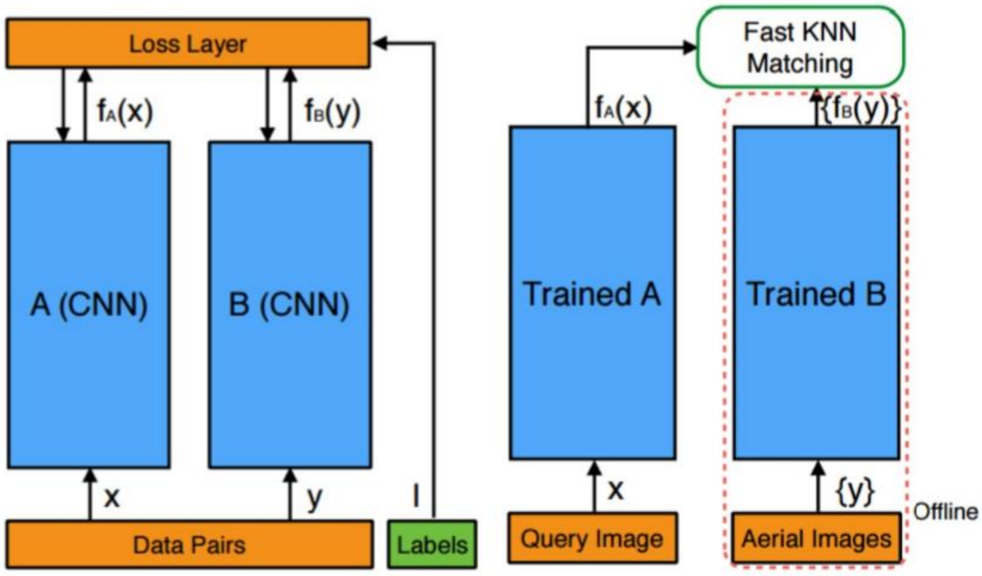
IFN680 Assignment 2 Project Siamese Network

Introduction

With rapidly developing of camera changes, Image classification and recognition have become a popular problem. Various types of neural network structure have been designed to classify and recognize different images. One of famous neural network is Siamese Neural Network, which contain two or more identical subnetworks. The input of a Siamese network includes a positive pair of picture and negative pair of picture. The two pictures belong to same equivalence classes is called positive pair, whereas it is called negative pair. In this report, we will build two kinds of Siamese network to recognize manta rays and compare their accuracy.

The dataset that contain the manta rays images load from MNIST dataset. There are ten classes dataset (0-9) in the MNIST dataset. The digits in [2, 3, 4, 5, 6, 7] will be used for training and testing. The digits in [0, 1, 8, 9] will be only used for testing. The structure of Siamese network is shown below:



(a) Training (b) Testing

We establish two Siamese network through two types of network. In first type Siamese network, we used three dense layer to build the base network. In second network, we use three convolution layer and three fully connect layer as complex CNN network. Then, the generalization capability of the network will be evaluated by three parts, which is testing pairs from the set of digits [2,3,4,5,6,7], [0-9] and [0,1,8,9]. Furthermore, we will compare the accuracy for these two Siamese network and justify a better neural network structure.

Methodology

To complete our program, there are four major steps to be followed. The first step is to split the whole dataset to training dataset (80%) and testing dataset (20%). And we select twenty present of training dataset as validation dataset. The second step is to create the positive pairs and negative pairs via adding two label (1, 0) for each pairs. The third step is to build the neural network through Tensor flow library and implement contrastive loss function. The final step is to test and evaluate our Siamese network. The whole program will run on Google Colaboratory with cloud GPU.

Experiment results

Base network

The structure of simple base network is shown in below picture:

```
model = keras.models.Sequential()
model.add(keras.layers.Dense(128, activation='relu'))
model.add(keras.layers.Dense(128, activation='relu'))
model.add(keras.layers.Dropout(0.2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(128, activation='relu'))
```

To evaluate the influence of epochs for accuracy, we set the epochs to 20 times and get the accuracy for from epochs 1 to 20. The below picture demonstrate the test result for three test pairs:

```
48432/48432 [==============================] - 37s 760us/step - loss: 0.0701 - accuracy: 0.9338 - val_loss: 0.0923 - val_accuracy: 0.8948
Epoch 19/20
48432/48432 [==============================] - 37s 760us/step - loss: 0.0694 - accuracy: 0.9350 - val_loss: 0.0945 - val_accuracy: 0.8890
Epoch 20/20
48432/48432 [==============================] - 37s 760us/step - loss: 0.0691 - accuracy: 0.9349 - val_loss: 0.0927 - val_accuracy: 0.8952
* Accuracy on training set: 94.24%
* Accuracy on test set: 88.94%
```

Above picture show the result for the first test pair. Since the first test pair of data also is tanning data, both training accuracy and test accuracy are very high. In particular, the accuracy on training set is 94.24% and the accuracy on test set is 88.94%.

