CLPS 1492 Computational Cognitive Neuroscience

Feedback and Feedforward Projections between V1 and IT cortex

Final Project Paper

Jinqian Li, Claire Diepenbrock, Zhuo Wang

Dec 12th 2023

1 Introduction

Originally, Chapter Six's object recognition model only considered bidirectional connections; there was a hierarchy, and the bidirectional connections ran between these and to the one above and below it. The V1 and IT cortex in the brain however, work much differently. In the actual brain, there very well may be projections from lower levels that project to higher levels. With this in mind, our group added new connections, changed the receptive field perspective, pattern rotation angles, line widths of object images, all in order to see how new objects would be recognized and were altered by our edits. We also created faces and new characters to see new interactions with the simulation.

2 Method

2.1 Different Line Widths and Rotations

2.2 Different Objects

2.2.1 Reason to build new objects

When we add a bidirectional connection between the V1 and IT layers of the network, it is natural to think that this may increase the network's ability to recognize objects. Under this assumption, we need more complex objects to train and test the network. The objects used in the homework are objects made up of horizontal and vertical lines of equal length. This is an object with relatively simple characteristics. In the initial test of the project, we found that if we used this object, it might be difficult to reflect the ability gap before and after the network change. Therefore, in order to increase the complexity of objects, we need to build a collection of objects with more features. When building new objects, we mainly build objects with different lengths of lines, oblique lines, and points to increase the complexity of the objects. Following these ideas, we constructed three new sets of objects: numbers, faces and Chinese numbers.

2.2.2 Coding for new objects

The way to build new objects is not complicated. Building new objects is mainly done by the drawline function in the lights.go code. This function can draw a line between two coordinates specified on a two-dimensional plane. With different combinations of coordinates, we can use this function to draw different lines, oblique lines, and points. By drawing different parts of an object multiple times using a function, we can build the object we want. In the process of building a new object, we first hand-draw the graph we want to build, then use the drawline function to build from the hand-drawn graph, and finally fine-tune the graph.

2.2.3 New objects and test method

Using the method in 3.2.2, we successfully built the following three new sets of objects.

The first set of objects are numbers. This set of figures consists of five numbers: 1,3,4,7,9. Each of these numbers has four different directions. This makes up a total of 20 different shapes. Compared to the original objects, these numbers are characterized by horizontal and vertical lines, but each line is not the same length. The object set is shown below.

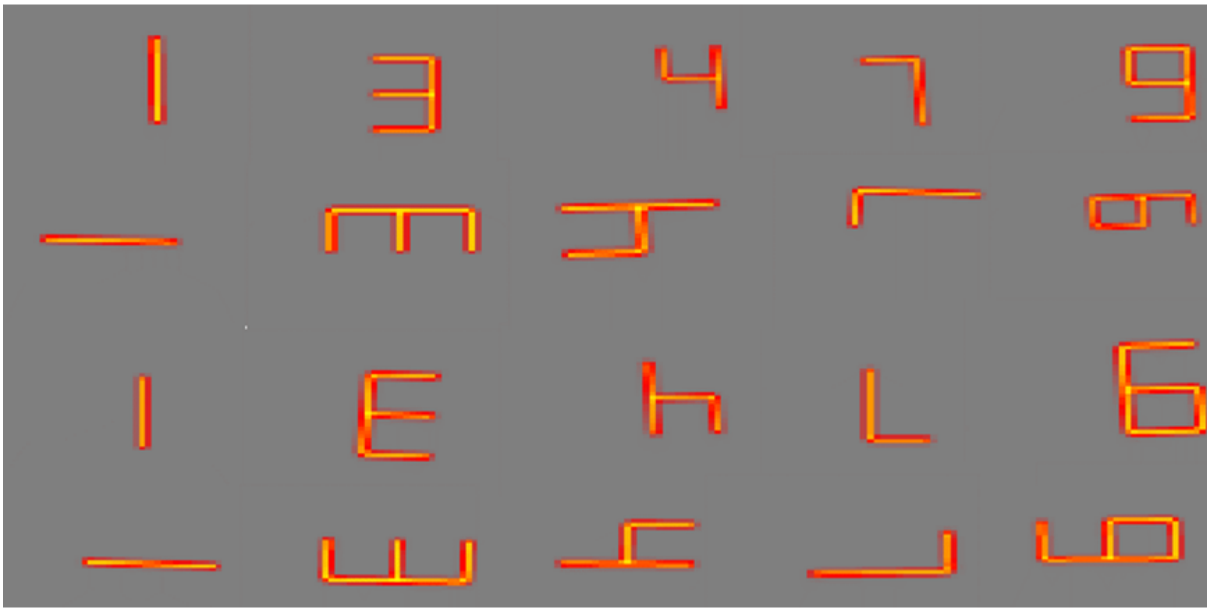

Figure 1: Objects set of number, consist of 1,3,4,7 and 9, each of them has 4 direction

The second set of objects is the face. The images were made up of five emojis: happy, unhappy, surprise, angry and sad. Each of these expressions has four different directions. This makes up a total of 20 different shapes. Compared to the original objects, these numbers are characterized by horizontal, vertical and diagonal lines. The object set is shown below.

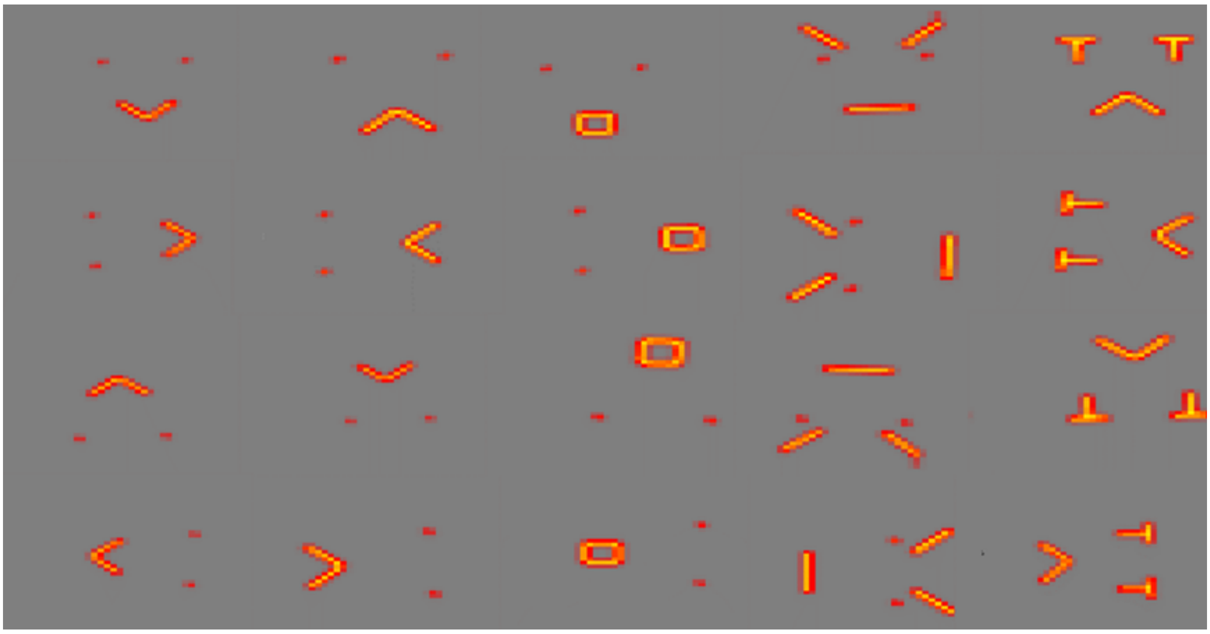

Figure 2: Objects set of face, consist of happy, unhappy, surprise, angry and sad, each of them has 4 direction

The third set of objects are Chinese numbers. This set of figures consists of five Chinese numbers, 2,3,5,6 and 8. Each of these numbers has four different directions. This makes up a total of 20 different shapes. Compared to the original objects, these numbers are characterized by horizontal, vertical and diagonal lines. The object set is shown below.

