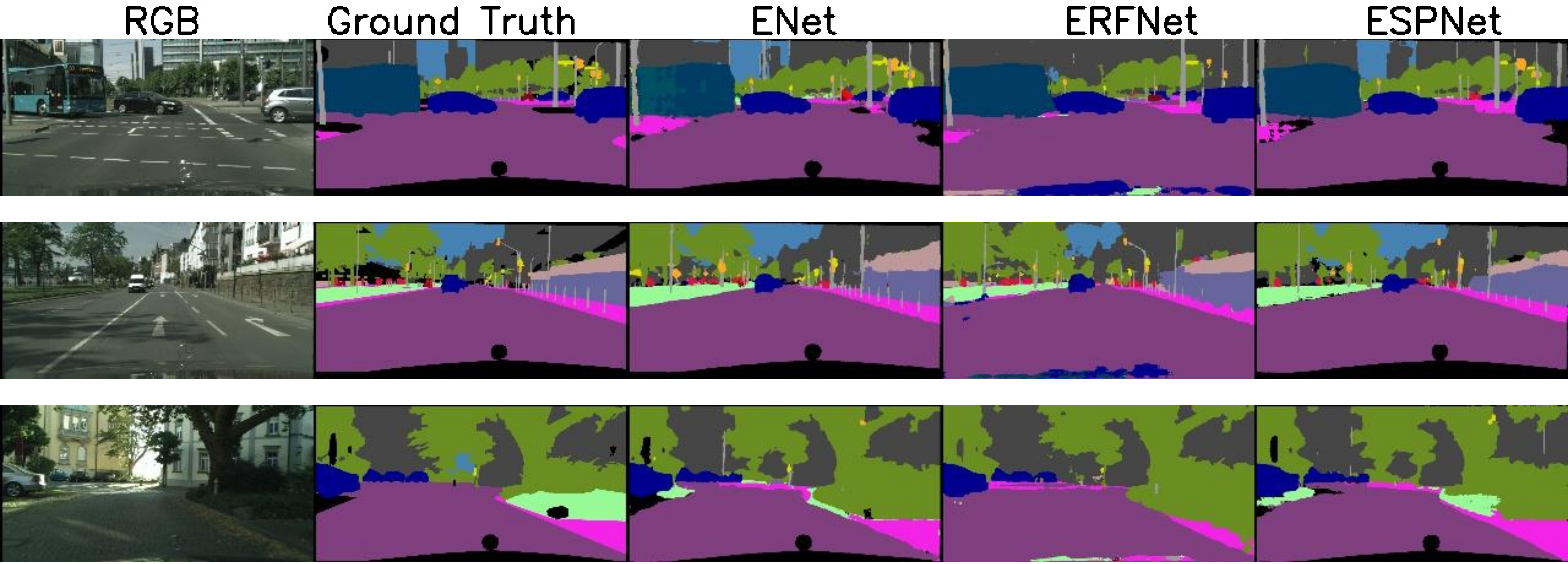
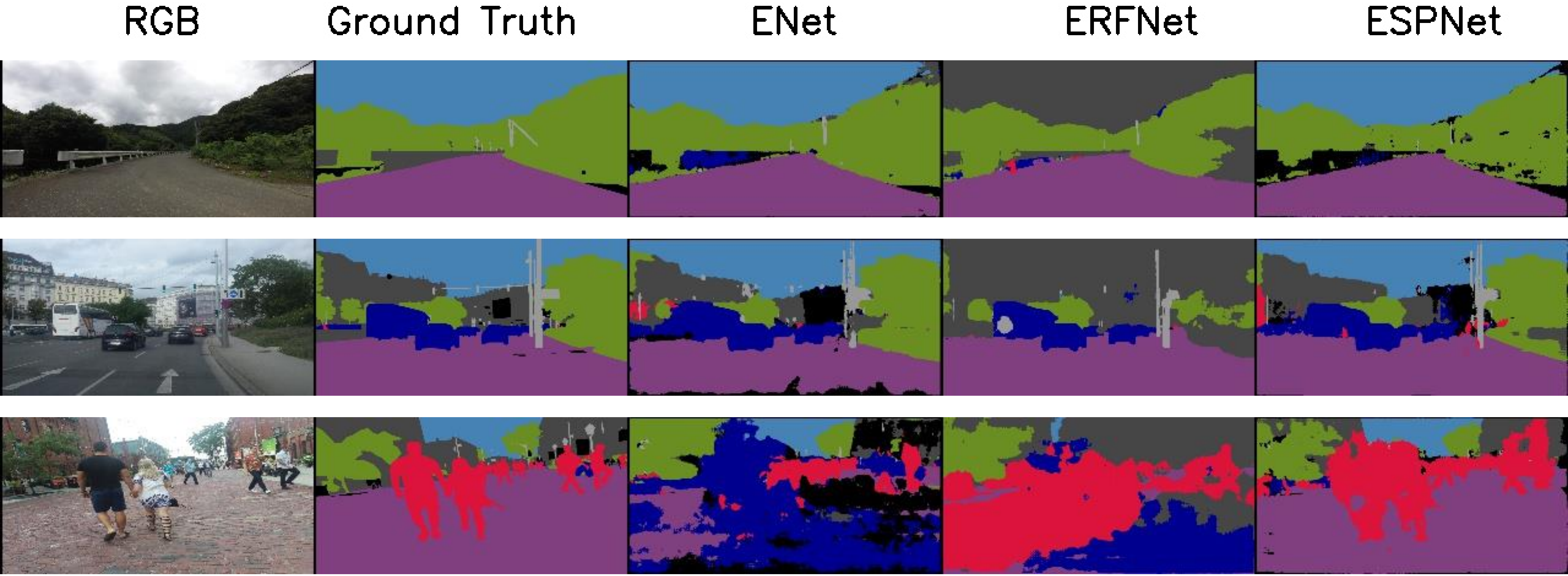


