Real-time Quadrilateral Object Corner Detection Algorithm Based on Deep Learning

Jinfeng Zhang, Zhibin Jiao, Xiangjing An and Yejun He*
Guangdong Engineering Research Center of Base Station Antennas and Propagation
Shenzhen Key Laboratory of Antennas and Propagation
College of Electronics and Information Engineering, Shenzhen University, 518060, China
Email: zhangjf@szu.edu.cn, jzb1631@163.com, 2172262984@email.szu.edu.cn, heyejun@126.com*

***Abstract*—In the field of computer vision, there are many quadrilateral objects such as slide, screen, book, etc, which usually has the characteristics of rotation, perspective, and beveling. Therefore, it becomes a key task to detect the corner points of the quadrilateral object to restore the content. This paper proposes an algorithm to solve the problem of quadrilateral object corner detection. Different from the previous methods, our method does not make any assumptions about the background, distance, content and other attributes of the quadrilateral object, which can be applied to a wider range of quadrilateral objects. We named the proposed method as Corner Detect Network (CDN). Simulation results show that the proposed algorithm can quickly and accurately extract the corner point of quadrilateral object.**

***Index Terms*—Deep learning, computer vision, object detection, corner detection**

I. Introduction

As one of the most important research fields of computer vision, object detection is widely used in autonomous driving, medical, industrial automation and other aspects. For object detection tasks, many researchers have developed a variety of object detection algorithms based on deep learning. Among them, the detection of quadrilateral objects often differs from other kinds of objects. The quadrilateral objects such as books, documents, screens, slides, etc. often contain a large amount of information, when the user tries to capture an image, it is often for the purpose of getting information from the object. Therefore, compared with general object detection, the central issue of quadrilateral object detection is to detect the corner point of the quadrilateral object that can restore the content from the object. Fig.1 shows the difference between the two tasks.

Classic object detection algorithms such as Faster RCNN [1], YOLO [2], SSD [3] etc. These algorithms show good detect efficiency and accuracy on general object detection problems, but when quadrilateral object detection is performed, these algorithms can only output the position of the rectangular box based on Boundingbox. Therefore, for a quadrilateral object with features such as rotation, perspective, and bevel in a natural image, these method cannot accurately output the position information of the quadrilateral object's corner. For a quadrilateral object captured in a real environment, how to get its corner point accurately and quickly is a challenging task.

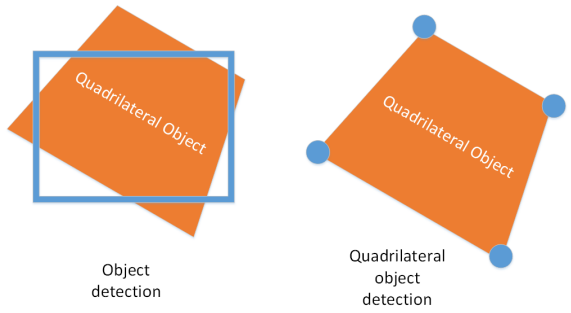


Fig. 1. Difference between object detection and quadrilateral object detection.

Our contributions include:
1) Generating semantic segmentation images on small-scale feature maps to speed up the network;
2) Presenting a simple and effective method for roughly extract quadrilateral corner points on semantically segment images;
3)Fully connected layers are used to precisely adjust the position of the corner points, which ensuring the speed of the network and improved the detection accuracy.

In view of the needs in practical applications, in this paper, we provide an end-to-end training method for single-object corner detection and an alternate optimization training method for multiple objects. referring to the accuracy evaluation standard of face key point detection [4], we define a similar accuracy evaluation standard. In the single-object task, our algorithm can reach 10 fps(on mobile device), and with failurerate 0.12. In multi-object task, the speed of our model is 3 fps, with the failurerate 0.23.

The detection of quadrilateral corner points has many algorithms in the aspect of computer vision. Lu and Chen proposed a method: By confidence measure for each quadrilateral [5], which uses the asymmetry variance of the edge histogram as the feature to detect quadrilateral. However,the content of quadrilatral is limited with Lu and Chen's method. To fix this problem, whiteboard scan [6] method developed by Zhang and He. Their method uses Sobel operator to generate image edges. Then, the Hough transform is used to detect the straight line in the image, and some of quadrilaterals combined by the straight lines are filtered according to a set of rules. Finally, according to the ratio of support edge number and the circumference, a

978-1-7281-1973-1/19/$31.00 ©2019 IEEE

quadrilateral having the largest value on the feature is selected as the output.