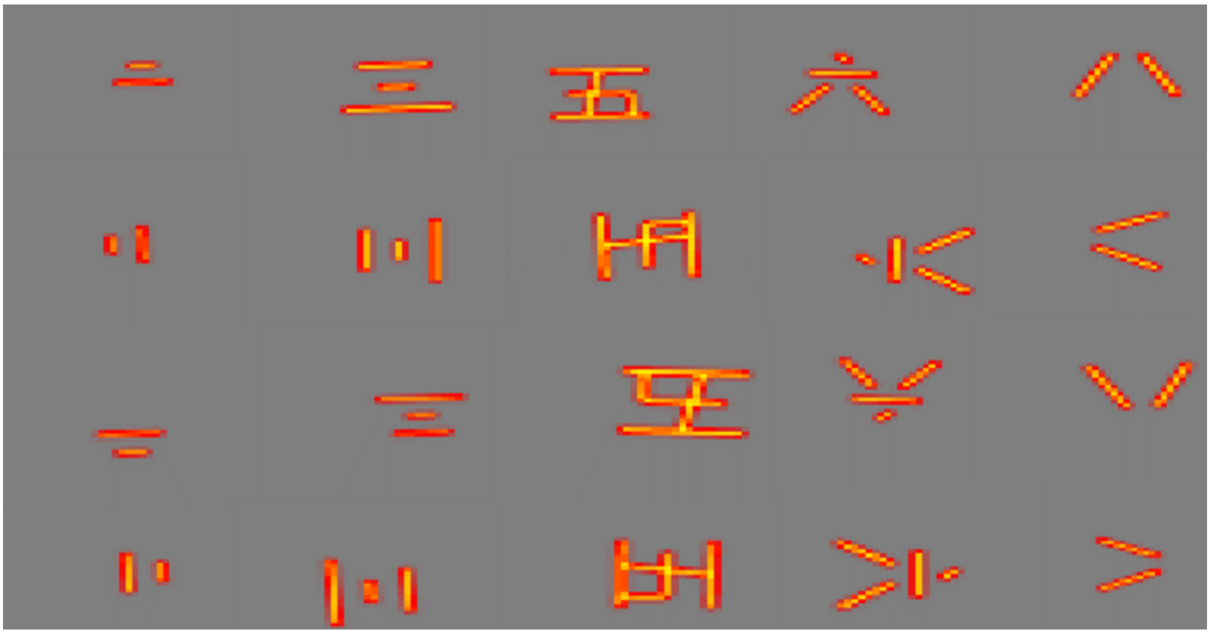

Figure 3: Objects set of chinese number, consist of 2,3,5,6 and 8, each of them has 4 direction

Using the object sets above, we can train the network and test its ability. For these new objects, we trained them using both the original network and the network with added connections. The training and testing parameters are the same as those used in the homework. For each set of objects, objects 1-18 are first trained and tested, and then objects 19 and 20 are trained and tested separately to test the generalization ability of the network. By comparing the above results, we can analyze the improved network.

2.3 Different Connection Sizes

In the description of the list of final projects we can learn from the clinical anatomical results that there is a projection from a lower level to a higher level (i.e. direct feedforward and feedback from V1 and V4/ IT). But the object recognition model we learned in course only considers the bidirectional connections between each level of the hierarchy and the levels above/below. Thus, we add these connections to different types of topographies and explore their impact on model object recognition. In this process, the influence of different connection sizes on the results is also an important exploration and discussion.

2.3.1 Change the Size of V1 Feedforward V4

We all know a too large connection size can lead to potential problems, such as confusion between different features, making neurons less selective for specific features. This can result in fuzzy feature representations, making it difficult for the network to accurately distinguish between different objects or features. For the network, too big connection size may increase the computational burden and complexity of the model. Furthermore, it requires more connection weights and information transfer, which can lead to wasted computing resources and reduced efficiency. In addition, it can also cause the network to overgeneralize, making the network overly sensitive to noise or irrelevant features and ignoring the really important features. This can reduce the model's ability to generalize, especially when dealing with new data or situations that have not been seen before. In terms of input features, a large connection size may make it more difficult for the network to capture and process local features. This may lead to the lack of sensitivity of the network to the local structure and details of the object, thus affecting the accuracy of the object recognition. Therefore, a large connection size may lead to information confusion, increased computational complexity, over-generalization, and difficulty in capturing local features.

In the process of trying to find the optimal connection size, I did not know whether the original network connection was too large, so I first tried to reduce the area of feedforward from V1 to V4 to 4*3 (row * column) for verification, so as to ensure the direction of my next experiment.

Algorithm Change

objrec.go

```
ss.V1V4Prjn = prjn.NewPoolTile()
ss.V1V4Prjn.Size.Set(4, 4)          ⇒          ss.V1V4Prjn.Size.Set(3, 4)
ss.V1V4Prjn.Skip.Set(2, 2)
ss.V1V4Prjn.Start.Set(-1, -1)
ss.V1V4Prjn.TopoRange.Min = 0.8
```

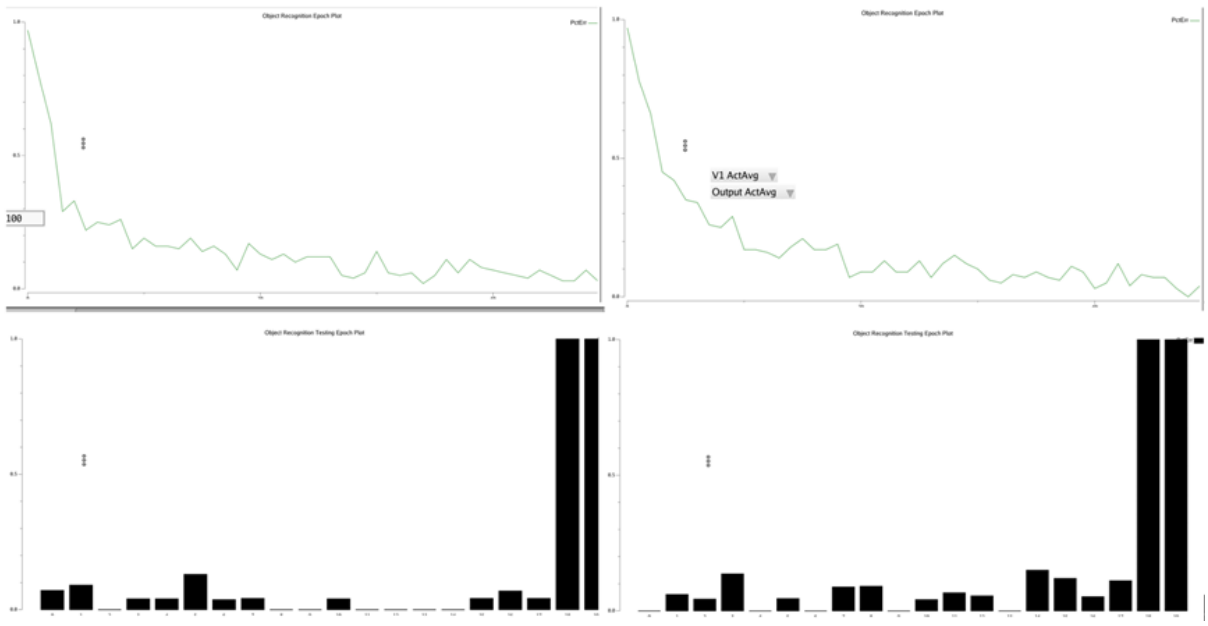

Figure 10. (*Upper left*) The prediction error about object recognition epoch of the original network. (*Upper right*) The prediction error about object recognition epoch of the 4*3 network. (*Lower left*) The prediction error about object recognition testing epoch of the original network. (*Lower right*) The prediction error about object recognition testing epoch of the 4*3 network.

After modifying the code, the model ran successfully and the connection size was also successfully changed, but the prediction error of reducing the connection size seemed to be larger than that before reducing the connection size, indicating that our next experimental direction was to try to increase the connection size to improve the robustness of the model.

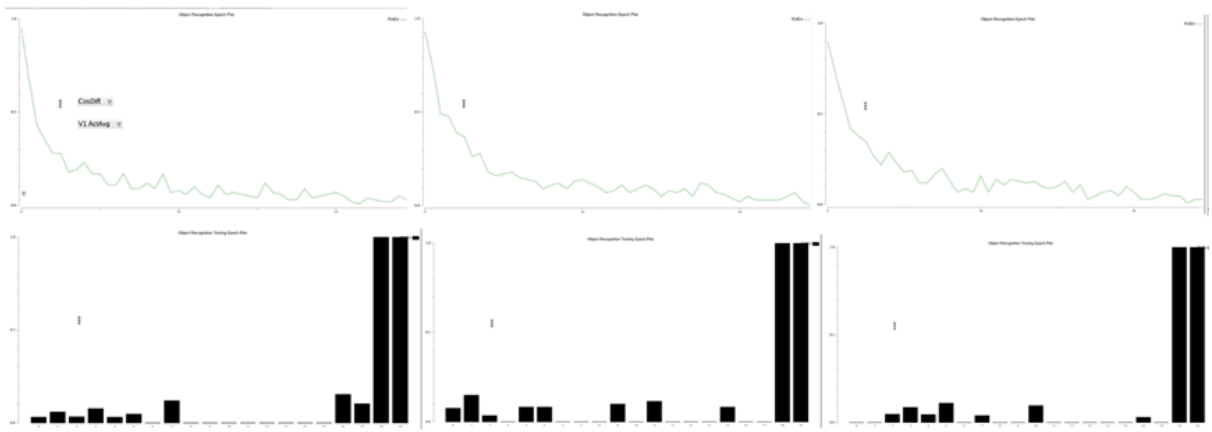

Figure 11. (*Left column*) The prediction error about object recognition epoch and object recognition testing epoch of the 6*5 network. (*Middle column*) The prediction error about object recognition epoch and object recognition testing epoch of the 8*6 network. (*Right column*) The prediction error about object recognition epoch and object recognition testing epoch of the 10*8 network.