Figure: Power consumption vs samples

Visual Results on the Cityscape validation set



Visual Results on unseen set



Results on Breast Biopsy Whole Slide Image Dataset

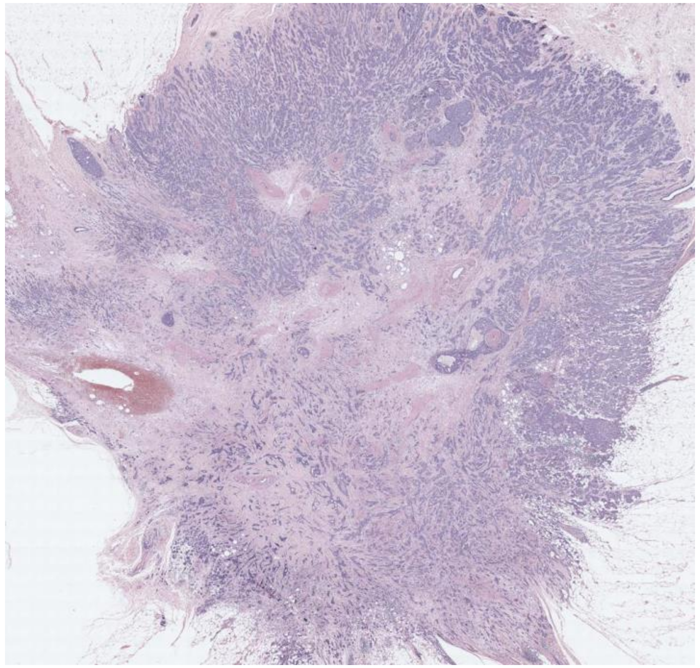
Results on Breast Biopsy dataset

- The average size of breast biopsy images is **10,000 x 12,000** pixels
- 58 images marked by expert pathologists into 8 different tissue categories were split into equal training and validation sets.
- ESPNet delivered the same segmentation performance while learning 9.46x lesser parameters than state-of-the-art networks.

