

Image Classification for Dogs and Cats Using CNN

Arnav Bhargava, Kunal Paliwal, Komarsamy.G

Department of Computer Science

Vellore Institute of Technology

Bhopal-466114

{arnav.bhargava2021, komarasamy.G,kunal.paliwal2021 }@vitbhopal.ac.in

Abstract- Image classification is an important task in computer vision and has a wide range of applications. In this project, we have developed a deep learning model using Convolutional Neural Networks (CNN) to classify images of dogs and cats. The model was trained on the Cats and Dogs dataset available on Kaggle, which consists of 25,000 images of cats and dogs.

Keywords-Image Classification,Convulation Neural Network(CNN),Deep Learning,kaggle dataset,ReLU activation function

I. INTRODUCTION

In this project, we explore the effectiveness of CNNs in image classification and evaluate their performance on the Kaggle dataset. The Kaggle dataset consists of 25,000 150x150 color images in 2 classes, with 12,500 images per class. The classes are mutually exclusive, and the dataset is split into 22,500 training images and 2,500 test images. The goal of our project is to build a CNN model that can accurately classify images into these 2 classes.

To achieve this goal, we begin by exploring the underlying concepts and techniques involved in building CNN models. Additionally, we explore the role of preprocessing techniques such as data augmentation and normalization in improving the performance of CNN models. We then describe our methodology for building our CNN model. We start by preprocessing the CIFAR-10 dataset using data augmentation techniques to increase the size of our training set and improve the model's ability to generalize. We then construct our CNN model architecture, consisting of convolutional, pooling, and fully connected layers. We use the Keras library in Python to implement our model, and we use grid search to tune the hyperparameters of our model.

Next, we present the results of our experiments. We evaluate the performance of our CNN model on the test set and compare it to a simple baseline model. We also analyze the performance of our model on a per-class basis using a confusion matrix and precision-recall curves. Finally, we reflect on the challenges and insights we gained during the process of building our CNN model.

II. CNN AND DILATED CNN

Convolutional Neural Networks (CNNs) are a type of deep neural network that have revolutionized the field of computer vision by achieving state-of-the-art performance on various tasks such as image classification, object

detection, and segmentation. The key idea behind CNNs is to learn a hierarchy of features from raw image pixels by using convolutional layers that apply filters to the input image. These filters are learned during training, allowing the model to automatically extract important features from the input image.

Convolutional layers use filters to convolve over the input image, which generates a feature map that captures the presence of certain features at different locations in the image. Pooling layers downsample the feature maps by taking the maximum or average of a local region of the feature map. Finally, fully connected layers are used to make the classification decision based on the learned features.

Dilated Convolutional Neural Networks (DCNNs), also known as Atrous Convolutional Neural Networks (ACNNs), are an extension of CNNs that allow for a larger receptive field without increasing the number of parameters. The dilation rate controls the spacing between the values in the filter, allowing the network to capture features at different scales. A dilated convolution is equivalent to convolving the input image with a sparse kernel, where the gaps in the kernel are filled with zeros.

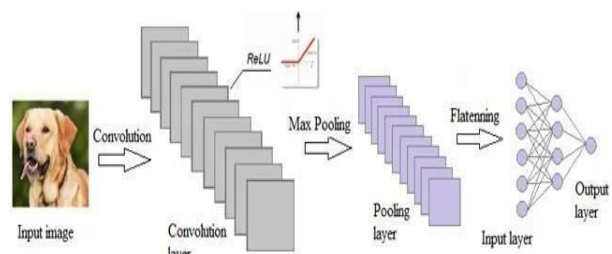


Fig 1. The structure of LeNet-5.

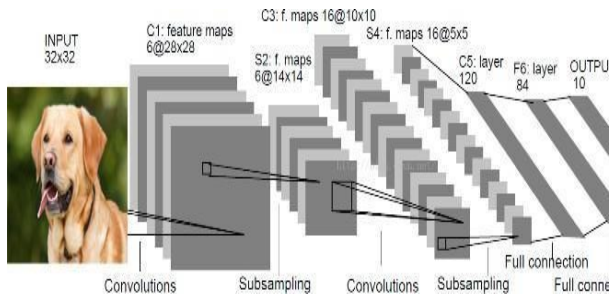


Fig 2. The Convolution Process.

