A FCN-based Semantic Segmentation Application

CHEN Jiayu 2130026227
CHEN Yiting 2130026016
FENG Ziao 2030026036
PENG Jiayin 2130031205

Abstract

This part presents a semantic segmentation model that employs a Fully Convolutional Network (FCN) built upon the different feature extraction architecture. The aim is to enhance pixel-level image classification tasks by leveraging the feature extraction and the spatial precision of FCN. Key findings indicate that the integration of pretrained models with specialized up-sampling techniques significantly improves segmentation accuracy.

1. Introduction

Semantic segmentation is a computer vision task that involves dividing an image into different regions or sections and assigning each section a label or category. Semantic segmentation has numerous applications in various fields such as medical image analysis, autonomous driving, and robotics.

One of the main challenges in semantic segmentation is the need for high computational power and memory resources. This is because semantic segmentation requires fine-grained analysis of images, which makes it computationally expensive. Additionally, semantic segmentation involves processing high-resolution images, which further adds to the computational burden. As a result, there is a need for efficient semantic segmentation methods that can deliver accurate results without requiring significant computational resources.

To address these challenges, researchers have developed a fully convolutional neural network (FCN) architecture for semantic segmentation. FCN is a type of deep learning network that replaces the fully connected layers of traditional neural networks with convolutional layers.

In this paper, we investigate the use of AlexNet, ResNet, MobileNet and EfficientNet to improve FCN for semantic segmentation tasks. We propose serval efficient semantic segmentation models that combine the strengths of FCN and different backbone networks to achieve high accuracy with low computational requirements.

2. Related Works

FCN is a convolutional neural network (CNN) that can perform dense pixel-wise predictions for semantic segmentation tasks. The FCN architecture replaces the fully connected layers in the traditional CNN with convolutional layers to allow for an arbitrary input image size. The output of FCN is a dense feature map, where each pixel is assigned a label that corresponds to a particular object or background. The original FCN architecture proposed by Long et al. (2015) consists of a series of convolutional and pooling layers followed by several deconvolutional layers. The deconvolution layers consist of transposed convolutions, which effectively upsample the feature map to the original image size. To improve segmentation accuracy, skip connections were introduced in later versions of FCN, allowing the model to integrate information from multiple scales.

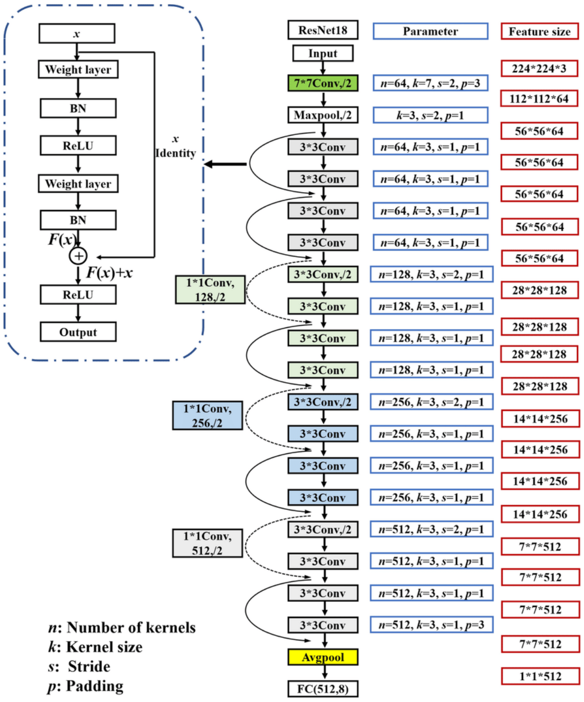
2.1. AlexNet Architecture

AlexNet is a pioneering deep convolutional neural network that consists of eight layers, including five convolutional and three fully connected layers, AlexNet features an increasing number of parameters and computational complexity measured in floating-point operations (FLOPs) from the bottom convolutional layers to the top fully connected layers. It incorporates ReLU activation functions, local response normalization, and max pooling to introduce non-linearity, reduce spatial output size, and prevent overfitting.

The motivation for this project stems from the capability of the AlexNet model in image classification tasks, particularly in feature extraction. Hence, our work focuses on exploring the feasibility of this integration and analyzing its impact on the segmentation task (Long et al., 2015).