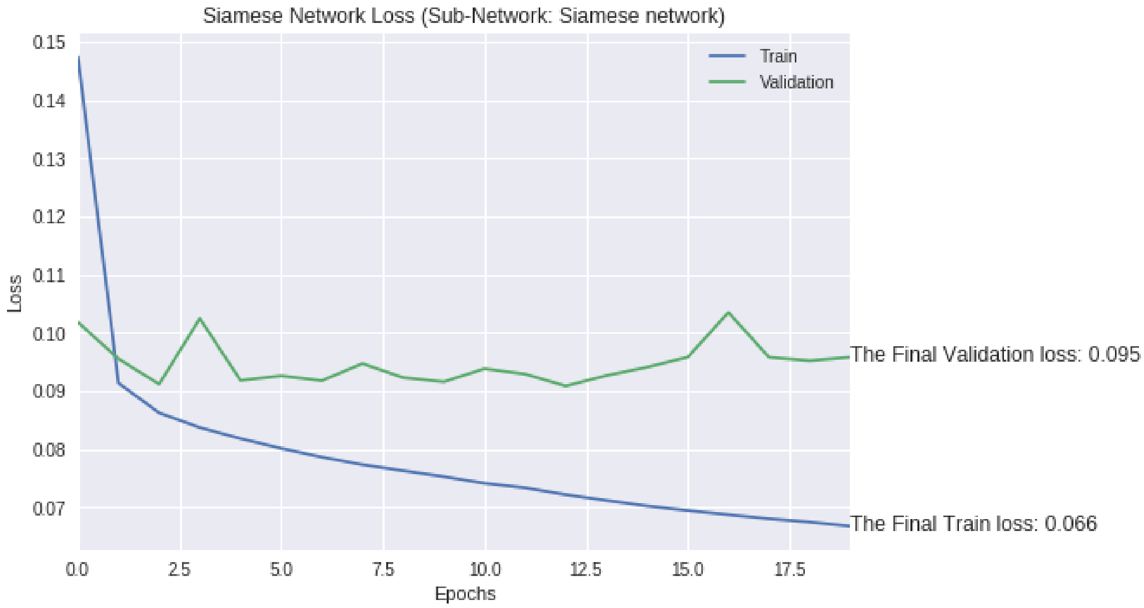
Then, we use the union pair of test data, which contain the training dataset ([2, 3, 4, 5, 6, 7]) and testing dataset ([0, 1, 8, 9]). The result demonstrate on below picture:

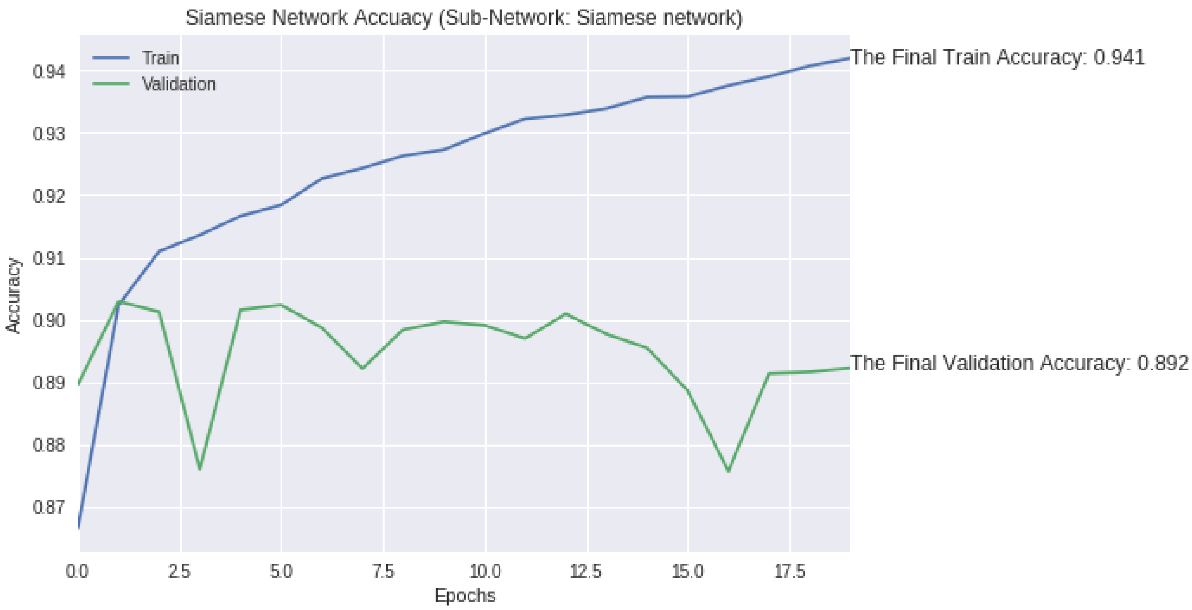
```
48432/48432 [==============================] - 37s 767us/step - loss: 0.0510 - accuracy: 0.9581 - val_loss: 0.0800 - val_accuracy: 0.9185
Epoch 19/20
48432/48432 [==============================] - 37s 767us/step - loss: 0.0503 - accuracy: 0.9595 - val_loss: 0.0804 - val_accuracy: 0.9135
Epoch 20/20
48432/48432 [==============================] - 37s 774us/step - loss: 0.0495 - accuracy: 0.9604 - val_loss: 0.0837 - val_accuracy: 0.9041
* Accuracy on training set: 96.15%
* Accuracy on test set: 82.65%
```

Based on the result, the accuracy of training set is still very high (96.15%). However, the accuracy on test set is 82.65%. Compare to the first pair result, the accuracy on test set has reduced from94.24% to 82.65%. The reason is that we add new dataset [0,1,8,9] to the test set and base neural network is easily to make mistake to recognize new image of dataset. Next, we will only use [0,1,8,9] to test the precision level of base network. The under picture show the result :

```
48432/48432 [==============================] - 37s 767us/step - loss: 0.0680 - accuracy: 0.9390 - val_loss: 0.0958 - val_accuracy: 0.8913
Epoch 19/20
48432/48432 [==============================] - 37s 760us/step - loss: 0.0675 - accuracy: 0.9407 - val_loss: 0.0952 - val_accuracy: 0.8916
Epoch 20/20
48432/48432 [==============================] - 37s 759us/step - loss: 0.0668 - accuracy: 0.9419 - val_loss: 0.0958 - val_accuracy: 0.8922
* Accuracy on training set: 94.75%
* Accuracy on test set: 67.08%
```

According to the above result, accuracy in test set is 67.08%. Moreover, we plot the training and validation error vs epoch to find a suitable epoch for our base network. The below two graph demonstrate the relationship.





According to above Siamese network loss graph, when epoch increase from 1 to 5, both validation loss and train loss decrease significantly. After epoch increase from 5 to 20, validation loss become stable around 0.095 and the training loss only reduce slightly. Based on the Siamese network accuracy, after epoch reaching 5 times, validation accuracy become stable around 0.89. And the train precision level increase slightly. Therefore, 5 epoch is enough for the base neural network. In next step we will set 5 epoch to base Simple network and run it 10 times to get the average accuracy. The accuracy table is shown below:

Base Network First Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	92.26%	93.78%	92.59%	92.59%	94.56%	91.62%	91.86%	91.76%	94.16%	92.67%
Test accuracy	90.05%	91.74%	90.85%	90.85%	92.55%	89.21%	89.61%	89.13%	92.27%	90.83%

Base network Second Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	91.58%	95.04%	92.92%	92.92%	91.36%	90.97%	92.68%	93.30%	93.49%	92.61%
Test accuracy	81.70%	83.25%	81.55%	81.55%	80.70%	80.25%	81.81%	81.58%	81.02%	82.21%

Base Network Third Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	93.24%	89.80%	98.99%	94.14%	92.92%	94.41%	88.55%	94.27%	93.81%	91.56%
Test accuracy	68.25%	66.95%	69.40%	68.82%	69.96%	67.84%	71.14%	69.94%	68.50%	69.79%

Base Network Average Accuracy Level

Test Dataset	Average training accuracy	Average testing accuracy
First pair (dataset ([2, 3, 4, 5, 6, 7])	92.47%	90.44%
Second pair (dataset[0-9])	92.10%	81.96%
Third pair (dataset [0,1,8,9])	93.24%	69.02%

