Exploring An Easy Way for Imbalanced Data Sets in Semantic Image Segmentation

Xiaoling Xia\textsuperscript{1,a} Qinyang Lu\textsuperscript{2,b\*} Xin Gu\textsuperscript{3,c}

\textsuperscript{1}College of Computer Science, Donghua University
Shanghai, China

\textsuperscript{2}College of Computer Science, Donghua University
Shanghai, China

\textsuperscript{3}College of Computer Science Donghua University
Shanghai, China

\textsuperscript{a}scherlysha@dhu.edu.cn \textsuperscript{b}2181818@mail.dhu.edu.cn \textsuperscript{c}guxin010@gmail.com

ABSTRACT: In recent years, deep convolutional neural networks have gradually become the preferred method for image processing. After the development of Classification, Detection and Segmentation, a large variety of state-of-the-art models and algorithms have emerged in the field. However, for some specific data sets or tasks, not all methods are applicable, which is inconvenient to researchers. This paper took the data set provided in the airbus ship detection challenge in Kaggle as an example to explore an easy and effective method for segmentation tasks of data sets with class imbalance. This paper used U-Net with a pre-trained ResNets model, and tried different methods to explore the feature of the set. In the process of training ResNets, this paper proposed a new convolutional block structure which is inspired by Fibonacci sequence, but the effect is not good. In the end, the mF2 values of the models this paper trained achieved good results, which is better than the model of the combined training of ResNets and ordinary U-Net\textsuperscript{34}. Moreover, the training parameters are less than that. This paper believe that this simple and effective training method will bring convenience to researchers in related fields.

CCS Concepts
Computing methodologies → Neural networks

1. INTRODUCTION
Semantic Image Segmentation has always been an important research direction in the field of computer vision. Unlike Classification and Object Detection, it requires accurate classification of each pixel of an image, which has higher requirements for model and parameter adjustment. This paper explores a simple and effective semantic segmentation method based on U-Net [1] structure for the data set which has imbalanced positive and negative samples, so that the other relative image segmentation task can be quickly realized.

Since the emergence of FCN [2], several state-of-the-art models have appeared in the segmentation field, such as Mask R-CNN [3] and U-net[1]. Their structure and principle are different, which also makes them not as good as the original text when using different types of data sets. There is a type of image data, whose target need to be detected is very small relative to the image. And the ratio of foreground and background pixels is often 1:100, which makes the performance of the state-of-the-art
model on this type of dataset is not good as it on COCO [4] or PASCAL VOC [5]. Therefore, for this problem, this paper has found a simple and reliable processing method for this type of dataset in image segmentation.

In this paper, the dataset is in a Kaggle online competition called Airbus Ship Detection, which is a group of satellite images of ships. The training set has a total of about 200,000 pieces, but the number of positive examples is only about 40,000. And the resolution of the image is 768 * 768. The proportion of pixels in the ship is also very low, which makes it not easy to apply a state-of-the-art model for the task. In this work, U-Net34 was used, which is very suitable for image segmentation, to segment the dataset. The model does not perform well, and mIoU only reaches 0.4, as shown in Figure 1. This is due to the class imbalance, which makes many negative seascapes without ships become easy example, and once these pictures are misclassified, they will dominate the loss function. This problem will cause easy examples dominant[6] that makes the segmentation effect greatly reduced.

![Figure 1. Schematic diagram of the segmentation results of the ship dataset.](image)

In contrast, for the problem of positive samples and negative samples imbalance, this paper has added a classification network based on Resnet34 [7]. There are two schemes: (1) Resnet directly uses the U-Net as encoder. This model is hereinafter referred to as U-Net with ResNet34. (2) With Resnet as a separate classification network, combined with the results of the segmentation network, this paper experimented with these two methods respectively. In addition, when building the classification network, this paper made a new attempt to join a new feature map connection, which is inspired by the Fibonacci sequence. Each feature map only connects the last two Layer, but the final result is not as good as the classic ResNets. This paper also made some different attempts at the details of the training, such as the initialization method, the loss function selection, the Dropout regularization, and the U-Net upsampling method. In the end, the effect of the model is significantly improved compared to the classic U-Net34, as shown in Figure 1.