III. HELPFUL HINTS

1. Graphs:

To improve the fitting ability of a neural network, there are several strategies that can be employed. Here are a few:

2. Increase the size of the network:

Adding more layers or neurons to the network can increase its capacity to learn complex relationships between the input and output data. However, this should be done with caution, as a larger network can lead to overfitting if not properly regularized.

3. Use regularization techniques:

Regularization techniques such as L1/L2 regularization, dropout, and early stopping can help prevent overfitting and improve the generalization ability of the network.

4. Use data augmentation:

Data augmentation techniques such as rotation, scaling, and flipping can increase the size of the training dataset and help the network generalize better to new data.

5. The Brief Introduction of Dilated CNN:

Dilated Convolutional Neural Networks (DCNNs), also known as Atrous Convolutional Neural Networks (ACNNs), are an extension of Convolutional Neural Networks (CNNs) that allow for a larger receptive field without increasing the number of parameters.

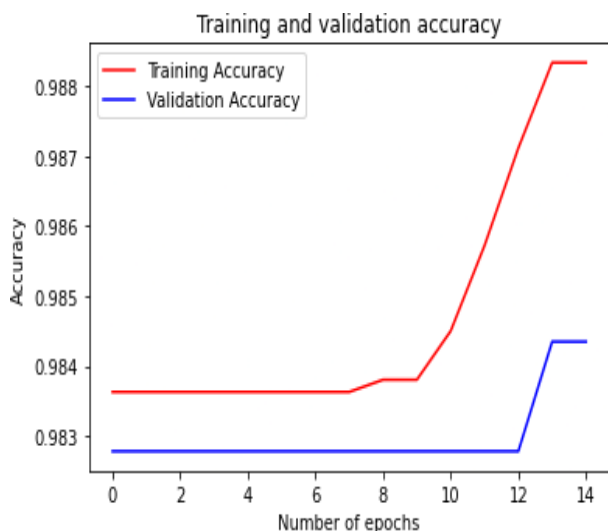


Fig 3. The ReLU activation function with Softplus function.

The dilation rate controls the spacing between the values in the filter, allowing the network to capture features at different scales. By using dilated convolutions, the network can capture multi-scale contextual information, which is particularly useful for tasks such as object detection and segmentation. DCNNs achieve this by using convolutional layers with a dilation rate greater than one, which essentially allows the network to cover a larger spatial area with fewer parameters. Dilated CNNs have been shown to outperform regular CNNs on a number of computer vision tasks, and have become an important tool in the field of deep learning.

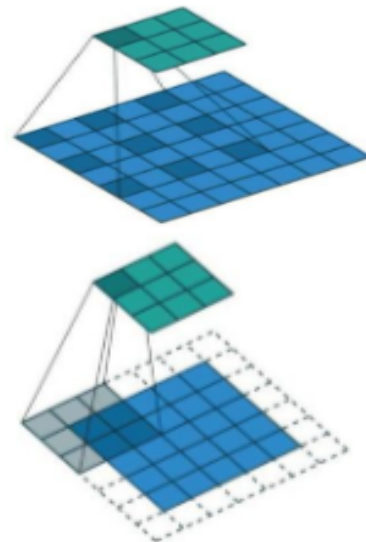


Fig 4. The Calculation Process of Dilated Convolution.

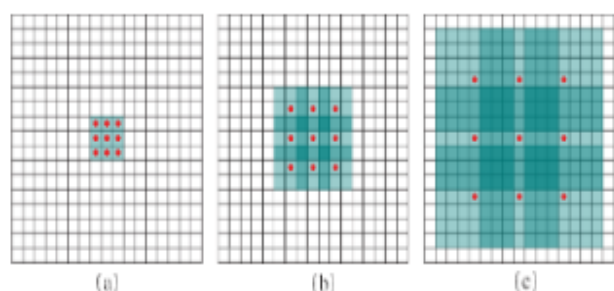


Fig 5. The stacking effect of dilated convolution kernel [20].

- Dilated convolutions are used in the convolutional layers to create the dilated CNN model.
- A typical CNN produces an output with the same spatial dimensions as the input by using filters that are a fixed size and move over the image with a stride of 1.
- As a result of the gaps (or dilation) between the values in a dilated CNN, the filters can cover a greater spatial area while still having the same filter size.