So far, in the task of quadrilateral detection, all method uses the computer vision traditional algorithm to extract the image features. So, the success rate of quadrilateral detection in complex background is not good enough. Therefore, use the deep learning method is necessary, and it becomes our work.

The rest of this paper is structured as follows:Corner detection algorithms are viewed in Section II, in SectionII part A, Single object network will be presented, and in part B, Multi object network is presented. Then in both part, we introduce the detail of our algorithm. next, in part C, we introduced the training method of algorithm. in partD, we pull in a evaluation standard. Comparison of the simulations with each algorithm are made in part E. Finally, the summary and the future works are given in the last Section.

II. Quadrilateral object corner detection with Corner Detect Network(CDN)

Different from the existing object detection algorithm based on BoundingBox, for the quadrilateral object, the semantic segment is used to locate the object region which mapped a quadrilateral object. Then, two sets of convolution kernels are utilized to locate corner points on object region, we named this part as Point Of Interests(POI) layer, and finally, the fully connected layer is used to corrected the position of corner point.

Considering that in some practical applications, user take a photo is only to get one slide, screen, or content in a book, that is, only need to get a single object, the algorithm of this paper is designed into two work modes, one is the single object mode and the other is the multi-object mode. Next, we introduce the design and training methods of the two modes, and show their test results on the data set.

A. Single object network design

For the single-object task scenario of quadrilateral object detection, the algorithm can be descript as the following parts: base network, Region Generator layer, POI layer and corner point regression layer. Algorithm 1 is the general process:

Algorithm 1 Framework of single object detection.

Input: Input image im size:[224,224,3];
Output: Four corner points, $Result$ size:[4,2];

1: Extracting the feature map fm use base network;
2: Perform the region generator on fm to get the semantically segment image seg_{im};
3: Roughly extract 4 corner points p by run POI layer on seg_{im};
4: Get the feature vector fv from fm by coordinate p, and run corner point regression layer on fv to generate the correction value reg of corner point;
5: $Result = (p + 0.5) \times 8 + reg$;
6: **return** $Result$;

Below are each parts we introduce.

1) Base network: In this part, we taking the ShuffleNetV2 [7] network and MobileNetV2 [8] network for compared, result shows in TABLE II. Based on the computation amount, the algorithm selects the MobileNetV2 network structure. After comprehensively examing the semantic information richness and computation complexity, the algorithm add a deconvolution layer [9] to the 13th Inverted Residual Block of MobileNetV2. the size of feature map that the inverted risidual block outputs is extended from [1,14,14,96] to [1,28,28,96]. By this operation, the object area is too small that can not extract corner point is avoid.

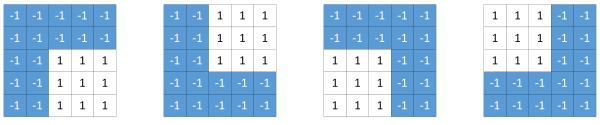
2) Region Generator layer: The objective of this layer is to achieve semantically segment the object area on a small size feature map. Inspired by the common algorithm FCN [10], this paper uses a convolution layer to generate a quadrilateral object segmentation image based on the feature map output from the base network. Different from the FCN algorithm, based on the computation amount of the algorithm, instead of generating semantically segment image on large size, our algorithm chooses to use a 1*1 size convolution kernel to make a convolution on the feature map. Generating an area segmentation image having the same size as the feature map.

3) Point of interests (POI) layer: This layer is used to roughly extract the corner point in the quadrilateral object area acquired by the Region Generator layer. Considering that the quadrilateral object may have two rotation modes as shown in Fig.2, this paper designs two sets of convolution kernels for the corner points extraction.

(a) mode 1:no rataion (b) mode 2:45°rotation

Fig. 2. Two rotaion mod of quadrilateral object

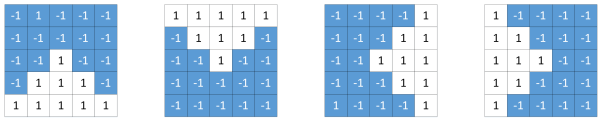
For mode one, as shown in Fig.3, the four convolution kernels are used to respectively detect the top left, bottom left, top right, and bottom right corners of the quadrilateral object.



(a) left top (b) left bottom (c) right top (d) right bottom

Fig. 3. Convolution kernal design for mode 1

For mode two, as shown in Fig.4, the four convolution kernels are designed to detect the four corners of the quadrilateral object, which is, top, bottom, left, and right corner. It should be noted that the size of the two sets of convolution kernels can be adjusted. For example, the convolution kernel size used in this experiment is 11*11.



(a) top (b) bottom (c) left (d) right

Fig. 4. Convolution kernal design for mode 2

Base on the two sets of convolution kernels, the steps of object corner point roughly extraction are as follows: First, the two sets of convolution kernels are respectively convolved with the image generated by the Region Generator layer, and for each convolution result, the position of maximum value and the maximum value is recorded. Each mode have 4 maximum values, then add them as a mode summation, next the two mode summation is compared, the mode with the larger mode summation is selected. Finally, output four points via the selected mode as the corner point of quadrilateral object.

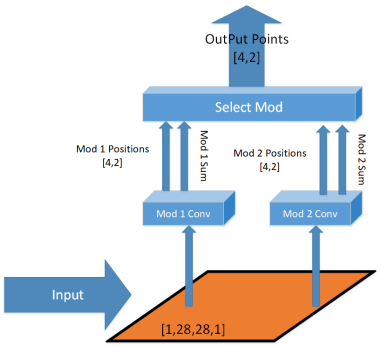


Fig. 5. Point of Interests(POI) layer.

4) Corner point regression layer: In this step, the corner point that get by POI layer is corrected using the corner point regression layer. To be more specific, first, the corner point is mapped in the feature map that generated by the base network, and the feature vector that corresponding to the point is extracted. Next, two fully-connected layers are used to output the horizental and vertical corrections of the four corner points on the 224*224 size image. The correction function are discribed as follows.

$$\begin{aligned} reg_y &= gt_y - (p_y + 0.5) \times 8 \\ reg_x &= gt_x - (p_x + 0.5) \times 8 \end{aligned} \tag{1}$$

where reg_y and reg_x represent the offset of a corner point in horizontal and vertical directions on the 224*224 size image. while gt_y and gt_x are the coordinates of ground truth on 224*224 size image, p_y and p_x are the coordinates that roughly extracted by POI layer on 28*28 size image.

The first fully connected layer consists of 96 neurons, the second fully connected layer consists of 2 neurons. Figure 6 shown the structure.

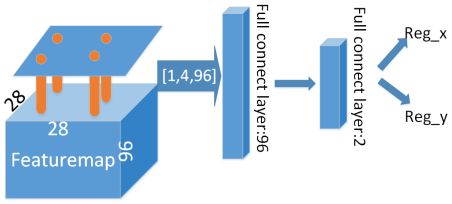


Fig. 6. Corner point regression layer of single object.

B. Multi-object task network design

The design for the multi-object task network is similar to that of the single-object task network. That is, the roughly corner point position of each quadrilateral is located by the object area, and the corner point is corrected by the corner point regression layer. Different from the single-object task network, for the scenerio of multi-object task, when the object corner point is extracted, each object area of the multi-object area needs to be separated, and then the POI layer is used to extract the corner point. Algorithm 2 is the general process:

Algorithm 2 Framework of multi-object detection

Input: Input image im size:[224,224,3];
Output: Output corner point $Result$ size:[N,4,2];

1: Run the base network to extracting the feature map fm;
2: Semantically segment the im on size [56,56] and get the segment image seg_{im} by perform region generator on fm;
3: Split the seg_{im} into N images by using connected area detection layer;
4: i=1
5: **repeat**
6: Roughly extract corner point p from seg_{im} using POI layer;
7: Get the vector fv from fm via coordinate p and using corner point regression layer to generate the correction value reg of corner point;
8: $Result[i] = (p + 0.5) \times 8 + reg$;
9: $i = i + 1$;
10: **until** $(i >= N)$

Each part of multi-object tasks are as follows.

1) Base network: Base network: Multi-object network uses the structure similar to the single-object network. However, for multi-object tasks, the network must ensure that the generated semantic segmentation is large enough, otherwise the near different object will be "stick together", and be identified as the same object by the two pass algorithm [11]. To avoid such mistake, the algorithm generates semantic segmentation on a large scale, and the different levels of inverted residual blocks are merged together by deconvolution as the output of the base network.

2) Region Generator layer: This layer is similar to that in single-object network, except that the size of output is [1,56,56,2].

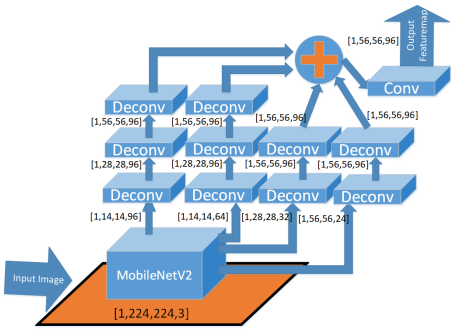


Fig. 7. Base network of multi object.

3) Connected area detection layer: Adding this layer to multi-object task is to achieve segmentation of multiple quadrilateral objects. In this layer, for the output of the Region Generator layer, a region extraction image is segmented into N images by the two-pass method [11], where N is the number of quadrilateral objects. The process is as follows.

1. Binarize the output of the Region Generator layer to obtain a region segmentation image on size 56* 56.
2. Binarized region image are segmented into some connected region images using two-pass method.
3. Count the number of pixels in each region obtained by step 2,and remove the regions which smaller than n pixels to exclude suspected objects,in our experiment n = 20.
4. Output the final N binary connected regions images[N, 56, 56] Here, [N, 56, 56] indicates that N images are generated, and each image has a connected region.

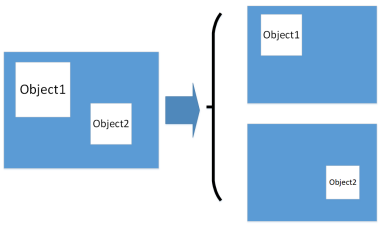


Fig. 8. Detect connected area and split into different images.

4) POI layer: In the multi-object network, the design of this layer is similar to the single-object network, but in order to adapt to the corner point extraction on the large-size segmentation image, the convolution kernel size in the POI layer is set to be 21*21.

5) Corner point regression layer: This part is similar to the corner point regression layer of the single object network, except that the input feature map is changed from [1, 28, 28, 1] of the single object network to [N, 56, 56, 1], and the output is from the single object network [4, 2] becomes [N, 4, 2], where N represents the number of objects.

C. Training

In this part, we provide two different training methods for the two networks. In the single object network, we adopt a joint training method. However, in multi object networks, because it is difficult to use the existing deep learning framework to detect connected areas, the whole network is divided into two parts using alternate training method.

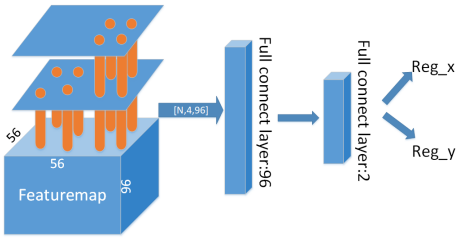


Fig. 9. Corner point regression layer of multi-object.

Single object network training: Loss function are set at the Region Generator layer and corner point regression layer, Setting Region Generator's loss are as follow.

$$\begin{aligned} \mathrm{L}\left(\mathrm{p}, \mathrm{p}^{*}\right) &= \frac{1}{\mathrm{N}_{cls}} \sum_{i} L_{cls}\left(p_i, p_i^{*}\right) \\ L_{cls}\left(p_i, p_i^{*}\right) &= p_i^{*} \times \log p_i \end{aligned} \tag{2}$$

where N_{cls} is the number of training samples, p_i is the prediction of network, p_i^{*} is the Ground Truth. Set the loss of corner point regression layer as:

$$\begin{aligned} \mathrm{L}\left(\mathrm{t}, \mathrm{t}^{*}\right) &= \frac{1}{\mathrm{N}_{reg}} \sum_{i} L_{reg}\left(t_i, t_i^{*}\right) \\ L_{reg}\left(t_i, t_i^{*}\right) &= \sqrt{t_i^2 - t_i^{*2}} \end{aligned} \tag{3}$$

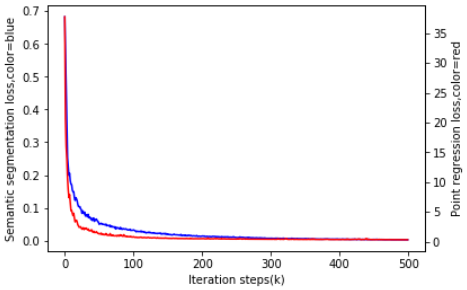
where the N_{reg} is number of corner points, t_i is the correction of predict, t_i^{*} is the Ground Truth.

Multi object network training: multi object network can be regarded as a two stage network, the first stage consist of base network and Region Generator, the second stage consist of connected region detector, POI layer and corner point regression layer, therefor, the alternative training are designed as follows.

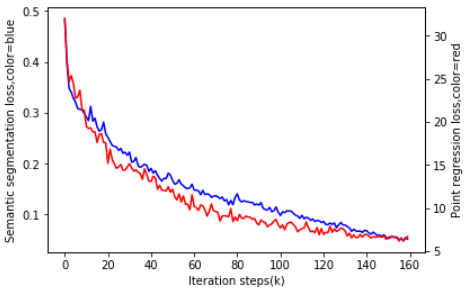
1.Optimize stage 1 and save the segmentation result of quadrilateral object area output by stage 1.
2.In stage2, first, check whether the number of connected areas is the same as the number of objects in Ground Truth. If the number of connected areas is the same as the number of objects in Ground Truth, go to 3, if not, turn to 1.
3.Because the object sequence of connected area detection output is not always the same as that of Ground Truth, it is necessary to match the corners points generated by POI layer in stage 2 with those in Ground Truth, and then calculate gt_y and gt_x (in Section 2.1.4) to train the corner point regression network in stage 2.

Figure 10 shows relationship between loss decreases and iteration steps.

From Fig 10, Zooming in on the multi-object network training process from 0 to 160k, Obviously that the decrease of point regression loss is slower than in the single object network.This is because the network extracts feature vectors



(a) Single object network



(b) Multi-object network

Fig. 10. The loss value decreases with the increase of iterations.

in the point regression layer. When there is only one object, the correspondence between the feature vector and the object is simpler. When there are multiple objects, the correspondence between the feature vector and the object is more blurred.

D. Evaluation standard

In the experiment, we use the evaluation standard of the key point detection in reference [4]: the inter-diagonal distance normalized error averages and failure rate to evaluate the method. Among them, inter-diagonal distance normalized error averages reflected the precision of prediction, describe it with NRMSE:

$$NRMSE = \frac{\frac{1}{n}\sum_{i=1}^{N} \|s_{p,i} - s_{g,i}\|_2}{mean\,(diagonal)} \tag{4}$$

another evaluation standard, Failure rate, reflects the recall rate of corner points, which is defined as follows:

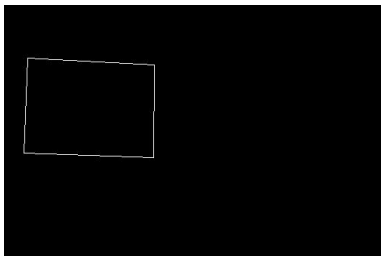
$$failurerate = \frac{f_N}{N} \times 100\% \tag{5}$$

any error above 10% is considered to be a failure. f_N means the failure number of testing images, while N is the total number of testing images.

E. Results on datasets

Our data set comes from the Slide/Document Corner Recognition Project of Shenzhen Ronghui Science and Technology Co., Ltd. The data set collects a large number of Slide/Document images from the Internet/real scene. On this basis, our data set screens out about 1000 Slide images for training. Figure 11 shows how to make labels for each image.

(a) Picture sample



(b) Label sample

Fig. 11. Dataset format.

In the [5], [6] articles related to this paper, Zhang and He's method is our comparative algorithm, because they do not make any assumptions about the content of the quadrilateral when detecting the quadrilateral, and have been widely used in the development of corresponding applications on the mobile terminal. On this data set, we measure the performance of the algorithm as TABLE 1. When evaluating the performance of the model, since the labeling of the data set tends to be marked in the internal measurement of the object contour, the label of the data set is subjective for the non-deep learning algorithm. For fairness, we set the threshold value of whiteboard scan [5] to 0.3 Because label with manual mark has certain subjective influence.

We count NRMSE and Failure rate on computer with a intel i5-7500 cpu and 16G RAM. For FPS, we use tensorflow generate a tflite file which run on Adroid device with Hisilicon Kirin 710 and 6G RAM to check algorithm perform.

TABLE I
SIMULATION RESULT

	Method	NRMSE	Failure rate	FPS(on mobile)
Single object	ours	0.035	0.12	>=10
	Whiteboard scan	0.26	0.8	-
Multi-object	ours	0.0605	0.23	3

TABLE II
BASE NETWORK TEST

	Base network	NRMSE	Failure rate	FPS(on mobile)
Single object	MobileNetV2	0.035	0.12	>=10
	ShuffleNetV2	0.06	0.2	6

From the data in the TABLE 1 above, our algorithm performs much better than Whiteboard Scan. This is mainly

because the Whiteboard Scan algorithm makes strict assumptions about the state of quadrilateral objects, such as the circumference of quadrilateral should not be less than (H+W)/2, and sometimes the screen is far away from the photographer. Moreover, in reality, the quadrilateral that needs to be photographed is often in a more complex environment, which leads to the algorithm based on line detection can not achieve higher accuracy. Some results on dataset are shown in Fig.12.

(a) Example1 (b) Example2 (c) Example3

(d) Example4 (e) Example5 (f) Example6

(g) Example7 (h) Example8 (i) Example9

Fig. 12. Some detected results on dataset

III. Conclusion

Aiming at the problem of quadrilateral detection widely existing in object detection, this paper proposes a new algorithm for corner detection of quadrilateral objects. Firstly, Mobilenet V2 network is used to extract image features, and then region generator layer is used to generate object regions. For the single object and multi-object task scenarios involved in quadrilateral detection, two network modes are designed. For the single object task, corner points are extracted directly from the object area, and then the corner point is regressed using the corner point regression layer. For multi-object networks, it is necessary to first run the two pass [11] algorithm to detecting connected areas, divide it into several images, then roughly extract the corner points of quadrilateral objects in each area, and finally, as for single object task, use the corner point regression layer to regress the corner point. It should be pointed out that in multi-object network, in order to avoid the "adhesion" of the object area, large-scale feature maps are needed to detect corner point, which slows down the speed of the network, and also has a certain impact on the detection success rate of the algorithm. So far, our algorithm has not solved the problem of overlapping different objects. So, our next goal of efforts are solve this problem and improve the detect success rate.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grants 61372077, 61801299, and 61871433, in part by the Shenzhen Science and Technology Programs under Grants GJHZ20180418190529516, JCYJ20170302150411789, JCYJ20170302142515949, JSGG20180507183215520, and GCZX2017040715180580, and in part by the Guangzhou Science and Technology Program under Grant 201707010490.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[4] Z. Deng, K. Li, Q. Zhao, and H. Chen, "Face landmark localization using a single deep network," in *Biometric Recognition*, Z. You, J. Zhou, Y. Wang, Z. Sun, S. Shan, W. Zheng, J. Feng, and Q. Zhao, Eds. Cham: Springer International Publishing, 2016, pp. 68–76.

[5] S. Lu, B. M. Chen, and C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations," *Image and Vision Computing*, vol. 23, no. 5, pp. 541 – 553, 2005.

[6] Z. Zhang and L.-W. He, "Whiteboard scanning and image enhancement," *Digital Signal Processing*, vol. 17, no. 2, pp. 414 – 432, 2007.

[7] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 122–138.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 4510–4520.

[9] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2528–2535.

[10] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[11] K. Wu, E. Otoo, and K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Pattern Analysis and Applications*, vol. 12, no. 2, pp. 117–135, Jun 2009.