The overview of the rest of the paper is as follows: in chapter 2 this paper will discuss related work in related fields; chapter 3 will detail the methods used and the reasons for their use; chapter 4 will describe the experimental process and show results.

2. RELATED WORK

CNN and Segmentation: In recent years, with the development of convolutional neural networks, lots of state-of-the-art models based on deep convolutional neural networks have emerged, which have achieved good results in the tasks of classification, detection and segmentation. In the field of semantic segmentation, there are also excellent models such as FCN [2] and Mask R-CNN [3].
are three main points for their excellent performance: (1) Convolutional layer due to the appearance of structures such as ResNets [7] The depth of the network can be made very deep, and the dimensions of the picture information that the model can extract are getting larger and larger. (2) The algorithm itself is no longer limited to a direct network path, but special processing and combination of the information extracted by other layers such as shortcut. (3) The appropriate loss function is used to make the convergence of the model better.

**Fully Convolutional Network:** FCN [2] is known as the pioneering work of convolutional neural networks in the field of image segmentation. It removes the fully connected layer that people liked to add to the convolutional layer before, and extracts features by downsampling. Then by upsampling to restore the size of the picture. This so-called "full-convolution network" achieves the pixel-level classification of the image, which also achieved good results in the segmentation task.

**U-Net:** A major innovation of FCN is that it utilizes information from different scales of the image. The segmentation task requires accurate judgment of the location information. Feature Pyramid Networks [8] mentioned that the shallow layer of the deep convolutional network contains more location information, and the depth contains more semantic information. The FCN experiment also shows that FCN-8x with more shallow information has better results in segmentation. Therefore, U-Net, which uses more layers of information, has emerged on the basis of FCN. U-Net is a symmetrical model of the downsampling process and the upsampling process, so that its structure looks like a letter "U", which uses skip-connection to connect the information of the downsampling layer and the upsampling layer in series [1]. This results in better semantic and location information extraction. Therefore, U-Net becomes a very popular baseline in segmentation field.

**Atrous Convolution:** Deeplab proposes a different approach to image segmentation, Atrous Convolution, which is actually Dilated Convolution [9]. The FCN-based segmentation task usually performs a set of downsampling-convolution-upsampling operations on the image. This is because the downsampling can increase the receptive field of the convolutional layer and extract more and deeper information. The resolution of the image is expanded to the size of the original image by upsampling. In general, the upsampling operation is implemented using conv_transpose or unpooling. The author of Atrous Convolution believes that this downsampling-upsampling operation will cause the image to lose a lot of information, so this new "hole-convolution" is used to increase the receptive field without reducing the image size which brings better results.

**Mask R-CNN:** Mask R-CNN [3] is the current leading framework in the field of segmentation, which is based on the framework of the object detection field, Faster R-CNN [10]. Faster R-CNN added the Region Proposal Network (RPN) based on the RoIPool proposed by Fast R-CNN[11] to further strengthen the position of the target anchor box and predict whether there is a object. For segmentation tasks, location information is equally important, so Mask R-CNN proposes a very amazing approach, which predicts a mask in parallel through ROI, and finally merges with mask and box information, so that detection and segmentation are simultaneously implemented. This framework has also achieved wonderful results.

3. METHOD

3.1 Models

**U-Net For Segmentation:** U-Net was used as the basic encoder-decoder model, because this paper used only one-object dataset. U-Net is enough to complete the task, without the need to use more complex instance segmentation like Mask R-CNN. In this dataset, the target ship is small. Due to the imbalance of positive and negative cases, a large number of negative examples actually become easy examples, and their error points will have a greater impact on the entire model. Therefore, for this point of view, this paper compares the effect of the entire data set and the data set with only positive examples, and the results show that the model trained only by the positive data set is better.