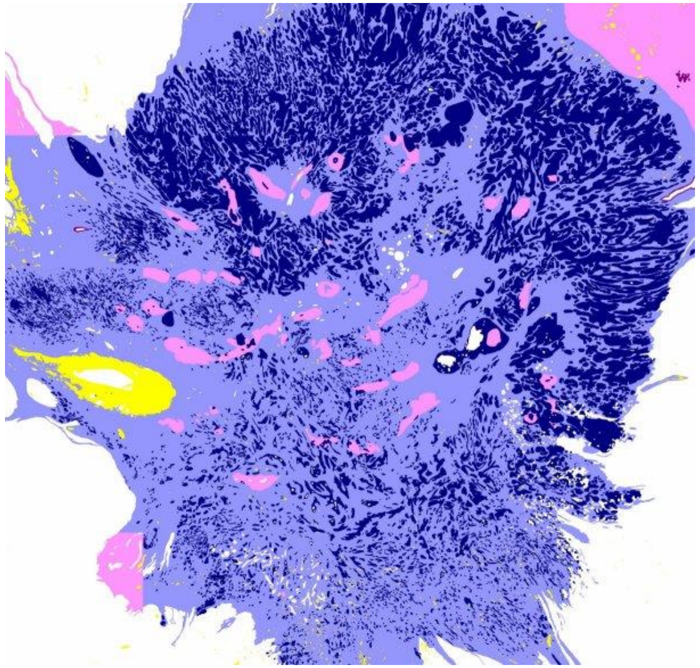
Model	**Module**	**mIOU**	**# Params**
ESPNet (Ours)*	ESP	44.03	**2.75**
SegNet [39]	VGG	37.6	12.80
Mehta *et al.* [36]	ResNet	**44.20**	26.03

Visual results

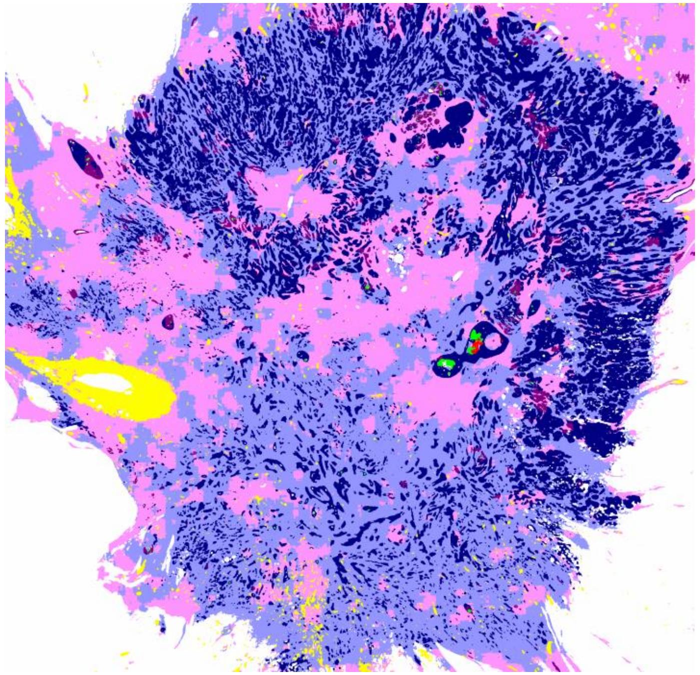
background | benign epithelium | normal stroma | secretion
malignant epithelium | desmoplastic stroma | blood | necrosis