Complex CNN network

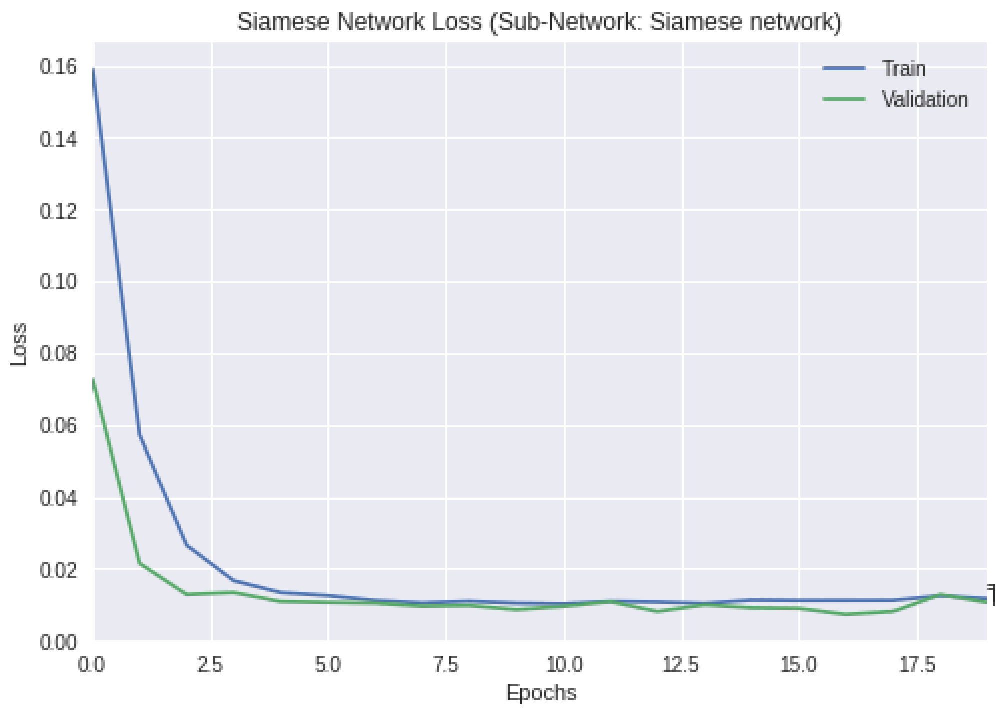
In previous research, several neural network models have been created to implement Siamese network, such as ImageNet, Alexnet and GoogLeNet. Our complex CNN network is based on the Alexnet neural network. In basic architecture of Alexnet, it contains first five convolutional layers and latter layers as three full connection layers. To improve the efficiency of our network, we remove one convolution layer and one fully connect layer.

To build a basic convolution layer, the Conv2D is usually layer 0. For layer 0 Conv2D, we set 16 kernels that the size is 5*5 and “relu” activation function to extract the features of training dataset. Also, we need to set the input shape parameter to the required shape of dataset. Then, we use MaxPooling2D as layer 2 to choose the highest value of a 2*2 matrix from previous layer .Next, using the same step, three convolution layer are created. In fully connect layer for avoiding overfitting, we create Dropout as layer to remove 30% and 40% data that come from previous respond layer. In addition, we utilize Flatten function to convert the previous layer data to two dimension layer. Finally, dense layer is utilized to represents 128 output classes. The whole CNN network is completed with three convolutional layer and three fully connect layer.