**Fine-tuning with ResNets:** At first, this paper used a simple classic U-Net structure, but the effect was average. The mIoU gradually no longer converged when it reached 36.4%. Because the positive
and negative proportions of the data set are extremely unbalanced, the poor effect of the model must mean that a large number of negative cases are predicted as positive examples. So rebuilding U-Net is the way to optimize, using a ResNet34-based classification network as the encoder on the left side of U-Net. Firstly let the model extract the information of "whether there is a ship" to enhance the performance of the model. The overall model is shown in Figure 2.

**ImageNet Pre-training:** In the process of training ResNets, inspired by "Rethinking ImageNet Pre-training [12]", this paper tried two ways: using the pre-training weights of the ImageNet dataset and using He initializer to initialize the weights. ImageNet has been a critical auxiliary task for the computer vision community to progress [12]. The results show that although the final convergence of the two methods is similar, the convergence rate of the model using ImageNet pre-training is four times faster than that of the model without it (see the experimental section for detailed experimental data). The purpose of the training of ResNets is to classify, and "ImageNet pre-training helps less if the target is more sensitive to localization than classification" [12], so it is necessary to use ImageNet.

![Figure 2. U-Net with ResNet34.](image)

Here is an example of the input of a 256x256 resolution picture. Each blue box is connected to an upsampled feature map of the same size, and the black box represents a copy of the blue box at the end of the black arrow. Finally, the sigmoid activation function is used to make the pixel between 0-1, which satisfies the condition that the data set has only one target.

![Figure 3. Convolution block based on Fibonacci sequence theory](image)
3.2 Things This Paper Tried That Didn’t Work

Since AlexNets[13], convolutional neural networks have been made deeper and deeper. One of the most important methods is Shortcut [7] proposed by ResNets. Some networks also use such as those used in GoogleNet [14]: The method of channel concatenate. In short, it is to use more layers to extract the information. DenseNet is a model with good classification [15]. Each convolution layer in each block is connected to the extraction result of all convolutional layers in this block, which is also an important reason for its high performance. But there is a problem with this, that is, the memory usage is too large. Therefore, this paper thinks about whether each convolution layer really needs information extracted by all other layers? This paper borrowed from the idea of the Fibonacci sequence, assuming that each convolutional layer might only need to extract two layers of information, and in the end the network can reach the appropriate output. This paper tried each convolutional layer to connect only the output of the first two layers. The schematic diagram of a block is shown in Figure 3, and ResNets is reconstructed, but the experimental results are not as good as the original ResNets. It may be because "two layers of information" does not refer to the first two layers of information, but may also be two kinds of "forms" of information, such as "content and style." It is not very convincing, but this paper will continue to study.

3.3 Implement Details

Convolution Block: This paper uses Batch normalization [16] + Relu + Conv as a basic block, as shown in Figure 4. Batch normalization is very helpful for the convergence of the model. When Batch normalization is not used, the convergence speed is very fast at the beginning of training, but after convergence to a certain extent, there is a shock, and this situation is improved after adding Batch normalization. This paper also uses Xavier [17] initialization and He[18] initialization to test. The results show that He uniform initialization method is more suitable for convolution module with Rectified Linear Unit (RELU) as the activation function.

![Convolution Block Diagram](image_url)

Up Sample and Transposed Convolution: In the U-Net upsampling process, this paper tried the UpSampling2D and Conv2DTranspose functions in Keras. They correspond to the upsampling method with no parameter repeating field values and the transposed convolution method with parameters. The Conv2DTranspose method produces a large number of trainable parameters, which undoubtedly adds a lot of burden to the training of the model. After the training with the same conditions, the transposition convolution is not better than the normal upsampling, only the mIoU of 0.02 is improved.
The reason is that the ratio of the data set's target to the whole picture is too small, so that the transposition convolution can not restore the picture information more clearly. Therefore, using ordinary upsampling under this data set is more worthy.

**Multi Scale Training:** Inspired by the failure of transposed convolution to achieve good results, this paper realized that using a native resolution for training on a data set with a highly unbalanced background and background ratio may not work well. Because the target is too small, it is difficult to locate the target at high resolution. In fact, when the model was firstly trained at 768*768 resolution, it converges very slowly and eventually does not converge to a workable point. When added a 3 stride to the drawing (ie, reduced the resolution to 256*256), the training effect was greatly improved, and it achieved good results by only 5 rounds of iterations. Therefore, this paper believes that in the case of low resolution, the model is easier to locate small targets. At the beginning, this paper chose to use a smaller resolution to train, so that the model has the ability to locate the target. When the model achieves a certain bottleneck, the large-resolution training is performed to make the model more suitable for the real data set. This method is effective, and continuing training at 768*768 resolution increases mIoU by 3 percentage points.

**Loss Function:** In a particular task, it is often important to choose a suitable loss function. When training the ResNet classification model, this paper used Binary Cross Entropy. When training U-Net, IoU (Interaction over Union) was tried firstly. The definition of IoU is as follows:

\[
\text{IoU} = \frac{\text{DetectionResult} \cap \text{GroundTruth}}{\text{DetectionResult} \cup \text{GroundTruth}}
\]  
(1)

The mIoU that averages the IoU of all samples is a good reflection of the overlap between Detection Result and Ground Truth. This paper also tried Dice coefficient, which is an evaluation criterion that reflects the similarity of sets. Unlike IoU, it focuses more on overlapping regions of two sets, which are defined as follows:

\[
\text{Dice} = \frac{2|\text{DetectionResult} \cap \text{GroundTruth}|}{|\text{DetectionResult}| + |\text{GroundTruth}|}
\]  
(2)

Finally, this paper also tried Focal Loss[6]. Focal Loss was originally designed to solve the problem of easy examples dominant, which will give hard examples a higher weight, so that the model is more focused on the identification of hard examples. After the experiment, mIoU works best, see section 4.3 Loss Function.

**Optimize Strategy:** In the optimization strategy, Adam algorithm with good performance was used as the optimizer. At the same time, this paper used SGDR (Stochastic Gradient Descent with Warm Restarts [19]), and set the learning rate back to the initial value after each cycle of complete training. This method does effectively improve the effect of convergence and solves the local optimal problem that plagues us. In addition, discriminative learning rate was used to make the shallow layer of the model have a smaller learning rate, and the deeper layer of the model has a higher learning rate. This is because the shallow layer of the model often extracts coarser information such as contour information [20], which determines the shape and position of the target, so a smaller learning rate is required to maintain the stability of the information. In addition, with the widespread use of Batch normalization, its regularization effect makes Dropout no longer necessary. But after used Dropout following the full connection layer of ResNets, it worked. This proves that when model is more complex, Dropout’s sparsity can still have a good regular effect.

**Joint Outcome:** Using the classification network as a U-Net encoder is an effective method. In addition, training a ResNet and a U-Net34 separately, and then combining their results is also a way to improve performance. The structure is shown in Figure 5. Intuitively, this method reduces the large number of False Negative examples predicted by U-Net and improves the performance of the model. This paper tried both the above methods and compared the pros and cons. Experiments show that Joint
Outcome only works when the overhead of training the classification network is small. The specific experimental results are shown in Section 4.4.

![Diagram of Sigmoid unit](image)

**Figure 5. Joint Outcome.**

Passes the output of ResNet34 through a Sigmoid unit, limiting output y to 0 and 1. If y=1, the segmentation result uses the output result of U-Net34; if y=0, the segmentation result is set to none.

| Model                              | Val_acc | Val_precision | Val_recall |
|------------------------------------|---------|---------------|------------|
| ResNet34 without Dropout           | 0.944   | 0.901         | 0.833      |
| ResNet34 with Dropout              | 0.965   | 0.927         | 0.902      |
| With ImageNet pre training(1 epoch)| 0.94    | -             | -          |
| With ImageNet pre training(3 epochs)| 0.976  | -             | -          |

**Table 1. The effect of Dropout and ImageNet pre-training.**

4. **EXPERIMENTS**

4.1 **Training ResNets**

**ImageNet Pre-training:** This paper respectively used the weights pre-trained on the ImageNet dataset and the weights without it to train. In the experiment, when using He initialization, the size of the mini-batch was set to 64, and Binary Cross Entropy was used as the loss function. After training four rounds on the entire data set, the accuracy reached 0.944. Similarly, using a 64-size mini-batch and loading the pre-trained weights on ImageNet, the model only trained on the entire data set for one epoch, and the accuracy reached 0.94 as shown in Table 1. Therefore, pre-training on ImageNet followed by Fine-tuning is necessary when classifying tasks with small objection.

**Dropout:** Since the advent of the Batch Normalization[16] algorithm, it has achieved regularization effects because it has many hyper-parameter calculations on each mini-batch, and it is also effective after being used in the convolutional layer. Therefore, some people think that Dropout can be replaced by Batch Normalization. However, this paper found that was not right. By adding Dropout after the full connection layer of ResNets, the accuracy increased from 0.944 to 0.96. It turns out that Dropout still has an indispensable sparseness and regularization effect for dense layers like the fully connected layer.

4.2 **Training U-Net**

**Evaluation Criteria:** This paper used two performance evaluation indicators, mIoU and the average value of F2 based on different IoU thresholds, which is also the official scoring method used by the
gam. See F2 value in Equation(3), where t is the the standard IoU threshold to judge whether the example is positive. mF2 (mean F 2 score) was used as the final performance evaluation index, which is shown in Equation(4).

\[ F_2(t) = \frac{5TP(t)}{5TP(t) + 4FN(t) + FP(t)} \]  
\[ mF2 = \frac{1}{|t|} \sum_{t} F_2(t) \]  

Drop negative examples: During the initial training, this paper tried to train the classic U-Net34 with all data sets. However, when mIoU converges to 0.3, it begins to oscillate. When all negative examples were eliminated, mIoU began to oscillate until it converges to 0.45. The experimental results are shown in Table 2. This is because in such a data set where the positive and negative samples and the image foreground and background are unbalanced, a lot of negative examples have become easy examples. And their misclassification has a great influence on the loss function, which easily causes the problem of examples dominant [6]. Therefore, under this data set, training the segmentation model using only positive examples is a very effective means.

Table 2. Pre- and post-effects of removing negative examples.

| Training images                  | mIoU |
|----------------------------------|------|
| Positive(40k)+Negative(150k)    | 0.3  |
| Positive(40k) only              | 0.45 |

Learning Rate Scheduling: Using the usual learning rate decay strategy is not very effective in practical experiments. The Fast-ai library based on Pytorch can easily achieve distributive learning rates. This paper used learning rates of ten times increment in the shallow, middle, and head layers of the model, and achieved good results. This also confirms that because the location and contour information extracted by the shallow network is more, it is not suitable for too much transformation [20]. The deeper the layer, the more semantic information is extracted, and the back propagation[21] needs bigger learning rate. In addition, SGDR[19] was also used, each two epoches restarted a learning rate. Using this method, based on the use of pre-trained ResNets, the model was only trained for one round, and mIoU reached 0.69.

Loss Function: This paper tried three loss functions, focal loss[6], dice coefficient, and IoU. Among them, the use of IoU as a loss function is slightly better than the use of dice coefficient, only 0.01. But when focal loss was used, mIoU decreased by 0.02. There are two reasons for analysis: (1) because mIoU is more sensitive to the loss function of IoU; (2) because of the effect of ResNets and the removal of negative examples in training, the model has good enough classification performance for positive and negative images. So focal loss is hard to help more in the easy examples.

Other Details: This paper used two kinds of upsampling methods: Upsample without parameters and Trans-posed convolution with it. The number of mini-batch is set to 16, the epoch step is set to 100, and the training is 60 rounds. Finally, Upsample is more than the mIoU value of Transposed convolution. It is 0.02 lower, but the mF2 value is 0.03 higher. This shows that Upsample is more advantageous under different thresholds of IoU. Plus its no-parameter characteristics, this paper think it would be better to use Upsample on such a small data set.

This experiment used multi-scale training, first training 2 epochs on 256*256 resolution images, and the mIoU reached 0.69. Then the resolution was increased to 384*384, and trained for 10 epochs, and mIoU increased to 0.79. Finally, this experiment used a picture with a resolution of 768*768, mIoU returns to 0.76. But since the model has been adapted to large resolution images at this time, this paper believes that the decline of mIoU is not equivalent to the performance degradation of the model.
After the final submission, the mF2 value reached 0.83, and the results are shown in Table 3. Therefore, multi-scale training has a significant improvement in the generalization ability of the model.

| Resolution | mIoU |
|------------|------|
| 256 x 256(stride=3) | 0.693 |
| 384 x 384(stride=2) | 0.796 |
| 768 x 768(stride=1) | 0.762 |

4.3 Comparison To Joint Outcome
This paper also tried to train a simple U-Net34 model with few parameters and then combined with ResNets to give the final prediction results. Using this method, the mF2 value has increased from 0.66 for U-Net to 0.77. Compared with the more complex U-Net using pre-trained ResNets, it is 6 percentage points lower but the training parameters are higher than U-Net with ResNet34. The result is shown in Table 4. Therefore, when the cost of training the classification network is similar to that of training U-Net with ResNet34, the method of separately training the classification network and combining it with the segmentation network is not desirable in terms of computational efficiency or prediction accuracy.

| Output                  | mF2 | Trainable parameters |
|-------------------------|-----|----------------------|
| U-Net34                 | 0.66| 492845               |
| Joint outcome           | 0.77| 13993985             |
| U-Net with ResNet34     | 0.83| 14209665             |

5. CONCLUSION
In this work, based on the Airbus ship detection challenge dataset in Kaggle, this paper finds an effective method for image segmentation tasks with positive and negative samples and extreme imbalance of image foreground and background as follows:
- Use U-Net with the classification network as the encoder.
- Use the full data set when training the classification network and only use positive examples when training U-Net.
- The upsampling process uses the most basic parameter-less up-sampling method, while using Batch Normalization, and uses Dropout after the fully connected layer.
- Use multi-scale training methods to increase image resolution from small to large and try different loss functions.
- The learning rate is selected using distributive learning rate, SG-DR, etc.

It has been proved by experiments that the U-Net with ResNet34 trained by the above method is improved by 17% compared with the ordinary U-Net, and the model combined with the results of U-Net34 and ResNet34 is improved by 6%, and trained for 13 epochs on only three scales of images, which took 12 hours on the Tesla K80 GPU. The improved U-Net model has a significant effect on the image segmentation task of this imbalanced training set. In practical applications, more accurate results can be obtained faster, and related research topics can be realized more quickly.
REFERENCES

[1] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[2] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, October). Mask r-cnn. In Computer Vision (ICCV), 2017 IEEE International Conference on (pp. 2980-2988). IEEE.

[4] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[5] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

[6] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence.

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[8] Lin, T. Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., & Belongie, S. J. (2017, July). Feature Pyramid Networks for Object Detection. In CVPR (Vol. 1, No. 2, p. 4).

[9] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.

[10] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[11] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[12] He, K., Girshick, R., & Dollár, P. (2018). Rethinking ImageNet Pre-training. arXiv preprint arXiv:1811.08883.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[14] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[15] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017, July). Densely connected convolutional networks. In CVPR (Vol. 1, No. 2, p. 3).

[16] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[17] Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

[18] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

[19] Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.
[20] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

[21] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533.