RGB Image



Ground Truth

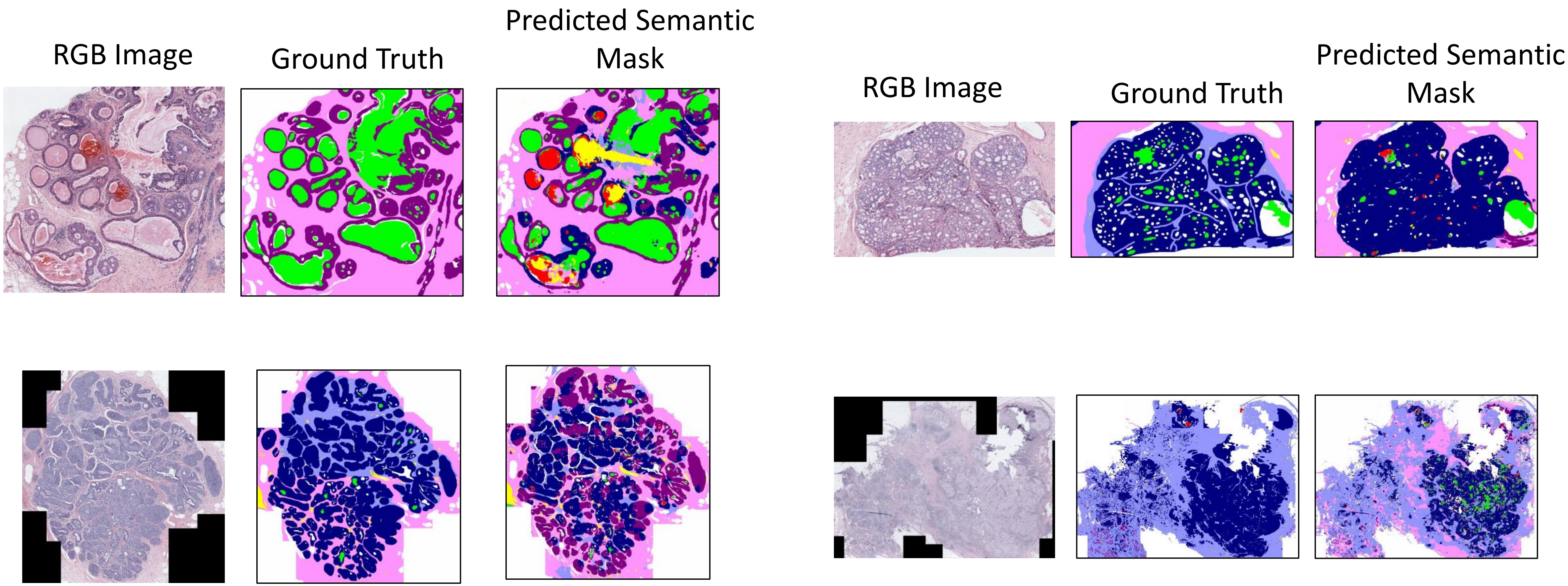


Predicted Semantic Mask

Visual results

background, benign epithelium, normal stroma, secretion

malignant epithelium, desmoplastic stroma, blood, necrosis



References

[1] (**PSPNet**) Zhao, Hengshuang, et al. "Pyramid scene parsing network." IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017.
[2] (**FCN-8s**) Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[3] (**SegNet**) Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE transactions on pattern analysis and machine intelligence 39.12 (2017): 2481-2495.
[4] (**DeepLab**) Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.
[5] (**SQNet**) Treml, Michael, et al. "Speeding up semantic segmentation for autonomous driving." MLITS, NIPS Workshop. 2016.
[6] (**ERFNet**) Romera, Eduardo, et al. "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation." IEEE Transactions on Intelligent Transportation Systems 19.1 (2018): 263-272.

References

[7] (**ENet**) Paszke, Adam, et al. "Enet: A deep neural network architecture for real-time semantic segmentation." arXiv preprint arXiv:1606.02147 (2016).
[8] (**MobileNet**) Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
[9] (**ShuffleNet**) Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." arXiv preprint arXiv:1707.01083 (2017).
[10] (**ResNext**) Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017.
[11] (**ResNet**) He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[12] (**Inception**) Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

Thank You