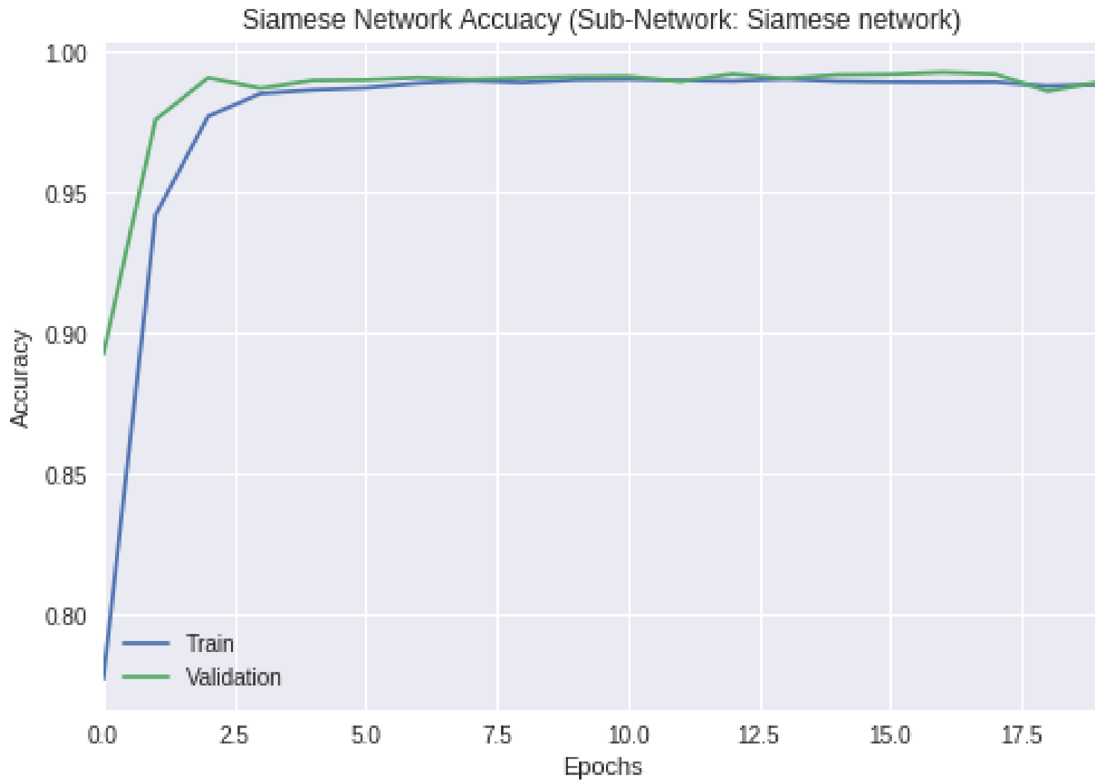
The whole structure is demonstrated on below picture:

```
model = keras.models.Sequential()
# Layer 1  convolutional layer
model.add(keras.layers.Conv2D(16, (5, 5), input_shape = input_shape, padding='same')) # 16 kernals
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2)))# pooling with size 2x2

# Layer 2 convolutional layer
model.add(keras.layers.Conv2D(32, (3, 3), padding='same'))# 32 kernals of size 3x3, case-senetive
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling2D(pool_size=(2, 2)))# pooling with size 2x2

# Layer 3 convolutional layer
model.add(keras.layers.ZeroPadding2D((1,1))) #zero padding with size 1x1
model.add(keras.layers.Conv2D(128, (1, 1), padding='same'))# 64 kernals of size 1x1, case-senetive
model.add(keras.layers.Activation('relu'))

# Layer 4  fully-connected layer
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(64, kernel_initializer=keras.initializers.glorot_normal(seed=None)))#
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.Dropout(0.3)) #avoid overfitting

# Layer 5 fully-connected layer
model.add(keras.layers.Dense(64, kernel_initializer=keras.initializers.glorot_normal(seed=None)))#
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.Dropout(0.4))

# Layer 6 fully-connected layer
model.add(keras.layers.Dense(128, kernel_initializer=keras.initializers.glorot_normal(seed=None)))
model.add(keras.layers.Activation('relu'))
```

To find the suitable epoch, we plot the training and validation error vs time graph that demonstrate in below picture.



Based on the above picture, the train and validation loss decrease significantly from 1 epoch to three epochs. When epochs is between 4 to 20, both train loss and validation loss is steady around 0.01. Thereby, three epochs is enough for our complex CNN network to diminish the loss. Furthermore, the below graph show the relationship between epochs and training accuracy:



Based on the above graph, the accuracy on training set raise significantly when epoch increase from 1 to 2 in both train and validation accuracy. For the complex CNN network, only run one epoch , the training accuracy have already reached 95% which is enough to recognize images. To improve this network efficiency, we decide to set the epoch to one. Then run this network 10 times to acquire the average accuracy. The below picture show one of the result for three pairs testing data.

```
----------------------------Testing First Pair:-----------------------------
Train on 48432 samples, validate on 12636 samples
Epoch 1/1
48432/48432 [==============================] - 15s 312us/step - loss: 0.1705 - accuracy: 0.7580 - val_loss: 0.0688 - val_accuracy: 0.8975
* Accuracy on training set: 90.04%
* Accuracy on test set: 89.21%
----------------------------Testing Second Union Pair:-----------------------------
Train on 48432 samples, validate on 12636 samples
Epoch 1/1
48432/48432 [==============================] - 16s 322us/step - loss: 0.1441 - accuracy: 0.8208 - val_loss: 0.0246 - val_accuracy: 0.9879
* Accuracy on training set: 99.05%
* Accuracy on test set: 86.14%
----------------------------Testing Third Pair:-----------------------------
Train on 48432 samples, validate on 12636 samples
Epoch 1/1
48432/48432 [==============================] - 16s 321us/step - loss: 0.1570 - accuracy: 0.7823 - val_loss: 0.0621 - val_accuracy: 0.9182
* Accuracy on training set: 92.16%
* Accuracy on test set: 76.27%
----------------------------------------------------------------------
```

CNN Network First Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	90.04%	95.36%	95.52%	97.24%	96.04%	94.84%	97.51%	91.90%	96.03%	94.54%
Test accuracy	89.21%	95.12%	95.22%	98.68%	95.54%	94.55%	97.09%	91.00%	96.03%	94.26%

CNN Network Second Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	99.05%	91.72%	94.02%	95.54%	92.47%	95.11%	91.95%	92.02%	92.46%	95.41%
Test accuracy	86.14%	84.13%	85.10%	84.66%	83.52%	84.21%	83.51%	83.38%	84.10%	85.36%

CNN Network Third Pair Accuracy Level

Time	1	2	3	4	5	6	7	8	9	10
Training accuracy	92.16%	95.48%	95.56%	95.63%	95.68%	93.13%	90.38%	98.55%	92.69%	95.57%
Test accuracy	76.27%	71.58%	78.18%	74.16%	76.33%	76.20%	76.37%	66.35%	76.27%	75.24%

CNN Network Average Accuracy Level

Test Dataset	Average training accuracy	Average testing accuracy
First pair (dataset ([2, 3, 4, 5, 6, 7])	92.29%	91.74%
Second pair (dataset[0-9])	97.23%	85.75%
Third pair (dataset [0,1,8,9])	93.87%	75.24%

Discussion

Time cost and accuracy level are two standard to consider these two Siamese network. From the experimental results, we can analyse performance for these two Siamese network. For both two Siamese network, the epoch times directly influence the training accuracy. According to the plot of training accuracy level vs epochs, the training accuracy level is raising when the epoch times is increased. However, after epochs reach a specific threshold, the training accuracy will be stabilized to a value. Hence, it is important to choose a suitable epoch times for balancing the accuracy level and time cost. Furthermore, for both base network and CNN network, the average test accuracy of first pair dataset is higher than the accuracy level of second dataset. The test accuracy level of second pair of dataset is higher than precision level of third pair of dataset. The reason is that the first test pair of dataset is training dataset. Thus, the test precision level of first pair is almost same as training accuracy level. The second pair of test dataset combine the training dataset([2,3,4,5,6,7)] and new test dataset[(0,1,8,9)] together, which cause the test precision level decrease. These two Siamese network will make some mistakes when they try to recognize new images. That is explain these two network have lowest average accuracy level of third pairs of dataset compose d by new test dataset ([0,1,8,9]) .

For base network, the average training precision is around 90%. The precision level of first, second and third pair of are 90.44%, 81.96% and 69.02%. The result show the base network can recognize well in first pair and second pair of test dataset. Compared to CNN network, the precision level of second and third pair of test dataset is less than the CNN network performance (85.75% and 75.34%).

For CNN network, the average training precision is around 94%. The average precision level of first, second and third pair dataset are 91.74%, 85.75% and 75.24%. Compared to base network, the performance of Complex CNN network is 6% higher than simple CNN network where it use third pair of test dataset. It is clear to notice the complex CNN network is better recognize new images than base network.

In conclusion, since the third pair of test data is new dataset, it is proper to use it to justify the performance of two Siamese network. As the complex CNN network have 75% test accuracy level for recognizing new images, which is higher than simple base network (69%). Also, this complex CNN network do not require high epochs to get high train accuracy and low loss value. It also can reduce the time cost to train this neural network. Therefore, we recommend the complex CNN network to recognize manta rays.

Group member:

Student1 name:

Student 1 number:

Student 2 name: Yixi Zhou

Student 2 number: n9599291

Student 3 name:

Student 3 number: