A Super-Fast Deep Network for Moving Object Detection

Bingxin Hou, Ying Liu, and Nam Ling
Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA 95053, USA
Email: {bhou, yliu15, nling}@scu.edu

***Abstract*—Deep learning methods have been actively applied to intelligent video surveillance for moving object detection in recent years and demonstrated impressive results. However, these models render superior accuracy at the cost of high computational complexity. In this work, we devised a new deep network structure that significantly improves inference speed, yet requires 10 times smaller model size and achieves 10 times reduction in floating-point operations as compared to existing deep learning models with tolerable accuracy loss.**

Keywords—moving object detection, video analytics, background subtraction, convolution neural network, deep learning, video surveillance

I. Introduction

With the rise of the Internet of Things, the spread of machine vision, and the large amount of video data, it is challenging and crucial to process or compress video data [1] at a fast speed. Efficiently reducing redundant information in videos such as the background and extracting meaningful foreground information like moving vehicles or pedestrians is a crucial step for video surveillance systems. Recently, deep learning-based moving object detection algorithms demonstrated superior detection accuracy as compared with traditional methods. However, existing deep models are computationally expensive and memory-intensive [2]. In this work, we demonstrate a new light-weight deep network model that achieved high detection accuracy, while it significantly accelerates the detection speed compared with current state-of-the-art deep models.

The paper is organized as follows. In section II, we introduce existing algorithms used for moving object detection. In section III, we elaborate on our proposed model in detail. Section IV describes our experimental setup and results compared with the current state-of-the-art models on the CDnet2014 dataset [3]. Section V concludes the paper.

II. Related Work

To detect moving objects in video scenes, conventional approaches basically include two components: First, background modeling and maintenance which initialize and update background scene over time by using average [4], temporal median filtering [5], or principal component analysis (PCA) [6]-[8]. Second, the classification of a new pixel as a foreground or background depends on whether the measured similarity between the new pixel and the corresponding reference pixel is above or below a predefined threshold. Traditional moving object detection methods combine these two components; examples include Gaussian Mixture Model (GMM) [9], IUTIS-5 [10] and SuBSENSE [11] that uses a feedback system to automatically adjust the background model based on the Local Binary Similarity Pattern (LBSP) features and pixel intensities [12].

Recently, deep learning methods are developed to replace the second component by using convolutional neural network (CNN) structures. In the first CNN-based moving object detection scheme ConvNets [13], the background is estimated by a temporal median filter, then the estimated backgrounds are stacked with the original video frames to form the input to a CNN that outputs the binary masks of detected objects. Some other deep learning methods combine these two components into one end-to-end network; examples include 3D CNN-LSTM [14] and MSFgNet [15]. Some methods skip the first component with a well-defined network structure that can replace the contribution from backgrounds; examples include the VGG-16 [16] based FgSegNet [17][18]. It contains a CNN that takes each video frame at three different scales in parallel as the input of the encoding network, then it adopts transposed convolutional layers in the decoding network to output a binary mask. However, the performance of all aforementioned deep learning-based moving object detection methods comes at a high cost of computation and slow speed due to complex network structures.

In this paper, we devised a new deep learning-based moving object detection approach with a simple network structure and simplified convolution operations. The proposed method does not require explicit background modeling and maintenance. It significantly accelerates inference speed and still achieves high detection accuracy.

III. Proposed Deep Network Model

The proposed network involves an encoder and a decoder shown in Fig. 1. We will elaborate on the details of our approach in the encoder network and the decoder network in the following.