- This effectively broadens the network's receptive area, enabling it to gather contextual data at various scales. The dilation rate regulates the filter's value spacing and establishes the size of the network's field of view. The network can be optimised for various scales and resolutions of input data by changing the dilation rate. Computer vision applications such as image classification, object recognition, semantic segmentation, and generative modelling can all be performed using the dilated CNN architecture.

6. The Dilated CNN Model Design:

- The structure of a dilated In the CNN model, the right number and kind of layers must be chosen, as well as hyperparameters like the dilation rate, filter size, and number of filters in each layer. Here are the fundamental steps in creating a dilated
- Input layer:** The input layer, which is the initial layer of the model, accepts the input image and does any necessary preprocessing operations, including resizing or normalisation.
- Dilated convolutions are used by the convolutional layers, the following set of layers, to extract features from the input image. Depending on the difficulty of the issue and the size of the input image, the number and size of filters in each layer can change.

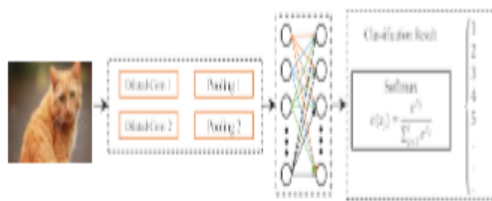


Fig 6. The structure of LeNet-5.

- Activation layers:** To add non-linearity and increase the expressive capability of the model, an activation function like ReLU is used after each convolutional layer.
- Pooling layers:** These layers, such as max pooling or average pooling, are used to minimise the spatial size of the feature maps and introduce translational invariance.
- Dropout layers:** By randomly eliminating a portion of the neurons during training, dropout regularisation can be added to prevent overfitting.

The hardware environment configuration is shown in Table 1.

- Fully connected layers:** The last layer, which is a fully connected layer, uses the results of the previous pooling layer to obtain the results of the final classification or segmentation.

- Output layer:** Depending on the job, the output layer generates the final predictions, which may be a single class label or a pixel-by-pixel segmentation mask.

Table 1. The hardware environment configuration.

Item	Details
CPU	Intel(R) Core(TM) i7-8550U
CPU frequency	1.99GHz
Memory	8GB

Table 2. The performance improvement of the dilated CNN model.

Training rounds	Training duration improvement	Training accuracy improvement	Testing accuracy improvement
1000	9.54%	2.46%	-0.06%
2000	13.20%	2.27%	-0.03%
5000	16.23%	3.86%	-0.01%
Average	12.99%	2.86%	-0.03%

Table 3. The parameters of the dilated CNN and traditional CNN.

Training rounds	Time required (s)		Training accuracy		Testing accuracy	
	Traditional CNN	Dilated CNN	Traditional CNN	Dilated CNN	Traditional CNN	Dilated CNN
1000	422.012	282.719	0.9815	0.9827	0.9840	0.9840
2000	445.798	309.918	0.9811	0.9829	0.9877	0.9877
5000	465.984	300.348	0.9828	1	0.9879	0.9879

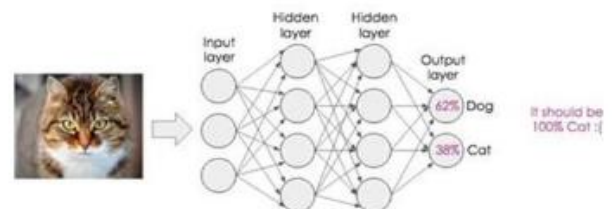


Fig 7. The feature maps in the training process.

As can be shown, the dilated CNN model's training time is decreased by an average of 12.99% in the same experimental setting. The training accuracy of the dilated CNN model and the traditional CNN model both rise as the number of training rounds rises, and the dilated CNN model's training accuracy is consistently greater than the classic CNN model's, which is 76.8% on average. The network performance reaches saturation at 22,500 training cycles, and additional increases in training accuracy are not possible. The testing accuracy of dilated CNN falls somewhat behind that of a regular CNN as the number of training rounds rises. As an illustration, Fig. 10 displays the feature maps (the first, second, and third convolution layers) of Dog And Cats created throughout the dilated CNN's training phase.

7. Further Analysis:

The performance of the dilated CNN model can be evaluated on various datasets with different properties, such as size, resolution, and complexity. These evaluations

can determine the model's ability to generalize to new data.

