Report of Final Project on Residual Attention Networks for Image Classification

Xin Huang
Department of Electrical Engineering, Columbia University
New York, USA
xh2510@columbia.edu

Qimeng Tao
Department of Electrical Engineering, Columbia University
New York, USA
qt2139@columbia.edu

Kangrui Li
Department of Electrical Engineering, Columbia University
New York, USA
kl3350@columbia.edu

***Abstract*—In this project, we re-implement the work in paper: Residual Attention Networks for Image Classification, we dived into F. Wang and other scholars' work and focused on the attention module primarily. By designing a model with shortcut, we successfully tested the residual unit's functionalities in the architecture. As a result of our implementation, we made very close results comparing to the original paper.**

***Index Terms*—Deep Learning, Residual Attention Network, Attention Mechanism, ResNet**

I. Introduction

In recent years, scholars have made important breakthroughs and achievements in the field of deep learning. In the field of computer vision, image recognition and natural language processing, deep learning neural models have achieved great success. With a series of sequence models such as CNN and RNN [1], and the proposal of AlexNet [2], VGG [3], ResNet [4] and a series of networks and frameworks, deep learning ushered in the culmination of research and application.

Sometimes a small but unique feature could catch people's attention. Extracting features of large images can dramatically decrease computationally expense in convolutional neural networks, and the nature of extracted attention has been studied in previous work [5]. Attention is found to be capable of not only selecting a focus position but improving various representations of objects at that position as well. Besides formulating attention drift to capture different attention aspects, another advance to enhance feedforward convolutional neural networks is to use 'very deep' structure.

To achieve better performance in image classification, a combined method of both attention mechanism and deep residual network [4] was proposed by the authors of the original paper. Residual network introduced 'shortcut connection' to traditional convolution network to prevent the gradient disappearance problem and solve the degradation problem that the training error rate and test error rate increase as the number of layers increases. Based on this, the authors designed a 'Residual Attention Network' in which attention-aware features can change adaptively as layers going deeper and this network can be easily scaled up to hundreds of layers.

Motivated by the aforementioned work, in our project, our objective is to successfully achieve image classification performance based on proposed attention module and residual blocks in the original paper. We plan to construct different Attention Residual Network and evaluate performance by making following changes: (1) Number of Attention Modules, it is said increasing attention modules could lead to consistent performance improvement; (2) Depth of Network, since deep residual network is used in our constructed network, shortcut connection would perform better as depth increases; (3) Data Augmentation, as learnt from lectures, we plan to add noise level or do flip and rotation process to original dataset.

For all the augmented and original datasets, several built network would be used to implement classification so there are lots of comparison experiments in our project. We also intend to test the performance of residual network so we would try to replace residual unit in the original paper by normal convolution unit with no 'shortcut connection'.

Obviously, we could encounter some technical challenges. The very first difficulty is that we cannot achieve acceptable accuracy while training and this could happen due to several reasons: (1) depth of network is not enough, generally number of attention modules and residual units should be increased so that attention-aware feature could be extracted better and receptive field of our network would increase; (2) lack of some layers, normalization and regularization process should be included in our network to make training process more generalized for different data; (3) data augmentation is not proper, sometimes data is slightly rotated or translated but sometimes augmentation degree would be too big so that dataset is contaminated which makes training meaningless; (4) hyperparameter is not set correctly, this can be addressed by several times of training by adjusting optimizer type, batch size, number of epoch, loss function type and learning rate. Another difficulty we may face is insufficient computing power because of GPU performance and memory is also insufficient if dataset is too big. This can be fixed by changing GPU in our virtual machine and dataset should be scaled to be smaller. Also, we may find our accuracy results are not as good as the original paper due to slight difference of constructed Attention Network or preprocessing to dataset and these issues can be solved by multiple experiments.