A. The Encoder Network

The encoder network extracts features from the RGB 3-channel input frames, so-called feature map 'encoding' by convolutions with filters, where different filters can capture different features. The encoder network consists of 8 blocks as shown in Fig. 2(a). Iteratively tuning the configuration during training helps to choose 8 to be the least required number of

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

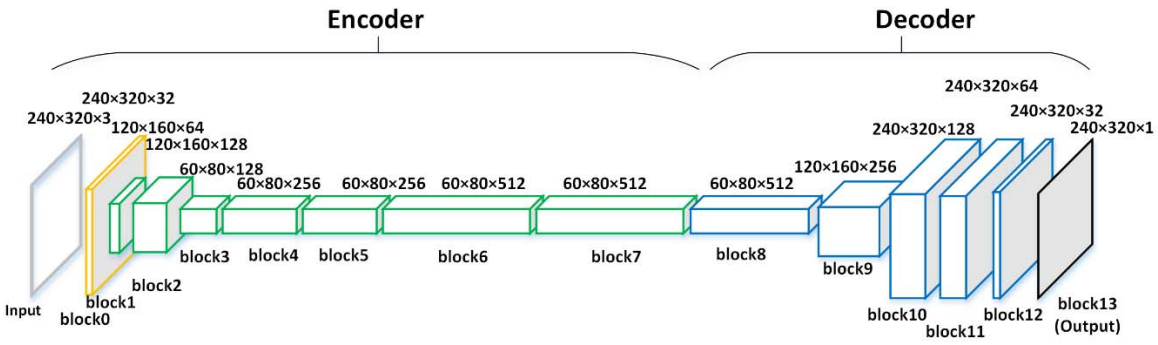
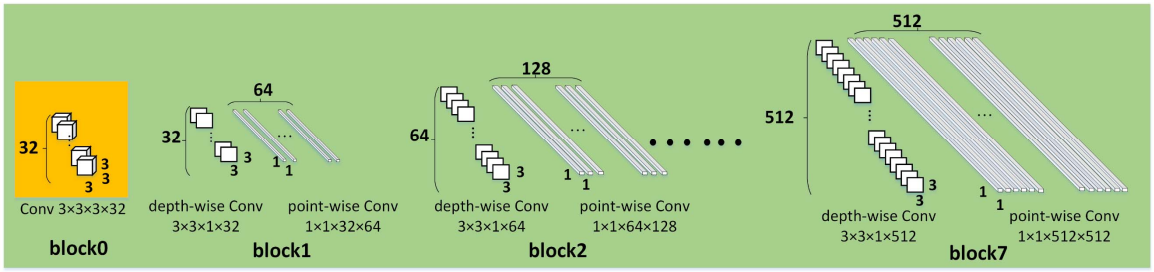
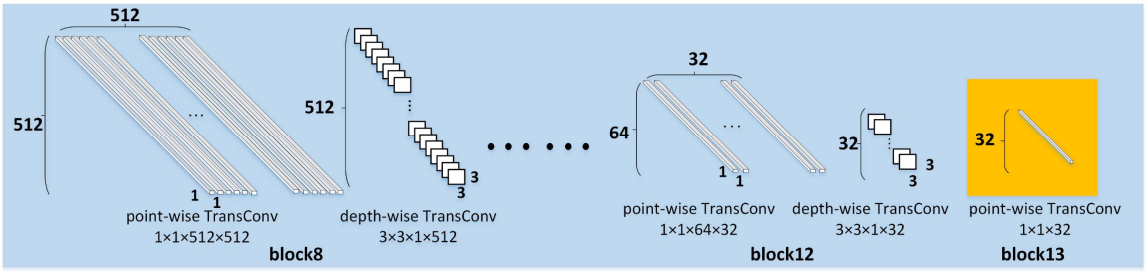


Fig. 1. The architecture of the proposed network



(a)



(b)

Fig. 2. Illustration of the filters used in (a) Encoder network (b) Decoder network

blocks to be able to achieve a high prediction accuracy where each block detects specific features present in the input data. In block 0, the regular convolution with 3D filters is adopted. 3×3×3×32 represents that the filter is 3×3 spatially, length-3 in the depth dimension, and the number of filters is 32, resulting in a layer with 32 channels. From block 1 to block 7, each block contains one depth-wise convolution, one batch normalization (BN) [19] and one point-wise convolution. For example, in block 1, the depth-wise convolution (depth-wise Conv) adopts 32 filters of dimension 3×3×1 and the point-wise convolution (point-wise Conv) adopts 64 filters of dimension 1×1×32. The filters of block 2 to block 7 are similarly defined. The notions of depth-wise convolution and point-wise convolution were first proposed in MobileNet [20]. The depth separable convolution replaces traditional convolution by applying an independent 2D filter for each input channel followed by point-wise operation using a 1×1×c (c is the number of input channels) convolution on each depth-column of the outputs from the depth-wise convolution [20]. The effect of this separation is to greatly reduce the amount of computation and model size [20]. The computational cost can be reduced to $\frac{1}{N}+\frac{1}{D_K^2}$ of a regular convolution, where N is the number of output channels and D_K is the spatial dimension of the $D_K \times D_K$ kernel [20]. In this paper, for the first time in the literature, we adopt such separated depth-wise and point-wise convolution in a moving object detection network.

B. The Decoder Network

Fig. 2(b) shows our proposed decoder network that expands the encoder output and generates a binary mask for the detected moving objects. The decoder has 6 blocks. We propose from block 8 to block 12 each of them consists of one point-wise transposed convolution and one depth-wise transposed convolution. Transposed convolution, informally called deconvolution, is a backward stride convolution for up-sampling optimally. In block 8, the point-wise transposed convolution (point-wise TransConv) adopts 512 filters of dimension 1×1×512, and the depth-wise transposed convolution (depth-wise TransConv) adopts 512 filters of dimension 3×3×1. The filters of block 9 to block 12 are similarly defined. In block 13, the point-wise transposed convolution adopts one filter of dimension 1×1×32. We propose to use a regular transposed convolution with 1×1×c (c is similarly defined as in the encoder network) filter and stride 2 for up-sampling as the point-wise transposed convolution, followed by a regular depth-wise

convolution to perform as a transposed convolution in depth-wise. Finally, a sigmoid activation function is appended to output a 0/1 binary mask (0 is background, 1 is foreground) with the same spatial dimension as the input images. For the first time in the literature, we propose the point-wise and depth-wise deconvolution in the decoder network symmetric to the encoder network for computational efficiency. Table I shows the details of the input and output shape and configuration in each layer.

IV. Experimental Results and Analysis

In this section, we evaluate the performance of our proposed deep moving object detection network on the CDnet2014 dataset [3]. It contains 11 categories of video sequences (such as baseline, bad weather, dynamic background, camera jitter, etc.), and each category contains 4 to 6 video sequences (such as highway, office, pedestrians, PETS2006 in the baseline category). In our experiments, 15 sequences covering 6 categories were selected for the training and testing. For each video sequence, 200 frames were selected randomly. The training was performed for every single video sequence using an Intel Xeon 8-core 3GHz CPU processor with an Nvidia Titan RTX 24G GPU. In the training phase, we use RMSprop optimizer and cross-entropy loss function at a learning rate $\alpha = 10^{-4}$ over 50 epochs in a batch size of 1. In the testing phase, we test more than 1,000 frames for each sequence with training frames excluded.

To evaluate the model performance, we calculate the *F-measure* between the predicted binary masks and the ground truth binary masks provided in the CDnet2014 dataset [3]. The F-measure (also called F_1 score or F score) is defined as:

$$\mathrm{F} - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (1)$$

where $precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$, $recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$. The inference speed measured in frames per second (fps) is also recorded.

Table II compares the F-measure scores and the inference speed of the proposed model, with the state-of-the-art FgSegNet (FgSegNet 1-scale, FgSegNet 3-scale) described in Section II, and the MobileNet+UNet model which is a MobileNet based Unet [21], recently proposed for semantic segmentation.

We observe that our proposed model achieves an inference speed of 149.81 fps, more than twice faster than those of the MobileNet+UNet and the FgSegNet 3-scale, and 1.8 times faster than that of the FgSegNet 1-scale. In terms of detection accuracy, our proposed model achieves an average F-measure of 0.9718, significantly higher than that of MobileNet+UNet, and only slightly lower than those of the two FgSegNet models.

TABLE I. The proposed network configuration. Encoder consists of blocks 0 to 7, Decoder consists of blocks 8 to 13.

		Layer Type / Stride	Filter Shape	Output Shape	Parameters #
Encoder	**block 0**			240×320×3 **(Input Image)**	
		Conv / s=1	3×3×3×32	240×320×32	896
	block 1	Conv dw / s=2	3×3×1×32 dw	120×160×128	352
		BN		120×160×128	64
		Conv pw/ s=1	1×1×32×64 pw	120×160×64	2048
	block 2	Conv dw / s=1	3×3×1×64 dw	120×160×64	704
		BN		120×160×64	128
		Conv pw/ s=1	1×1×64×128 pw	120×160×128	8192
	block 3	Conv dw / s=2	3×3×1×128 dw	60×80×128	1408
		BN		60×80×128	256
		Conv pw/ s=1	1×1×128×128 pw	60×80×128	16384
	block 4	Conv dw / s=1	3×3×1×128 dw	60×80×128	1408
		BN		60×80×128	256
		Conv pw/ s=1	1×1×128×256 pw	60×80×256	32768
	block 5	Conv dw / s=1	3×3×1×256 dw	60×80×256	2816
		BN		60×80×256	512
		Conv pw/ s=1	1×1×256×256 pw	60×80×256	65536
	block 6	Conv dw / s=1	3×3×1×256 dw	60×80×256	2816
		BN		60×80×256	512
		Conv pw/ s=1	1×1×256×512 pw	60×80×512	131072
	block 7	Conv dw / s=1	3×3×1×512 dw	60×80×512	5632
		BN		60×80×512	1024
		Conv pw/ s=1	1×1×512×512 pw	60×80×512	262144
Decoder	**block 8**	TransConv pw / s=1	1×1×512×512 pw	60×80×512	262656
		Conv dw / s=1	3×3×1×512 dw	60×80×512	4608
	block 9	TransConv pw/ s=2	1×1×512×256 pw	120×160×256	131328
		Conv dw / s=1	3×3×1×256 dw	120×160×256	2304
	block 10	TransConv pw / s=2	1×1×256×128 pw	240×320×128	32896
		Conv dw / s=1	3×3×1×128 dw	240×320×128	1152
	block 11	TransConv pw / s=1	1×1×128×64 pw	240×320×64	8256
		Conv dw / s=1	3×3×1×64 dw	240×320×64	576
	block 12	TransConv pw/ s=1	1×1×64×32 pw	240×320×32	2080
		Conv dw / s=1	3×3×1×32 dw	240×320×32	288
	block 13	TransConv pw/ s=1	1×1×32 pw	240×320×1	33
		Activation		240×320×1 **(Binary Mask)**	
Total					983105

TABLE II. COMPARISON OF F-MEASURE AND INFERENCE SPEED WITH OTHER MODELS ON DIFFERENT CATEGORIES AND SEQUENCES

Category	Sequence	F-Measure				Inference Speed (fps) on PC			
		MobileNet +UNet	*FgSegNet (3-scale)*	*FgSegNet (1-scale)*	*Proposed model*	*MobileNet +UNet*	*FgSegNet (3-scale)*	*FgSegNet (1-scale)*	*Proposed model*
No.1	**baseline(highway)**	0.9698	0.9970	0.9975	0.9931	56.02	70.42	80.19	143.35
No.1	**baseline(office)**	0.9737	0.9934	0.9936	0.9839	62.19	66.42	86.28	158.58
No.1	**baseline(pedestrians)**	0.0373	0.9743	0.9778	0.9596	51.36	67.28	86.73	153.00
No.1	**baseline(PETS2006)**	0.0796	0.9948	0.9958	0.9714	69.74	72.73	85.98	154.08
No.2	**badWeather(skating)**	0.9525	0.9930	0.9941	0.9839	66.62	66.12	78.99	139.08
No.2	**badWeather(bilzzard)**	0.6424	0.9945	0.8241	0.9545	63.45	70.57	82.37	152.44
No.2	**badWeather(snowFall)**	0.1638	0.9820	0.9290	0.9619	46.06	70.98	78.19	155.04
No.2	**badWeather(wetSnow)**	0.8563	0.9733	0.9669	0.9384	56.75	70.64	85.76	147.15
No.3	**dynamicBackground(boats)**	0.9797	0.9982	0.9982	0.9945	60.42	65.99	84.53	161.29
No.3	**dynamicBackground(canoe)**	0.9606	0.9991	0.9994	0.9964	63.65	67.35	79.94	137.82
No.4	**lowFramerate(tunnelExit_0_35fps)**	0.7515	0.9911	0.9928	0.9803	49.55	65.22	86.06	161.55
No.4	**lowFramerate(turnpike_0_5fps)**	0.9536	0.9959	0.9969	0.9912	56.95	66.16	81.10	143.88
No.5	**shadow(bungalows)**	0.9215	0.9957	0.9958	0.9834	64.43	72.34	79.05	135.50
No.6	**nightVideos(busyBoulvard)**	0.8841	0.9645	0.9561	0.9403	51.81	71.35	80.78	160.36
No.6	**nightVideos(winterStreet)**	0.9389	0.9856	0.9887	0.9444	50.43	71.72	78.65	144.09
Average		0.7377	0.9888	0.9738	**0.9718**	57.96	69.02	82.31	**149.81**

TABLE III. SUMMARY OF COMPARISON BETWEEN THE PROPOSED MODEL AND OTHER MODELS

	Training Scheme	F-Measure	Parameters #	Flops(millions/s)	Model Size (bytes)	Inference Speed (fps)
FgSegNet 3-scale	single-scene, RGB	0.9888	15,857,665	16.20	60MB	69.02
FgSegNet 1-scale	single-scene, RGB	0.9738	9,358,593	9.63	36MB	82.31
MobileNet+UNet	single-scene, RGB	0.7377	4,217,397	4.18	16.8MB	57.96
Proposed Model	single-scene, RGB	**0.9718**	**983,105**	**1.42**	**3.8MB**	**149.81**

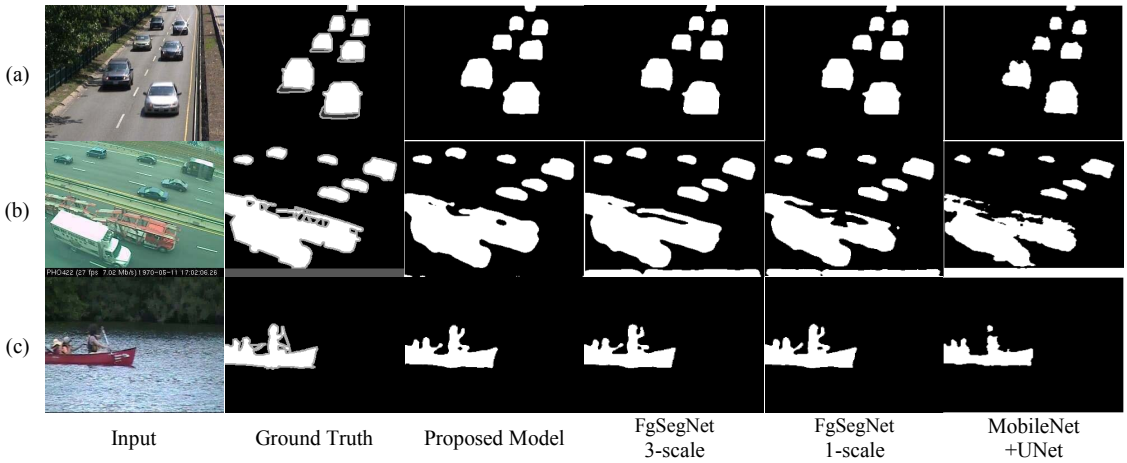


Fig. 3. A comparison results on 3 out of 15 sequences (a) baseline: highway (b) lowFramerate: turnpike_0_5fps (c) dynamicbackground: canoe . The first column is input image, the second column is ground truth, third-sixth columns are the proposed model, FgSegNet 3-scale, FgSegNet 1-scale, MobileNet+UNet respectively.

Table III shows the overall comparisons on F-measure, the number of model parameters, floating-point operations (FLOPs in millions/second), the model space (the storage space size of weights in megabytes (MB)), and the inference speed. The proposed model requires the least number of model parameters, the least number of floating-point operations, the smallest model storage space size, while it offers the fastest average inference speed and a competitive moving object detection accuracy in terms of the F-measure.

Fig. 3 shows the visual results for three sequences (from 15 sequences in the experiment) in three categories: baseline, low frame rate, and dynamic background. Our method has detected clear backgrounds and refined boundaries in the foreground regions, which is competitive with the results of FgSegNet 3-scale and FgSegNet 1-scale, and much better than the result of MobileNet+UNet.

V. CONCLUSION

In this paper, we devise an efficient deep network model based on depth-wise and point-wise convolution and deconvolution for moving object detection in surveillance video. The proposed model uses a simple network structure with simplified convolution operations. Experimental studies demonstrated the effectiveness of our proposed model. For future study, we intend to incorporate temporal information and upgrade the network to achieve better performance.

REFERENCES

[1] Licheng Xiao, Hairong Wang, and Nam Ling, "Image Compression with Deeper Learned Transformer," Proceedings of the APSIPA Annual Summit and Conference 2019, pp. 53-57, Lanzhou, China, Nov 18-21, 2019.

[2] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang, "A survey of model compression and acceleration for deep neural networks," arXiv:1710.09282, 2017.

[3] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., pp. 393–400, 2014.

[4] Y. Zheng and L. Fan, "Moving object detection based on running average background and temporal difference," Proc. 2010 IEEE Int. Conf. Intell. Syst. Knowl. Eng. ISKE 2010, pp. 270–272, 2010.

[5] Q. Zhou and J. K. Aggarwal, "Tracking and classifying moving objects using single or multiple cameras," Handb. Pattern Recognit. Comput. Vision, 3rd Ed., vol. 0012, pp. 499–524, 2005.

[6] Liu Y and Pados D. A, "Compressed-sensed-domain l1-pca video surveillance," IEEE Trans. Multimedia, 18(3), 351–363 (2016).

[7] Pierantozzi M, Liu Y, Pados D. A, and Colonnese S, "Video background tracking and foreground extraction via l1-subspace updates," Proc. SPIE Commercial + Scientific Sensing and Imaging, Compressive Sensing V: From Diverse Modalities to Big Data Analytics 9857, 985708 (2016).

[8] Ying Liu, Zachary Bellay, Payton Bradsky, Glen Chandler, and Brandon Craig, "Edge-to-fog computing for color-assisted moving object detection," Proc. SPIE 10989, Big Data: Learning, Analytics, and Applications, 1098903 (13 May 2019)

[9] T. S. F. Haines and T. Xiang, "Background subtraction with dirichlet process mixture models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 4, pp. 670–683, 2014.

[10] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10484 LNCS, pp. 96–107, 2017.

[11] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," IEEE Trans. Image Process., vol. 24, no. 1, pp. 359–373, 2015.

[12] T. Akilan, "A foreground inference network for video surveillance using multi-view receptive field," arXiv:1801.06593, 2018.

[13] Marc Braham and Marc Van Droogenbroeck. "Deep background subtraction with scene-specific convolutional neural networks," International Conference on Systems, Signals and Image Processing (IWSSIP), pages 1–4. IEEE, 2016.

[14] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-based image-to-image foreground segmentation," IEEE Trans. Intell. Transp. Syst., pp. 1–13, 2019.

[15] P. Patil and S. Murala, "MSFgNet: a novel compact end-to-end deep network for moving object detection," IEEE Transactions on Intelligent Transportation Systems, December 2018.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representation, 2015.

[17] L. Lim and H. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," Pattern Recognition Letters, 112:256–262, 2018.

[18] L. Lim, l. Ang, and H. Keles, "Learning multi-scale features for foreground segmentation," arXiv:1808.01477, 2018.

[19] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning, pp. 448–456, 2015.

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

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," CoRR, vol. abs/1505.04597, 2015.