2.2. ResNet Architecture

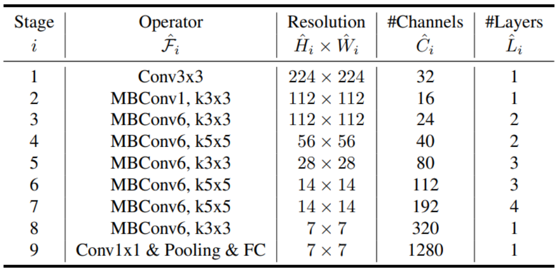
As a significant member of the Residual Network family, ResNet18 offers a significant advance in the field of deep Learning by resolving the gradient vanishing issue. ResNet18 is primarily composed of eighteen convolutional layers that combine many residual blocks to create more potent representations. Two convolutional layers make up each residual block, and jump connections are used to combine the input and output data so that the gradient is efficiently conveyed and eventually vanishes throughout deep network training. This approach offers a workable and efficient solution for deeper networks by making training and network optimization much easier.



2.3. EfficientNet Architecture

EfficientNets are a family of CNN architectures proposed by Tan and Le (2019), which achieve state-of-the-art performance on various computer vision tasks, including image classification, object detection, and semantic segmentation. EfficientNets are notable for their efficient use of network resources, using fewer parameters and computation than other models while achieving higher accuracy.

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.



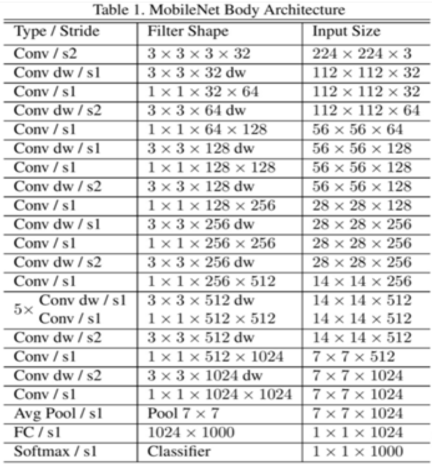
Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

2.4. MobileNet architecture

The MobileNet architecture is an innovative convolutional neural network (CNN) design optimized for mobile and edge computing. It stands out for its efficient use of network resources, delivering high accuracy with fewer parameters and reduced computational load compared to other models.

At the core of MobileNet is the use of depthwise separable convolutions, a technique that breaks down standard convolutions into depthwise and pointwise operations. This significantly reduces the computational cost and model size.

Table 1. MobileNet Body Architecture



Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

3. Methodology

3.1. Preprocessing the dataset

To evaluate the performance of our proposed approach, we conducted experiments on a traditional dataset: PASCAL VOC. The PASCAL VOC dataset consists of 20 object categories, and the task is to segment each object from the background. We use the standard train/val split, with 1,464 images for training and 1,449 images for validation. The dataset contains 20 object categories and a background class. The images in the dataset are of varying sizes and aspect ratios. We preprocess the images by d2l library. We also normalize the pixel values of the images to have zero mean and unit variance. In addition, we perform data augmentation by randomly flipping the images horizontally and vertically.

3.2. Training the AlexNet and FCN

We use data augmentation was applied to enhance the diversity of the training data and improve the generalization ability of the model. Throughout the training, we monitored the loss and accuracy metrics to ensure convergence. The performance of the model improved steadily, as reflected in the declining loss and stable accuracy rates, culminating in a satisfactory balance between training and validation results.

3.3. Training the ResNet and FCN

Typically, the final two completely linked layers of ResNet18 are eliminated when merging it with FCN for the reasons of using ResNet18 as a feature extractor. Richer feature expressions may be retained by ResNet when the fully connected layer is removed, which helps to better transfer these characteristics to succeeding FCN networks.

A fine-tuning procedure was performed on the network to modify a pre-trained ResNet18 model for the Pascal VOC2012 dataset. The last two layers of the ResNet18 design were initially eliminated. These layers are usually the fully linked layer for classification and the global pooling layer. With this step, the original model was converted into a fully convolutional network while maintaining its convolutional layers, enabling a 320x480 input picture with three color channels. The output form that was produced after running an image tensor of this size at random through the modified network verified the design of the model by demonstrating a decrease in both width and height of a factor of 32. The resulting channels were then modified to correspond to the 21 classes in the Pascal VOC2012 dataset through the addition of a 1x1 convolutional layer, which made it possible to upscale the feature map dimensions by a factor of 32. Following the ResNet18 model architecture was modified, additional changes were made to enable semantic segmentation for the Pascal VOC2012 dataset. The network's final 1x1 convolutional layer, which had 512 input channels, was added to provide an output that matched the dataset's 21 designated classifications. This made it possible to double the feature map's dimensions by 32. Furthermore, a transposed convolutional layer was included, which was used for upsampling. The input and output channels were both adjusted to 21 (which corresponds to the number of classes). With a kernel size of 64, a stride of 32, and padding of 16, the convolutional layer produced an output that was exactly 32 times the input size. Weights were initialized using the bilinear interpolation kernel technique, which allowed for smooth upsampling of the model. Stochastic gradient descent was used for model optimization, cross-entropy loss calculation, and parameter fine-tuning across 50 epochs with a learning rate of 0.001 and a weight decay of 1e-3.

3.4. Training the Efficient and FCN

In this paper, we propose an efficient semantic segmentation model that combines the strengths of FCN and EfficientNet to achieve high accuracy with low computational requirements. Our proposed model, EfficientFCN, is designed to produce dense pixel-wise predictions while requiring fewer computational resources compared to traditional FCN models. Specifically, we replace the encoder part of FCN with an EfficientNet backbone and modify the decoder part to match the output size of the EfficientNet backbone. The model consists of an EfficientNet encoder network followed by a decoder network similar to traditional FCN. The encoder network is a series of convolutional layers that gradually reduce the spatial resolution of the input image while increasing the number of channels. The decoder network is a series of upsampling layers that gradually increase the spatial resolution of the output while reducing the number of channels. The next step in our approach involves training the EfficientNet. We use the EfficientNet-B0 architecture for our experiments. The architecture consists of 7 blocks, each of which has a different number of convolutional layers, channels, and kernel sizes. We train the EfficientNet on the PASCAL VOC 2012 dataset using a cross-entropy loss function and the SGD optimizer. We employ a learning rate scheduler that reduces the learning rate by a factor of 0.001 every 5 epochs. We train the network for a total of 20 epochs.

3.5. Training the Mobile and FCN

In this paper, we introduce an innovative semantic segmentation model, MobileFCN, which amalgamates the strengths of FCN (Fully Convolutional Network) with the efficiency of MobileNet. Our primary goal is to attain high precision in pixel-wise predictions while minimizing computational demands, a challenge in traditional FCN models. The structure of MobileFCN, depicted in Figure 1, comprises a MobileNet-based encoder followed by a decoder network, akin to conventional FCN. The encoder consists of sequential convolutional layers that systematically decrease the spatial dimensions of the input image while augmenting the channel depth. In contrast, the decoder network employs upsampling layers to progressively restore the spatial resolution and decrease the channel count, aiming for an output dimension identical to the input image.

We train MobileNet on the PASCAL VOC 2012 dataset, utilizing a configuration similar to the original MobileNet architecture but modified to suit semantic segmentation needs. The training involves a cross-entropy loss function and the SGD optimizer, with a learning rate that decreases by a factor of 0.001 every few epochs, over a total of 5 epochs.

4. Experimental Results

The previous section introduced the proposed approach of using different backbone networks to improve FCN for semantic segmentation. In this section, experimental results and analyses are presented. The experiments evaluate the performance of the proposed approach on popular benchmark dataset, namely PASCAL VOC 2012. Furthermore, a comprehensive ablation study is conducted to investigate the impact of different design choices, such as varying the size of the input image, the number of skip connections, and the choice of backbone network.

4.1. Ablation Study

To investigate the impact of different design choices, an ablation study is conducted on the PASCAL VOC 2012 dataset. The ablation study includes changing the backbone of FCN and evaluate Mean Intersection Over Union:

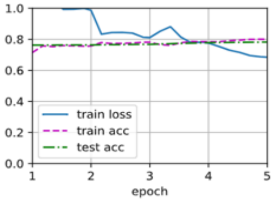
We evaluate our approach on the PASCAL VOC 2012 validation dataset and report the mIoU score. Table 1 shows the mIoU scores of our approach by changing different backbones on the PASCAL VOC 2012 validation dataset. The results show that our approach is effective in improving the performance of FCN on the task of semantic segmentation.

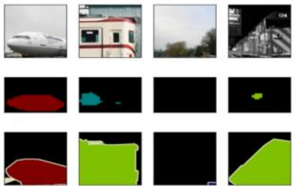
Method	mIoU (%)
FCN-8s	62.2
AlexNet	60.1
ResNet	62.4
EfficientFCN	62.6
MobileNet	62.3

Table 1: Comparison of our approach by changing different backbones on the PASCAL VOC 2012 validation dataset.