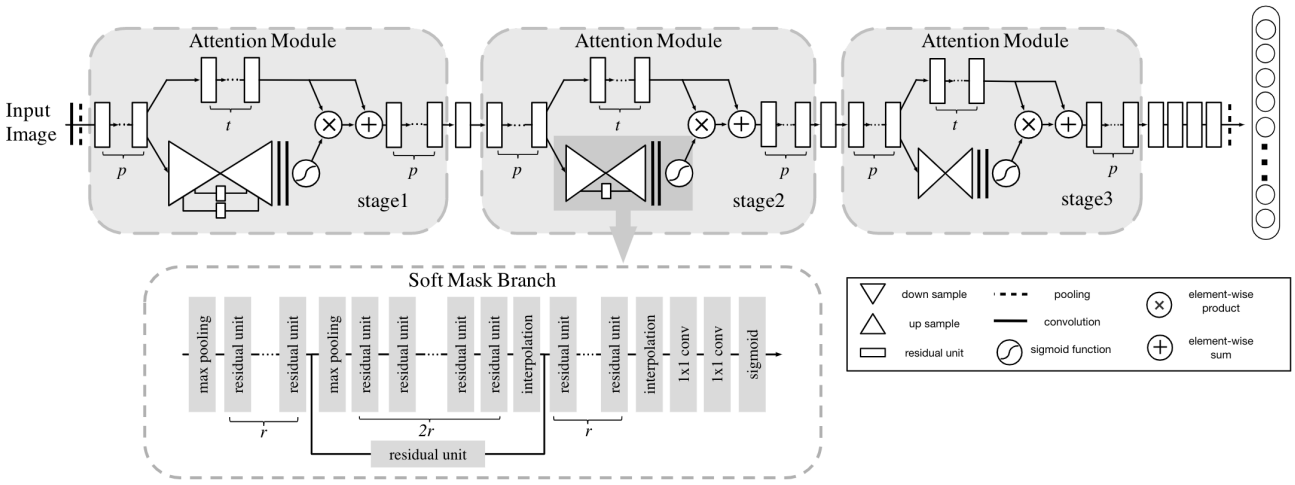

Fig. 1. Example architecture of Soft Mask Branch, with the construction of residual units, comparing to the general branch with convolution modules. The hyper-parameters r and 2r denote the corresponding amount of residual units or convolution modules between adjacent pooling layer in the branch.

II. Summary of Original Paper

A. Methodology

The authors note that although attention has been studied for many years, previous work has not applied attention mechanisms to feedforward network structures to achieve the highest performance for image classification tasks [5]. After Kaiming He proposed deep residual networks, the depth of convolutional neural networks was greatly increased and the performance of image classification tasks was significantly improved. Based on this, the authors of this paper have come up with a network that mixes the attention module with the residual network to enhance the network performance.

The authors' proposed network is shown in Fig.1 The network is constructed by stacking multiple attention modules. Each attention module is divided into two branches: the mask branch and the trunk branch. Trunk branch performs feature processing functions and can be adapted to any recent network structure. The mask branch proposed in the article consists of a fast feed-forward scan and a top-down feedback step. The former operation quickly collects global information about the whole image, and the latter operation combines the global information with the original feature map, a structure that mimics the process of fast feed-forward and feedback attention.

However, simply stacking attention modules can cause network performance to degrade in a deeper network, so they cleverly introduced identity mappings in residual networks to solve this problem. The authors introduced shortcut connections in each attention module. With this connection, residual learning can maintain good performance of the original features, and allow these features to bypass the soft mask branch and advance to the top layer, thus weakening the feature selection ability of the mask branch.

For the soft mask branch, starting from the input, several max-pooling is performed to rapidly increase the receptive field after a small number of residual units. After reaching the minimum resolution, the global information is expanded by a symmetric top-down architecture to guide the input features at each location. Linear interpolation of the output is upsampled after residual units. The number of bilinear interpolations is the same as the number of max-pooling to make the output size the same as the input feature map. Then, after two consecutive convolutional layers, sigmoid-type layers normalize the output range.

In this way, the attention-perception function from different modules changes adaptively as the module goes deeper which can be easily expanded to hundreds of layers.

B. Key Results

In the experiment, the authors evaluated the performance of proposed Residual Attention Network on several benchmark datasets, which are CIFAR-10, CIFAR-100 and ImageNet. The experiment is composed of two parts. First is analysis of the effectiveness of each component in the Residual Attention Network including residual attention learning and different architectures of soft mask branch. And the second is about noise resistance evaluation.

The first important result is the comparison between Residual Attention Network and Naïve Attention Network to test the effectiveness of proposed architecture. It is shown that ARL could achieve lower Top-1 error rate with 5.52% in Attention-56 module (56 means the number of layers in attention module) than NAL with 5.89% in the same metric and module. And in ARL, increasing depth of attention module obtains consistent improvements in Top-1 error rate.

The second key result is the noise resistance performance. The authors compared the ResNet-164 network with the Attention-92 network at different noise levels. The test error of the Attention-92 network is significantly lower than that of the ResNet-164 network at the same noise level, with only 5.15%

error in the Attention-92 network at 10% noise level, which is lower than the 5.93% error in the ResNet-164 network. In addition, when they increased the noise ratio, the test error of Attention-92 will slowly decrease compared with the ResNet-164.

Moreover, the authors compared Top-1 error and number of parameters on both CIFAR-10 and CIFAR-100 under state-of-the-art methods like ResNet and WRN. Attention-92 network achieved 4.99% test error on CIFAR10 and 21.71% test error on CIFAR-100 compared with 5.46% and 24.33% test error on CIFAR-10 and CIFAR100 using ResNet-164 network under similar parameter size. In addition, Attention-236 outperforms ResNet-1001 using only half of the parameters. It suggests that the attention module and attention residual learning scheme can effectively reduce the number of parameters in the network while improving the classification performance.

For ImageNet, the authors evaluated their model using single crop scheme on the ImageNet validation set and it is shown that attention-56 network outperforms ResNet-152 by a large margin with a 0.4% reduction on top-1 error and a 0.26% reduction on top-5 error. More importantly, attention-56 network achieves better performance with only 52% parameters and 56% FLOPs compared with ResNet-152, which suggests that the proposed attention mechanism can significantly improve network performance while reducing the model complexity.

III. Methodology

In our project, by following the proposed network architecture, we built our own deep network by stacking layers, including attention module and residual unit, in which attention modules contain different soft mask branches with different number of shortcut connections. There will be slightly different about the number of each module and unit in our own residual attention network because we do not need so deep architecture. We reconstructed Attention56 and Attention92 network and compared performance of them with simple ResNet network on CIFAR-10 and CIFAR-100. We also plan to add noise to our dataset to test the noise resistance of constructed network. In our project, we designed a meaningful experiment to test the performance of shortcut connection proposed in deep residual network to justify the effectiveness of this paper.

A. Objectives and Technical Challenges

Our project includes several parts. For the first part, we build our own residual attention network based on this paper. We defined residual unit by our own, including batch normalization, activation, convolution operation. Then we defined the attention block and it could be applied to different stage because the 'encoder' parameter can be defined later when calling this function, so our attention module would be like Fig 2. Then we apply pre-built attention56 and attention92 model to train CIFAR-10 and CIFAR-100 to get an acceptable results of test accuracy.

The second section of work is to apply original ResNet on our training and compare their performance when all hyperparameters are set to be the same. Then in our third part of the project, we add noise level of different degree to training dataset and test the noise resistance of our proposed model. Finally, we test the performance of shortcut connection by comparing it with simple network without shortcut and the architecture is given later.

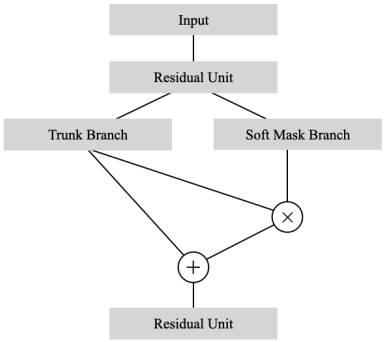

Fig. 2. Attention Module Architecture.

In our project, we encountered some technical challenges. The very first difficulty is that we cannot achieve acceptable test accuracy while training with two CIFAR datasets. And that could occur because of some reasons: 1) depth of network is not enough, we did not stack enough residual unit inside our model; 2) lack of some layers, normalization, dropout and regularization operation is not included in our network; 3) data augmentation is not applied in our design, at first we used original dataset so that the difference of training accuracy and test accuracy is really; 4) hyperparameter is not set correctly, the optimizer type, batch size, number of epoch, loss function type and learning rate is not set to achieve good performance. Another difficulty we may face is insufficient computing power because of GPU performance and memory is also not enough if dataset is too big. Also, we find our accuracy results are not as good as the original paper due to slight difference of constructed Attention Network or preprocessing to dataset.

B. Problem Formulation and Design Description

1) Attention Module: In our project, the most important part is the design of attention module. Each attention module is composed of two branches, trunk branch and mask branch. For trunk branch, we applied residual unit to perform feature extracting and get the trunk branch output T(x). For soft mask branch, it could be divided as two down-sampling and up-sampling process as given in Fig 3. so we can get soft mask branch output M(x). The output for a single attention module would be:

$$H_{i,c}(x) = M_{i,c}(x) * T(x) = (1 + M_{i,c}(x)) * F_{i,c}(x) \quad (1)$$

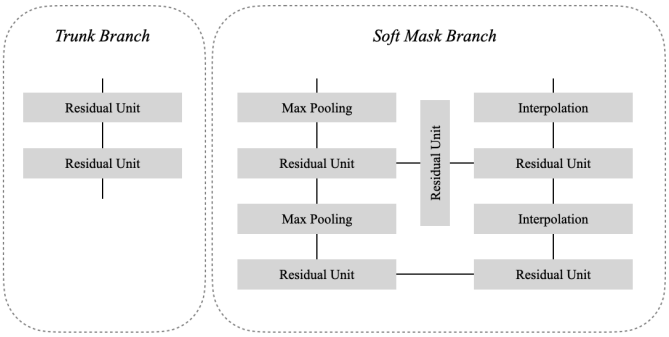

Fig. 3. Structure of two branches.

And attention mask can serve as a gradient update filter during back propagation, the gradient of mask for input feature is like this:

$$\frac{\partial M(x,\theta)T(x,\phi)}{\partial \phi} = M(x,\theta)\frac{\partial T(x,\phi)}{\partial \phi} \tag{2}$$

2) Residual Unit and Shortcut Connection: The key idea in ResNet is the introduction of 'shortcut connection', which helps to solve the 'degradation problem' when network becomes really deep. In our project, because of high level of depth, we applied lots of shortcut connection in our network architecture.

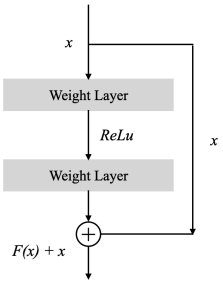

Fig. 4. Shortcut Connection

3) System Architecture: Our network including attention module is like this:

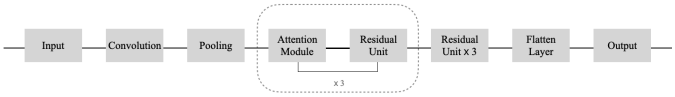

Fig. 5. System Architecture.

And to justify the performance of shortcut connection in ResNet we also designed a network replacing residual unit, like this:

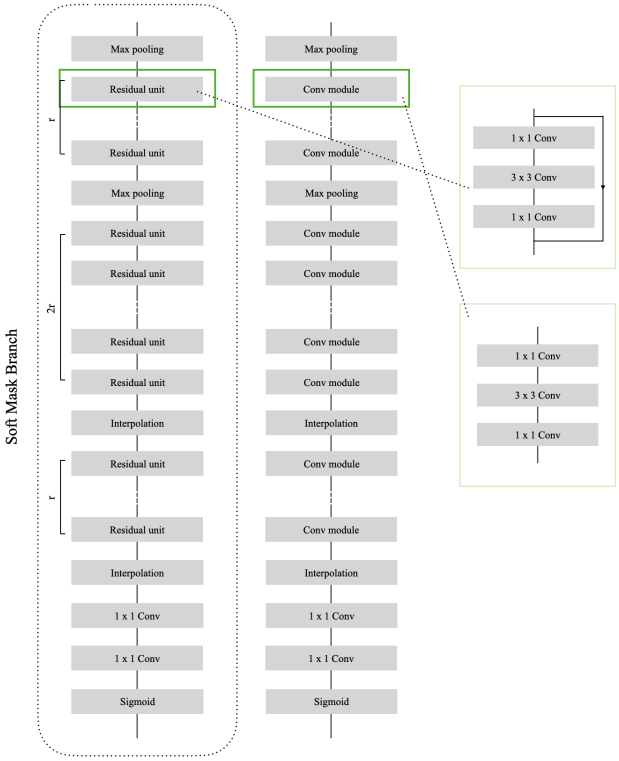

Fig. 6. Example architecture of Soft Mask Branch, with the construction of residual units, comparing to the general branch with convolution modules. The hyper-parameters r and 2r denote the corresponding amount of residual units or convolution modules between adjacent pooling layer in the branch.

IV. Implementation

In Chapter 4, we will discuss the architecture of the Residual Attention Network. We completed the model training on Google Colab, the graphics card used is Tesla P100, and the TensorFlow version is 2.5.

A. Data

The CIFAR-10 and CIFAR-100 data sets have a total of 60,000 color images. These images are 32x32 and are divided into 10/100 categories, each with 6000/600 images. There are 50,000/500 images for training, forming 5 training batches, each batch of 10000/100 images; the other 10000/100 for testing, forming a single batch. The test set selects each category from 10/100 categories, and randomly selects 1000/10 images from each category. The remaining photos are randomly arranged to form the training set.

B. Deep Learning Network

In our project, we mainly constructed two architectures, attention 56 and attention 92. And the only difference between them is the number of attention module inside them, with 3 in attention56 and 6 in attention92 respectively. So only the structure of attention56 will be described.

Our attention56 network consists of a residual unit, three different attention modules, flatten layers and there are also two residual units between two attention modules, flatten layer, activation function, like Fig.1.

Each attention module consists of a residual unit, a trunk branch composed of two residual units, a soft mask branch including two down sampling, two up sampling, two convolution layers and a sigmoid activation function. Each residual unit consists of a batch normalization layer, a ReLU activation function and a convolution layer. In both attention module and residual unit, shortcut connection is introduced. Zero-padding is used in all convolution layers to preserve size. All pooling layers have the step and stride of 2 to avoid overly reduce the size of representation. The fully connected layer has 10 units each and our final activation function is softmax function.

The table I shows the architecture of the Residual Attention Network corresponding to attention56 and attention92.

TABLE I
OUR NETWORK STRUCTURE OF ATTENTION 56 AND ATTENTION 92.

Layers	Attention 56	Attention 92	Output Shape
Conv2D	5x5x32, stride = 3		(None, 32, 32, 32)
MaxPool2D	2x2, stride = 2		(None,16, 16, 32)
Residual Unit	(32x32x128)		(None,16, 16, 128)
Attention Module	x1	x1	(None,16, 16, 128)
Residual Unit	(128x128x256)		(None,16, 16, 128)
Attention Module	x1	x2	(None, 8, 8, 256)
Residual Unit	(256x256x512)		(None, 4, 4, 512)
Attention Module	x1	x3	(None, 4, 4, 512)
Residual Unit	(512x512x1024) x 3		(None,16, 16, 128)
AveragePooling2D	4x4, stride = 1		(None, 1, 1, 1024)
Flatten			(None, 1024)
FC, Softmax			(None, 10)
Total params	54120138	385083684	

C. Software Design

We preprocess the dataset in the following method: firstly, we do some change on original dataset including translation, scale, rotation, flip with 'DataGenerator' function which can increase model performance; secondly, we map training and testing label to 10 class with 'to_categorical' function for training.

In our project, we set several experiments to test noise influence on our network, so there will be such a preprocessing in 'DataGenerator' function to add noise level to dataset. In the original paper, the author uses SGD as the optimization, the learning rate is set to 0.1, and use a weight decay of 0.0001 with a momentum of 0.9 and set the initial learning rate to 0.1, and decay the learning rate in the later stage. The epoch is 160k (reference literature). But considering our computing power and training time, it is unrealistic to use the author's hyperparameters, so we use ADAM as the optimization algorithm, batch size is set to 64, epoch is 100, the initial learning rate is set to 0.0001, and the learning rate is set Attenuation, the factor is 0.75.

V. RESULTS

Based on the above work, we completed the establishment of the model and used the CIFAR-10 and CIFAR-100 data sets to verify the accuracy of the model.

A. Project Results

First of all, we use the data set on CIFAR-10, first train the model on the network of Attention 56 and Attention 92 according to the above-mentioned hyperparameters. After 3 hours training time, the training error is 2.11% and the Val error is 5.52%. It can be seen from Table II, III, IV, V that we spent less computing power and time, and got a conclusion very close to the original author.

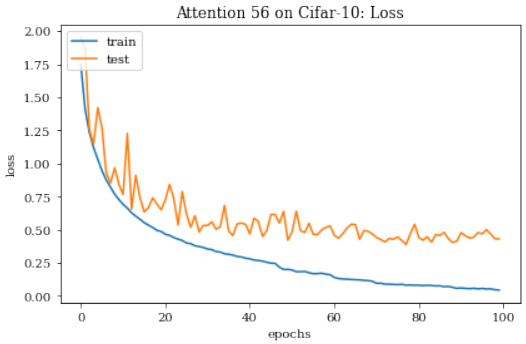

Fig. 7. Attention56 on Cifar-10: Loss

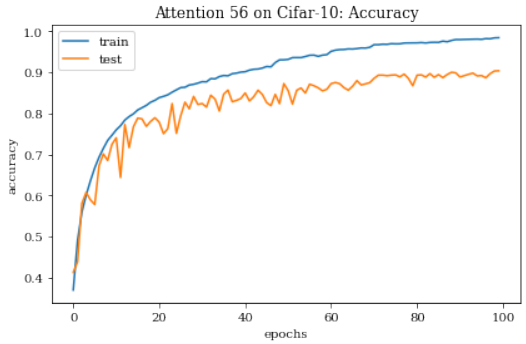

Fig. 8. Attention56 on Cifar-10: Accuracy

TABLE II
ATTENTION 56 ERROR ON CIFAR 10.

Network Structure	Error	Reference Error	Traning Times
With Shortcut	6.71%	5.52%	107s / epochs
Without Shortcut	6.68%	5.52%	98s / epochs

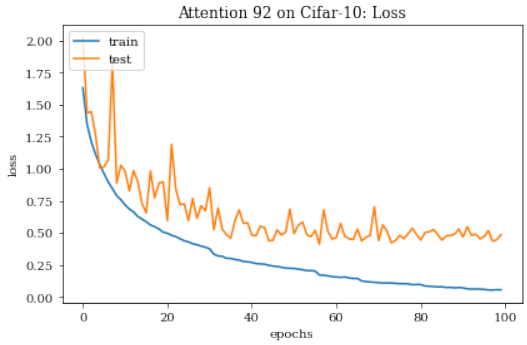

Fig. 9. Attention92 on Cifar-10: Loss

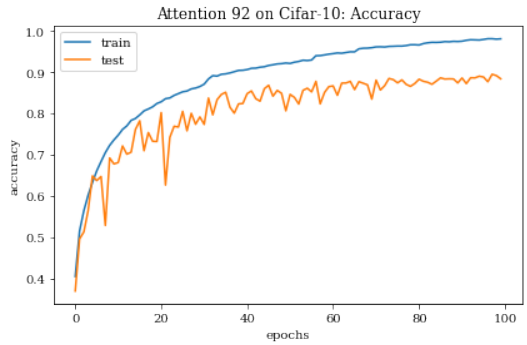

Fig. 10. Attention92 on Cifar-10: Accuracy

TABLE III
ATTENTION 92 ERROR ON CIFAR 10.

Network Structure	Error	Reference Error	Traning Times
With Shortcut	5.91%	4.99%	109s / epochs
Without Shortcut	6.12%	4.99%	109s / epochs

Next, we use the data set on CIFAR-100 to train the model on the network of Attention 56 and Attention 92. After we spent xx time training, the training error is and the Val error is. These results are also consistent with the original author's conclusion.

TABLE IV
ATTENTION 56 ERROR ON CIFAR 100.

Network Structure	Error	Reference Error	Traning Times
With Shortcut	32.24%	Not provided	172s / epochs
Without Shortcut	37.98%	Not provided	172s / epochs

TABLE V
ATTENTION 92 ERROR ON CIFAR 100.

Network Structure	Error	Reference Error	Traning Times
With Shortcut	25.8%	20.71%	424s / epochs
Without Shortcut	30.3%	20.71%	424s / epochs

Finally, we verify the effect of noise on the results, we set noise = 10%, 30%, 50% and 70%.

TABLE VI
ATTENTION 56 AND 92 ERROR ON CIFAR 10.

Noise	ResNet 164	Attention 56	Attention 92
10%	5.93%	9.46%	7.35%
30%	6.61%	10.43%	8.21%
50%	8.35%	13.10%	10.31%
70%	17.21%	21.3%	18.3%

It can be seen from the Table VI that when the noise is same, compared to the ResNet-164 network, the errors of Attention 56 and Attenion 92 have smaller errors. These results show that our Residual Attention Network can be trained well with high rates of noisy data. When we add noise, the corresponding mask can prevent gradient updates caused by label errors. In this way, only the trunk branch is learning the wrong supervision information.

B. Comparison of the Results Between the Original Paper and Students' Project

The core of the original paper is that using Attention Residual Learning (ARL) can achieve better performance than using ResNet. According to the data given in the paper, the error of ResNet-164 in CIFAR-10 / CIFAR-100 is 5.46% / 24.33%, and the error of ResNet-1001 is 4.92% / 22.71%. We built the Attention 56 / Attention 92 model according to the paper. According to our experiments, when using Attention 56 / Attention 92, the error of CIFAR-10/ CIFAR-100 is 6.71% / 5.91% Regarding noise, when the author of the paper uses Attention 92, the noise is 10%, the 30% error is 5.15%, and 5.79%. According to our experiments, when using Attention 92, the noise is 10% and 30%, and the error in CIFAR-10 / CIFAR-100 is 7.35% / 8.21%. This result is very close to the original paper.

C. Discussion of Insights Gained

The Attention 56 / Attention 92 model performs well in CIFAR-10. This may be because in CIFAR-10, each category has 6000 pictures, and most importantly, these categories are completely mutually exclusive, independent of each other, and there will be no overlap. So during training, each category has a large amount of original training data, so the effect is very good.

The author found that only stacking attention modules will significantly reduce the accuracy of the model, so he proposed the idea of attention residual learning, which is to build a residual attention network by stacking multiple attention modules to optimize the residual attention network. This is also through a residual method, so that very deep models can be easily optimized and learned. We also show through experiments that adding more attention modules can linearly improve the classification performance of the network. In other

words, we show through experiments that whether it is CIFAR-10 or CIFAR-100, Attention 92 has better performance than Attention 56.

The Residual Attention Network can be used in most deep networks. Stacking through a Residual Attention Module structure enables the network model to easily reach a very deep level. And it can reduce the amount of calculation while ensuring equal accuracy.

Bottom-up and top-down structure are combined, bottom-up is mainly for image feature extraction, and top-down is for generating Attention Map.

VI. Conclusion

In this project, we reproduced the main results in the original paper Residual Attention Networks for Image Classification. Due to the urgency of time and insufficient hardware resources, we did not reproduce the effect of the original model on the ImageNet data set, but mainly reproduced the main work on the Cifar10 and Cifar100 data sets. At the same time, since the original paper did not mention the detailed structure of the corresponding neural network for Cifar10 and Cifar100, we optimized our model parameters many times and finally achieved good results (accuracy).

In the process of completing this course project, we have figured out the key theories and processes in the original work through reading the original papers and reading related documents and materials. At the same time, we used what we learned in the ECBM 4040 Deep Learning & Neural Networks course to restore the original work. Through this project, the three of us not only have a detailed understanding of the residual unit and attention mechanism, but also have a deeper understanding of the development of neural network architecture and the current state of academic frontier research.

Acknowledgment

Here, we need to thank Professor Zoran Kostic and the teaching assistants of this course for their help and teaching, and in the process of reproducing this classic article, we also read a lot of materials on the Internet to be able to penetrate the residual unit and The real core of the attention mechanism is also grateful to the evangelists of these academic knowledge.

References

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

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

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[5] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Appendix

A. Code

The code of this re-implementation has been uploaded on Github:

https://github.com/ecbme4040/e4040-2021Fall-Project-REAL-xh2510-qt2139-kl3350

B. Team Member and Contribution

TABLE VII
Team Member and Contribution

Name	Xin Huang	Qimeng Tao	Kangrui Li
UNI	xh2510	qt2139	kl3350
Contribution Fraction	1/3	1/3	1/3
Details 1	Building Model	Building Model	Building Model
Details 2	Training Model	Training Model	Training Model
Details 3	Writing Paper	Writing Paper	Writing Paper