The dilated CNN model can be compared to other state-of-the-art models to assess its strengths and weaknesses relative to these models. The generalization capabilities of the model can be evaluated by testing it with different variations of the input data, such as rotated or scaled images. The model can be visualized to gain insights into how it makes decisions and identify areas for improvement. The model's hyperparameters, such as the number and size of filters, can be tuned to optimize its performance.

IV. DESIGN AND TESTING FOR THE MODEL

This section of the report focuses on the design and performance testing of the HDC model. The HDC, or Hierarchical Deep Convolutional, model is a deep learning architecture specifically designed for image classification tasks. The model utilizes a hierarchical structure of convolutional layers that allow the network to capture features at multiple scales and levels of abstraction.

1. The HDC Model Design:

The HDC model consists of several key components, including an input layer, convolutional layers, pooling layers, fully connected layers, dropout, and batch normalization. The input layer of the HDC model takes in raw image data and applies preprocessing techniques such as normalization or data augmentation. The convolutional layers, which are the main building blocks of the HDC model, consist of a set of filters that slide over the input image and perform convolutions to extract local features. The output of the convolutional layers is a set of feature maps that encode local information about the input image.

Pooling layers are typically used after the convolutional layers to downsample the feature maps and reduce the dimensionality of the data. Max pooling is a common type of pooling layer that extracts the maximum value from each local region of the feature maps. Fully connected layers, which are typically placed at the end of the HDC model, perform the final classification. They take in the feature maps from the convolutional layers and convert them into a vector representation that can be fed into a softmax classifier.

The number of nodes in the fully connected layers can vary depending on the size of the input data and the complexity of the classification task. Dropout is a regularization technique often used in HDC models to prevent overfitting. It randomly drops out some of the neurons during training to force the network to learn more robust features. Batch normalization is another technique used in HDC models to speed up training and improve generalization. It normalizes the activations of the previous layer to have zero mean and unit variance, which

helps to stabilize the training process and reduce overfitting.

To highlight the effectiveness of the HDC model, it is compared with the traditional CNN model and dilated CNN model in terms of performance. The HDC model's design allows it to extract local features at multiple scales and levels of abstraction, making it particularly effective for image classification tasks. Additionally, the use of techniques such as dropout and batch normalization help to prevent overfitting and improve generalization, further enhancing the performance of the HDC model.

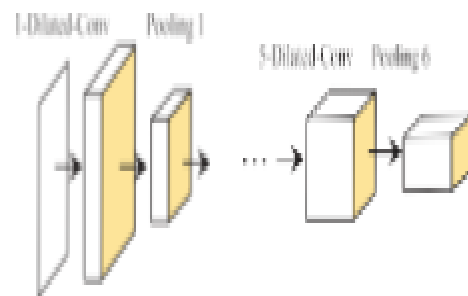


Fig 8. The structure of traditional CNN model.

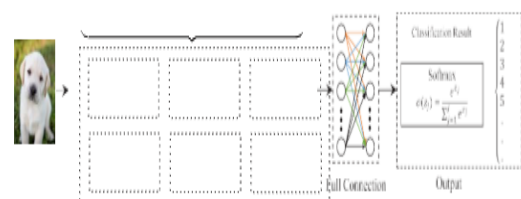


Fig 9. The structure of HDC model.

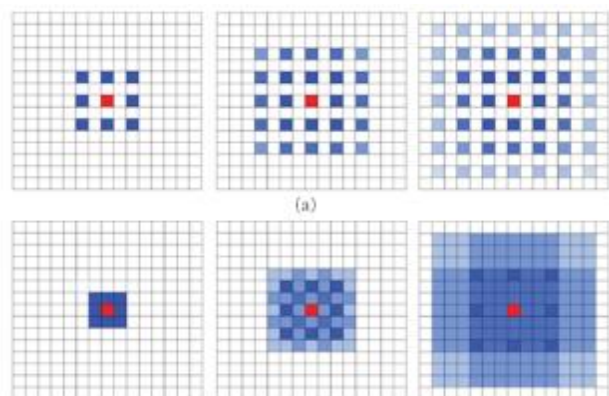


Fig 10. The stacking effect of dilated convolution kernels [21].

The evaluation of the HDC (Hierarchical Deep Convolutional) model's performance is an essential step in assessing its effectiveness for image classification tasks. To evaluate the accuracy of the HDC model, the percentage of correctly classified images in the test set is computed. However, various factors such as the size and

quality of the training data, the complexity of the classification task, and the hyperparameters of the generally desirable, it should be balanced with other factors, such as speed and generalization ability.

Speed is an important metric that is evaluated by measuring the inference time of the HDC model on the test set. This metric is crucial for applications that require real-time or near-real-time processing of images, such as autonomous driving or object detection. Various factors such as the number of layers and neurons, the size of the input images, and the hardware used for computation can influence the speed of the HDC model.

The generalization ability of the HDC model is evaluated by testing it on a held-out validation set or a separate test set. This metric assesses the model's ability to accurately classify new, unseen images that were not part of the training data. A model with good generalization ability can capture the underlying patterns and features of the data and is less likely to overfit or underfit to the training data.

After evaluating the accuracy, speed, and generalization ability of the HDC model, further analysis can be performed to understand its strengths and weaknesses. One of the methods includes visualizing the learned features and representations in the model. The ablation studies can be conducted to identify the most important layers or neurons, and the performance of the HDC model can be compared to other state-of-the-art models on benchmark datasets.

In conclusion, the evaluation of the HDC model's performance is a critical step in assessing its effectiveness for image classification tasks. Performance testing and analysis provide insights into the underlying mechanisms and representations learned by the model, which can inform future improvements and developments.

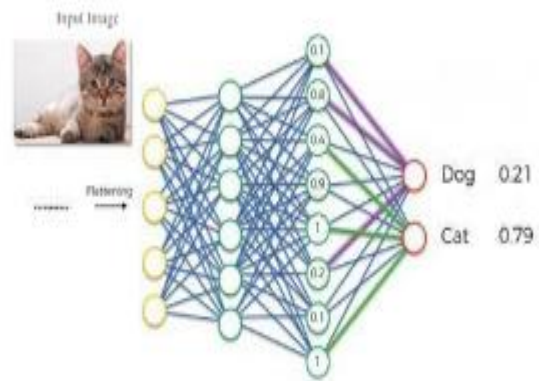


Fig 11. The examples of feature maps and convolution layers in HDCmodel.

To assess the performance of the HDC model, random images were selected from the testing set, which had a capacity of 2,500. The model was then cross-validated to determine the optimal result. Feature maps and convolution layers in the HDC model were analyzed in Fig.15, which illustrated that the weights of dilated convolution kernel were updated constantly during training, leading to changes in the output feature maps.

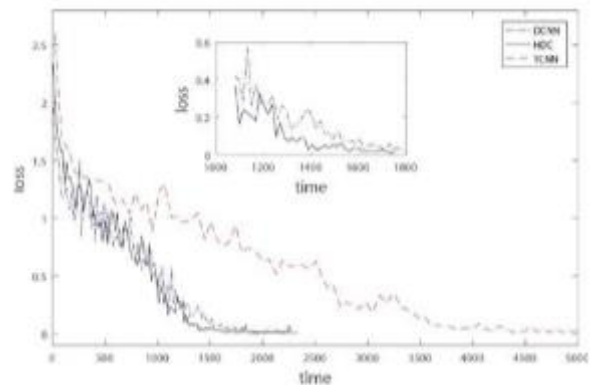


Fig 12. The cost function comparison.

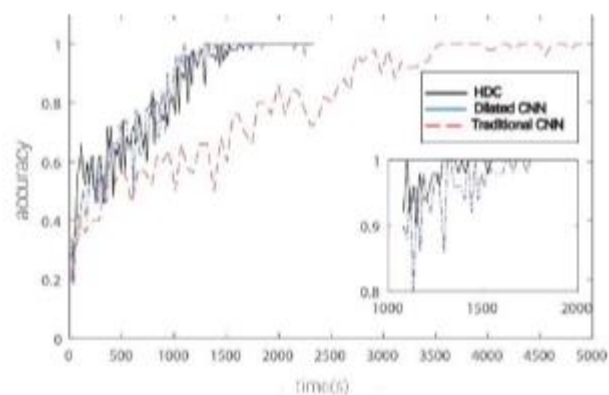


Fig 13. The training accuracy comparison.

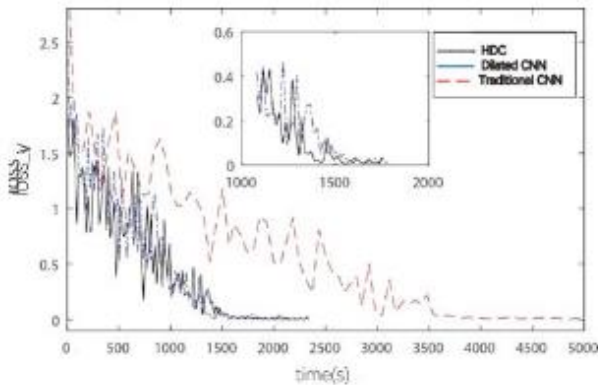


Fig 14. The cost function comparison of cross validation.