Then I followed the above algorithm adjustment method. The influence of 6*5, 8*6 and 10*8 connection sizes on the performance of the model were continuously tried. And the corresponding prediction error figures were obtained.

Obviously, 10*8 V1 to V4 connection size is the best result, so we determined a robust V1 to V4 connection size, and on the basis of this parameter we can proceed to our next exploration to try to find the most robust V1 to IT connection size.

2.3.2 Change the Size of V1 Feedforward IT

As mentioned above, the clinical anatomical evidence shows that there is a certain degree of connection between V1 and IT, so a suitable connection is of great significance. Therefore, we first tried to quarter V1 and the IT layer and connect them to verify this view.

<u>Algorithm Change</u>

objrec.go

```
V1V4Prjn *prjn.PoolTile          ⇒          V1V4Prjn *prjn.PoolTile
                                            V1ITPrjn *prjn.PoolTile
……
    ss.V1V4Prjn.TopoRange.Min = 0.8
  ⇓
    ss.V1V4Prjn.TopoRange.Min = 0.8
    ss.V1ITPrjn = prjn.NewPoolTile()
    ss.V1ITPrjn.Size.Set(5, 5)
    ss.V1ITPrjn.Skip.Set(5, 5)
    ss.V1ITPrjn.Start.Set(0, 0)
    ss.V1ITPrjn.TopoRange.Min = 0.8
……
    it := net.AddLayer2D("IT", 10, 10, emer.Hidden)
  ⇓
    it := net.AddLayer4D("IT", 2, 2, 5, 5, emer.Hidden)
……
    net.ConnectLayers(v1, v4, ss.V1V4Prjn, emer.Forward)
  ⇓
    net.ConnectLayers(v1, v4, ss.V1V4Prjn, emer.Forward)
```

```
net.ConnectLayers(v1, it, ss.V1ITPrjn, emer.Forward)
```

After modifying the code, the model ran successfully and the connection size was also successfully changed. However, the prediction error rate has increased significantly. We hypothesized that by connecting parts of V1 to their counterparts in IT, parts of the information were lost or passed incomplete to higher-level neurons. As a result, the model cannot make full use of the features extracted from the V1 layer for effective object recognition. In addition, connecting parts of V1 to IT may cause the model to lose the integrity of the overall feature representation. The information at layer V1 may contain important local features of the object, and partial connections may not provide enough information to support accurate identification of the complete object. There may also be other underlying topological factors.

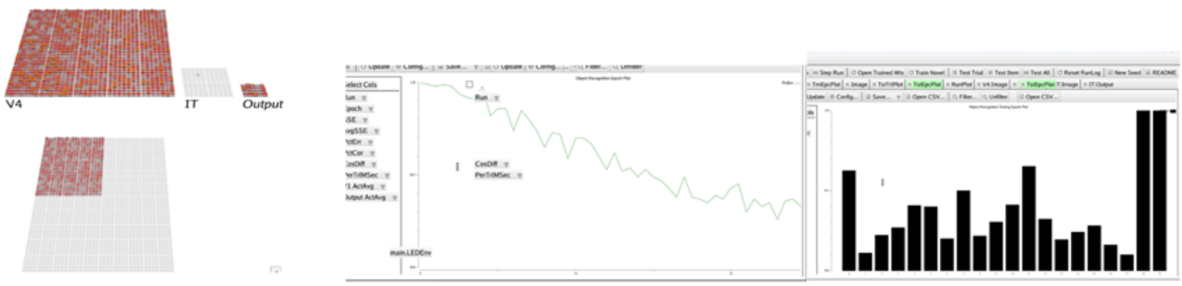

Figure 12. (*Left*) Connection size. (*Middle*) The prediction error about object recognition epoch of the V1 to IT counterpart connection network. (*Right*) The prediction error about object recognition testing epoch of the V1 to IT counterpart connection network.

Then I followed the above algorithm adjustment method. Three different connection sizes of 8*8, 9*9 and full connection were tried, and it was found that the partial connection of 9*9 performed best on the raw dataset, and the full connection followed.

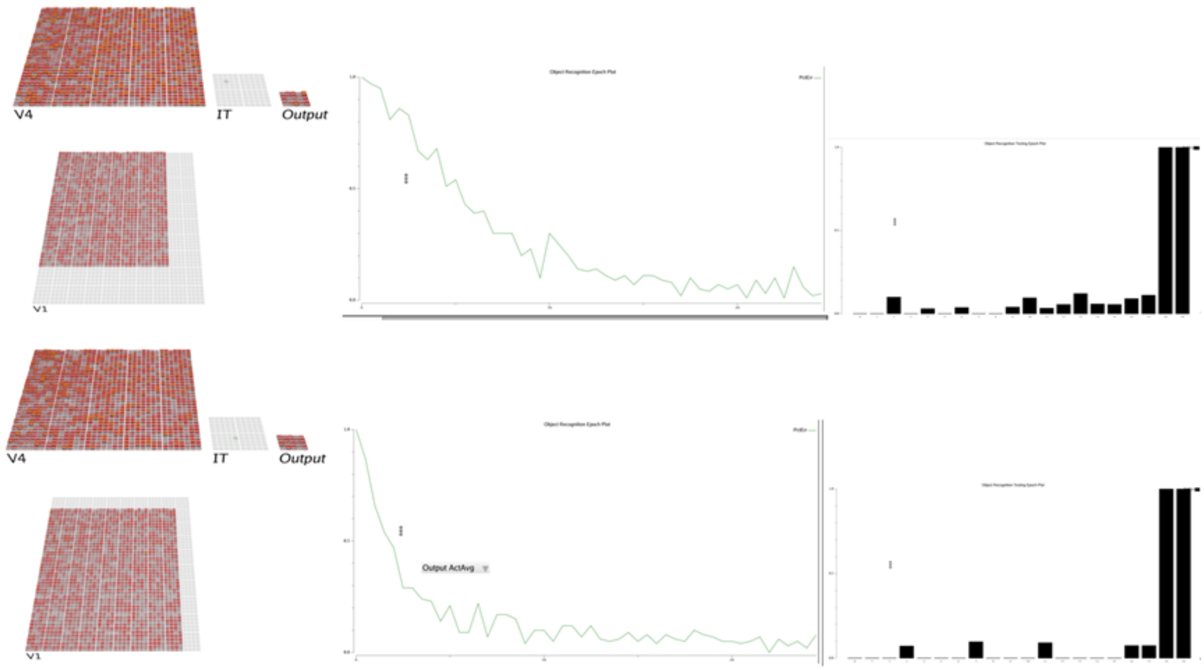

Figure 13. Comparison of the results of 8*8 and 9*9 connections

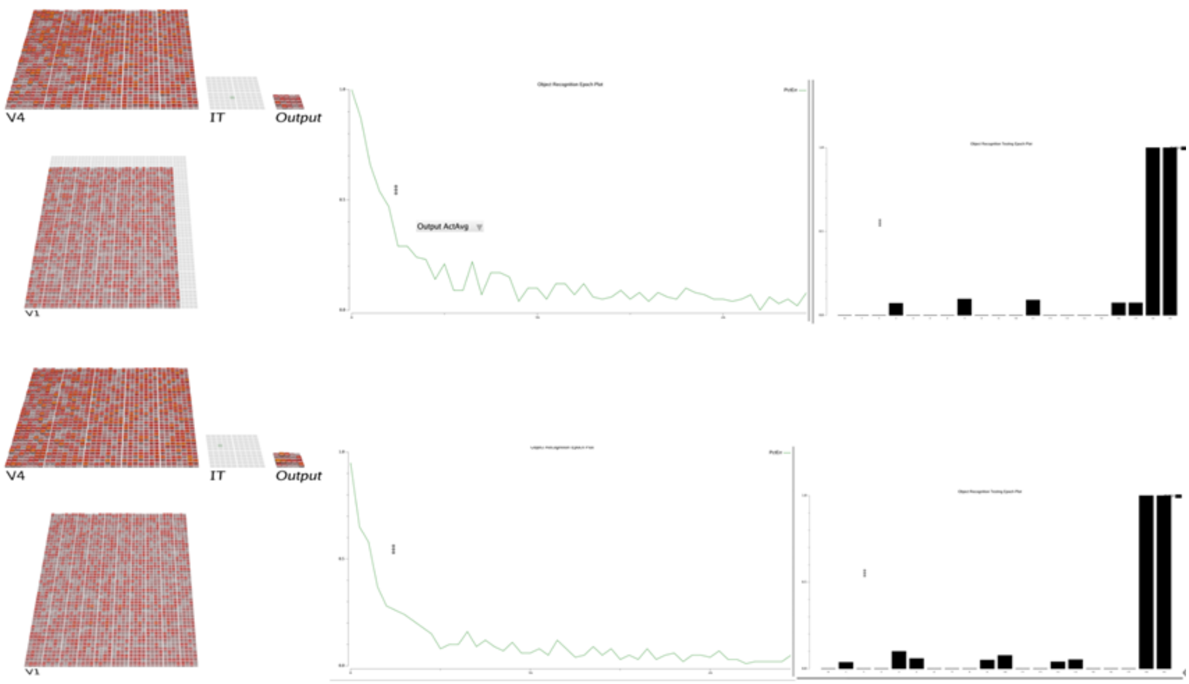

Figure 14. Comparison of the results of 9*9 and full connections

In order to improve the generalization ability of the validation model, we tested the 9*9 connected and fully connected models on number, face and Chinese numbers datasets respectively, and obtained the experimental results successfully.