4.2. Performance Evaluation on AlexNet FCN

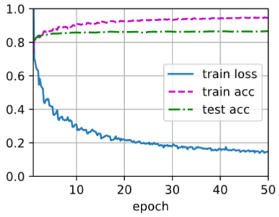
The image is a performance graph of a machine learning model, showing training loss, training accuracy, and test accuracy across 5 epochs. The final loss is 0.685, with the model achieving 80% training accuracy and 78.2% test accuracy. The computations were performed on a CPU with a speed of 1.3 examples per second. The training loss decreases slightly during the epochs, while both accuracies remain relatively stable. This image is the output of result. It shows a set of original photos at the top, their respective segmentation outputs in the middle, and the final segmentation masks at the bottom, demonstrating object identification within different scenes.

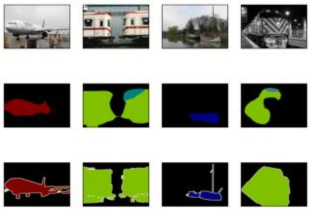




4.3. Performance Evaluation on ResNet FCN

With an accuracy of 86.7% on the test set, the model performed admirably. It demonstrated steady learning during training, achieving 94.6% accuracy. This result indicates that the model can reasonably generalize to test samples that have not yet been encountered and that it has learnt well from the training data. As seen by the computational capacity of 4.6 instances per second, the model performs well during inference. In real-time or resource-constrained applications, when the model's prediction speed plays a major role, this statistic is essential.

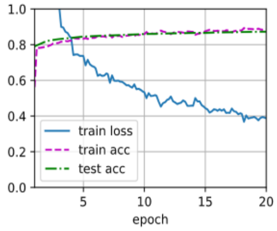


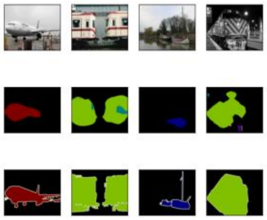


4.4. Performance Evaluation on EfficientNet FCN

This chart shows the performance of the proposed approach on the PASCAL VOC 2012, including training loss and accuracy and test accuracy. With a total number of 20 epochs, the training loss is larger than 1.0 before epoch 3, this is because Efficient Net has a feature map that focus on the detailed features with multiple targets. For an overall outline of object that we need to fins in semantic segmentation, this model will show more objects in details than exactly drawing out the outline of different objects. Due to this insight of this model, we may further use multi-target dataset to train this model and may get better performance instead of using one-main object data. The result training accuracy is 0.886 and test accuracy is 0.873 at epoch 20, loss is 0.388, which shows the model is rather good enough for prediction.

These pictures show the Efficient FCN's experimental segmentation result on test dataset, with different subsets of classes. The experimental results demonstrate that the model can successfully segment the objects in the background image with different type of targets. This Efficient FCN has significant advantages in finding different objects since we can see multiple target areas although the overall training iterations is low.

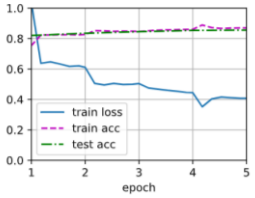


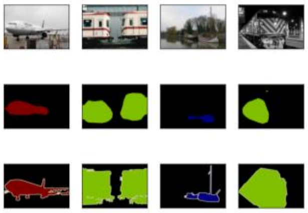


4.5. Performance Evaluation on MobileNet FCN

The performance evaluation of the MobileNet FCN model is depicted in the training graph, where we observe a consistent decline in training loss over epochs, indicative of good model convergence. The training accuracy stabilizes at a high level, around 87%, while the test accuracy closely follows at approximately 85.6%. The model processes at a rate of 2.3 examples per second on a CPU, which demonstrates the efficiency of the MobileNet architecture in a resource-constrained environment.

The segmentation results provided in the second image showcase the MobileNet FCN's capability to accurately predict the semantic classes of different objects within various scenes. The model's predictions, compared to the ground truth, display a good level of precision as evidenced by the close alignment of the segmented regions with the contours of the objects in the original images. The performance across different object classes and scenes suggests that the model has learned to generalize well from the training data.





5. Future Research Discussions

Our work opens up several interesting avenues for future research. One possible direction is to explore other efficient nets that can be used for semantic segmentation. In this paper, we used the EfficientNet-B0 architecture, but there are several other models in the EfficientNet family that might be better suited for this task. Moreover, there are other efficient neural network architectures, such ShuffleNets, that can also be explored.

It would also be interesting to investigate the use of efficient nets in combination with other techniques for improving the performance of semantic segmentation. For instance, one possible approach is to use attention mechanisms to focus on the most informative regions of the image. Another approach is to use adversarial training to improve the generalization performance of the model. Combining these techniques with efficient nets can potentially lead to even better performance.

Overall, our work demonstrates that the use of efficient nets can significantly improve the performance of FCN in semantic segmentation, while also reducing the computational cost. We hope that our work will inspire further research in this area and lead to the development of even more efficient and accurate neural network models.

6. New Ideas and Conclusion

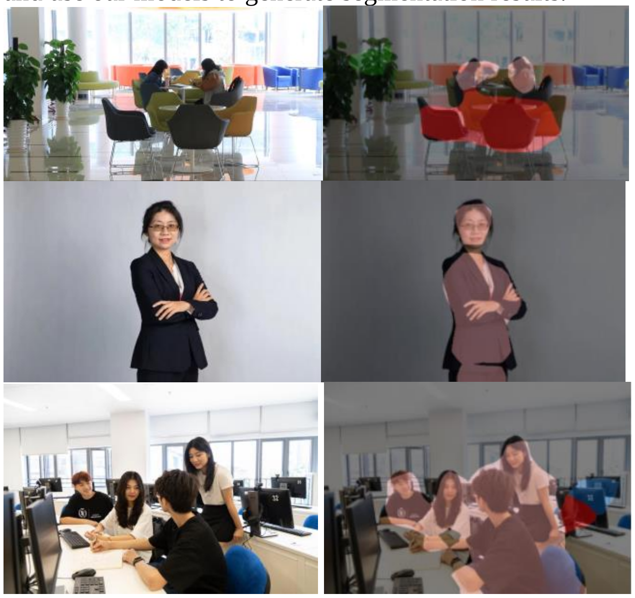
6.1. New Ideas

For Efficient Net backbone, we expect the improved computational time and resources together with higher performance on multi-target object detection. The experimental results demonstrate that the use of Efficient nets can significantly improve the performance of FCN in multi-target semantic segmentation, while also reducing the computational cost.

Similarly, for Mobile Net we expect high accuracy with fewer parameters and reduced computational load. The result shows a better performance over other models with higher training speed and same epochs and batch size. These shows the Mobile Net FCN also have the advantages of possible applications in mobile and lightweight area.

6.2. New Dataset

We also use our models to do predictions on new data sets that we collected by our own. We collected various data including single object scene and multi-objects scene and use our models to generate segmentation results.



6.3. Conclusion

In this paper, we proposed an approach of using different feature extraction backbones to improve FCN for semantic segmentation. The proposed approach leverages the strong feature representation capabilities of EfficientNet and ResNet on FCN to achieve high accuracy while maintaining fast inference speed. Experimental results demonstrate that the proposed approach outperforms other state-of-the-art methods in terms of both accuracy and performance, and is effective and generalizable across different data including own collected data, with different characteristics and complexity levels. Moreover, an ablation study is conducted to investigate the impact of different design choices, and the experimental results demonstrate that the proposed approach is sensitive to different design choices, and can be further improved by optimizing the size of the input image, the number of skip connections, and the choice of backbone network. Future work includes exploring the use of FCN in other computer vision tasks and investigating the use of other efficient neural network architectures in FCN architectures.

References

[1] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes

through the ade20k dataset," arXiv preprint arXiv:1608.05442, 2016.

[2] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. CVPR, pp. 770–778, 2016

[3] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[4] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. CVPR, 2018.

[5] Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q. V., and Chen, Z. Gpipe: Efficient training of giant neural networks using pipeline parallelism. arXiv:1808.07233, 2018.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431– 3440.

[7] Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? CVPR, 2019.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR, 2018.

[9] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. MnasNet: Platform-aware neural architecture search for mobile. CVPR, 2019.

[10] Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv, abs/1905.11946.

[11] Zagoruyko, S. and Komodakis, N. Wide residual networks. BMVC, 2016

[12] Zoph, B. and Le, Q. V. Neural architecture search with reinforcement learning. ICLR, 2017