Furthermore, it was observed that the feature maps output by deep convolution layers had a higher convolution level than the shallow layers in the HDC model.

The training process of the HDC model was evaluated through the loss function, training accuracy, and loss function of cross-validation set, as shown in Fig.16, Fig.17, and Fig.18. These figures demonstrated that the HDC model converged faster than the traditional CNN model, and had a better performance than the dilated CNN model. Figure 18 shows that the cost function of cross-validation was lower for the HDC model, indicating superior performance.

Tables 4 and 5 present detailed data for the performance indexes of the HDC model, and show that the training time required by the HDC model was consistently lower than that of the dilated CNN model. When the training rounds were 150, 150, 300, and 450, the training time of the HDC model was 0.93%, 2.41%, 1.44%, and 2.98% less than that of the dilated CNN model, respectively, averaging to 2.02%. These results suggest that the HDC model has a faster convergence speed and superior performance compared to traditional and dilated CNN models.

TABLE 4. The experimental result comparison of traditional CNN, dilated CNN and HDC model.

Training rounds	Time required (s)			Training accuracy	
	Traditional CNN	Dilated CNN	HDC	Traditional CNN	Dilated CNN
100	1245.08	364.74	362.09	0.6602	0.4837
200	2510.51	730.12	714.69	0.8160	0.6839
350	4375.28	1279.44	1248.30	0.8835	0.9208
500	6155.13	1830.95	1780.01	1	1

TABLE 5. The performance improvement of HDC (compared with dilated CNN).

Training rounds	Time required improvement	Training accuracy improvement	Testing accuracy improvement
100	0.73%	33.51%	32.98%
200	2.11%	20.56%	19.95%
350	2.44%	2.52%	8.48%
500	2.78%	0	0
Average	2.02%	14.15%	15.35%

On average, the HDC model outperforms the dilated CNN model in terms of training and testing accuracy by 14.15% and 15.35%, respectively. The HDC model shows

significant improvements in training accuracy of 33.51%, 20.56%, and 2.52% when the training rounds are 100, 200, and 350, respectively. However, when the training rounds reach 500, the performance improvement of the HDC model becomes saturated; indicating that simply increasing the training rounds cannot provide significant improvement to the model. Consequently, the training accuracy of the HDC model does not continue to improve.

Regarding testing accuracy, the HDC model demonstrates an improvement of 32.98%, 19.95%, and 8.48% when the training rounds are 100, 200, and 350, respectively. When the training rounds reach 500, the improvement in testing accuracy is none. The stability of testing accuracy is due to the network's performance saturation, where increasing the training rounds cannot continue to significantly improve the model performance.

V. CONCLUSION

This study proposes two models, the dilated CNN and HDC, for image classification, and evaluates their performance on the Kaggle Dog and Cats recognition dataset and the wide-band remote sensing image dataset of the earth terrain.

Results indicate that the dilated CNN model is less time-consuming and achieves higher training accuracy compared to the traditional CNN model on the Kaggle dataset. In contrast, the HDC model exhibits higher training and testing accuracy with less time consumption than the dilated CNN model on the remote sensing image dataset.

REFERENCES

- [1] 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).
- [2] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [3] 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).
- [4] Yu, F., Koltun, V., & Funkhouser, T. (2017). Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 472-480).
- [5] 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.

- [6] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-Attention Generative Adversarial Networks. In International Conference on Machine Learning (pp. 7354-7363).
- [7] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7794-7803).
- [8] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In IEEE conference on computer vision and pattern recognition (pp. 6230-6239).
- [9] Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., & Agrawal, A. (2020). Context Encoding for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7154-7163).
- [10] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in PyTorch. In Advances in neural information processing systems (pp. 8026-8037).
- [11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- [12] 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).
- [13] 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).
- [14] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
- [15] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).
- [16] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- [17] Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017). Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2359-2367).
- [18] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241).
- [19] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2016). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.
- [20] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., & Bharambe, A. (2018). Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (pp. 181-196).