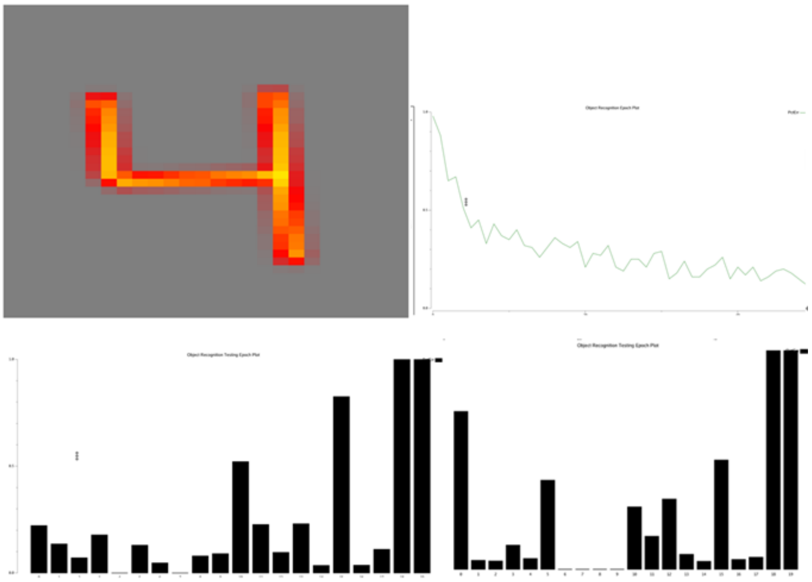

Figure 15. (*Upper left*) The image of the number dataset. (*Upper right*) The prediction error about object recognition epoch of the 9*9 network test on the number dataset. (*Lower left*) The prediction error about object recognition testing epoch of the 9*9 network. (*Lower right*) The prediction error about object recognition testing epoch of the full connection network.

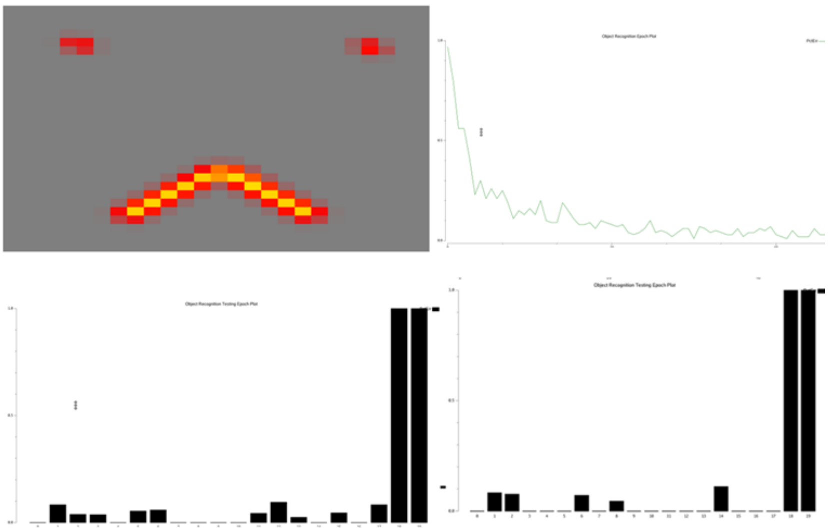

Figure 16. (*Upper left*) The image of the face dataset. (*Upper right*) The prediction error about object recognition epoch of the 9*9 network test on the face dataset. (*Lower left*) The

prediction error about object recognition testing epoch of the 9*9 network. (*Lower right*) The prediction error about object recognition testing epoch of the full connection network.

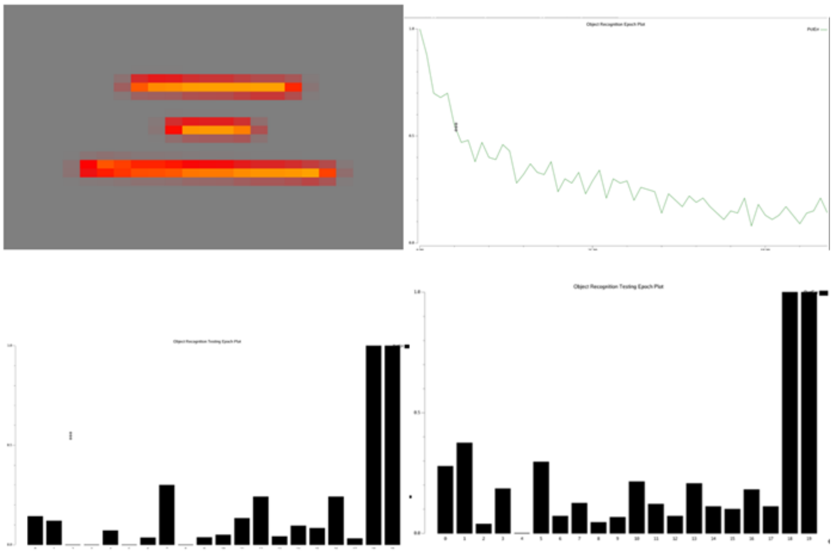

Figure 17. (*Upper left*) The image of the Chinese number dataset. (*Upper right*) The prediction error about object recognition epoch of the 9*9 network test on the Chinese number dataset. (*Lower left*) The prediction error about object recognition testing epoch of the 9*9 network. (*Lower right*) The prediction error about object recognition testing epoch of the full connection network.

2.3.3 Change the feedback route for V1/V4

In the project description, we noticed that there might be feedback from V4 to V1, so we tried to add feedback paths to the robust model and conducted tests on the raw dataset to compare the result with the robust model without adding feedback paths.

Algorithm Change

objrec.go

V1V4Prjn *prjn.PoolTile	⇒	V1V4Prjn *prjn.PoolTile
V1ITPrjn *prjn.PoolTile		V1ITPrjn *prjn.PoolTile
		V4V1Prjn *prjn.PoolTile

……

```
ss.V1ITPrjn.TopoRange.Min = 0.8
```

⇓

```
ss.V1ITPrjn.TopoRange.Min = 0.8
ss.V4V1Prjn = prjn.NewPoolTile()
ss.V4V1Prjn.Size.Set(5, 5)
ss.V4V1Prjn.Skip.Set(0, 0)
ss.V4V1Prjn.Start.Set(0, 0)
ss.V4V1Prjn.TopoRange.Min = 0.8
```

……

```
net.ConnectLayers(v1, it, ss.V1ITPrjn, emer.Forward)
```

⇓

```
net.ConnectLayers(v1, it, ss.V1ITPrjn, emer.Forward)
net.ConnectLayers(v4, v1, ss.V4V1Prjn, emer.Back)
```

After modifying the code, the model ran successfully and the connection size was also successfully changed. And we managed to get the results, but it seems that the results of adding feedback were not as good as we expected, which we will discuss in the discussion section.

	1	*2*	*3*	*4*	*5*	*6*	*7 …*	*20*	*Avg*
Feedback	0.036	0.048	0	0	0	0	0	1	0.129
Non-Feedback	0	0	0	0.036	0	0	0.143	1	0.128

Table 1. Comparison table of error rates of feedback and non-feedback models.

3 Result

3.1 Different line widths and Rotations

3.2 Different Objects

3.2.1 Number

Through training and testing, we get the training graphs and test results of different networks for this object set of numbers. The training and testing plot is shown below.

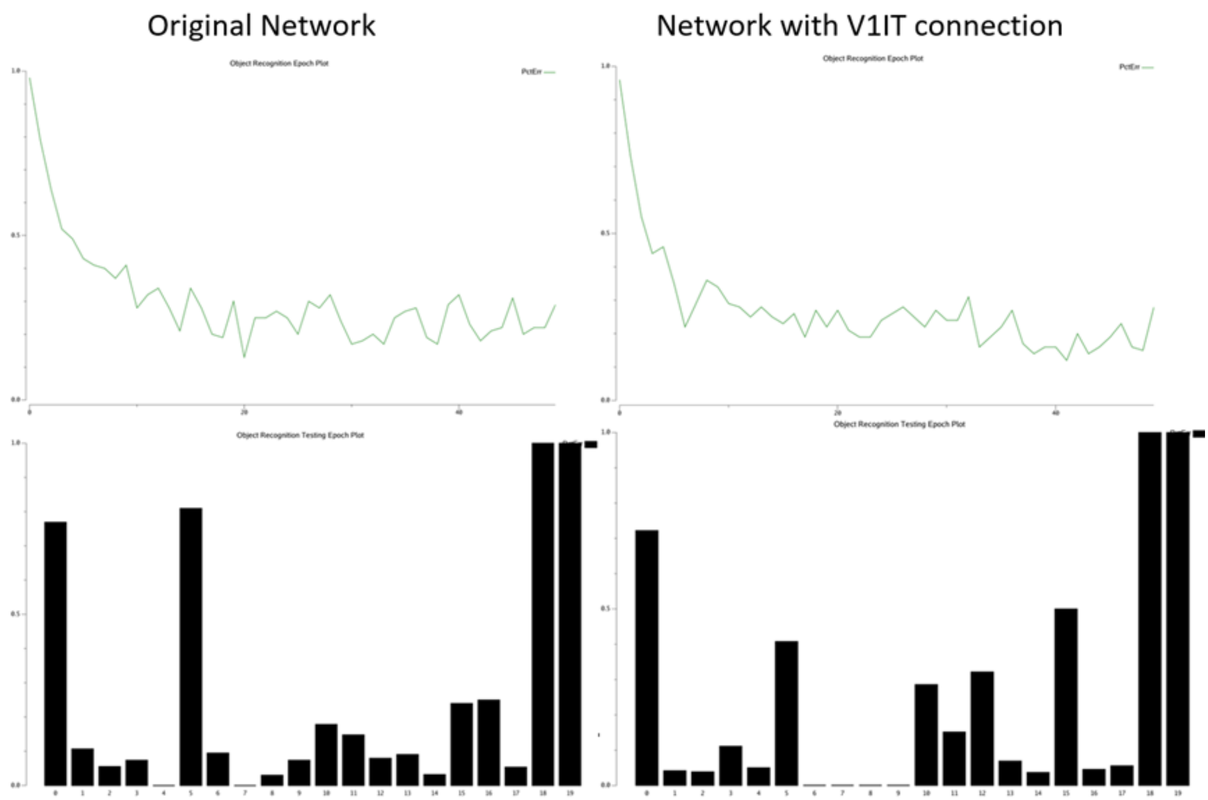

Figure 4: Training and Testing plot of numbers, left: original network right: edited network

The left side is the result of the original network, and the right side is the result after adding the V1IT connection. It can be found that the improved network can be trained to achieve lower errors and identify objects more accurately. As can be seen from the figure, objects 18 and 19 have a large error in both networks, which is due to the fact that the network has not trained them. After individual training, on the original network, objects 18 and 19 have errors of 0.96 and 0.52, respectively. On the improved network, the error of object 18 and 19 is 0.71 and 0.45 respectively. It can be seen from the results that the improved network generalization ability has a slight improvement.

In addition to training the effect, we also tested the receptive field of the V4 and IT layers. The result is shown below.

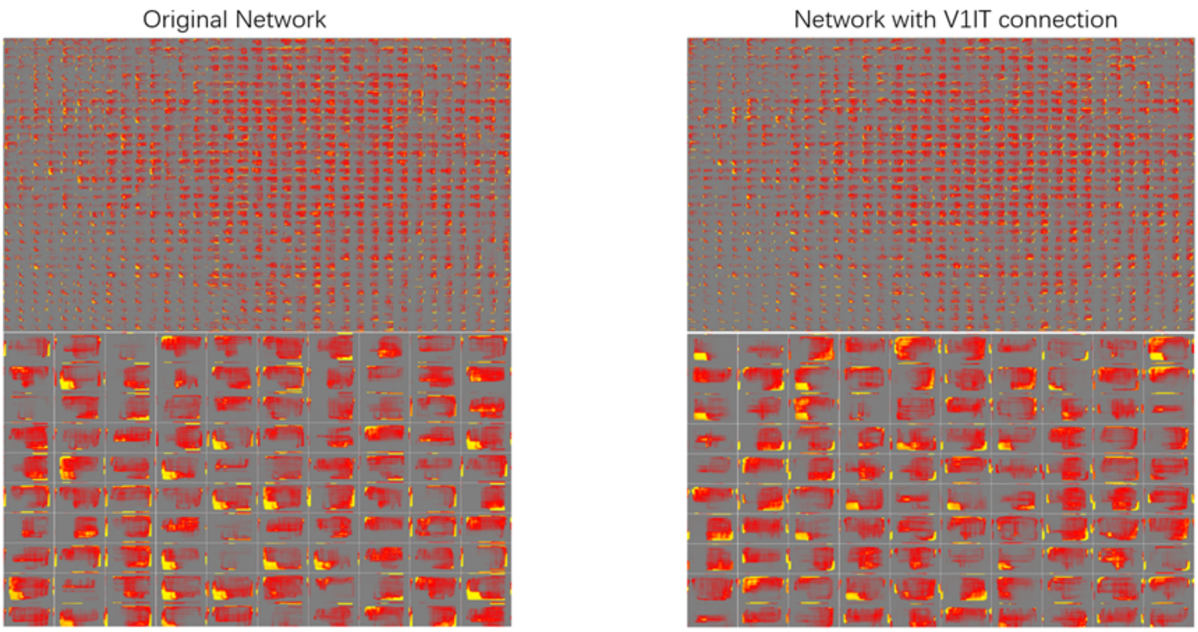

Figure 5: Receptive field test of numbers, left: original network right: edited network

As you can see from the figure, the V4 layer seems to be most sensitive to the number 7. The receptive field of the IT layer of the improved network seems to have more units that are more active. Compared with the original network IT layer, the improved IT layer has more obvious input object characteristics. This is possibly because of increased V1 connections to IT.

3.2.2 Face

Through training and testing, we get the training graphs and test results of different networks for this object set of faces. The training and testing plot is shown below.

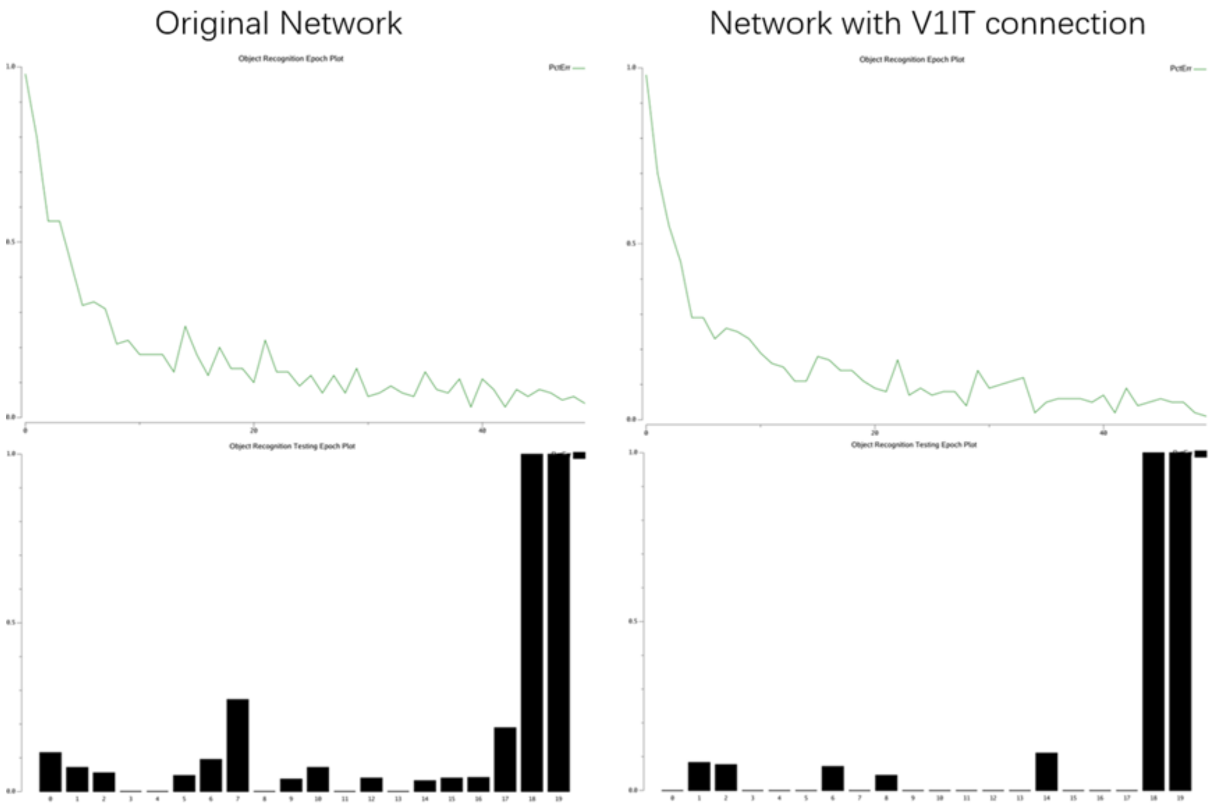

Figure 6: Training and Testing plot of faces, left: original network right: edited network

The left side is the result of the original network, and the right side is the result after adding the V1IT connection. It can be found that the improved network can be trained to achieve lower errors and identify objects more accurately. As can be seen from the figure, objects 18 and 19 have a large error in both networks, which is due to the fact that the network has not trained them. After individual training, on the original network, objects 18 and 19 have errors of 0.75 and 0.62, respectively. On the improved network, the error of object 18 and 19 is 0.71 and 0.68 respectively. It can be seen from the results that there is no significant difference in network generalization ability before and after improvement.

In addition to training the effect, we also tested the receptive field of the V4 and IT layers. The result is shown below.

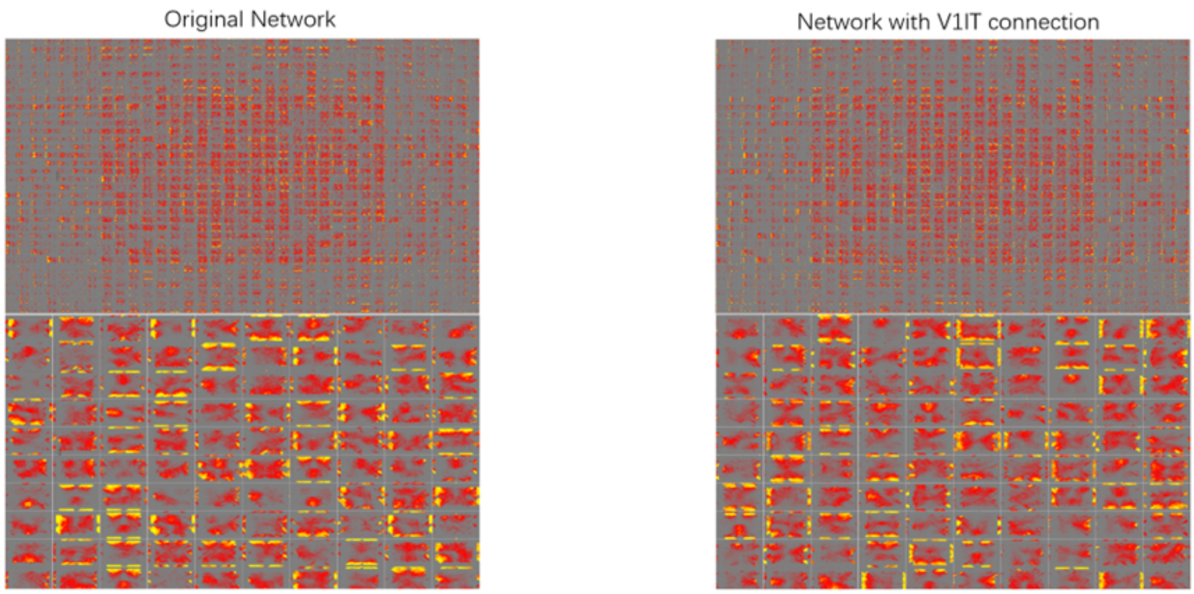

Figure 7: Receptive field test of faces, left: original network right: edited network

As you can see from the figure, the V4 layer seems to be most sensitive to the mouth of each expression. The receptive field of the IT layer of the improved network seems to have more units that are more active. Compared with the original network IT layer, the improved IT layer has more obvious input object characteristics. This is possibly because of increased V1 connections to IT.

3.2.3 Chinese number

Through training and testing, we get the training graphs and test results of different networks for this object set of Chinese numbers. The training and testing plot is shown below.

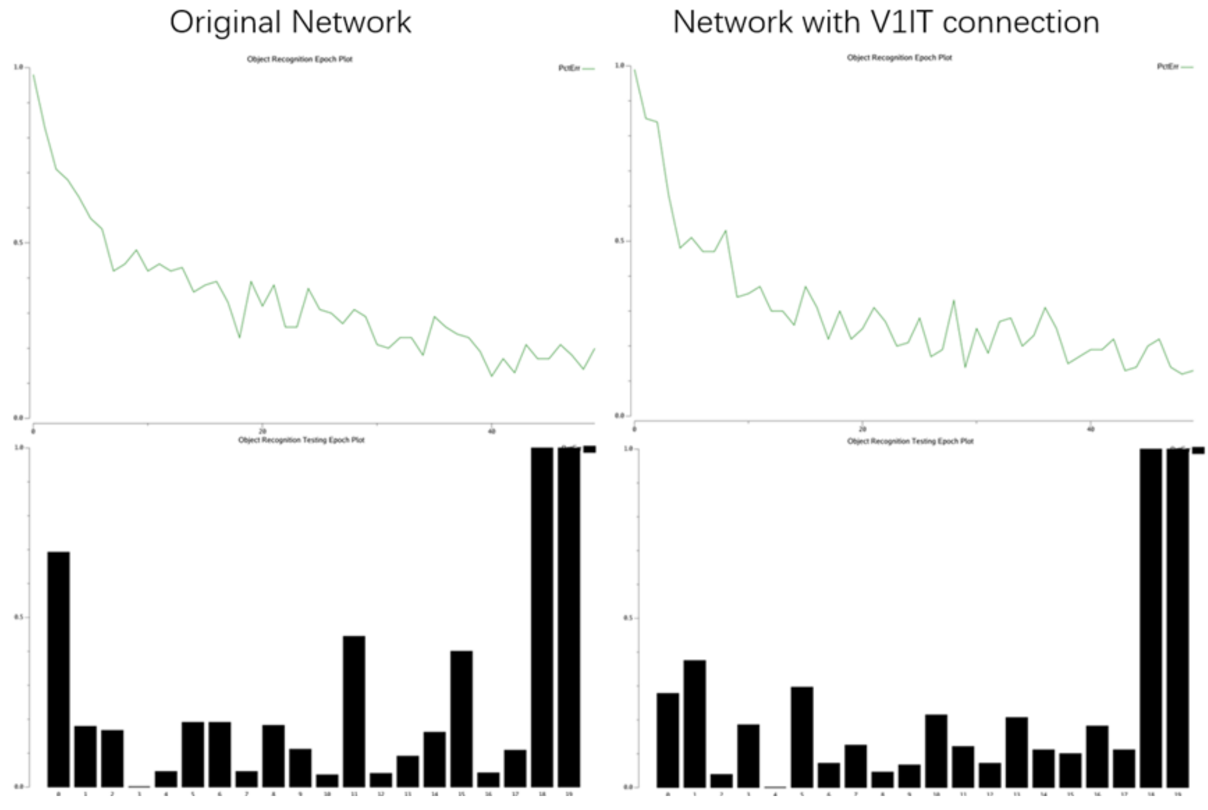

Figure 8: Training and Testing plot of Chinese numbers, left: original network right: edited network

The left side is the result of the original network, and the right side is the result after adding the V1IT connection. It can be found that the improved network can be trained to achieve lower errors and identify objects more accurately. As can be seen from the figure, objects 18 and 19 have a large error in both networks, which is due to the fact that the network has not trained them. After individual training, on the original network, objects 18 and 19 have errors of 0.82 and 0.5, respectively. On the improved network, the error of object 18 and 19 is 0.75 and 0.72 respectively. It can be seen from the results that there is no significant difference in network generalization ability before and after improvement.

In addition to training the effect, we also tested the receptive field of the V4 and IT layers. The result is shown below.

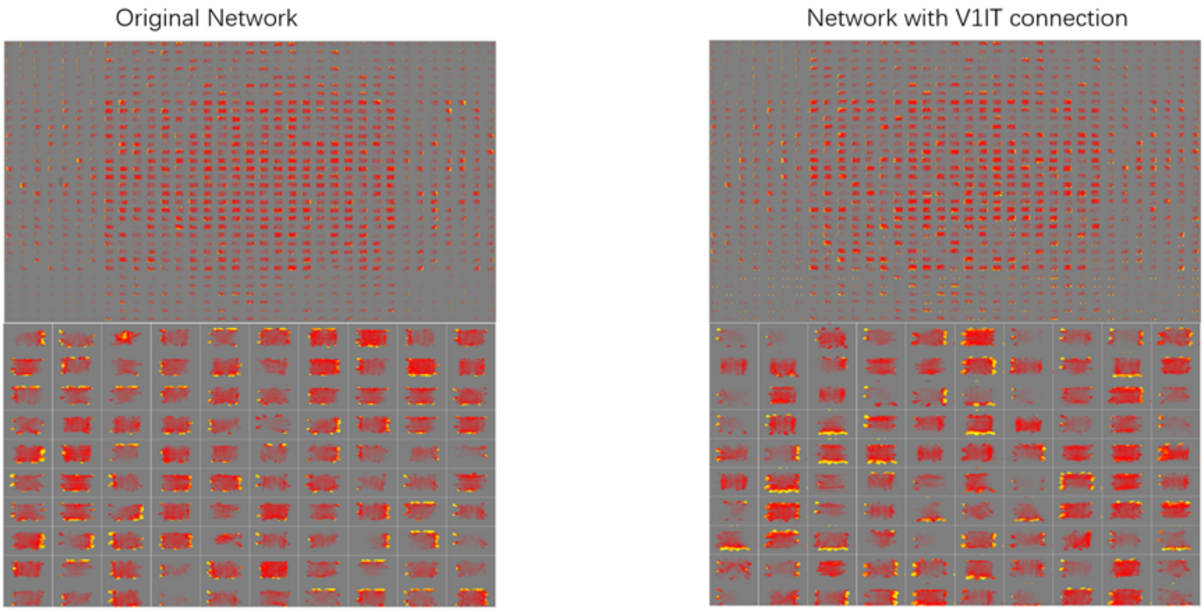

Figure 9: Receptive field test of Chinese numbers, left: original network right: edited network

As you can see from the figure, the V4 layer seems to be most sensitive to the two slash lines per number 6 and 8. The receptive field of the IT layer of the improved network seems to have more units that are more active. Compared with the original network IT layer, the improved IT layer has more obvious input object characteristics. This is possibly because of increased V1 connections to IT.

3.3 Different Connection Sizes

3.3.1 Change the Size of V1 Feedforward V4

After several experiments, we obtained a bar chart of prediction error results for different V1 to V4 connection sizes. The most frequent prediction error rate of 0 is the 10*8 model, which predicts the error rate of 0 in 11 objects, much higher than the 10 of 8*6 and the 9 of 6*5. The best numerical results were achieved.

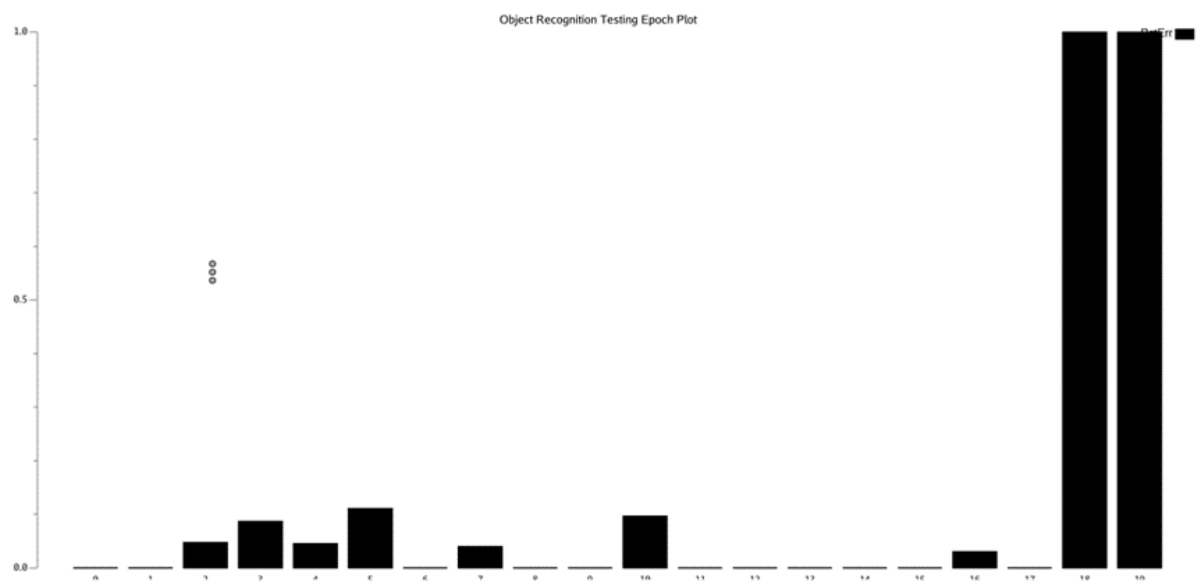

Figure 18. The prediction error about object recognition testing epoch of the 10*8 network.

3.3.2 Change the Size of V1 Feedforward IT

To verify the robustness of the 9*9 (Robust) model and the full connection model, we tested the two models on datasets of numbers, faces and Chinese numbers respectively, and obtained the prediction error rate results for all tests. In order to better visualize the data analysis, we plot the box diagram and the violin diagram of the test results on all the data sets. Also plot all the results of the mean box plot and violin plot.

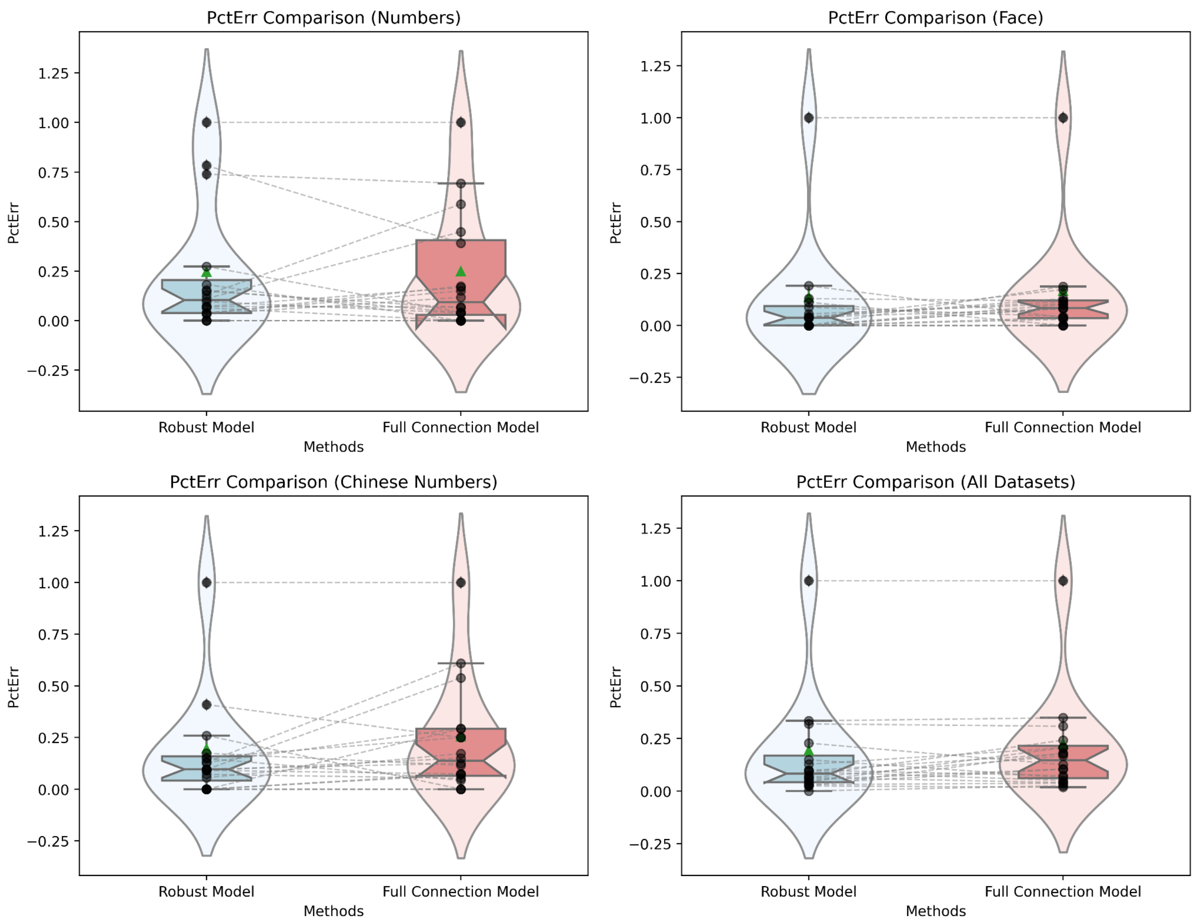

Figure 19. Box and violin plots of the robust (9*9) model and the full connection model testing results on different datasets

By visualizing the data, we can clearly find that compared with the full connection model, the box diagram of the robust (9*9) model is more obviously compact, and the error score of the full connection model in the upper quartile of the number dataset and the Chinese number dataset and the average result is also higher than that of the robust model.

For the numerical result, we calculated the average of the overall error rate for the 20 objects in the average result data. The average error rate of the robust model is 0.193 and the average error rate of the full connection model is 0.224. The error rate of a robust model is obviously lower.

3.3.3 Change the feedback route for V1/V4

According to our error rate results for models with and without feedback paths from V1 to V4 validated on the raw dataset. We also plot an intuitive box figure and a violin figure. We can find that the error rate of the model with feedback path is slightly higher than that of the model without feedback path. The small green triangle marks the average error rates of the two models. The mark on the left is slightly higher than the mark on the right. From the recognition error rate line of the corresponding object, there are more oblique lines tilted upward towards the side with feedback.

Numerically, the average error rate of the model with the feedback path is 0.129, while the average error rate of the model without the feedback path is 0.128. This is proved numerically.

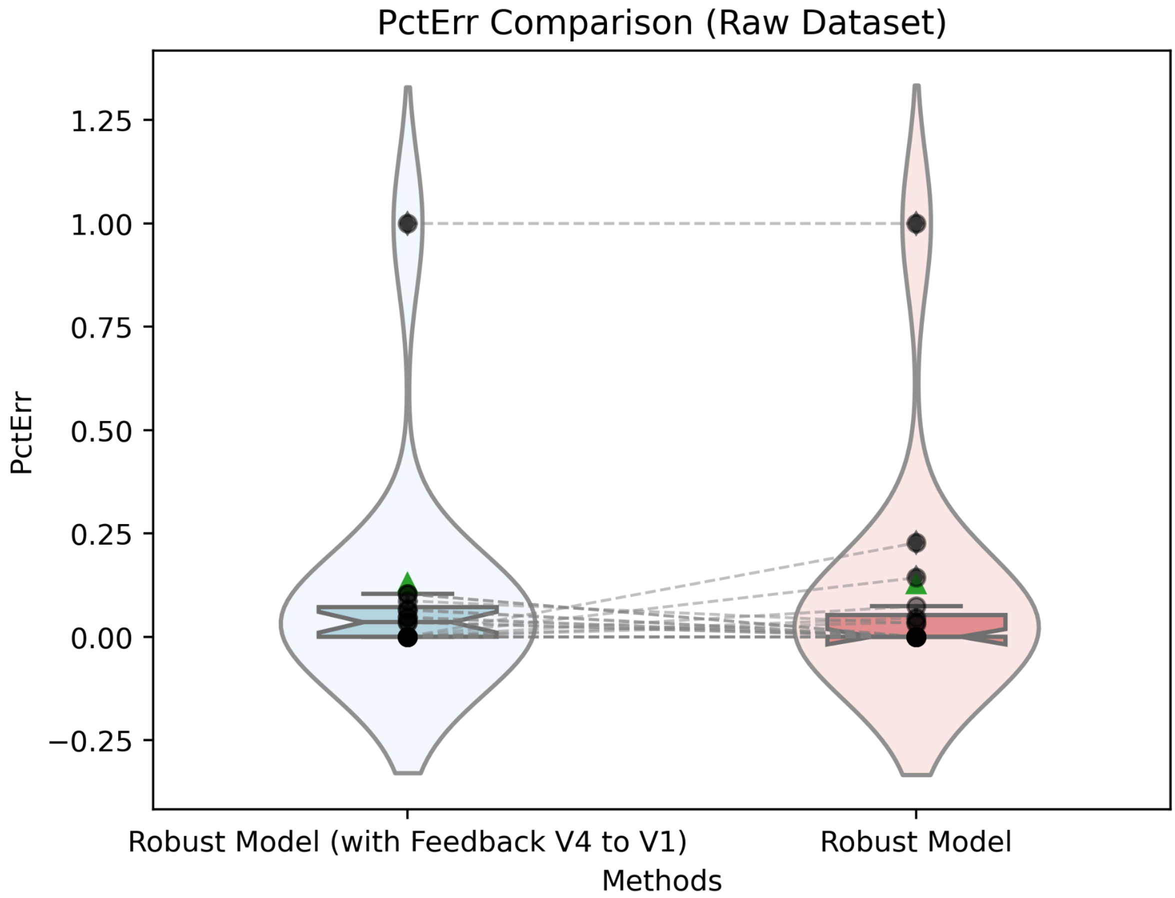

Figure 20. Box and violin plots of the robust model (with full feedback connection V4 to V1) and the robust model testing results on raw dataset

4 Conclusion and Discussion

4.1 Different line widths and Rotations

4.2 Different Objects

In this project, we successfully built three different groups of new objects, which are number, face and chinese number. Using the original network and a network with added bidirectional connectivity between V1 and the IT layer, we trained and tested these three different sets of objects separately. Through testing and comparison, we can draw the following conclusions about the network with added bidirectional connection between V1 and IT layer:

1. The network with added connections can achieve lower error in the test, which indicates that the improved network has more powerful object recognition capability.

2. Networks with added connections did not perform better on generalization tests. Of the three groups of graphs, only one group achieved slightly better generalization in the improved network, while the other two groups showed no significant improvement. Therefore, under the current test, we can not conclude that the network generalization ability is improved.

3. Through the test of receptive field, we can find that after training, the IT layer's receptive field of the improved network exhibits more features of a single object than the receptive field of the original network, rather than the feature set of multiple objects at a higher level. This may be due to the direct connection between V1 and the IT layer, resulting in the trained

IT layer weights having object features directly from V1, rather than the object features simplified by V4 layer.

In the testing of different objects in this project, there are still some deficiencies and areas that can be improved.

First, when building objects, we want to build new objects that are more complex. Our main strategy is to list some common objects and subjectively judge the complexity of these objects. Then pick out objects that may have more complex features. In the process, we did not find a set of methods to quantitatively assess the complexity of objects and systematically build new objects. Therefore, if we want to improve the experiment, we will need to find a reasonable way to build new objects in the future.

Second, when performing the generalization ability test, we only tested two objects. Too small a number of objects may not be able to statistically represent the error change before and after the network change. Therefore, in the future test, we should add more objects for the generalization test, in order to get accurate conclusions.

Finally, when adding the connection between V1 and the IT layer, we currently fully connect V1 and the IT layer. But a fully connected network may not work best. In future tests, we can only partially connect like layer V1 to layer V4, which may make the network achieve better results.

4.3 Different Connection Sizes

In the course of the project, through iterative exploration and experimentation, optimal connection sizes from V1 to V4 and from V1 to IT were successfully identified. Additionally, a fully connected feedback network from V4 to V1 was introduced. Subsequently, these three experimental setups were individually trained, tested, and the results thoroughly examined across various datasets. The conclusions drawn from these experiments are as follows:

1. The experimental results of the model became better after increasing the connection size from V1 to V4.
2. The network part connects V1 to IT better than V1 fully connects to IT.
3. The effect is worse after adding full-connection feedback from V4 to V1 to the robust model than without. (Limited to our experiments, results need further discussion.)

As for the first result, we believe that the increase of the connection size between V1 and V4 leads to the increase of information richness, and more connections mean that more information can be transferred from one region to another. This can increase the information richness of the model, allowing it to handle more complex visual features and patterns. In addition, increasing the connection size can promote a higher level of feature representation and extraction, thereby improving the model's understanding and generalization ability of visual information.

However, we also note that in this experiment, we only changed the connection size from V1 to V4 on the raw dataset. In future experiments, we can try to change the connection size from V1 to V4 on different datasets, and the generalization of different connection sizes has been verified.

As for the second result, we believe that the direct connection between V1 and IT can reduce the path length of information transfer. Since information does not need to pass through multiple intermediate layers, this direct connection helps to deliver and process information more quickly. Promote a higher level of visual feature integration and abstraction. This helps the model capture richer and more abstract visual features, improving the ability to understand complex visual patterns. However, full connection may bring about overfitting and sensitivity loss of local features, so the results of partial connection are better than those of full connection.

But we didn't try a random proportional partial connection pattern, which is something we could try in the future.

As for the result of adding the feedback connection from V4 to V1, we guessed that the feedback connection may introduce too much feedback information, which may interfere with or confuse the information propagated forward. Too many or inappropriate feedback signals can cause the learning of the model to become unstable, thus affecting performance. But feedback from the anatomy suggests that V4 to V1 is present. Our results need more verification.

Our experiment only tried fully connected feedback from V4 to V1, and in the future we can try partially connected feedback and test it on different datasets.

5 Author contributions

JL: Experiment (Different Objects Part), Methodology (Different Objects Part), Writing—manuscript (Different Objects Part), Validation and Investigation. CD: Experiment (Different Line Widths and Rotations Part), Methodology (Different Line Widths and Rotations Part), Writing—manuscript (Different Line Widths and Rotations & Introduction Parts), Validation and Investigation. ZW: Experiment (Different Connection Sizes Part), Methodology (Different Connection Sizes Part), Writing—manuscript (Different Connection Sizes Part), Validation, Investigation and Review.

6 Availability of Code

Code and models: https://github.com/cho-wang001/CLPS1492_Final_Project Frok: https://github.com/CompCogNeuro/sims.git