CAB420 - Assignment 1B

Callum McNeilage - n10482652

Connor Smith - n9991051

Queensland University of Technology

Problem 1: Training and Adapting Deep Networks

Network Design

The design of the network we chose for question 1, matches that used in the week 4 example 2 code for training a CNN. The only difference is that the input layer and output layers have been changed to match the data we've been given. The design is small consisting of three 2d convolutions with a max pooling in between. It ends with a flatten layer and 2 dense layers to classify the image. The full structure can be seen in appendix 1. Due to issues with struggling to get the code to work in time, for the pre-trained network we used the vgg_2stage_CIFAR_bigger model from blackboard and trained it. The model is quite deep consisting of 24 layers and has been pre-trained on CIFAR dataset. CIFAR is a dataset also used for classification but used for different object types like automobile, airplane, etc. The model had the last layer removed so that a dense layer could be added for classification. The full layout of the network can be seen in appendix 2.

Type of Augmentation

The data was augmented before being passed through to the rest of the model. The types of augmentation used were random rotations and random zoom. Both amounts of change were kept relatively small, as to avoid confusing the model with similar numbers i.e. rotate a 5 too much and it could look like a 2. The pictures are also small, only being 32x32 so a small value of zoom was used. Random translations were not used due to the small image size and random flips were not used as it would cause confusion. Examples of the augmented data are shown below

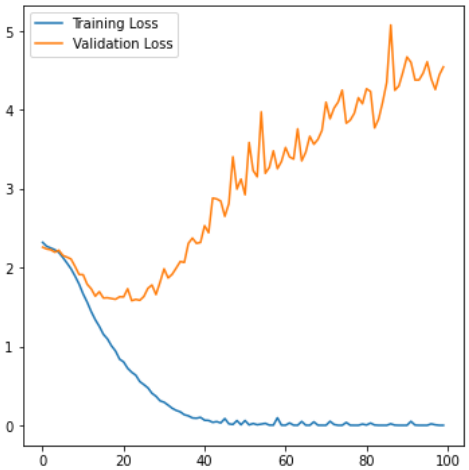
Figure 1

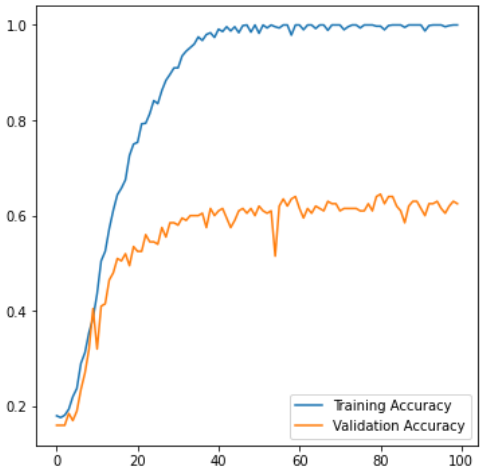
Computational Constraints

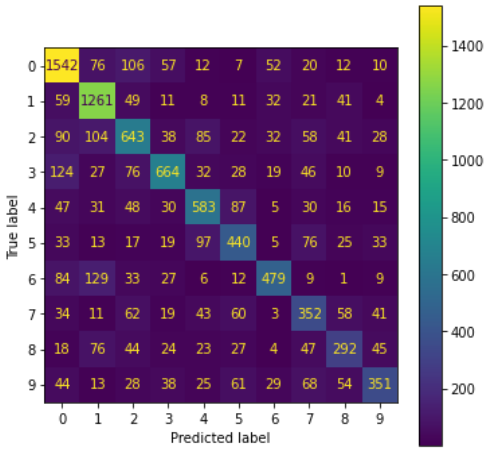
Due to having a decently powerful PC to train the models, all models were trained for 100 epochs and with a batch size of 32. Despite having sufficient power, a smaller model was chosen for the first two sections as it allowed for faster training and testing of the model.

Comparisons of Models

After training the basic model with non-augmented data, it produced accuracy of 100% on the training set







and 62% with validation. The training performance can be seen below:

The basic model with augmented data produced an accuracy of 98% on the training set and 75% on validation. The training performance can be seen below:

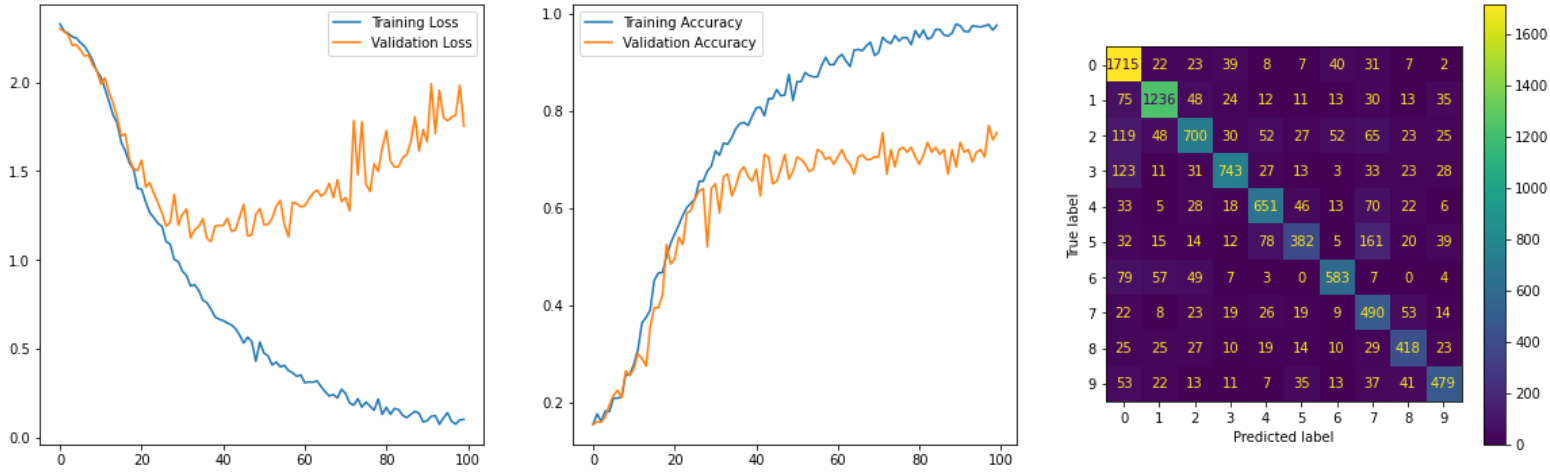


Figure 3

Comparing the non augmented data vs the augmented data, the model with the augmented data performed better resulting in 10% more accuracy in validation. The model without augmented data seemed to have overfit to the dataset with the last few epochs showing 100% accuracy on training with no increase to validation.

```
Epoch 95/100
25/25 [==============================] - 0s 7ms/step - loss: 6.5398e-05 - accuracy: 1.0000 -
val_loss: 4.4679 - val_accuracy: 0.6300
Epoch 96/100
25/25 [==============================] - 0s 6ms/step - loss: 3.6226e-05 - accuracy: 1.0000 -
val_loss: 4.6105 - val_accuracy: 0.6150
Epoch 97/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0027 - accuracy: 0.9995 -
val_loss: 4.3929 - val_accuracy: 0.6050
Epoch 98/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0089 - accuracy: 0.9981 -
val_loss: 4.2590 - val_accuracy: 0.6200
Epoch 99/100
25/25 [==============================] - 0s 6ms/step - loss: 2.1359e-04 - accuracy: 1.0000 -
val_loss: 4.4443 - val_accuracy: 0.6300
Epoch 100/100
25/25 [==============================] - 0s 6ms/step - loss: 5.5059e-05 - accuracy: 1.0000 -
val_loss: 4.5467 - val_accuracy: 0.6250
```

The augmented data prevented the overfitting. The model never hit 100% accuracy and the increased validation accuracy shows this.

```
Epoch 95/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1128 - accuracy: 0.9688 -
val_loss: 1.8018 - val_accuracy: 0.7150
Epoch 96/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1132 - accuracy: 0.9748 -
val_loss: 1.7821 - val_accuracy: 0.7200
Epoch 97/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0605 - accuracy: 0.9845 -
val_loss: 1.8031 - val_accuracy: 0.7050
Epoch 98/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1006 - accuracy: 0.9696 -
val_loss: 1.8132 - val_accuracy: 0.7700
Epoch 99/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0673 - accuracy: 0.9723 -
val_loss: 1.9831 - val_accuracy: 0.7400
Epoch 100/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0806 - accuracy: 0.9814 -
val_loss: 1.7520 - val_accuracy: 0.7550
```

Augmenting the data has produced better results, allowing for the model to learn from ‘new’ data. Whilst actual new data would produce increased results, the small dataset made this not possible, thus augmenting the data is the better option. In terms of speed, both models were reasonably quick with the non augmented model being slightly faster at 7ms/step vs 17ms/step for the augmented.

Looking at the pre-trained model, it produced a disappointing 14% on the training set and 9.5% on validation. The training performance can be seen below:

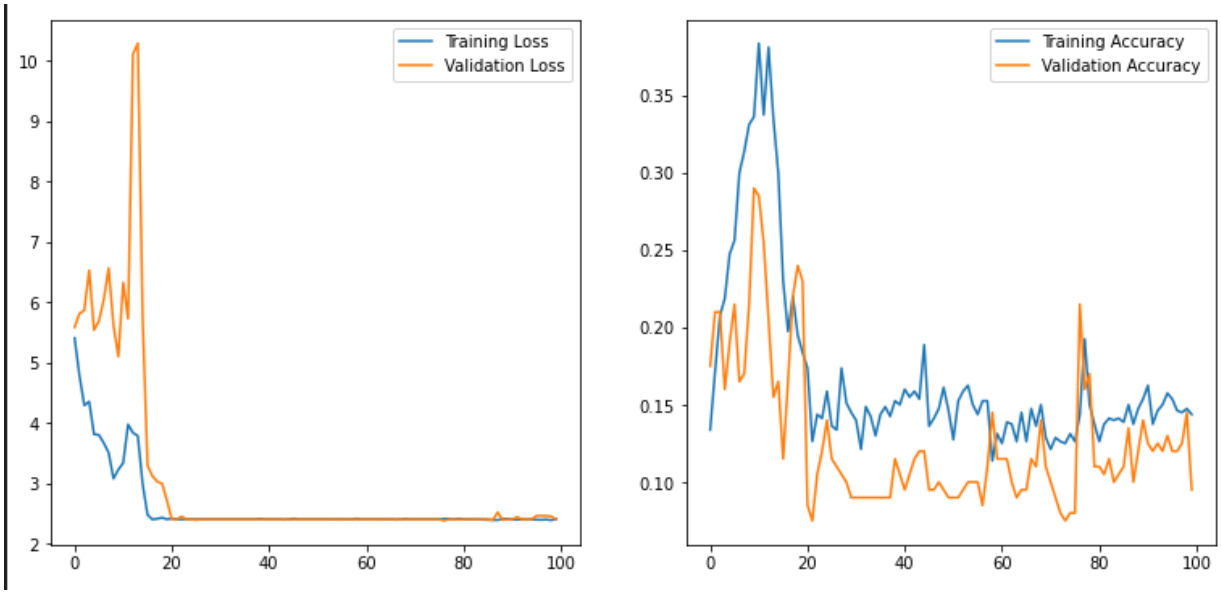


Figure 4

The bad accuracy could be due to the previous model being used to classify objects rather than numbers. The lack of data could also indicate the bad performance. For around 15 epochs performance was as expected, slowly rising before both training and validation accuracy sharply dropped.

```
Epoch 15/100
25/25 [==============================] - 2s 88ms/step - loss: 3.1901 - accuracy: 0.3148 -
val_loss: 5.7288 - val_accuracy: 0.1650
Epoch 16/100
25/25 [==============================] - 2s 86ms/step - loss: 2.5232 - accuracy: 0.2516 -
val_loss: 3.2861 - val_accuracy: 0.1150
Epoch 17/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3934 - accuracy: 0.2013 -
val_loss: 3.1159 - val_accuracy: 0.1650
Epoch 18/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3929 - accuracy: 0.2358 -
val_loss: 3.0206 - val_accuracy: 0.2200
Epoch 19/100
25/25 [==============================] - 2s 88ms/step - loss: 2.4584 - accuracy: 0.2222 -
val_loss: 2.9889 - val_accuracy: 0.2400
Epoch 20/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1897 -
val_loss: 2.7053 - val_accuracy: 0.2300
```

Augmented data was not used so the lack of new samples may not have been enough to sufficiently retrain the model. There is also the possibility that human error is involved. Compared to the other models, this model performed significantly worse in both accuracy and time. The time is due to the increased depth of the model though and cannot be compared to the other simple models.

Overall, the simple model with augmented data performed the best on this limited dataset. It allowed for more ‘new’ data for the model to learn off of and helped to prevent overfitting.

Problem 2: Person Re-Identification

This task required us to match a detected person to a gallery of previously seen people, and determine their identity. For this task we have developed a non-deep learning model and a deep learning model for the provided portion of the Market-1501 dataset. As we were provided with a folder of images rather than a .mat file for this task we used a combination of glob and cv2.imread functions in order to append each image into training, gallery and probe arrays.

```
train = []
gnd = []
files = glob.glob('Data/Q2/Q2/Q2/Training/*.jpg')
for myfile in files:
    image = cv2.imread(myfile, 0)
    gnd.append(myfile[23:27])
    train.append(image)

train = np.array(train)
gnd = np.array(gnd)

print('Training shape: ', train.shape)
print('Training gnd: ', gnd[1:10])
```

Evaluation of Non-Deep Learning method

To develop a non-deep learning method to solve this problem, we decided to follow the Week 6 PCA examples. We opted to use the Gallery dataset as our test dataset so that we could use the Probe dataset for validation later.

We reshaped our feature sets for the three datasets into a 2D dataset by combining the x and y values to allow us to use the decomposition.PCA() function to fit our data on the training set. We then used the developed model as a pca.transform on each of our three datasets.

```
nsamples, nx, ny = train_fea.shape
d2_train_dataset = train_fea.reshape((nsamples,nx*ny))

nsamples, nx, ny = test_fea.shape
d2_test_dataset = test_fea.reshape((nsamples,nx*ny))
```

```
nsamples, nx, ny = test_pro.shape
d2_probe_dataset = test_pro.reshape((nsamples,nx*ny))

pca = decomposition.PCA()
pca.fit(d2_train_dataset)
transformed = pca.transform(d2_train_dataset)
transformed_test = pca.transform(d2_test_dataset)
transformed_probe = pca.transform(d2_probe_dataset)
```

As a sanity check at this stage, we completed a 90%, 95% and 99% reconstruction of image in position pca.components_[0] as well as generating a cumulative sum graph of the explained variance.

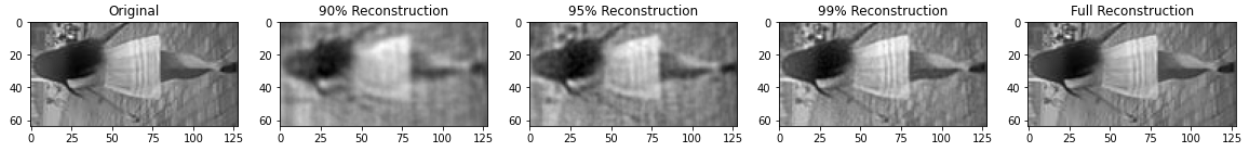


Figure 5

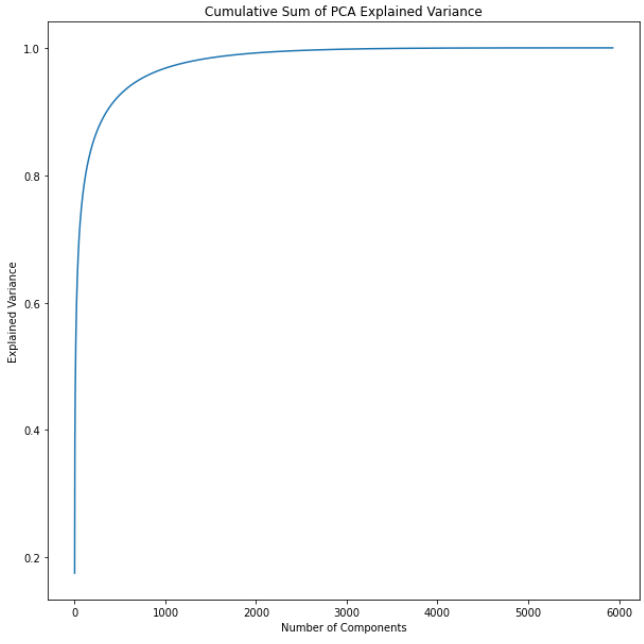


Figure 6

Finally, we graphed a CMC Curve for the 90%, 95% and 99% datasets respectively in order to compare performance of the model given a slightly reduced dataset.

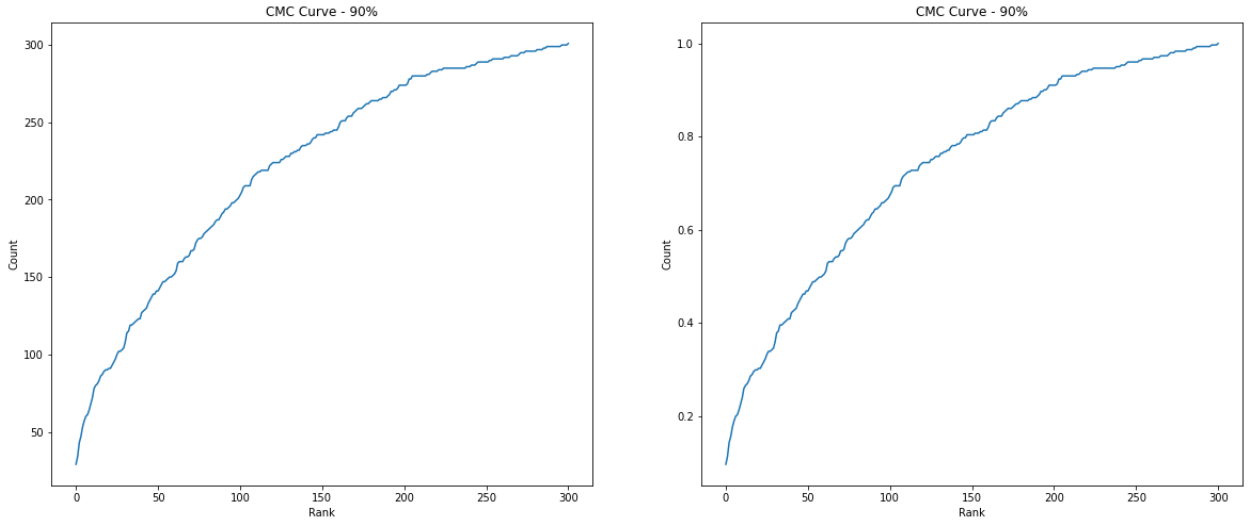


Figure 7

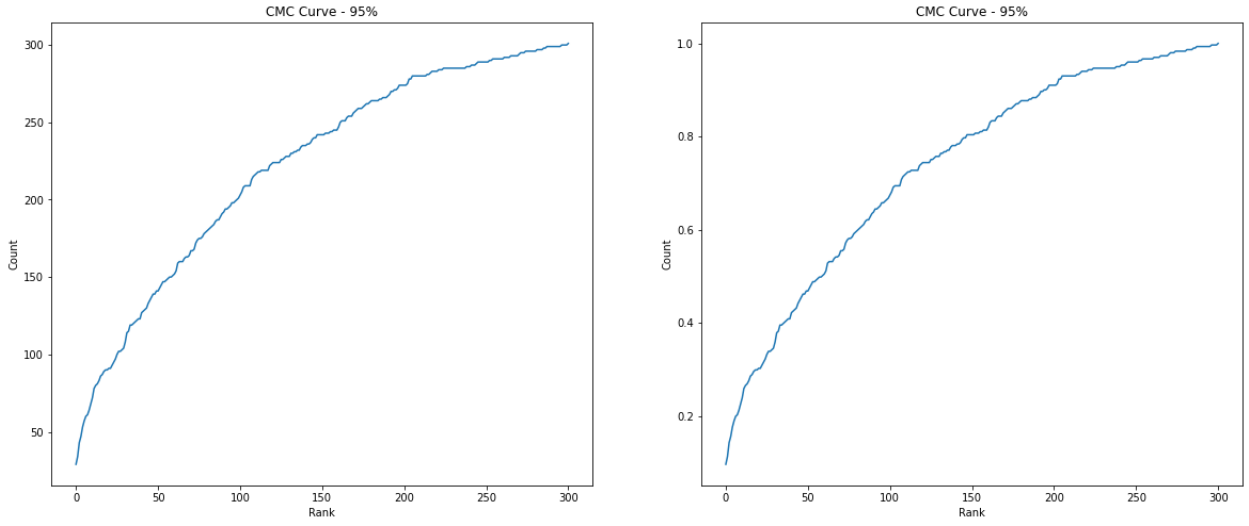


Figure 8

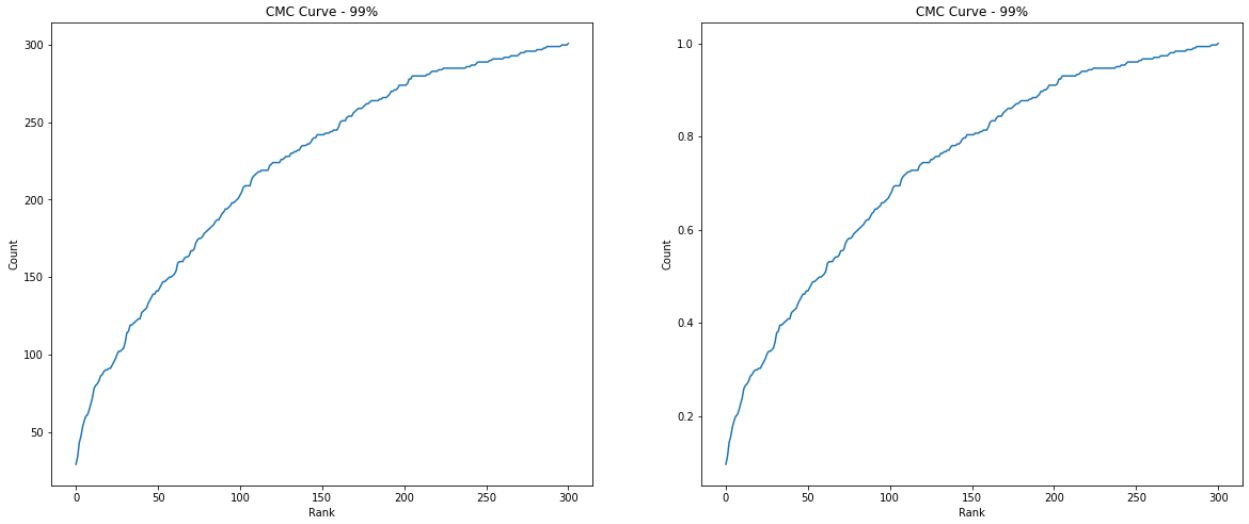


Figure 9

Evaluation of Deep Learning Method

For our Deep Learning method, it was decided to follow the Siamese model completed in Week 7 Example 1 (See Appendix 3 for model and network summary).

We used the PairGenerator code from the example to find pairs of images in the training dataset with the same gnd (ID number) before training the Siamese Network 100 times with a batch size of 32.

```
batch_size = 32
training_gen = PairGenerator(train_fea, gnd, batch_size)

siamese_test_x, siamese_test_y = GetSiameseData(test_fea, test_gnd, 5933)
siamese_network.fit(training_gen, steps_per_epoch = 256 // batch_size, epochs=100, validation_data =
(siamese_test_x, siamese_test_y))
```

We then used the trained network to predict gallery and probe values.

```
gal_x, gal_y = GetSiameseData(test_fea, test_gnd, 301)
gal_predict = siamese_network.predict(gal_x)

pro_x, pro_y = GetSiameseData(test_pro_fea, test_pro_gnd, 301)
pro_predict = siamese_network.predict(pro_x)
```

Finally, we generated a CMC Curve for the network. Unfortunately, due to time constraints we were not able to complete this for Top-1, Top-5 and Top-10 datasets, instead the CMC is for whole generated dataset with no information removed.

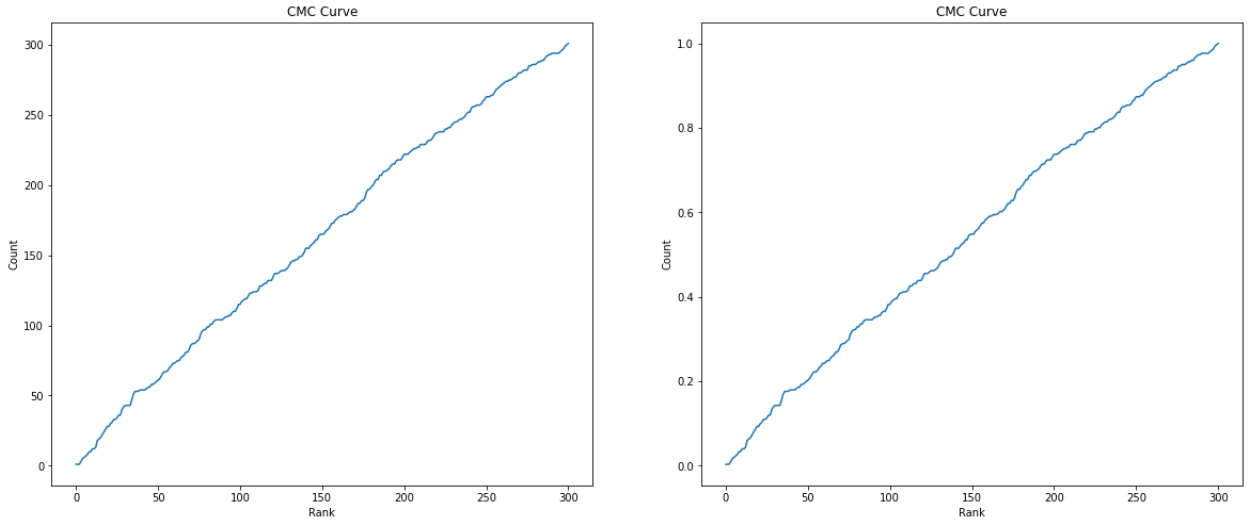


Figure 10

Comparison of Performance

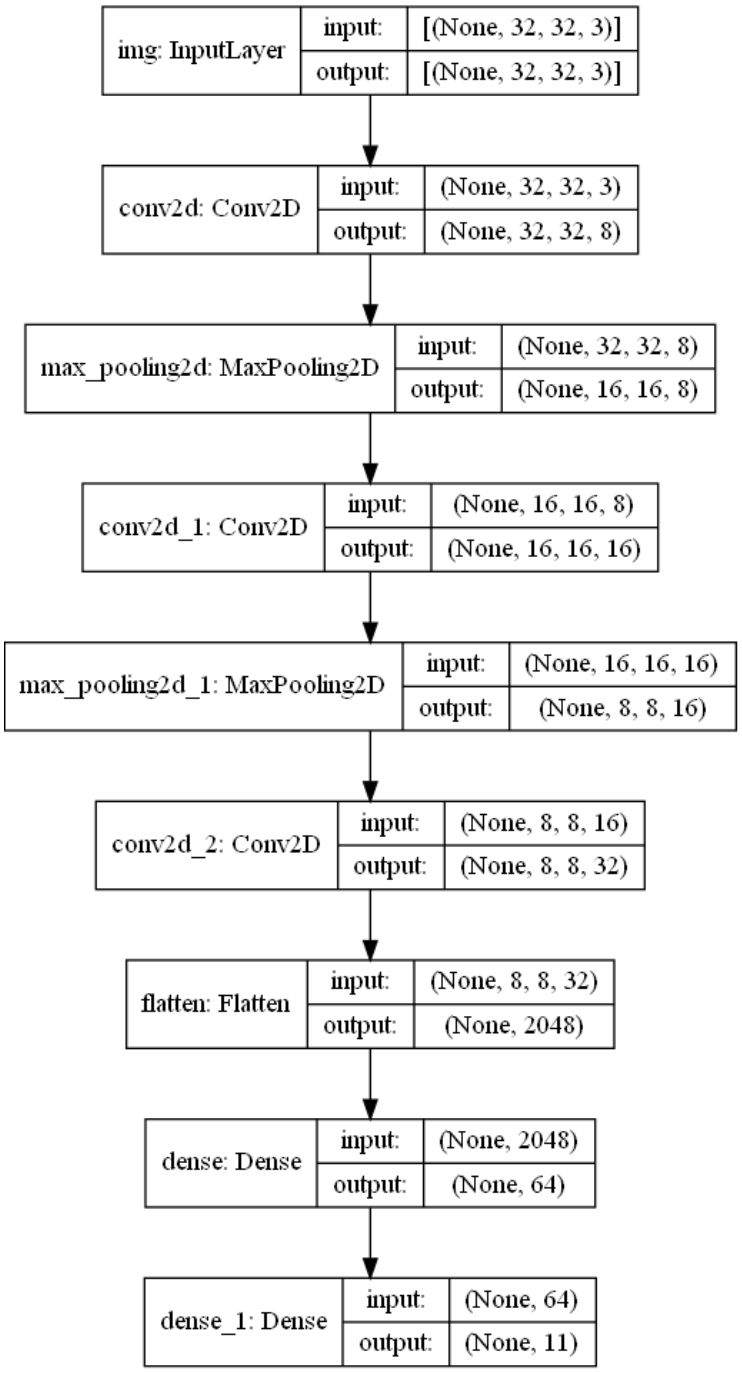
Both Deep Learning and Non-Deep Learning approaches have their strengths and weaknesses. A Deep Learning model is perfect for large datasets of unstructured and unlabeled data, however, if there is not enough data for the model to train on or if there is a lot of similar data, a Deep Learning model will be highly inaccurate or in the worst case will be no different from randomly selecting any ID in the dataset for a given image. For our dataset, our Deep Learning model was only achieving between 50% and 60% accuracy (See Appendix 4). We believe this is because of a combination of only being provided a subset of the entire Market-1501 dataset as well as using greyscale images that, at times, could be particularly low quality. As can be seen when

comparing the CMC Curves of the Deep Learning solution in Figure 10 with our PCA models in Figures 7, 8 & 9, this resulted in significantly less accuracy generated in our Deep Learning model. However, this does not mean our PCA model is the best model possible as the CMC Curves generated are less than ideal for an accurate model, where the optimal curve would approach 1.0 as quickly as possible.

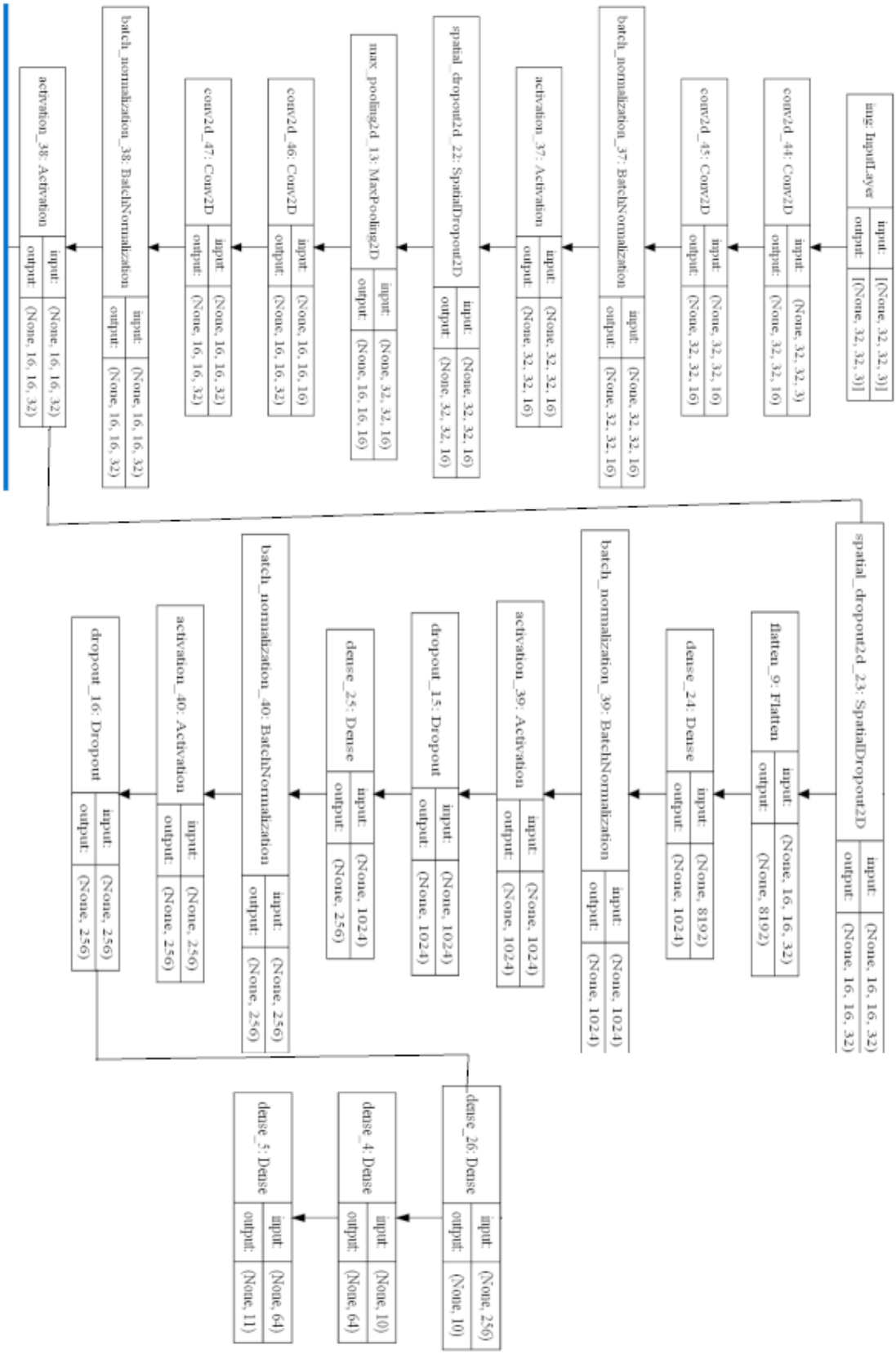
We believe that if we were able to use colour images instead of greyscale, this would provide a wider range of diversity within images that are seen by the model and would lead to less false positive matches and improve overall accuracy of the model. Additionally, having access to the entire dataset may improve accuracy as well.

Appendix

Appendix 1



Appendix 2


ing: InputLayer
input: [(None, 32, 32, 3)]
output: [(None, 32, 32, 3)]
conv2d_44: Conv2D
input: (None, 32, 32, 3)
output: (None, 32, 32, 16)
conv2d_45: Conv2D
input: (None, 32, 32, 16)
output: (None, 32, 32, 16)
batch_normalization_37: BatchNormalization
input: (None, 32, 32, 16)
output: (None, 32, 32, 16)
activation_37: Activation
input: (None, 32, 32, 16)
output: (None, 32, 32, 16)
spatial_dropout2d_22: SpatialDropout2D
input: (None, 32, 32, 16)
output: (None, 32, 32, 16)
max_pooling2d_13: MaxPooling2D
input: (None, 32, 32, 16)
output: (None, 16, 16, 16)
conv2d_46: Conv2D
input: (None, 16, 16, 16)
output: (None, 16, 16, 32)
conv2d_47: Conv2D
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
batch_normalization_38: BatchNormalization
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
activation_38: Activation
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
spatial_dropout2d_23: SpatialDropout2D
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
flatten_9: Flatten
input: (None, 16, 16, 32)
output: (None, 8192)
dense_24: Dense
input: (None, 8192)
output: (None, 1024)
batch_normalization_39: BatchNormalization
input: (None, 1024)
output: (None, 1024)
activation_39: Activation
input: (None, 1024)
output: (None, 1024)
dropout_15: Dropout
input: (None, 1024)
output: (None, 1024)
dense_25: Dense
input: (None, 1024)
output: (None, 256)
batch_normalization_40: BatchNormalization
input: (None, 256)
output: (None, 256)
activation_40: Activation
input: (None, 256)
output: (None, 256)
dropout_16: Dropout
input: (None, 256)
output: (None, 256)
dense_26: Dense
input: (None, 256)
output: (None, 10)
dense_4: Dense
input: (None, 10)
output: (None, 64)
dense_5: Dense
input: (None, 64)
output: (None, 11)

Appendix 3

```
Model: "SiameseBranch"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 64, 1)]      0
_________________________________________________________________
conv2d (Conv2D)              (None, 128, 64, 8)        80
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 64, 8)        584
_________________________________________________________________
batch_normalization (BatchNo (None, 128, 64, 8)        32
_________________________________________________________________
activation (Activation)      (None, 128, 64, 8)        0
_________________________________________________________________
spatial_dropout2d (SpatialDr (None, 128, 64, 8)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 32, 8)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 32, 16)        1168
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 64, 32, 16)        2320
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 32, 16)        64
_________________________________________________________________
activation_1 (Activation)    (None, 64, 32, 16)        0
_________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 64, 32, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 16, 16)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 16, 32)        4640
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 32, 16, 32)        9248
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 16, 32)        128
_________________________________________________________________
activation_2 (Activation)    (None, 32, 16, 32)        0
_________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 32, 16, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 16384)             0
_________________________________________________________________
dense (Dense)                (None, 256)               4194560
_________________________________________________________________
batch_normalization_3 (Batch (None, 256)               1024
_________________________________________________________________
activation_3 (Activation)    (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                8224
=================================================================
Total params: 4,222,072
Trainable params: 4,221,448
Non-trainable params: 624
_________________________________________________________________
```

```
Model: "SiameseNetwork"
__________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==========================================================================
InputA (InputLayer)             [(None, 128, 64, 1)] 0
__________________________________________________________________________
InputB (InputLayer)             [(None, 128, 64, 1)] 0
__________________________________________________________________________
SiameseBranch (Functional)      (None, 32)           4222072     InputA[0][0]
                                                                 InputB[0][0]
__________________________________________________________________________
concatenate (Concatenate)       (None, 64)           0           SiameseBranch[0][0]
                                                                 SiameseBranch[1][0]
__________________________________________________________________________
dense_2 (Dense)                 (None, 128)          8320        concatenate[0][0]
__________________________________________________________________________
dense_3 (Dense)                 (None, 1)            129         dense_2[0][0]
==========================================================================
Total params: 4,230,521
Trainable params: 4,229,897
Non-trainable params: 624
__________________________________________________________________________
```

Appendix 4

```
Epoch 91/100
8/8 [==============================] - 24s 3s/step - loss: 0.6609 - accuracy: 0.6099 - val_loss: 0.6991 - val_accuracy: 0.5036
Epoch 92/100
8/8 [==============================] - 24s 3s/step - loss: 0.6250 - accuracy: 0.6646 - val_loss: 0.7367 - val_accuracy: 0.5004
Epoch 93/100
8/8 [==============================] - 24s 3s/step - loss: 0.6300 - accuracy: 0.6229 - val_loss: 0.7089 - val_accuracy: 0.5458
Epoch 94/100
8/8 [==============================] - 24s 3s/step - loss: 0.6381 - accuracy: 0.6497 - val_loss: 0.6773 - val_accuracy: 0.5626
Epoch 95/100
8/8 [==============================] - 24s 3s/step - loss: 0.6322 - accuracy: 0.6455 - val_loss: 0.7090 - val_accuracy: 0.5326
Epoch 96/100
8/8 [==============================] - 25s 4s/step - loss: 0.6311 - accuracy: 0.6260 - val_loss: 0.6744 - val_accuracy: 0.5840
Epoch 97/100
8/8 [==============================] - 26s 4s/step - loss: 0.6104 - accuracy: 0.6881 - val_loss: 0.7293 - val_accuracy: 0.5233
Epoch 98/100
8/8 [==============================] - 27s 4s/step - loss: 0.6583 - accuracy: 0.6302 - val_loss: 0.6909 - val_accuracy: 0.5788
Epoch 99/100
8/8 [==============================] - 25s 3s/step - loss: 0.6313 - accuracy: 0.5985 - val_loss: 0.7182 - val_accuracy: 0.5029
Epoch 100/100
8/8 [==============================] - 25s 4s/step - loss: 0.5951 - accuracy: 0.6626 - val_loss: 0.6781 - val_accuracy: 0.5581

<tensorflow.python.keras.callbacks.History at 0x172c8467be0>
```

Assignment 1B - Question 1

Training and Adapting Deep Networks

In [1]:
```
import pandas
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import scipy.stats as stats
from scipy.io import loadmat

import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from tensorboard import notebook
from tensorflow.keras.preprocessing.image import Iterator

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
```

```
INFO:tensorflow:Enabling eager execution
INFO:tensorflow:Enabling v2 tensorshape
INFO:tensorflow:Enabling resource variables
INFO:tensorflow:Enabling tensor equality
INFO:tensorflow:Enabling control flow v2
```

Load data from training and testing sets

Use scipy.io loadmat function

In [2]:
```
train = loadmat('Assessment 1B/Data/Q1/q1_train.mat')
test = loadmat('Assessment 1B/Data/Q1/q1_test.mat')
```

Sanity check

In [3]:
```
(x_test, y_test) = (test['test_X'], test['test_Y'])
x_test = x_test.astype('float32') / 255
x_test = np.transpose(x_test, (3,0,1,2))
fig = plt.figure(figsize=[20, 20])
for i in range(100):
    ax = fig.add_subplot(10, 10, i + 1)
    ax.imshow(x_test[i,:,:,:])
```

In [4]:

```
(x_train, y_train) = (train['train_X'], train['train_Y'])
x_train = x_train.astype('float32') / 255
x_train = np.transpose(x_train, (3,0,1,2))
fig = plt.figure(figsize=[20, 20])
for i in range(100):
    ax = fig.add_subplot(10, 10, i + 1)
    ax.imshow(x_train[i,:,:,:])
```

Keras Model

Taken from Week 4 example 2

In [7]:
```
def get_model():
    inputs = keras.Input(shape=(32, 32, 3, ), name='img')
    x = layers.Conv2D(filters=8, kernel_size=(3,3), activation='relu', padding='same
    x = layers.MaxPool2D(pool_size=(2, 2))(x)
    x = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='sam
    x = layers.MaxPool2D(pool_size=(2, 2))(x)
    x = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='sam
    x = layers.Flatten()(x)
    x = layers.Dense(64, activation='relu')(x)
    outputs = layers.Dense(11, activation='softmax')(x)

    model_cnn = keras.Model(inputs=inputs, outputs=outputs, name='SVHN_CNN_Model')

    return model_cnn
```

In [8]:
```
model_cnn = get_model()
model_cnn.compile(loss=keras.losses.SparseCategoricalCrossentropy(from_logits=False)
                  optimizer=keras.optimizers.RMSprop(),
                  metrics=['accuracy'])
history = model_cnn.fit(x_train, y_train,
```

```
                    batch_size=32,
                    epochs=100,
                    validation_split=0.2, verbose=True)
```

```
Epoch 1/100
25/25 [==============================] - 1s 12ms/step - loss: 2.3392 - accuracy: 0.1
842 - val_loss: 2.2591 - val_accuracy: 0.1600
Epoch 2/100
25/25 [==============================] - 0s 8ms/step - loss: 2.2829 - accuracy: 0.16
01 - val_loss: 2.2361 - val_accuracy: 0.1600
Epoch 3/100
25/25 [==============================] - 0s 7ms/step - loss: 2.2669 - accuracy: 0.16
93 - val_loss: 2.2271 - val_accuracy: 0.1600
Epoch 4/100
25/25 [==============================] - 0s 7ms/step - loss: 2.2311 - accuracy: 0.18
52 - val_loss: 2.1957 - val_accuracy: 0.1850
Epoch 5/100
25/25 [==============================] - 0s 7ms/step - loss: 2.1943 - accuracy: 0.21
89 - val_loss: 2.2205 - val_accuracy: 0.1700
Epoch 6/100
25/25 [==============================] - 0s 7ms/step - loss: 2.1290 - accuracy: 0.24
07 - val_loss: 2.1540 - val_accuracy: 0.1900
Epoch 7/100
25/25 [==============================] - 0s 7ms/step - loss: 2.0824 - accuracy: 0.28
23 - val_loss: 2.1329 - val_accuracy: 0.2350
Epoch 8/100
25/25 [==============================] - 0s 7ms/step - loss: 1.9888 - accuracy: 0.31
46 - val_loss: 2.1094 - val_accuracy: 0.2700
Epoch 9/100
25/25 [==============================] - 0s 7ms/step - loss: 1.9191 - accuracy: 0.35
44 - val_loss: 2.0151 - val_accuracy: 0.3200
Epoch 10/100
25/25 [==============================] - 0s 7ms/step - loss: 1.8385 - accuracy: 0.35
35 - val_loss: 1.9130 - val_accuracy: 0.4050
Epoch 11/100
25/25 [==============================] - 0s 7ms/step - loss: 1.6995 - accuracy: 0.41
10 - val_loss: 1.9100 - val_accuracy: 0.3200
Epoch 12/100
25/25 [==============================] - 0s 7ms/step - loss: 1.6047 - accuracy: 0.48
85 - val_loss: 1.7891 - val_accuracy: 0.4100
Epoch 13/100
25/25 [==============================] - 0s 7ms/step - loss: 1.4571 - accuracy: 0.52
57 - val_loss: 1.7312 - val_accuracy: 0.4150
Epoch 14/100
25/25 [==============================] - 0s 7ms/step - loss: 1.3687 - accuracy: 0.57
31 - val_loss: 1.6379 - val_accuracy: 0.4650
Epoch 15/100
25/25 [==============================] - 0s 7ms/step - loss: 1.2993 - accuracy: 0.61
05 - val_loss: 1.6950 - val_accuracy: 0.4800
Epoch 16/100
25/25 [==============================] - 0s 7ms/step - loss: 1.1661 - accuracy: 0.63
79 - val_loss: 1.6146 - val_accuracy: 0.5100
Epoch 17/100
25/25 [==============================] - 0s 6ms/step - loss: 1.0976 - accuracy: 0.66
25 - val_loss: 1.6179 - val_accuracy: 0.5050
Epoch 18/100
25/25 [==============================] - 0s 6ms/step - loss: 1.0123 - accuracy: 0.67
89 - val_loss: 1.6076 - val_accuracy: 0.5200
Epoch 19/100
25/25 [==============================] - 0s 7ms/step - loss: 0.8959 - accuracy: 0.73
85 - val_loss: 1.5985 - val_accuracy: 0.4950
Epoch 20/100
25/25 [==============================] - 0s 7ms/step - loss: 0.8545 - accuracy: 0.74
17 - val_loss: 1.6301 - val_accuracy: 0.5350
Epoch 21/100
25/25 [==============================] - 0s 6ms/step - loss: 0.7934 - accuracy: 0.75
45 - val_loss: 1.6284 - val_accuracy: 0.5250
Epoch 22/100
25/25 [==============================] - 0s 6ms/step - loss: 0.7433 - accuracy: 0.78
```

```
35 - val_loss: 1.7321 - val_accuracy: 0.5250
Epoch 23/100
25/25 [==============================] - 0s 7ms/step - loss: 0.6867 - accuracy: 0.79
18 - val_loss: 1.5809 - val_accuracy: 0.5600
Epoch 24/100
25/25 [==============================] - 0s 7ms/step - loss: 0.6205 - accuracy: 0.81
84 - val_loss: 1.5974 - val_accuracy: 0.5450
Epoch 25/100
25/25 [==============================] - 0s 6ms/step - loss: 0.5628 - accuracy: 0.83
86 - val_loss: 1.5859 - val_accuracy: 0.5450
Epoch 26/100
25/25 [==============================] - 0s 6ms/step - loss: 0.5082 - accuracy: 0.83
86 - val_loss: 1.6338 - val_accuracy: 0.5400
Epoch 27/100
25/25 [==============================] - 0s 6ms/step - loss: 0.4736 - accuracy: 0.86
72 - val_loss: 1.7374 - val_accuracy: 0.5750
Epoch 28/100
25/25 [==============================] - 0s 6ms/step - loss: 0.3597 - accuracy: 0.90
59 - val_loss: 1.7793 - val_accuracy: 0.5550
Epoch 29/100
25/25 [==============================] - 0s 7ms/step - loss: 0.3229 - accuracy: 0.91
54 - val_loss: 1.6590 - val_accuracy: 0.5850
Epoch 30/100
25/25 [==============================] - 0s 6ms/step - loss: 0.2849 - accuracy: 0.92
45 - val_loss: 1.8151 - val_accuracy: 0.5850
Epoch 31/100
25/25 [==============================] - 0s 7ms/step - loss: 0.2735 - accuracy: 0.90
43 - val_loss: 1.9845 - val_accuracy: 0.5800
Epoch 32/100
25/25 [==============================] - 0s 7ms/step - loss: 0.2474 - accuracy: 0.94
00 - val_loss: 1.8692 - val_accuracy: 0.5950
Epoch 33/100
25/25 [==============================] - 0s 7ms/step - loss: 0.1882 - accuracy: 0.96
31 - val_loss: 1.9158 - val_accuracy: 0.5900
Epoch 34/100
25/25 [==============================] - 0s 7ms/step - loss: 0.2097 - accuracy: 0.94
67 - val_loss: 1.9954 - val_accuracy: 0.6000
Epoch 35/100
25/25 [==============================] - 0s 6ms/step - loss: 0.1757 - accuracy: 0.96
07 - val_loss: 2.0778 - val_accuracy: 0.6000
Epoch 36/100
25/25 [==============================] - 0s 7ms/step - loss: 0.1155 - accuracy: 0.98
51 - val_loss: 2.0665 - val_accuracy: 0.6000
Epoch 37/100
25/25 [==============================] - 0s 6ms/step - loss: 0.1074 - accuracy: 0.97
46 - val_loss: 2.3078 - val_accuracy: 0.6050
Epoch 38/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0901 - accuracy: 0.98
13 - val_loss: 2.3750 - val_accuracy: 0.5750
Epoch 39/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0796 - accuracy: 0.99
03 - val_loss: 2.3080 - val_accuracy: 0.6150
Epoch 40/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0757 - accuracy: 0.98
66 - val_loss: 2.3198 - val_accuracy: 0.6000
Epoch 41/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0479 - accuracy: 0.99
73 - val_loss: 2.5324 - val_accuracy: 0.6100
Epoch 42/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0483 - accuracy: 0.99
34 - val_loss: 2.4425 - val_accuracy: 0.6150
Epoch 43/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0384 - accuracy: 0.99
72 - val_loss: 2.8828 - val_accuracy: 0.5950
Epoch 44/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0429 - accuracy: 0.99
05 - val_loss: 2.8712 - val_accuracy: 0.5750
Epoch 45/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0446 - accuracy: 0.99
```

```
32 - val_loss: 2.8440 - val_accuracy: 0.5900
Epoch 46/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0485 - accuracy: 0.99
25 - val_loss: 2.6509 - val_accuracy: 0.6100
Epoch 47/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0262 - accuracy: 0.99
76 - val_loss: 2.8116 - val_accuracy: 0.6150
Epoch 48/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0107 - accuracy: 1.00
00 - val_loss: 3.4063 - val_accuracy: 0.6050
Epoch 49/100
25/25 [==============================] - 0s 7ms/step - loss: 0.1114 - accuracy: 0.96
98 - val_loss: 2.9967 - val_accuracy: 0.6150
Epoch 50/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0078 - accuracy: 1.00
00 - val_loss: 3.1230 - val_accuracy: 0.6000
Epoch 51/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0568 - accuracy: 0.98
53 - val_loss: 2.9246 - val_accuracy: 0.6200
Epoch 52/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0062 - accuracy: 1.00
00 - val_loss: 3.5863 - val_accuracy: 0.6100
Epoch 53/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0118 - accuracy: 0.99
83 - val_loss: 3.2322 - val_accuracy: 0.6050
Epoch 54/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0088 - accuracy: 1.00
00 - val_loss: 3.1525 - val_accuracy: 0.6100
Epoch 55/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0042 - accuracy: 0.99
97 - val_loss: 3.9751 - val_accuracy: 0.5150
Epoch 56/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0427 - accuracy: 0.98
79 - val_loss: 3.1969 - val_accuracy: 0.6200
Epoch 57/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0027 - accuracy: 1.00
00 - val_loss: 3.2716 - val_accuracy: 0.6350
Epoch 58/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0021 - accuracy: 1.00
00 - val_loss: 3.4809 - val_accuracy: 0.6200
Epoch 59/100
25/25 [==============================] - 0s 7ms/step - loss: 0.1551 - accuracy: 0.96
49 - val_loss: 3.2550 - val_accuracy: 0.6350
Epoch 60/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0017 - accuracy: 1.00
00 - val_loss: 3.3469 - val_accuracy: 0.6400
Epoch 61/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0011 - accuracy: 1.00
00 - val_loss: 3.5242 - val_accuracy: 0.6150
Epoch 62/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0253 - accuracy: 0.99
16 - val_loss: 3.4038 - val_accuracy: 0.5950
Epoch 63/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0011 - accuracy: 1.00
00 - val_loss: 3.3770 - val_accuracy: 0.6150
Epoch 64/100
25/25 [==============================] - 0s 7ms/step - loss: 6.8874e-04 - accuracy:
1.0000 - val_loss: 3.7608 - val_accuracy: 0.6050
Epoch 65/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0422 - accuracy: 0.99
35 - val_loss: 3.3547 - val_accuracy: 0.6200
Epoch 66/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0014 - accuracy: 1.00
00 - val_loss: 3.4650 - val_accuracy: 0.6150
Epoch 67/100
25/25 [==============================] - 0s 6ms/step - loss: 5.2853e-04 - accuracy:
1.0000 - val_loss: 3.6665 - val_accuracy: 0.6100
Epoch 68/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0176 - accuracy: 0.99
```

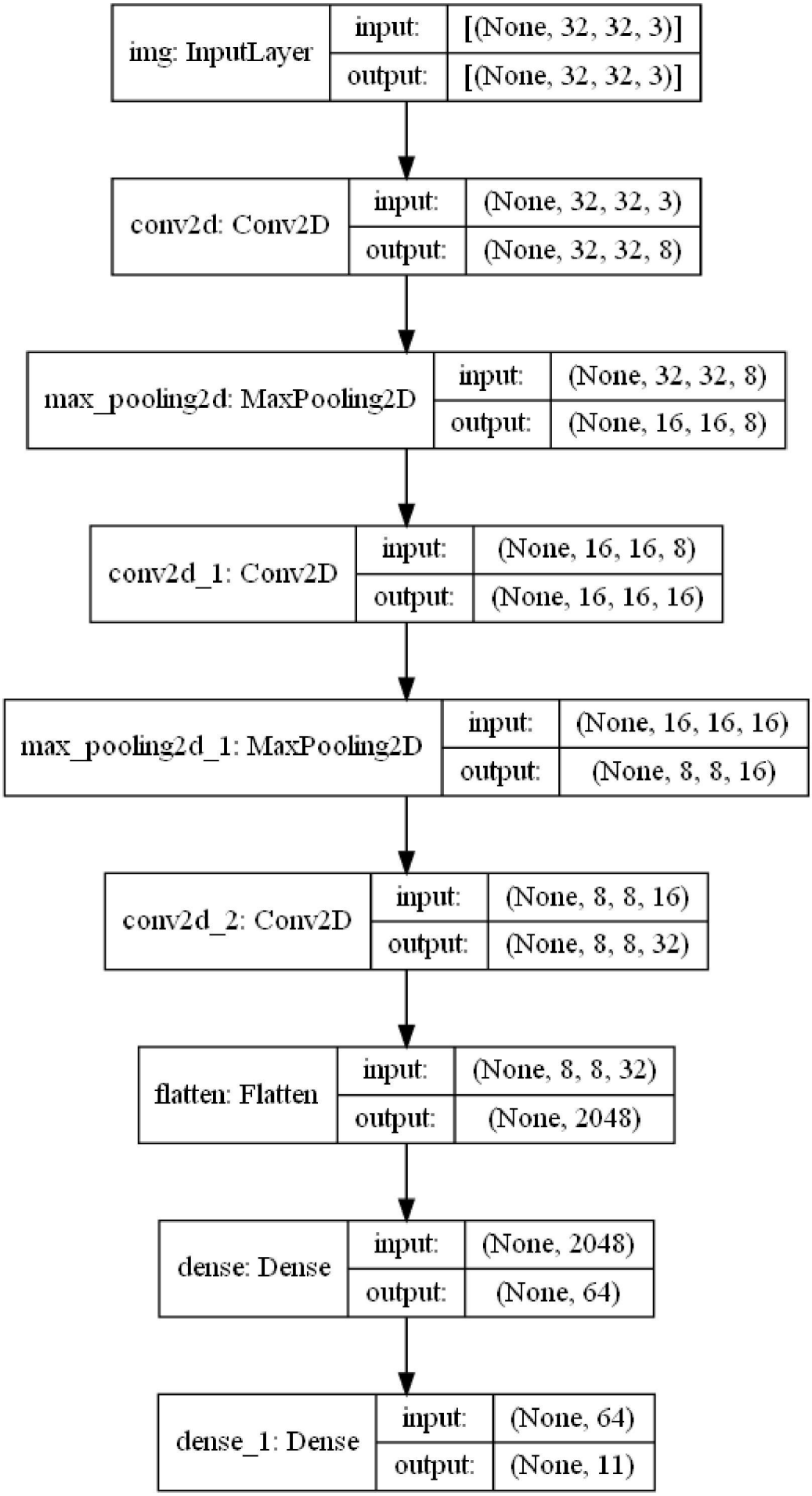
```
54 - val_loss: 3.5647 - val_accuracy: 0.6300
Epoch 69/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0018 - accuracy: 1.00
00 - val_loss: 3.6264 - val_accuracy: 0.6250
Epoch 70/100
25/25 [==============================] - 0s 7ms/step - loss: 5.5660e-04 - accuracy:
1.0000 - val_loss: 3.7398 - val_accuracy: 0.6250
Epoch 71/100
25/25 [==============================] - 0s 7ms/step - loss: 2.7192e-04 - accuracy:
1.0000 - val_loss: 4.0987 - val_accuracy: 0.6100
Epoch 72/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0147 - accuracy: 0.99
72 - val_loss: 3.8884 - val_accuracy: 0.6150
Epoch 73/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0116 - accuracy: 0.99
58 - val_loss: 4.0240 - val_accuracy: 0.6150
Epoch 74/100
25/25 [==============================] - 0s 7ms/step - loss: 4.4911e-04 - accuracy:
1.0000 - val_loss: 4.1002 - val_accuracy: 0.6150
Epoch 75/100
25/25 [==============================] - 0s 6ms/step - loss: 2.3179e-04 - accuracy:
1.0000 - val_loss: 4.2516 - val_accuracy: 0.6150
Epoch 76/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0128 - accuracy: 0.99
76 - val_loss: 3.8321 - val_accuracy: 0.6100
Epoch 77/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0016 - accuracy: 1.00
00 - val_loss: 3.8717 - val_accuracy: 0.6100
Epoch 78/100
25/25 [==============================] - 0s 7ms/step - loss: 2.8195e-04 - accuracy:
1.0000 - val_loss: 3.9587 - val_accuracy: 0.6250
Epoch 79/100
25/25 [==============================] - 0s 6ms/step - loss: 1.5004e-04 - accuracy:
1.0000 - val_loss: 4.1553 - val_accuracy: 0.6100
Epoch 80/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0037 - accuracy: 0.99
93 - val_loss: 4.0792 - val_accuracy: 0.6400
Epoch 81/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0031 - accuracy: 0.99
85 - val_loss: 4.2685 - val_accuracy: 0.6450
Epoch 82/100
25/25 [==============================] - 0s 8ms/step - loss: 0.0121 - accuracy: 0.99
68 - val_loss: 4.2351 - val_accuracy: 0.6250
Epoch 83/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0070 - accuracy: 0.99
72 - val_loss: 3.7722 - val_accuracy: 0.6400
Epoch 84/100
25/25 [==============================] - 0s 7ms/step - loss: 2.8698e-04 - accuracy:
1.0000 - val_loss: 3.8851 - val_accuracy: 0.6400
Epoch 85/100
25/25 [==============================] - 0s 7ms/step - loss: 1.5842e-04 - accuracy:
1.0000 - val_loss: 4.0996 - val_accuracy: 0.6200
Epoch 86/100
25/25 [==============================] - 0s 7ms/step - loss: 8.9892e-05 - accuracy:
1.0000 - val_loss: 4.3499 - val_accuracy: 0.6100
Epoch 87/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0149 - accuracy: 0.99
68 - val_loss: 5.0779 - val_accuracy: 0.5850
Epoch 88/100
25/25 [==============================] - 0s 7ms/step - loss: 0.0048 - accuracy: 1.00
00 - val_loss: 4.2498 - val_accuracy: 0.6200
Epoch 89/100
25/25 [==============================] - 0s 7ms/step - loss: 1.3800e-04 - accuracy:
1.0000 - val_loss: 4.3040 - val_accuracy: 0.6300
Epoch 90/100
25/25 [==============================] - 0s 7ms/step - loss: 8.5759e-05 - accuracy:
1.0000 - val_loss: 4.4793 - val_accuracy: 0.6300
Epoch 91/100
25/25 [==============================] - 0s 7ms/step - loss: 4.2072e-05 - accuracy:
```

```
1.0000 - val_loss: 4.6720 - val_accuracy: 0.6150
Epoch 92/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0318 - accuracy: 0.99
18 - val_loss: 4.6031 - val_accuracy: 0.6000
Epoch 93/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0067 - accuracy: 0.99
66 - val_loss: 4.3799 - val_accuracy: 0.6250
Epoch 94/100
25/25 [==============================] - 0s 6ms/step - loss: 1.1916e-04 - accuracy:
1.0000 - val_loss: 4.3813 - val_accuracy: 0.6250
Epoch 95/100
25/25 [==============================] - 0s 7ms/step - loss: 6.5398e-05 - accuracy:
1.0000 - val_loss: 4.4679 - val_accuracy: 0.6300
Epoch 96/100
25/25 [==============================] - 0s 6ms/step - loss: 3.6226e-05 - accuracy:
1.0000 - val_loss: 4.6105 - val_accuracy: 0.6150
Epoch 97/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0027 - accuracy: 0.99
95 - val_loss: 4.3929 - val_accuracy: 0.6050
Epoch 98/100
25/25 [==============================] - 0s 6ms/step - loss: 0.0089 - accuracy: 0.99
81 - val_loss: 4.2590 - val_accuracy: 0.6200
Epoch 99/100
25/25 [==============================] - 0s 6ms/step - loss: 2.1359e-04 - accuracy:
1.0000 - val_loss: 4.4443 - val_accuracy: 0.6300
Epoch 100/100
25/25 [==============================] - 0s 6ms/step - loss: 5.5059e-05 - accuracy:
1.0000 - val_loss: 4.5467 - val_accuracy: 0.6250
```

In [9]:

```
keras.utils.plot_model(model_cnn, show_shapes=True)
```

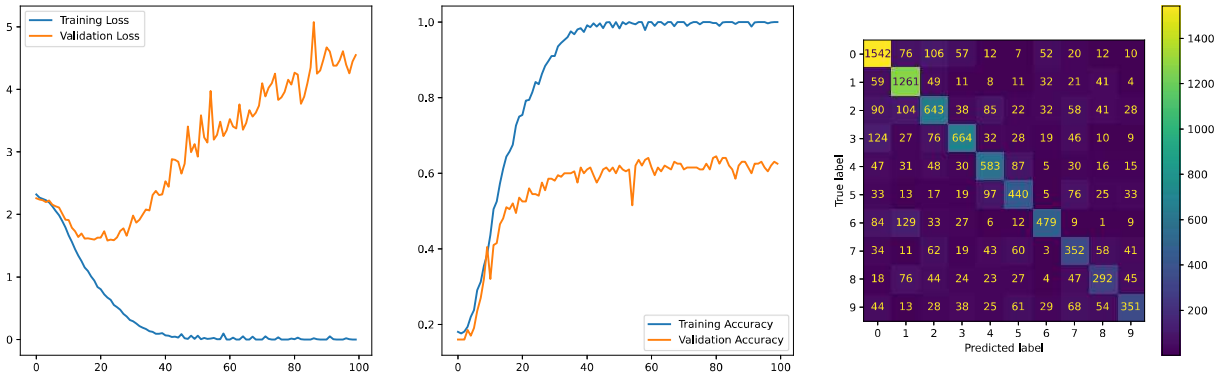
Out[9]:



In [10]:

```
def plot_training(history, model, x_test, y_test):
    fig = plt.figure(figsize=[20, 6])
    ax = fig.add_subplot(1, 3, 1)
```

```
    ax.plot(history.history['loss'], label="Training Loss")
    ax.plot(history.history['val_loss'], label="Validation Loss")
    ax.legend()

    ax = fig.add_subplot(1, 3, 2)
    ax.plot(history.history['accuracy'], label="Training Accuracy")
    ax.plot(history.history['val_accuracy'], label="Validation Accuracy")
    ax.legend()

    pred = model.predict(x_test)
    indexes = tf.argmax(pred, axis=1)
    i = tf.cast([], tf.int32)
    indexes = tf.gather_nd(indexes, i)

    cm = confusion_matrix(y_test, indexes)
    ax = fig.add_subplot(1, 3, 3)
    c = ConfusionMatrixDisplay(cm, display_labels=range(10))

    c.plot(ax = ax)

plot_training(history, model_cnn, x_test, y_test)
```

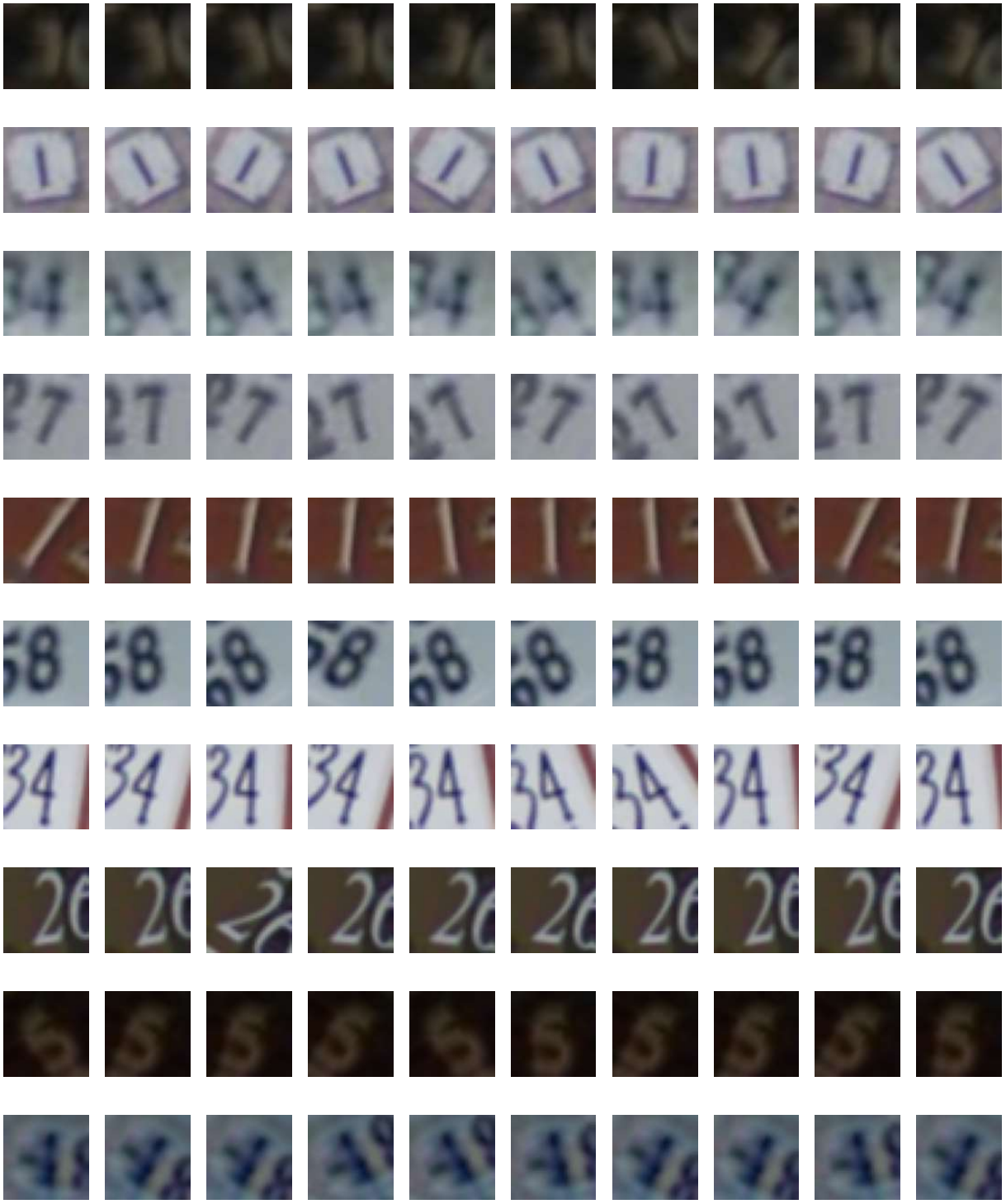


Data Augmentation

Week 5 example 3

In [11]:

```
data_augmentation = keras.Sequential([
  layers.experimental.preprocessing.RandomRotation(0.1),
  layers.experimental.preprocessing.RandomZoom(0.03),
])

fig = plt.figure(figsize=[20, 25])
for i in range(10):
    for j in range(10):
        ax = fig.add_subplot(10, 10, i*10 + (j + 1))
        augmented_image = data_augmentation(tf.expand_dims(x_train[i,:,:,:],0))
        plt.imshow(augmented_image[0])
        plt.axis("off")
```



In [12]:

```
def get_model_augment():
    inputs = keras.Input(shape=(32, 32, 3, ), name='img')
    augmented = data_augmentation(inputs)
    x = layers.Conv2D(filters=8, kernel_size=(3,3), activation='relu', padding='same
    x = layers.MaxPool2D(pool_size=(2, 2))(x)
    x = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='sam
    x = layers.MaxPool2D(pool_size=(2, 2))(x)
    x = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='sam
    x = layers.Flatten()(x)
    x = layers.Dense(64, activation='relu')(x)
    outputs = layers.Dense(11, activation='softmax')(x)

    model_cnn_aug = keras.Model(inputs=inputs, outputs=outputs, name='SVHN_CNN_Model

    return model_cnn_aug
```

In [13]:
```
model_cnn_aug = get_model_augment()
model_cnn_aug.compile(loss=keras.losses.SparseCategoricalCrossentropy(from_logits=Fa
                  optimizer=keras.optimizers.RMSprop(),
                  metrics=['accuracy'])
history = model_cnn_aug.fit(x_train, y_train,
                        batch_size=32,
                        epochs=100,
                        validation_split=0.2, verbose=True)
```

```
Epoch 1/100
25/25 [==============================] - 1s 17ms/step - loss: 2.3611 - accuracy: 0.1
351 - val_loss: 2.2990 - val_accuracy: 0.1550
Epoch 2/100
25/25 [==============================] - 0s 13ms/step - loss: 2.2841 - accuracy: 0.1
697 - val_loss: 2.2840 - val_accuracy: 0.1600
Epoch 3/100
25/25 [==============================] - 0s 13ms/step - loss: 2.2784 - accuracy: 0.1
601 - val_loss: 2.2666 - val_accuracy: 0.1600
Epoch 4/100
25/25 [==============================] - 0s 13ms/step - loss: 2.2597 - accuracy: 0.1
939 - val_loss: 2.2062 - val_accuracy: 0.1700
Epoch 5/100
25/25 [==============================] - 0s 13ms/step - loss: 2.2449 - accuracy: 0.1
889 - val_loss: 2.2097 - val_accuracy: 0.1950
Epoch 6/100
25/25 [==============================] - 0s 13ms/step - loss: 2.2137 - accuracy: 0.2
091 - val_loss: 2.1801 - val_accuracy: 0.2150
Epoch 7/100
25/25 [==============================] - 0s 13ms/step - loss: 2.1927 - accuracy: 0.2
111 - val_loss: 2.1471 - val_accuracy: 0.2250
Epoch 8/100
25/25 [==============================] - 0s 13ms/step - loss: 2.1468 - accuracy: 0.2
146 - val_loss: 2.1513 - val_accuracy: 0.2100
Epoch 9/100
25/25 [==============================] - 0s 13ms/step - loss: 2.0988 - accuracy: 0.2
675 - val_loss: 2.0972 - val_accuracy: 0.2650
Epoch 10/100
25/25 [==============================] - 0s 13ms/step - loss: 2.0544 - accuracy: 0.2
617 - val_loss: 2.0659 - val_accuracy: 0.2550
Epoch 11/100
25/25 [==============================] - 0s 13ms/step - loss: 1.9974 - accuracy: 0.2
939 - val_loss: 1.9888 - val_accuracy: 0.2700
Epoch 12/100
25/25 [==============================] - 0s 12ms/step - loss: 1.9378 - accuracy: 0.3
270 - val_loss: 2.0238 - val_accuracy: 0.3000
Epoch 13/100
25/25 [==============================] - 0s 12ms/step - loss: 1.9018 - accuracy: 0.3
729 - val_loss: 1.9429 - val_accuracy: 0.2900
Epoch 14/100
25/25 [==============================] - 0s 12ms/step - loss: 1.8419 - accuracy: 0.3
723 - val_loss: 1.8821 - val_accuracy: 0.2750
Epoch 15/100
25/25 [==============================] - 0s 13ms/step - loss: 1.7679 - accuracy: 0.3
810 - val_loss: 1.8009 - val_accuracy: 0.3550
Epoch 16/100
25/25 [==============================] - 0s 13ms/step - loss: 1.7114 - accuracy: 0.4
357 - val_loss: 1.6978 - val_accuracy: 0.3950
Epoch 17/100
25/25 [==============================] - 0s 13ms/step - loss: 1.6610 - accuracy: 0.4
688 - val_loss: 1.7075 - val_accuracy: 0.3950
Epoch 18/100
25/25 [==============================] - 0s 12ms/step - loss: 1.5225 - accuracy: 0.4
864 - val_loss: 1.5675 - val_accuracy: 0.4200
Epoch 19/100
25/25 [==============================] - 0s 12ms/step - loss: 1.4689 - accuracy: 0.5
288 - val_loss: 1.5136 - val_accuracy: 0.5250
Epoch 20/100
25/25 [==============================] - 0s 13ms/step - loss: 1.3610 - accuracy: 0.5
```

```
550 - val_loss: 1.5028 - val_accuracy: 0.4850
Epoch 21/100
25/25 [==============================] - 0s 12ms/step - loss: 1.4306 - accuracy: 0.5
386 - val_loss: 1.5604 - val_accuracy: 0.4950
Epoch 22/100
25/25 [==============================] - 0s 12ms/step - loss: 1.2617 - accuracy: 0.5
918 - val_loss: 1.4098 - val_accuracy: 0.5400
Epoch 23/100
25/25 [==============================] - 0s 12ms/step - loss: 1.2355 - accuracy: 0.5
933 - val_loss: 1.4327 - val_accuracy: 0.5250
Epoch 24/100
25/25 [==============================] - 0s 12ms/step - loss: 1.2325 - accuracy: 0.6
020 - val_loss: 1.3720 - val_accuracy: 0.5900
Epoch 25/100
25/25 [==============================] - 0s 13ms/step - loss: 1.1795 - accuracy: 0.6
256 - val_loss: 1.3202 - val_accuracy: 0.5950
Epoch 26/100
25/25 [==============================] - 0s 13ms/step - loss: 1.1790 - accuracy: 0.6
112 - val_loss: 1.2624 - val_accuracy: 0.6200
Epoch 27/100
25/25 [==============================] - 0s 13ms/step - loss: 1.1255 - accuracy: 0.6
421 - val_loss: 1.1893 - val_accuracy: 0.6350
Epoch 28/100
25/25 [==============================] - 0s 12ms/step - loss: 1.1297 - accuracy: 0.6
381 - val_loss: 1.2111 - val_accuracy: 0.6400
Epoch 29/100
25/25 [==============================] - 0s 13ms/step - loss: 1.0262 - accuracy: 0.6
829 - val_loss: 1.3689 - val_accuracy: 0.5200
Epoch 30/100
25/25 [==============================] - 0s 13ms/step - loss: 0.9644 - accuracy: 0.6
912 - val_loss: 1.1946 - val_accuracy: 0.6400
Epoch 31/100
25/25 [==============================] - 0s 13ms/step - loss: 0.8941 - accuracy: 0.7
296 - val_loss: 1.2556 - val_accuracy: 0.6500
Epoch 32/100
25/25 [==============================] - 0s 13ms/step - loss: 0.9476 - accuracy: 0.6
893 - val_loss: 1.2855 - val_accuracy: 0.5900
Epoch 33/100
25/25 [==============================] - 0s 12ms/step - loss: 0.8226 - accuracy: 0.7
434 - val_loss: 1.1242 - val_accuracy: 0.6650
Epoch 34/100
25/25 [==============================] - 0s 13ms/step - loss: 0.8941 - accuracy: 0.7
185 - val_loss: 1.1683 - val_accuracy: 0.6700
Epoch 35/100
25/25 [==============================] - 0s 13ms/step - loss: 0.7546 - accuracy: 0.7
673 - val_loss: 1.1871 - val_accuracy: 0.6250
Epoch 36/100
25/25 [==============================] - 0s 13ms/step - loss: 0.7312 - accuracy: 0.7
719 - val_loss: 1.2313 - val_accuracy: 0.6450
Epoch 37/100
25/25 [==============================] - 0s 13ms/step - loss: 0.7554 - accuracy: 0.7
787 - val_loss: 1.1204 - val_accuracy: 0.6750
Epoch 38/100
25/25 [==============================] - 0s 13ms/step - loss: 0.7248 - accuracy: 0.7
749 - val_loss: 1.1022 - val_accuracy: 0.6850
Epoch 39/100
25/25 [==============================] - 0s 13ms/step - loss: 0.6571 - accuracy: 0.7
889 - val_loss: 1.1894 - val_accuracy: 0.6650
Epoch 40/100
25/25 [==============================] - 0s 12ms/step - loss: 0.6257 - accuracy: 0.8
054 - val_loss: 1.1935 - val_accuracy: 0.6550
Epoch 41/100
25/25 [==============================] - 0s 13ms/step - loss: 0.6476 - accuracy: 0.8
090 - val_loss: 1.1927 - val_accuracy: 0.6800
Epoch 42/100
25/25 [==============================] - 0s 13ms/step - loss: 0.6240 - accuracy: 0.8
108 - val_loss: 1.2340 - val_accuracy: 0.6250
Epoch 43/100
25/25 [==============================] - 0s 13ms/step - loss: 0.6196 - accuracy: 0.7
```

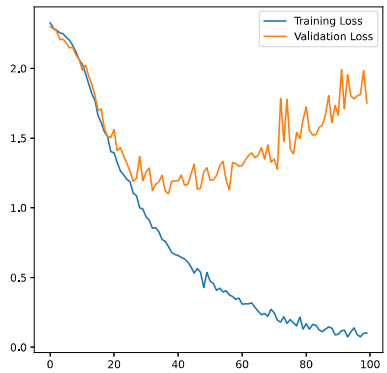
```
961 - val_loss: 1.1613 - val_accuracy: 0.7100
Epoch 44/100
25/25 [==============================] - 0s 13ms/step - loss: 0.6307 - accuracy: 0.8
157 - val_loss: 1.1673 - val_accuracy: 0.7050
Epoch 45/100
25/25 [==============================] - 0s 13ms/step - loss: 0.5386 - accuracy: 0.8
390 - val_loss: 1.2400 - val_accuracy: 0.6500
Epoch 46/100
25/25 [==============================] - 0s 12ms/step - loss: 0.5174 - accuracy: 0.8
565 - val_loss: 1.3123 - val_accuracy: 0.6550
Epoch 47/100
25/25 [==============================] - 0s 12ms/step - loss: 0.5909 - accuracy: 0.8
348 - val_loss: 1.1346 - val_accuracy: 0.6800
Epoch 48/100
25/25 [==============================] - 0s 13ms/step - loss: 0.4638 - accuracy: 0.8
633 - val_loss: 1.1399 - val_accuracy: 0.7100
Epoch 49/100
25/25 [==============================] - 0s 12ms/step - loss: 0.4080 - accuracy: 0.8
856 - val_loss: 1.2560 - val_accuracy: 0.6600
Epoch 50/100
25/25 [==============================] - 0s 12ms/step - loss: 0.4919 - accuracy: 0.8
366 - val_loss: 1.2873 - val_accuracy: 0.6750
Epoch 51/100
25/25 [==============================] - 0s 12ms/step - loss: 0.4688 - accuracy: 0.8
480 - val_loss: 1.1972 - val_accuracy: 0.7050
Epoch 52/100
25/25 [==============================] - 0s 12ms/step - loss: 0.4252 - accuracy: 0.8
738 - val_loss: 1.1997 - val_accuracy: 0.7000
Epoch 53/100
25/25 [==============================] - 0s 12ms/step - loss: 0.3951 - accuracy: 0.8
919 - val_loss: 1.2330 - val_accuracy: 0.6900
Epoch 54/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3920 - accuracy: 0.8
849 - val_loss: 1.3015 - val_accuracy: 0.6750
Epoch 55/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3720 - accuracy: 0.8
802 - val_loss: 1.3337 - val_accuracy: 0.6800
Epoch 56/100
25/25 [==============================] - 0s 14ms/step - loss: 0.3506 - accuracy: 0.8
950 - val_loss: 1.1962 - val_accuracy: 0.7200
Epoch 57/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3355 - accuracy: 0.9
087 - val_loss: 1.1292 - val_accuracy: 0.7150
Epoch 58/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3536 - accuracy: 0.9
140 - val_loss: 1.3237 - val_accuracy: 0.7000
Epoch 59/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3047 - accuracy: 0.9
103 - val_loss: 1.3170 - val_accuracy: 0.7050
Epoch 60/100
25/25 [==============================] - 0s 13ms/step - loss: 0.3079 - accuracy: 0.9
126 - val_loss: 1.2977 - val_accuracy: 0.6900
Epoch 61/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2741 - accuracy: 0.9
209 - val_loss: 1.3036 - val_accuracy: 0.7050
Epoch 62/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2834 - accuracy: 0.9
246 - val_loss: 1.3426 - val_accuracy: 0.7200
Epoch 63/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2669 - accuracy: 0.9
231 - val_loss: 1.3743 - val_accuracy: 0.7000
Epoch 64/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2634 - accuracy: 0.9
115 - val_loss: 1.3927 - val_accuracy: 0.6900
Epoch 65/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2946 - accuracy: 0.9
225 - val_loss: 1.3578 - val_accuracy: 0.6700
Epoch 66/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2473 - accuracy: 0.9
```

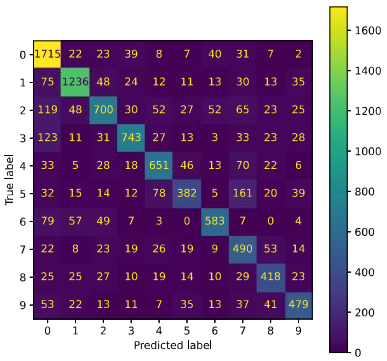
```
326 - val_loss: 1.3773 - val_accuracy: 0.7050
Epoch 67/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2471 - accuracy: 0.9
232 - val_loss: 1.4303 - val_accuracy: 0.7100
Epoch 68/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2375 - accuracy: 0.9
362 - val_loss: 1.3506 - val_accuracy: 0.7000
Epoch 69/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1810 - accuracy: 0.9
456 - val_loss: 1.4504 - val_accuracy: 0.7000
Epoch 70/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2962 - accuracy: 0.8
992 - val_loss: 1.3268 - val_accuracy: 0.7050
Epoch 71/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2347 - accuracy: 0.9
168 - val_loss: 1.3500 - val_accuracy: 0.7050
Epoch 72/100
25/25 [==============================] - 0s 13ms/step - loss: 0.2230 - accuracy: 0.9
516 - val_loss: 1.2757 - val_accuracy: 0.7550
Epoch 73/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1789 - accuracy: 0.9
443 - val_loss: 1.7832 - val_accuracy: 0.6700
Epoch 74/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2107 - accuracy: 0.9
375 - val_loss: 1.4771 - val_accuracy: 0.7200
Epoch 75/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1355 - accuracy: 0.9
624 - val_loss: 1.7759 - val_accuracy: 0.6850
Epoch 76/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2285 - accuracy: 0.9
338 - val_loss: 1.4263 - val_accuracy: 0.7200
Epoch 77/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1778 - accuracy: 0.9
497 - val_loss: 1.3853 - val_accuracy: 0.7250
Epoch 78/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1616 - accuracy: 0.9
470 - val_loss: 1.5406 - val_accuracy: 0.7150
Epoch 79/100
25/25 [==============================] - 0s 12ms/step - loss: 0.2607 - accuracy: 0.9
206 - val_loss: 1.4975 - val_accuracy: 0.7250
Epoch 80/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1402 - accuracy: 0.9
608 - val_loss: 1.6311 - val_accuracy: 0.7050
Epoch 81/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1586 - accuracy: 0.9
590 - val_loss: 1.7252 - val_accuracy: 0.6900
Epoch 82/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1110 - accuracy: 0.9
736 - val_loss: 1.5538 - val_accuracy: 0.7100
Epoch 83/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1684 - accuracy: 0.9
472 - val_loss: 1.5221 - val_accuracy: 0.7350
Epoch 84/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1252 - accuracy: 0.9
568 - val_loss: 1.5246 - val_accuracy: 0.7150
Epoch 85/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1303 - accuracy: 0.9
678 - val_loss: 1.5745 - val_accuracy: 0.7250
Epoch 86/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1054 - accuracy: 0.9
707 - val_loss: 1.5936 - val_accuracy: 0.7100
Epoch 87/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1447 - accuracy: 0.9
502 - val_loss: 1.6659 - val_accuracy: 0.7200
Epoch 88/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1511 - accuracy: 0.9
558 - val_loss: 1.8055 - val_accuracy: 0.6700
Epoch 89/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1422 - accuracy: 0.9
```

```
537 - val_loss: 1.6116 - val_accuracy: 0.7200
Epoch 90/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0746 - accuracy: 0.9
812 - val_loss: 1.7322 - val_accuracy: 0.6850
Epoch 91/100
25/25 [==============================] - 0s 12ms/step - loss: 0.0920 - accuracy: 0.9
796 - val_loss: 1.6639 - val_accuracy: 0.7350
Epoch 92/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1302 - accuracy: 0.9
571 - val_loss: 1.9906 - val_accuracy: 0.7150
Epoch 93/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1002 - accuracy: 0.9
718 - val_loss: 1.7114 - val_accuracy: 0.7200
Epoch 94/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0629 - accuracy: 0.9
796 - val_loss: 1.9540 - val_accuracy: 0.6950
Epoch 95/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1128 - accuracy: 0.9
688 - val_loss: 1.8018 - val_accuracy: 0.7150
Epoch 96/100
25/25 [==============================] - 0s 12ms/step - loss: 0.1132 - accuracy: 0.9
748 - val_loss: 1.7821 - val_accuracy: 0.7200
Epoch 97/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0605 - accuracy: 0.9
845 - val_loss: 1.8031 - val_accuracy: 0.7050
Epoch 98/100
25/25 [==============================] - 0s 13ms/step - loss: 0.1006 - accuracy: 0.9
696 - val_loss: 1.8132 - val_accuracy: 0.7700
Epoch 99/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0673 - accuracy: 0.9
723 - val_loss: 1.9831 - val_accuracy: 0.7400
Epoch 100/100
25/25 [==============================] - 0s 13ms/step - loss: 0.0806 - accuracy: 0.9
814 - val_loss: 1.7520 - val_accuracy: 0.7550
```

In [14]:

```
def plot_training(history, model, x_test, y_test):
    fig = plt.figure(figsize=[20, 6])
    ax = fig.add_subplot(1, 3, 1)
    ax.plot(history.history['loss'], label="Training Loss")
    ax.plot(history.history['val_loss'], label="Validation Loss")
    ax.legend()

    ax = fig.add_subplot(1, 3, 2)
    ax.plot(history.history['accuracy'], label="Training Accuracy")
    ax.plot(history.history['val_accuracy'], label="Validation Accuracy")
    ax.legend()

    pred = model.predict(x_test)
    indexes = tf.argmax(pred, axis=1)
    i = tf.cast([], tf.int32)
    indexes = tf.gather_nd(indexes, i)

    cm = confusion_matrix(y_test, indexes)
    ax = fig.add_subplot(1, 3, 3)
    c = ConfusionMatrixDisplay(cm, display_labels=range(10))
    c.plot(ax = ax)

plot_training(history, model_cnn_aug, x_test, y_test)
```





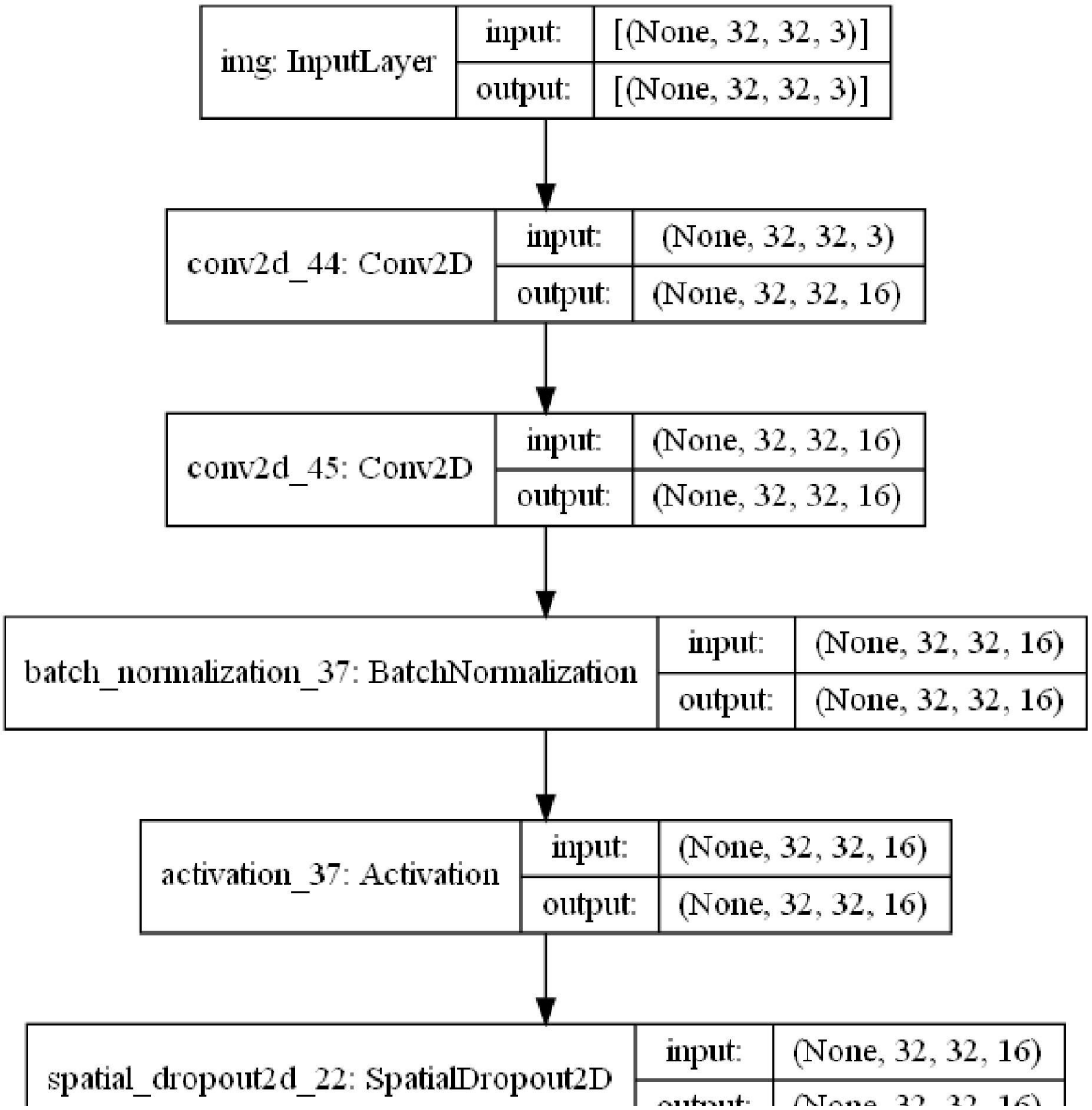
Fine Tuning on vgg

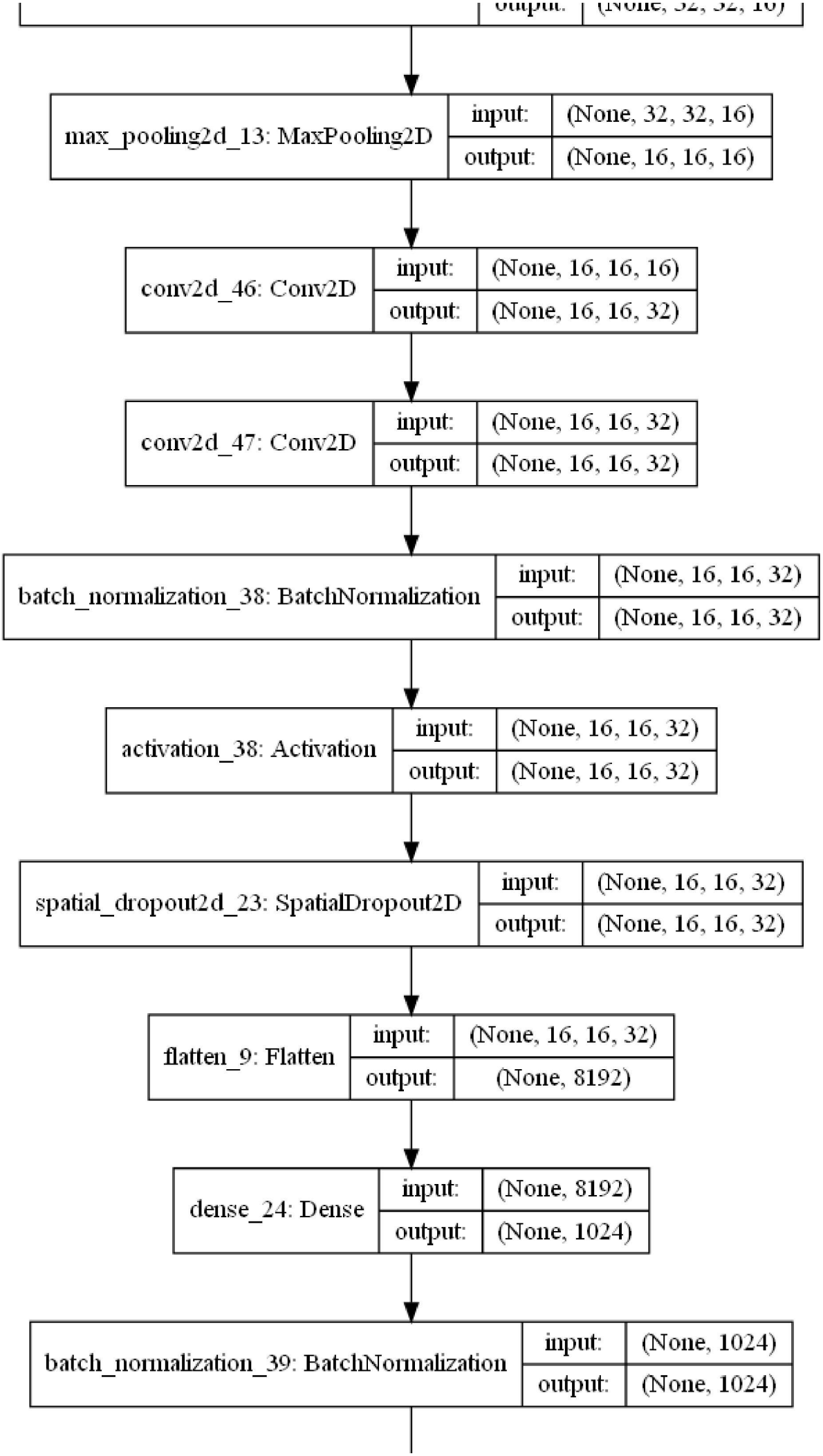
Week 5 example 3

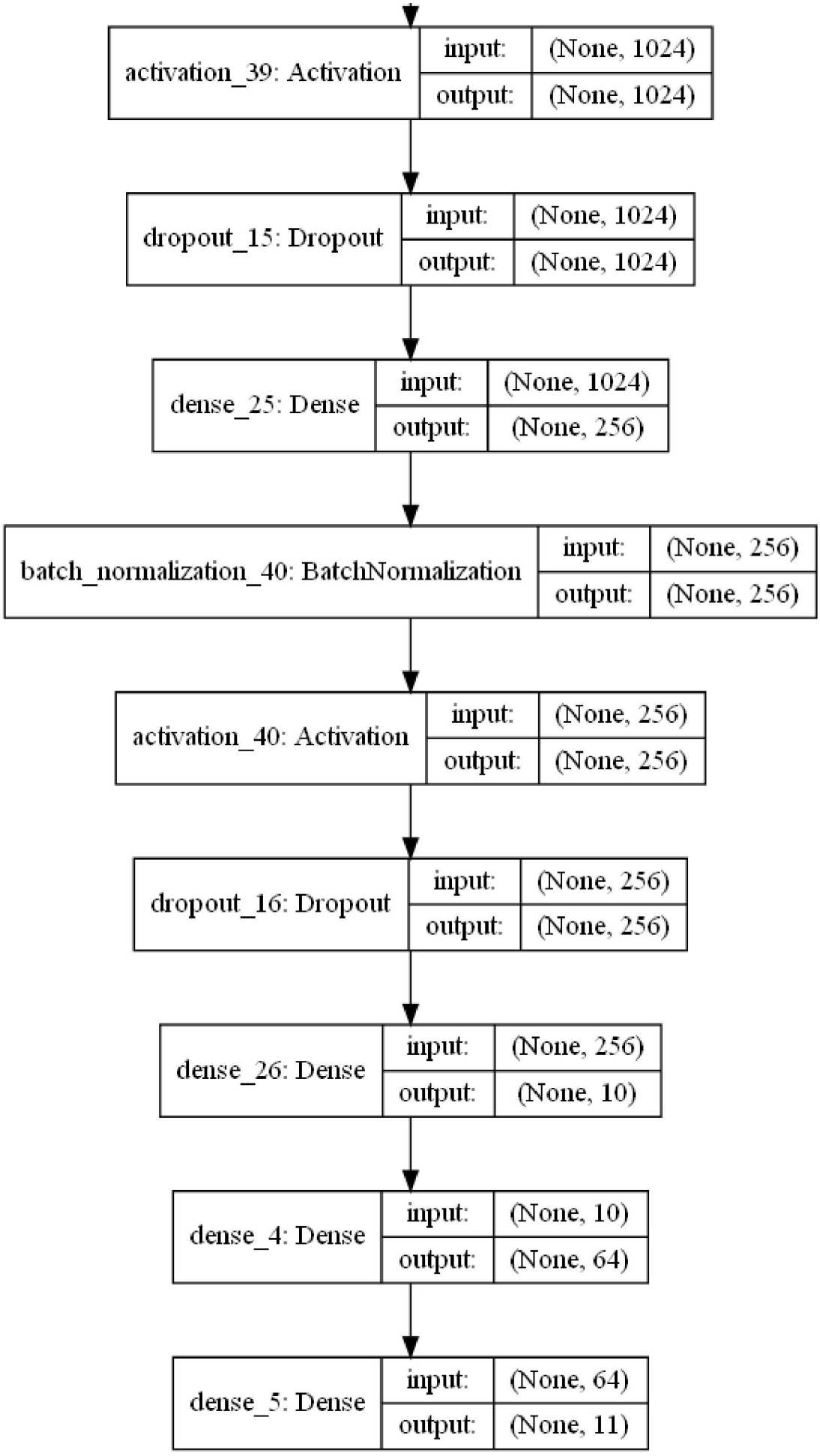
In [12]:

```
model = keras.models.load_model('Assessment 1B/vgg_2stage_CIFAR_bigger.h5')

x = layers.Dense(64, activation='relu')(model.layers[-1].output)
# output layer, single value, as we only have 1 output - the
outputs = layers.Dense(11)(x)

new_model = keras.Model(inputs=model.input, outputs=outputs)
keras.utils.plot_model(new_model, show_shapes=True)
```

Out[12]:




max_pooling2d_13: MaxPooling2D
input: (None, 32, 32, 16)
output: (None, 16, 16, 16)
conv2d_46: Conv2D
input: (None, 16, 16, 16)
output: (None, 16, 16, 32)
conv2d_47: Conv2D
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
batch_normalization_38: BatchNormalization
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
activation_38: Activation
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
spatial_dropout2d_23: SpatialDropout2D
input: (None, 16, 16, 32)
output: (None, 16, 16, 32)
flatten_9: Flatten
input: (None, 16, 16, 32)
output: (None, 8192)
dense_24: Dense
input: (None, 8192)
output: (None, 1024)
batch_normalization_39: BatchNormalization
input: (None, 1024)
output: (None, 1024)



In [11]:
```
# train the model
new_model.compile(loss=keras.losses.SparseCategoricalCrossentropy(from_logits=False)
                  optimizer=keras.optimizers.RMSprop(),
```

```
                  metrics=['accuracy'])

history = new_model.fit(x_train, y_train,
                        batch_size=32,
                        epochs=100,
                        validation_split=0.2, verbose=True)
```

```
Epoch 1/100
25/25 [==============================] - 4s 92ms/step - loss: 5.9352 - accuracy: 0.1
232 - val_loss: 5.5801 - val_accuracy: 0.1750
Epoch 2/100
25/25 [==============================] - 2s 86ms/step - loss: 4.9253 - accuracy: 0.1
695 - val_loss: 5.8066 - val_accuracy: 0.2100
Epoch 3/100
25/25 [==============================] - 2s 86ms/step - loss: 4.1194 - accuracy: 0.2
139 - val_loss: 5.8597 - val_accuracy: 0.2100
Epoch 4/100
25/25 [==============================] - 2s 85ms/step - loss: 4.0632 - accuracy: 0.1
967 - val_loss: 6.5230 - val_accuracy: 0.1600
Epoch 5/100
25/25 [==============================] - 2s 85ms/step - loss: 3.8324 - accuracy: 0.2
203 - val_loss: 5.5341 - val_accuracy: 0.1900
Epoch 6/100
25/25 [==============================] - 2s 85ms/step - loss: 3.5539 - accuracy: 0.2
458 - val_loss: 5.6826 - val_accuracy: 0.2150
Epoch 7/100
25/25 [==============================] - 2s 86ms/step - loss: 3.4527 - accuracy: 0.3
122 - val_loss: 6.0353 - val_accuracy: 0.1650
Epoch 8/100
25/25 [==============================] - 2s 86ms/step - loss: 3.4701 - accuracy: 0.2
952 - val_loss: 6.5614 - val_accuracy: 0.1700
Epoch 9/100
25/25 [==============================] - 2s 87ms/step - loss: 3.0635 - accuracy: 0.3
305 - val_loss: 5.6132 - val_accuracy: 0.2150
Epoch 10/100
25/25 [==============================] - 2s 87ms/step - loss: 2.8268 - accuracy: 0.3
660 - val_loss: 5.0943 - val_accuracy: 0.2900
Epoch 11/100
25/25 [==============================] - 2s 85ms/step - loss: 3.0488 - accuracy: 0.3
758 - val_loss: 6.3208 - val_accuracy: 0.2850
Epoch 12/100
25/25 [==============================] - 2s 85ms/step - loss: 3.6904 - accuracy: 0.3
332 - val_loss: 5.7211 - val_accuracy: 0.2550
Epoch 13/100
25/25 [==============================] - 2s 86ms/step - loss: 3.5569 - accuracy: 0.3
682 - val_loss: 10.1054 - val_accuracy: 0.2050
Epoch 14/100
25/25 [==============================] - 2s 88ms/step - loss: 3.8160 - accuracy: 0.3
289 - val_loss: 10.2844 - val_accuracy: 0.1550
Epoch 15/100
25/25 [==============================] - 2s 88ms/step - loss: 3.1901 - accuracy: 0.3
148 - val_loss: 5.7288 - val_accuracy: 0.1650
Epoch 16/100
25/25 [==============================] - 2s 86ms/step - loss: 2.5232 - accuracy: 0.2
516 - val_loss: 3.2861 - val_accuracy: 0.1150
Epoch 17/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3934 - accuracy: 0.2
013 - val_loss: 3.1159 - val_accuracy: 0.1650
Epoch 18/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3929 - accuracy: 0.2
358 - val_loss: 3.0206 - val_accuracy: 0.2200
Epoch 19/100
25/25 [==============================] - 2s 88ms/step - loss: 2.4584 - accuracy: 0.2
222 - val_loss: 2.9889 - val_accuracy: 0.2400
Epoch 20/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
897 - val_loss: 2.7053 - val_accuracy: 0.2300
Epoch 21/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3990 - accuracy: 0.2
```

```
029 - val_loss: 2.3979 - val_accuracy: 0.0850
Epoch 22/100
25/25 [==============================] - 2s 88ms/step - loss: 2.3979 - accuracy: 0.1
025 - val_loss: 2.3979 - val_accuracy: 0.0750
Epoch 23/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
431 - val_loss: 2.4440 - val_accuracy: 0.1050
Epoch 24/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
407 - val_loss: 2.3979 - val_accuracy: 0.1200
Epoch 25/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
457 - val_loss: 2.3979 - val_accuracy: 0.1400
Epoch 26/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
201 - val_loss: 2.3859 - val_accuracy: 0.1150
Epoch 27/100
25/25 [==============================] - 2s 88ms/step - loss: 2.3979 - accuracy: 0.1
162 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 28/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
769 - val_loss: 2.3979 - val_accuracy: 0.1050
Epoch 29/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
389 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 30/100
25/25 [==============================] - 2s 89ms/step - loss: 2.3979 - accuracy: 0.1
537 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 31/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
461 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 32/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
149 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 33/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
303 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 34/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
657 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 35/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
344 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 36/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
390 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 37/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
373 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 38/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
337 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 39/100
25/25 [==============================] - 2s 88ms/step - loss: 2.4045 - accuracy: 0.1
659 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 40/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
396 - val_loss: 2.3979 - val_accuracy: 0.1050
Epoch 41/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
544 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 42/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
526 - val_loss: 2.3979 - val_accuracy: 0.1050
Epoch 43/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
552 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 44/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3891 - accuracy: 0.1
```

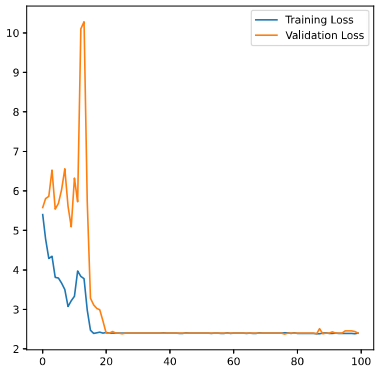
```
454 - val_loss: 2.3979 - val_accuracy: 0.1200
Epoch 45/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3928 - accuracy: 0.1
642 - val_loss: 2.3979 - val_accuracy: 0.1200
Epoch 46/100
25/25 [==============================] - 2s 85ms/step - loss: 2.4049 - accuracy: 0.1
474 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 47/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
475 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 48/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
413 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 49/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
635 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 50/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
505 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 51/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
327 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 52/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
484 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 53/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
476 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 54/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3921 - accuracy: 0.1
736 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 55/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
510 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 56/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
559 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 57/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3911 - accuracy: 0.1
611 - val_loss: 2.3979 - val_accuracy: 0.0850
Epoch 58/100
25/25 [==============================] - 2s 88ms/step - loss: 2.3909 - accuracy: 0.1
433 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 59/100
25/25 [==============================] - 2s 86ms/step - loss: 2.4093 - accuracy: 0.1
187 - val_loss: 2.3979 - val_accuracy: 0.1450
Epoch 60/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3970 - accuracy: 0.1
423 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 61/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
329 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 62/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
489 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 63/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
311 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 64/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
227 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 65/100
25/25 [==============================] - 2s 88ms/step - loss: 2.3946 - accuracy: 0.1
512 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 66/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
412 - val_loss: 2.3979 - val_accuracy: 0.0950
Epoch 67/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3951 - accuracy: 0.1
```

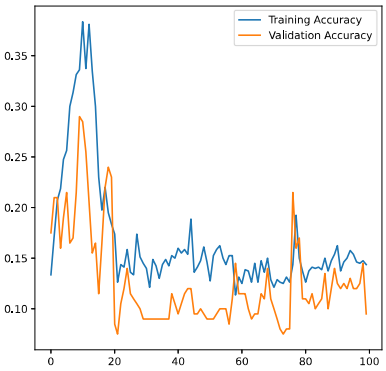
```
586 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 68/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3934 - accuracy: 0.1
297 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 69/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3956 - accuracy: 0.1
464 - val_loss: 2.3979 - val_accuracy: 0.1400
Epoch 70/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
398 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 71/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
381 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 72/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3979 - accuracy: 0.1
329 - val_loss: 2.3979 - val_accuracy: 0.0900
Epoch 73/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
255 - val_loss: 2.3979 - val_accuracy: 0.0800
Epoch 74/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
260 - val_loss: 2.3979 - val_accuracy: 0.0750
Epoch 75/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3979 - accuracy: 0.1
215 - val_loss: 2.3979 - val_accuracy: 0.0800
Epoch 76/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3979 - accuracy: 0.1
156 - val_loss: 2.3979 - val_accuracy: 0.0800
Epoch 77/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3973 - accuracy: 0.1
322 - val_loss: 2.3739 - val_accuracy: 0.2150
Epoch 78/100
25/25 [==============================] - 2s 83ms/step - loss: 2.3951 - accuracy: 0.1
928 - val_loss: 2.3979 - val_accuracy: 0.1600
Epoch 79/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3938 - accuracy: 0.1
532 - val_loss: 2.3979 - val_accuracy: 0.1700
Epoch 80/100
25/25 [==============================] - 2s 87ms/step - loss: 2.4052 - accuracy: 0.1
523 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 81/100
25/25 [==============================] - 2s 88ms/step - loss: 2.3959 - accuracy: 0.1
243 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 82/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3911 - accuracy: 0.1
630 - val_loss: 2.3979 - val_accuracy: 0.1050
Epoch 83/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3975 - accuracy: 0.1
452 - val_loss: 2.3979 - val_accuracy: 0.1150
Epoch 84/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3972 - accuracy: 0.1
569 - val_loss: 2.3979 - val_accuracy: 0.1000
Epoch 85/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3944 - accuracy: 0.1
473 - val_loss: 2.3979 - val_accuracy: 0.1050
Epoch 86/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3862 - accuracy: 0.1
401 - val_loss: 2.3979 - val_accuracy: 0.1100
Epoch 87/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3895 - accuracy: 0.1
349 - val_loss: 2.3859 - val_accuracy: 0.1350
Epoch 88/100
25/25 [==============================] - 2s 87ms/step - loss: 2.3886 - accuracy: 0.1
152 - val_loss: 2.5131 - val_accuracy: 0.1000
Epoch 89/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3985 - accuracy: 0.1
415 - val_loss: 2.3859 - val_accuracy: 0.1200
Epoch 90/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3999 - accuracy: 0.1
```

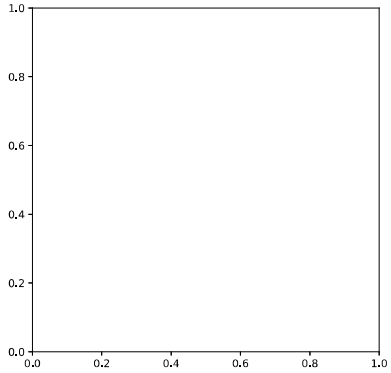
```
412 - val_loss: 2.3979 - val_accuracy: 0.1400
Epoch 91/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3922 - accuracy: 0.1
655 - val_loss: 2.3979 - val_accuracy: 0.1250
Epoch 92/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3962 - accuracy: 0.1
260 - val_loss: 2.4353 - val_accuracy: 0.1200
Epoch 93/100
25/25 [==============================] - 2s 84ms/step - loss: 2.3979 - accuracy: 0.1
517 - val_loss: 2.3979 - val_accuracy: 0.1250
Epoch 94/100
25/25 [==============================] - 2s 83ms/step - loss: 2.3962 - accuracy: 0.1
521 - val_loss: 2.3979 - val_accuracy: 0.1200
Epoch 95/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3938 - accuracy: 0.1
571 - val_loss: 2.3979 - val_accuracy: 0.1300
Epoch 96/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3936 - accuracy: 0.1
475 - val_loss: 2.4559 - val_accuracy: 0.1200
Epoch 97/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3811 - accuracy: 0.1
528 - val_loss: 2.4554 - val_accuracy: 0.1200
Epoch 98/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3970 - accuracy: 0.1
445 - val_loss: 2.4551 - val_accuracy: 0.1250
Epoch 99/100
25/25 [==============================] - 2s 85ms/step - loss: 2.3815 - accuracy: 0.1
426 - val_loss: 2.4437 - val_accuracy: 0.1450
Epoch 100/100
25/25 [==============================] - 2s 86ms/step - loss: 2.3955 - accuracy: 0.1
492 - val_loss: 2.3979 - val_accuracy: 0.0950
```

In [13]:

```
def plot_training(history, model, x_test, y_test):
    fig = plt.figure(figsize=[20, 6])
    ax = fig.add_subplot(1, 3, 1)
    ax.plot(history.history['loss'], label="Training Loss")
    ax.plot(history.history['val_loss'], label="Validation Loss")
    ax.legend()

    ax = fig.add_subplot(1, 3, 2)
    ax.plot(history.history['accuracy'], label="Training Accuracy")
    ax.plot(history.history['val_accuracy'], label="Validation Accuracy")
    ax.legend()

    pred = model.predict(x_test)
    indexes = tf.argmax(pred, axis=1)
    i = tf.cast([], tf.int32)
    indexes = tf.gather_nd(indexes, i)

    cm = confusion_matrix(y_test, indexes)
    ax = fig.add_subplot(1, 3, 3)
    c = ConfusionMatrixDisplay(cm, display_labels=range(10))

    #c.plot(ax = ax)
plot_training(history, new_model, x_test, y_test)
```


Training Loss
Validation Loss


Training Accuracy
Validation Accuracy



In [ ]:

Assignment 1B - Question 2

Person Re-Identification

In [1]:

```
import pandas
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import scipy.stats as stats
from scipy.io import loadmat

import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from tensorboard import notebook
from tensorflow.keras.preprocessing.image import Iterator

from sklearn import decomposition
from sklearn import discriminant_analysis
from sklearn import datasets
from sklearn.manifold import TSNE
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import plot_confusion_matrix
import matplotlib.pyplot as plt

import cv2
import os
import glob
import random
```

```
INFO:tensorflow:Enabling eager execution
INFO:tensorflow:Enabling v2 tensorshape
INFO:tensorflow:Enabling resource variables
INFO:tensorflow:Enabling tensor equality
INFO:tensorflow:Enabling control flow v2
```

Load data

Uses opencv's cv2.imread function on all files in a directory to load into a numpy array

In [2]:

```
train = []
gnd = []
files = glob.glob('Data/Q2/Q2/Q2/Training/*.jpg')
for myfile in files:
    image = cv2.imread(myfile, 0)
    gnd.append(myfile[23:27])
    train.append(image)

train = np.array(train)
gnd = np.array(gnd)

print('Training shape: ', train.shape)
print('Training gnd: ', gnd[1:10])
```

```
Training shape:  (5933, 128, 64)
Training gnd:  ['0001' '0001' '0001' '0001' '0001' '0001' '0001' '0001' '0
001']
```

In [3]:

```
test_gal = []
test_gal_gnd = []
files = glob.glob('Data/Q2/Q2/Q2/Testing/Gallery/*.jpg')
for myfile in files:
    image = cv2.imread(myfile, 0)
    test_gal_gnd.append(myfile[30:34])
    test_gal.append(image)

test_gal = np.array(test_gal)

print('Training shape: ', test_gal.shape)
print('Training gnd: ', test_gal_gnd[1:10])
```

```
Training shape:  (301, 128, 64)
Training gnd:  ['1201', '1202', '1203', '1204', '1205', '1206', '1207', '1
208', '1209']
```

In [4]:

```
test_pro = []
test_pro_gnd = []
files = glob.glob('Data/Q2/Q2/Q2/Testing/Probe/*.jpg')
for myfile in files:
    image = cv2.imread(myfile, 0)
    test_pro_gnd.append(myfile[28:32])
    test_pro.append(image)

test_pro = np.array(test_pro)

print('Training shape: ', test_pro.shape)
print('Training gnd: ', test_pro_gnd[1:10])
```

```
Training shape:  (301, 128, 64)
Training gnd:  ['1201', '1202', '1203', '1204', '1205', '1206', '1207', '1
208', '1209']
```

In [5]:

```
train_fea = np.reshape(train.transpose(), (64, 128, 1, len(train)))
fig = plt.figure(figsize=[20, 8])
for i in range(40):
    ax = fig.add_subplot(4, 10, i + 1)
    ax.imshow(train_fea[:,:,0,i].transpose(), cmap=plt.get_cmap('gray'))
    ax.set_axis_off()
```

In [6]:

```
test_fea_img = np.reshape(test_gal.transpose(), (64, 128, 1, len(test_gal)))
fig = plt.figure(figsize=[20, 8])
for i in range(40):
    ax = fig.add_subplot(4, 10, i + 1)
    ax.imshow(test_fea_img[:,:,0,i].transpose(), cmap=plt.get_cmap('gray'))
    ax.set_axis_off()
```

In [7]:

```
print(test_gal.shape)
```

```
(301, 128, 64)
```

In [8]:

```
test_pro_img = np.reshape(test_pro.transpose(), (64, 128, 1, len(test_pro)))
fig = plt.figure(figsize=[20, 8])
for i in range(40):
    ax = fig.add_subplot(4, 10, i + 1)
    ax.imshow(test_pro_img[:,:,0,i].transpose(), cmap=plt.get_cmap('gray'))
    ax.set_axis_off()
```

PCA

Code taken from Week 6 Example 5 Eigenfaces

In [9]:

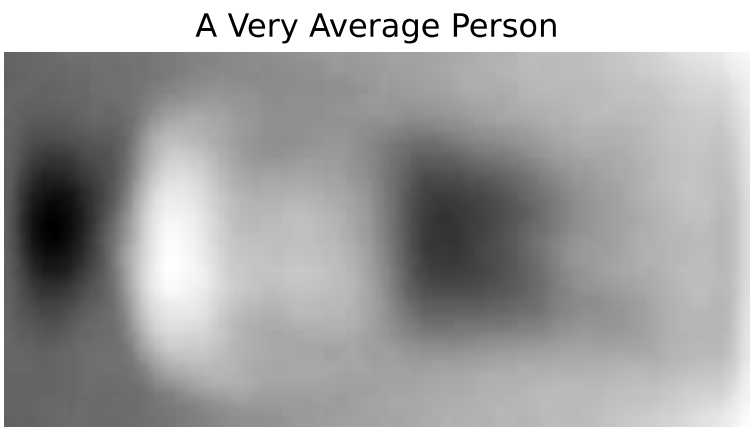
```
train_fea = train
train_gnd = gnd
test_fea = test_gal
test_gnd = test_gal_gnd
print(np.shape(train_fea))
print(np.shape(test_fea))
```

```
(5933, 128, 64)
(301, 128, 64)
```

In [10]:

```
meanperson = np.reshape(np.mean(train_fea, axis=0), (-1, 1));
print(meanperson.shape)
fig = plt.figure(figsize=[5, 5])
ax = fig.add_subplot(1, 1, 1)
meanperson_im = np.reshape(meanperson, (128, 64))
ax.imshow(meanperson_im.transpose(), cmap=plt.get_cmap('gray'))
ax.set_axis_off()
ax.set_title('A Very Average Person');
```

```
(8192, 1)
```

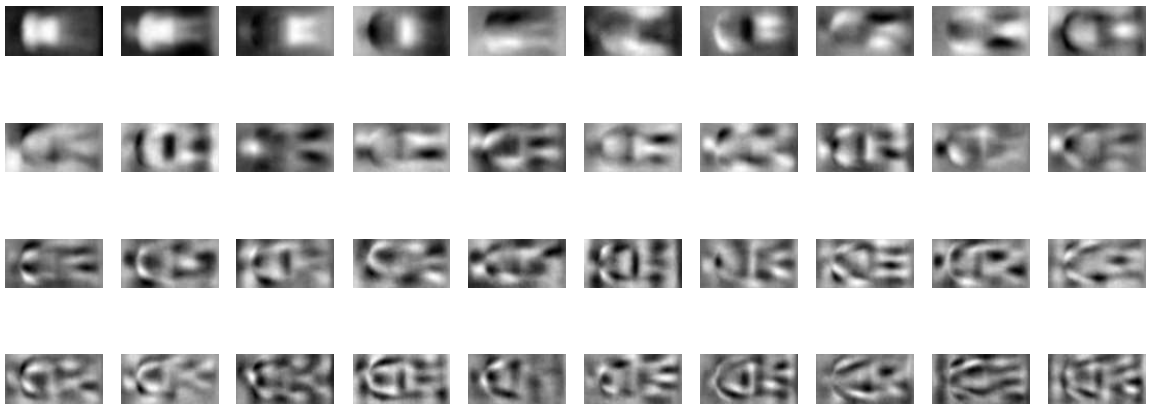


In [11]:

```
nsamples, nx, ny = train_fea.shape
d2_train_dataset = train_fea.reshape((nsamples,nx*ny))

nsamples, nx, ny = test_fea.shape
d2_test_dataset = test_fea.reshape((nsamples,nx*ny))

nsamples, nx, ny = test_pro.shape
d2_probe_dataset = test_pro.reshape((nsamples,nx*ny))

pca = decomposition.PCA()
pca.fit(d2_train_dataset)
transformed = pca.transform(d2_train_dataset)
transformed_test = pca.transform(d2_test_dataset)
transformed_probe = pca.transform(d2_probe_dataset)
```

In [12]:

```
fig = plt.figure(figsize=[20, 8])
for i in range(40):
    ax = fig.add_subplot(4, 10, i + 1)
    pc = np.reshape(pca.components_[i,:], (128, 64))
    ax.imshow(pc.transpose(), cmap=plt.get_cmap('gray'))
    ax.set_axis_off()
```

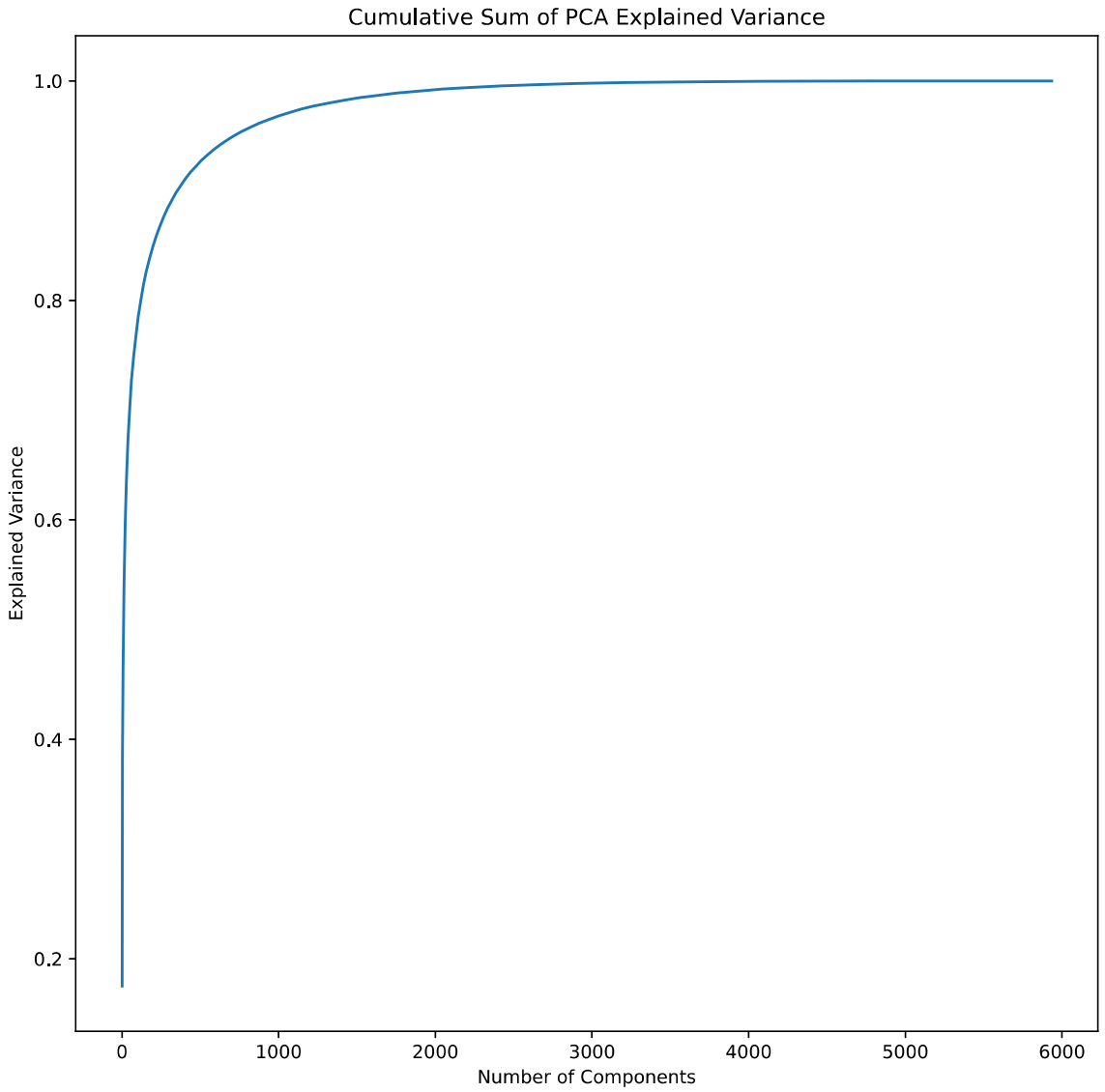


In [13]:

```
cumulative_sum = np.cumsum(pca.explained_variance_ratio_, axis=0)
fig = plt.figure(figsize=[10, 10])
ax = fig.add_subplot(1, 1, 1)
ax.plot(cumulative_sum)
ax.set_title('Cumulative Sum of PCA Explained Variance')
ax.set_ylabel('Explained Variance')
ax.set_xlabel('Number of Components')
```

Out[13]:

```
Text(0.5, 0, 'Number of Components')
```



In [14]:

```
top90 = np.where(cumulative_sum > 0.90)[0][0]
print('90% in ' + '%d' % (top90+1) + ' components')
top95 = np.where(cumulative_sum > 0.95)[0][0]
print('95% in ' + '%d' % (top95+1) + ' components')
top99 = np.where(cumulative_sum > 0.99)[0][0]
print('99% in ' + '%d' % (top99+1) + ' components')
```

```
90% in 355 components
95% in 715 components
99% in 1832 components
```

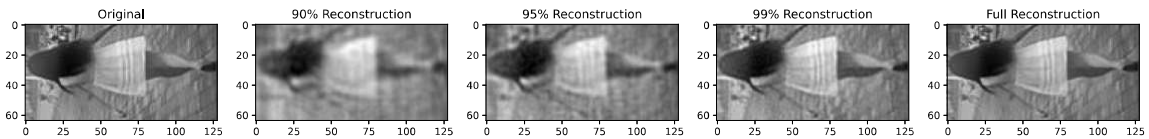
In [15]:

```
print(train_fea.shape)
```

```
(5933, 128, 64)
```

In [16]:

```
fig = plt.figure(figsize=[20, 5])
ax = fig.add_subplot(1, 5, 1)
ax.imshow(np.reshape(d2_train_dataset[0,:] - pca.mean_, (128, 64)).transpose(), cmap=pl
t.get_cmap('gray'))
ax.set_title('Original');

ax = fig.add_subplot(1, 5, 2)
ax.imshow(np.reshape(pca.components_[0:top90,:].transpose().dot(np.reshape(transformed[
0,0:top90], (-1, 1))), (128, 64)).transpose(), cmap=plt.get_cmap('gray'))
ax.set_title('90% Reconstruction');

ax = fig.add_subplot(1, 5, 3)
ax.imshow(np.reshape(pca.components_[0:top95,:].transpose().dot(np.reshape(transformed[
0,0:top95], (-1, 1))), (128, 64)).transpose(), cmap=plt.get_cmap('gray'))
ax.set_title('95% Reconstruction');

ax = fig.add_subplot(1, 5, 4)
ax.imshow(np.reshape(pca.components_[0:top99,:].transpose().dot(np.reshape(transformed[
0,0:top99], (-1, 1))), (128, 64)).transpose(), cmap=plt.get_cmap('gray'))
ax.set_title('99% Reconstruction');

ax = fig.add_subplot(1, 5, 5)
ax.imshow(np.reshape(pca.components_[:,:].transpose().dot(np.reshape(transformed[0,:],
(-1, 1))), (128, 64)).transpose(), cmap=plt.get_cmap('gray'))
ax.set_title('Full Reconstruction');
```



In [17]:

```
transformed_train_90 = transformed[:, 0:top90]

transformed_test = pca.transform(d2_test_dataset)
transformed_test_90 = transformed_test[:, 0:top90]

transformed_probe = pca.transform(d2_probe_dataset)
transformed_probe_90 = transformed_probe[:, 0:top90]

distance = np.sqrt(np.sum((transformed_train_90 - transformed_test_90[0,:])**2, axis=1
))

index_min = np.argmin(distance)

print(train_gnd[index_min])
print(test_gnd[0])
```

```
0231
1200
```

CMC for PCA - 90

In [18]:

```
vector = np.vectorize(np.int)

ID_1 = vector(np.array(test_gnd))
ID_2 = vector(np.array(test_pro_gnd))
Feat_1 = transformed_test_90
Feat_2 = transformed_probe_90
print(Feat_1)
print(Feat_2)
```

```
[[-8.49220617e+02  1.83749737e+02 -6.54313054e+02 ... -4.58680802e+01
  -1.09986638e+02 -7.86400808e+01]
 [-3.64926407e+02 -1.03960456e+03  1.04404176e+01 ... -3.31126700e+01
   7.65057545e-01 -5.08426061e+01]
 [ 1.88868883e+03  2.64765370e+03  1.05595416e+02 ...  2.46982324e+01
  -5.00788847e+01  8.52737718e+01]
 ...
 [-1.70274140e+03 -1.14014197e+03 -9.44224282e+02 ...  1.27089824e+02
  -9.76317538e+00 -2.13296224e+01]
 [-5.11923554e+02  6.92347505e+02 -7.08931015e+02 ... -3.66523329e+01
  -9.46441361e+01  7.22179543e+01]
 [ 8.68886202e+02 -4.64037805e+02  2.11727342e+03 ... -7.33490915e+00
  -2.98766253e+01 -1.13083337e+02]]
[[-1.01270063e+03  1.55429417e+02 -6.78405830e+02 ...  9.97964129e+01
   2.40295361e+01 -4.31671586e-01]
 [-2.78922823e+03  2.67592527e+03  1.25071424e+03 ...  7.20875117e+01
  -6.13149251e+01 -2.35690161e+01]
 [ 3.16695051e+03  5.79911683e+02 -8.05360941e+02 ... -4.89490968e+01
   5.54554675e+01 -1.22451800e+02]
 ...
 [ 5.79157766e+02 -1.24738571e+03 -1.50131049e+03 ... -1.73412096e+02
   1.48557748e+02  4.14689706e+01]
 [ 2.73372195e+03  2.29472244e+02 -1.27164897e+03 ...  1.61053500e+01
   8.37480288e+00 -6.51915946e+01]
 [-9.61418338e+02  2.55551297e+02  9.86341587e+02 ...  3.59300290e+01
  -1.09596689e+02  2.23187980e+01]]
```

In [19]:

```
# storage for ranked histogram
ranked_histogram = np.zeros(len(test_gnd))

# loop over all IDs in the probe set
for i in range(len(ID_2)):
    # get the true ID of this sample
    true_ID = ID_2[i]
    print('Searching for ID %d' % (true_ID))

    # get the distance between the current probe and the whole gallery, L1 distance her
e. Note that L1
    # may not always be the best choice, so consider your distance metric given your pr
oblem
    dist = np.sqrt(np.sum(((Feat_2[:] - Feat_1[i])**2), axis=1)) #This one needs to cha
nge - Too large?
    print('Distance between source feature and all target features:')
    print(dist)

    # get the sorted order of the distances
    a = np.argsort(dist)
    # apply the order to the gallery IDs, such that the first ID in the list is the clo
sest, the second
    # ID is the second closest, and so on
    ranked = ID_1[a]
    print('Ranked IDs for query:')
    print(a)

    # find the location of the True Match in the ranked list
    ranked_result = np.where(ranked == true_ID)[0][0]
    print(ranked_result)

    # store the ranking result in the histogram
    ranked_histogram[ranked_result] += 1
    print('')

print(ranked_histogram)
```

```
009952 6114.93845201
 6266.9726034  6878.03048626 5789.24522253 8712.48237164 6210.23814123
 6220.2664637  6507.41174164 4770.2778351  7267.97085443 5522.22533897
 5702.19139662 4889.42695119 6642.17449823 4936.06796975 5300.63874506
 5355.49511911 6589.41344298 5854.43812857 5427.66251618 5585.26508857
 8177.25032289 5415.62065228 6188.04224032 6601.65500431 5707.74718017
 6093.11256623 4651.43321237 6561.3571771  6558.41829815 8420.91705063
 6107.52400439 6972.88941827 6471.6148855  5657.88268721 5213.11514206
 6343.68468744 7265.22050182 6662.37638822 5660.13185779 4827.90542267
 6472.73494776 4909.46663359 6441.98589465 6016.63402064 4930.98465906
 4953.82593793 5747.98961603 5826.82492682 5615.4109062  5246.64424561
 5487.32739373]
Ranked IDs for query:
[116 105 171  19 147   3  73 199 245 117 238 276 128 187 180 120 235 161
 119 129 257 151  30 241  91 289  12  82 236 261  95 158 291 176 294 263
  79 295 153  38 208 227 149 111  45  34  48 122  17 103 178  72 181  31
  28 198 209 131  88 168 102 284 154  60 299 127  27 225 148  93 240 205
 264  86 172 196 215 200 243  11  92 265   4  74 101 195  98 242 206 271
 268  53  66  54 300  56 191 183  50 110   0 184 259 140 118 231 222 192
 232  36 269 207 170 298  90 146  70  26 283 288  41 224  87 229 260 274
 237  71  40 137 296  14 218 252 177 175  85 297 150 211  39  55 267   5
 234  59 142  65 230 210 115   2 155  49   8  61 124 293  29 201  52 109
  44 275 159 280 249  23 247  58  68  57  16  63 213 272 254 255  24  21
 141 162 250 104  64  25 239 285  80 194  13 193  75 248  97 106  37 292
 246 160  43 282 290 125 133 203 202  20 256 219 157 156  78 278 277 134
 130  46 266 273 185  32 262 287 221 226 233 144 165  10 166  18  67 204
 152  51 217 197   6 251  35 107 212 223 121 214 281  83 182  84 138 108
  77  96   7 164 190 112 167  33 173 216 286 258 220 179 132  62 169  81
 123  89  22 145 100   9 113 188 163 126   1 174 189  94  99  15 186 143
 228  76 139 135  47 114 270 244  42 136  69 279 253]
137

Searching for ID 1498
Distance between source feature and all target features:
[4769.11920122 6383.03225568 7542.55162014 5461.59939271 4758.93015124
 4525.23800913 5429.73644537 6510.98174194 4878.44176079 5259.67694557
 5522.36831873 7195.01645956 7221.75318468 5505.74935224 5141.7682297
 6212.44689428 5760.98491324 4830.17434072 4449.34857922 6129.68067495
 4554.68470074 4752.6519907  5252.52144723 6137.64790497 4919.25614282
 4737.05308449 6648.96476012 5441.64869411 6912.1853546  5845.60275122
 4998.75634617 6128.17915525 6173.76290707 7098.36144415 5594.474193
 5978.71616344 6439.51317675 7659.76932264 5571.13842284 6173.66602713
 5005.26512689 5090.7520303  6892.41181706 5321.44129064 4756.96836201
 4871.76853047 5280.06595128 7329.54328752 4403.06911347 4367.92722797
 4915.81966796 6341.47163142 5353.26084252 4609.30605631 6156.42346433
 6300.30687496 7022.86511363 4671.86719175 8117.02861366 4967.20962681
 6098.33514561 4883.65994633 5231.3058021  4435.04478967 4429.63659408
 5918.8881855  5002.85382662 7666.12638023 5218.08508425 6175.26821861
 4615.96688006 4888.39406826 5709.67640702 5410.80759363 6289.44542563
 6775.9108819  6714.00213646 6002.6958951  4371.33467376 5873.1655486
 7826.90824666 4982.82743491 5251.93467675 6020.97100476 6105.68249339
 6825.75044207 4276.87460343 4382.8669824  6501.0416463  8249.92558241
 4768.58457039 4806.87799223 4847.93815516 4823.14532494 6727.65173982
 4332.94898755 5233.67214818 4786.48593489 5943.58938505 7465.03070636
 6120.99819815 4953.3938561  5269.03053616 5454.63492838 6254.96249167
 5906.4790874  3856.3973388  4686.79211542 4890.74766188 4719.92691786
 6062.12875865 7126.74511215 5966.66758378 5878.54783306 9283.97403929
 7263.61860526 5639.94121107 5495.46202476 5046.01155724 6169.39199707
 7488.08698069 5032.87917646 4368.95942544 6524.6742911  4392.69767207
 5140.64132749 6359.82402363 6014.03606129 6357.08006148 5003.7024815
 6053.6861966  5388.331544   6391.0800076  5260.81015074 7031.19759127
```

```
 7127.71491923 5867.01956998 6934.43847931 5858.87506503 7438.84907527
 5456.42774007 7246.58355343 4983.83852535 6066.37115359 5713.55784367
 9101.5295083  4495.48468754 5758.55260959 5591.87870131 6737.53493052
 5413.72074548 5963.10972729 5326.16150445 6533.17351584 7261.53747831
 5556.32716403 5968.39640461 6364.09748926 5109.59815883 7510.1963714
 7179.74485527 6525.6015072  5122.26197022 8352.34964643 6920.08289239
 7057.54888164 7373.00226489 5721.45966164 5417.36439416 4832.95386956
 5987.01543958 5484.78622812 7279.39314785 5383.59718935 6483.09860764
 4368.83059221 5462.60703452 5201.8454082  5715.50339692 5747.31795136
 4709.96307102 5683.87494544 5075.40090045 6462.30075999 6253.58098806
 5778.67849951 7821.0445502  5670.48202318 5071.78751246 6666.5307645
 7449.39996534 5250.7076178  5799.37758295 4389.5619478  5434.39962631
 4484.42631591 5259.35368695 6363.19749801 7731.49785904 5730.55076307
 5230.37903837 5518.08891908 5881.92180249 4298.15530109 6218.64804572
 5922.85104972 4919.33988228 4990.70755094 5548.11732313 4771.24938666
 4491.71591924 5647.73284118 5199.76105948 6213.79172612 4925.57270313
 4682.02947152 5627.41712592 4501.75918007 4703.60955814 5398.47561943
 6527.26204644 4912.16199978 6567.22332344 4345.5005912  5601.00556195
 4922.55357284 6033.71318841 5131.65260149 5748.06282082 4414.1506663
 5574.81531146 4656.11615632 8026.14915795 5040.59493778 5712.84352546
 5006.17903968 5765.23575127 5142.81954002 5477.65416907 4668.59803571
 4500.4027123  5072.87927679 6150.39593792 4203.15977983 7470.40016155
 5897.60720012 4938.79921156 3805.44080935 3826.96957468 4642.66738834
 7338.03840216 5789.62132692 4788.78839039 6753.3230159  6922.06803059
 4158.13125729 4878.0251065  4762.45884721 6772.43195684 6077.43528197
 5141.54885976 8139.80659042 5712.91321875 5033.61723766 4468.47849419
 5846.96798831 5562.91410316 5634.46049593 4409.2431651  6006.64908334
 6512.96608824 4340.7631233  5154.18623138 6312.84480328 5149.60689762
 6468.75776234 4218.28554511 5602.47709996 4467.45891035 6577.90719931
 7293.34464693 4891.87314128 4940.30487648 6841.79222534 6778.2882922
 5396.44244158 5393.96528526 4768.29473962 5013.1917369  5072.597718
 7542.52247045 7113.34178441 4581.50912756 4917.91791274 4499.1872816
 6485.69469804 5514.01829803 7565.67912552 4714.17243089 6703.06216482
 4991.678484  ]
Ranked IDs for query:
[247 248 106 255 243 276  86 203  95 271 223  49 175 122  78  87 193 124
  48 268 229  64  63  18 278 264 195 210 146 294 240 217   5  20 292  53
  70 249 231 239  57 215 107 218 180 298 109  25  21  44   4 257 287  90
   0 209  97 252  91  93  17 169  92  45 256   8  61  71 108 281 221  50
 293  24 206 225 214 246 282 101  59  81 142 207 300  30  66 129  40 235
 288 121 263 233 118 188 289 241 182  41 158 162 227 125 260  14 237 274
 272 212 177  68 200  62  96 191  82  22 196   9 133 102  46  43 152  52
 173 131 286 285 219  73 150 168   6 194  27 103 140   3 176 238 171 117
  13 296 201  10 208 155 266  38 230 148  34 224 277 216 267 116 211 187
 181  72 234 262 144 178 167 199 179 228 147  16 236 185 251 192  29 265
 138 136  79 113 202 245 105  65 205  98 151 112 156  35 170  77 269 127
  83 226 130 110 143 259  60  84 100  31  19  23 242  54 119  39  32  69
  15 213 204 184 104  74  55 273  51 128 126 197 157   1 132  36 183 275
 174 295  88   7 270 123 161 220 153 222 279  26 189 299  76  94 149 253
 258  75 284  85 283  42  28 164 254 137  56 134 165  33 291 111 135 160
  11  12 141 154 115 172 280  47 250 166 139 190  99 244 120 159 290   2
 297  37  67 198 186  80 232  58 261  89 163 145 114]
45

Searching for ID 1499
Distance between source feature and all target features:
[4108.16410418 5911.15536052 6588.95472756 4437.79964932 4398.19558477
 5015.29888309 5308.17057637 5255.31578711 4737.16792309 5348.18643421
 5347.82543269 6171.97273503 6046.01142056 5127.62456974 5446.16244525
 6133.20463708 5185.87303461 4474.10434416 5289.20318488 5602.47832398
 4787.51852049 4904.73861536 5003.84250975 6905.70401047 4953.26371972
```

```
 5201.64796192 5006.75952073 5401.6738461  6534.32186368 5377.96389735
 4656.89465022 4911.22297281 5387.55546948 5376.77784553 4393.11111696
 4945.36689831 5279.19981884 6848.70598795 4947.75595743 4666.48650384
 4021.8138791  4969.98675473 6203.63468554 6086.30226017 5066.21195315
 5114.80623664 5460.57199638 6319.1300203  4550.53070132 5729.1629992
 5330.20809774 6519.52291546 5560.45327176 5522.47965752 5911.5063926
 5006.09356479 6676.42266763 5280.04374488 7571.14184489 6074.96135603
 5882.36516399 5344.55200912 5082.44913844 5779.33354448 5377.31400593
 6452.81178969 4408.42073427 6461.49512891 5766.2180786  5752.23509379
 4879.60917357 5568.81063984 5269.70167952 4782.33395435 5033.73069091
 5409.30201902 6065.22720692 5508.4873737  4801.12286217 5388.77779546
 6297.56890944 4988.65732989 5395.77129941 6099.89759815 5789.33234301
 5021.31843058 4192.66962287 4768.56001918 5600.13467913 6686.28367628
 5166.85489858 4468.11630436 4806.25139447 5119.69195425 5875.8718801
 4170.14718009 4869.7481447  4397.04119605 5846.18267134 6720.97973217
 5503.78090658 5724.88040927 5043.74985135 5288.65058725 6242.53156306
 5166.78940035 4455.62933986 5172.11763295 5545.46444113 5222.36601704
 5888.09159376 6154.19843623 5287.4022151  5730.31391426 7676.44208045
 6249.45722264 4810.23692748 4773.29098995 5812.80530925 5444.13930433
 6330.09189467 5194.50289133 4440.07661143 6371.33517462 4637.96754429
 4761.87118952 5439.41699828 6018.96767336 5577.50674328 4236.99994485
 6576.51147158 5000.14093976 5706.58876672 6110.63031148 5476.0958187
 7263.0833007  5791.93963792 5674.84887734 5108.09715347 5957.60778458
 4179.17319139 5761.78051622 5162.48008602 6194.09350456 6162.33654347
 7384.74974366 5083.04054941 4962.94202932 5170.20121132 6558.3513186
 5287.51005231 5538.36394109 5927.89044333 5514.96748061 6808.09038217
 4920.18026323 5454.56842393 5774.7372006  4992.52495217 5804.84483528
 6556.39007378 5405.70007164 5756.64086486 6600.92644031 5352.08666973
 6766.15814166 5769.05137498 5349.54890603 5015.89023111 5329.49586061
 5312.9717157  5096.14118614 6034.79276374 4955.40878744 6031.04628326
 5492.38103831 4421.44022406 5107.50627882 5308.97443418 5910.11741887
 4490.880087   4751.28927681 6060.41163971 5181.78798607 5765.06955709
 5021.9742683  8425.92130724 5403.01161569 5479.90628634 5356.77515953
 6073.80868505 4891.98039523 5134.07762324 4561.69552702 4792.94954996
 4436.59379023 5841.49153429 5682.51559107 6192.57415533 5109.2100271
 4135.10477759 5110.80292932 4917.96014791 4882.43108745 5377.73284515
 4972.3748959  5030.01475144 4371.76677221 5491.9925511  4895.18382976
 4893.01776077 5757.52807015 6223.09129595 5372.08055486 5738.93318274
 4511.00369704 5545.18253646 5299.00632493 4828.46999099 5594.70833879
 5628.51491104 4481.41385631 4695.66442343 5111.31275204 5662.20981021
 5142.62708777 4929.56300494 4924.67525652 5425.9043373  4490.38058173
 5565.79437174 4555.63610284 7273.92205684 5897.26382429 5402.35588249
 3952.39603324 5352.7809687  4854.18477353 5088.20841951 5018.60151664
 4958.55178938 5168.91519434 5815.43771182 4356.17362785 6200.88211458
 4730.73232114 5438.10708055 5529.3289125  5289.14454882 5367.66370411
 7320.99982638 5891.75528143 5494.08688647 6492.94660778 6115.15559164
 4518.53611612 4770.24636129 4521.2455312  5561.81564759 5431.33872366
 5851.12653529 6960.98211221 5708.982936   5380.34043209 5232.52509684
 5553.91613062 4703.2298637  6183.36975967 3722.33823568 5073.2336761
 6222.25295203 4804.84889788 5381.66585157 5814.77261616 4685.42502306
 5734.16519708 4267.73810762 4669.51127884 5042.61769553 5917.82635572
 5610.18033802 5327.94493298 6000.29771556 6207.95407515 7221.05709761
 4795.01117647 5310.19461473 4802.41731467 5719.12116891 4702.55710568
 6033.26856609 5774.57867243 5588.76234613 4952.93336998 5165.81266269
 6288.3679527  5861.79640957 5900.20144359 5154.35812719 5964.82743412
 4566.86094325]
Ranked IDs for query:
[268 235  40   0 200  95 140  86 129 276 243 207  34  97   4  66 176 195
   3 122 106  91  17 221 229 180 215 255 257  48 231 193 300 124  30  39
 277 274 222 289 266 245   8 181 125  87 256 117  73  20 194 285  78 287
 271  92 116 218 237  96  70 203 191 210 209  21  31 202 155 227 226  35
```

```
  38 293  24 173 240 147  41 205  81 158 131  22  55  26   5 168 239  85
 185 206  74 278 102  44 269  62 146 238 171 177 138 199 201 223  45  93
  13 192 225 298 142 294 105  90 241 148 107 183  16 121  25 109 264   7
  72  36  57 112 150 103 248  18 217   6 178 286 170 281 169  50  61  10
   9 167 164 236 189 249 213  33  64 204  29 263 272  32  79  82  27 234
 187 161  75 228 259 246 126 119  14 156  46 134 188 208 175 252 100  77
 153  53 247 151 216 108 265  52 258 230  71 128 292 219  88  19 280 220
 224 137 197 132 262 288 101  49 113 275 214  69 162 211 141 184  68 166
 291 157  63  84 136 159 118 273 242 196  98 260 296  94  60 110 251 233
 297 179   1  54 279 152 139 299 282 127 174 290 172  12 182  76 190  59
  43  83 133 254  15 111 144  11 267 198 143 244  42 283 270 212 104 115
 295  80  47 120 123  65  67 253  51  28 160 149 130   2 163  56  89  99
 165 154  37  23 261 284 135 232 250 145  58 114 186]
241

Searching for ID 1500
Distance between source feature and all target features:
[4983.21840189 6065.64286489 6180.20255619 3948.77364065 4641.25489815
 5063.00246578 6344.70300796 6797.53749228 5151.0335208  6123.54155244
 6279.60678962 5984.11325915 4883.08756573 5120.07594596 5289.95123869
 6383.93079907 5478.75713817 5273.44787706 6108.94836956 4559.35778475
 6169.37181362 5261.25536175 6432.83783848 5905.93417918 5451.04454653
 6084.75882089 5702.47644029 5379.43573346 5115.32912    5769.94879121
 4457.76030458 4202.12499442 6282.89025037 6773.80331112 4008.61001221
 6310.60625955 5649.42203408 6151.95806354 4598.55286983 5502.80863489
 4442.98163368 4904.65398985 7775.64325771 5852.8939449  5168.66363438
 4932.77503512 6460.7554949  6919.72857121 4464.91106808 5505.45407356
 4792.04231809 5197.30477439 5085.46732205 4408.60942303 5561.48629127
 6009.76703065 5441.7998477  5064.84609109 5680.24027161 5319.39334376
 4633.14249138 4910.89593698 6332.86806389 5282.74979802 5466.15511961
 4763.507063   5539.99650035 7418.19151969 4860.29164468 7322.75590779
 4924.03120784 4824.87591289 4479.66921427 5571.77387232 5749.61855646
 6296.88486799 6413.00326464 7343.52047498 5319.47146437 4807.48771885
 6338.14493297 6564.91294956 3960.15067061 5928.12909253 6403.11027927
 5660.23761753 5107.62323411 5160.73205975 4213.79884263 6737.50791232
 5014.95907641 4367.57764845 4676.84452031 4703.17606235 6864.87015104
 4829.70513097 6073.9667105  5312.93669716 6135.89903883 6487.92084892
 6422.92198767 4996.91435326 3693.01962046 4389.77595003 4543.51530314
 4269.31210518 5281.11296053 5551.71514062 5645.20016002 5216.18304106
 5376.67712531 5205.4233334  5901.45043052 6659.1026677  6814.9442576
 6608.43297767 4857.49285549 3954.07838886 4909.01344974 4775.9113004
 4735.66794036 6320.97700282 5063.22741926 6185.43243536 5135.06608221
 5345.05047953 7238.81320228 5836.82169788 4641.3225395  4013.66558839
 5176.4281785  4397.6365464  6669.58466148 5002.62310734 5751.49382652
 6610.42687401 7177.11305742 6283.90714284 6032.88632298 7402.51040828
 5506.43545041 5203.64464745 5792.12346495 6782.94620182 6095.09922783
 7075.53485266 5007.23811036 4875.26142904 4004.9274539  4947.64563511
 5157.13072233 4375.56465798 5620.61328565 4988.39964801 5103.5395028
 5494.34204754 4914.81376937 4748.84283566 4773.30040593 5603.45495822
 6401.00230939 5451.99071817 5704.6065425  7185.27366489 6288.87144988
 5677.10223039 7383.81998631 5686.50736072 4051.80808967 6468.5226368
 5023.34829801 4714.44723927 6239.43504815 6250.85446843 6286.37307638
 5056.18816269 4356.10082061 5621.76599639 5672.5256352  5791.52785721
 4213.0987223  5703.25065577 5357.41693641 4881.17771203 4512.53733343
 4859.59140246 7831.49163669 4871.74320706 6433.34817397 7114.4685455
 6544.93747574 4821.03509265 5768.87864134 5302.72793431 5116.74106846
 5423.90074521 5540.9397105  5676.78693348 5955.34729139 4230.37851067
 5209.76501258 5992.48812971 6382.54305048 5175.96508771 7010.82079373
 4841.33256866 4561.29010942 5216.155227   4054.02477657 4298.90878269
 4949.57827115 5168.51238554 6181.0240047  5482.49253199 6054.783188
 5101.85537553 5757.52583707 5654.7845469  4583.66954504 4293.71203129
```

```
 5578.75828661 5850.82560484 5535.71764188 6294.02205561 4044.56344236
 4736.43770061 5745.33549531 3844.47908882 6538.15212    4473.60738574
 4675.16335944 4872.06345788 6122.07249763 5783.98170595 5903.0093815
 4228.83290766 5595.38019378 5476.54749097 4228.8909983  6162.34213118
 4560.36990361 4619.46342232 5824.91558791 4098.65469662 7708.22292545
 4050.21019562 5828.98335951 5167.9784617  5868.41727007 5353.87522765
 5030.94032321 5744.59572863 4844.82050782 7416.38676131 6357.86521246
 5714.71283651 5702.8016211  4688.04010517 6660.57166577 3732.98315374
 5141.8782879  5601.06399802 6199.91574262 4088.71266782 4427.89780879
 5730.77715688 6148.17111163 4644.83505308 5133.48817932 5071.01653245
 6836.38528015 5416.07228908 4579.50318992 4941.82567193 4573.19811181
 6361.057206   4258.95450172 5514.32885354 6678.85051413 7458.76662998
 6241.1711345  6153.97992918 5672.01461986 5214.75103834 5425.1154187
 4975.23269817 6903.6058645  5503.88834342 4495.49498639 5241.79106341
 6539.30530883 6241.82881018 5886.10591284 5375.91763261 4912.44262002
 5193.31916097 4944.34001533 6586.64702927 5852.36501183 6507.78569065
 4191.12405296]
Ranked IDs for query:
[102 259 227   3 117  82 148  34 129 224 245 168 208 263 243 300  31 180
  88 235 238 199 276 105 219 209 176  91 151 103 131  53 264  40  30  48
 229  72 288 184 104  19 240 206 274 272 218  38 241  60   4 128 267 230
  92 257  93 171 120 225 157  65 158 119  50  79 191  71  95 205 252 116
 185  68 187 231 147 183  12  41 118  61 294 156  70  45 273 296 149 210
 285   0 153 101 133 146  90 170 250 175   5 122  57 269  52 215 154  86
  28 194  13 268 124 260   8 150  87 247 211  44 203 130 295  51 141 111
 200 283 207 109 289  21  17 106  63  14 193  97  59  78 125 249 182 293
 110  27 271 195 284  56  24 161  64 237  16 213 155  39 287  49 140 277
 222  66 196 107  54  73 220 236 261 159 152 177 108  36 217  85 282 178
 197 165  58 167  26 256 181 162 255 265 251 226  74 134 216 192  29 233
 179 142 242 246 127 221 298  43 248 292 112 234  23  83 198  11 201  55
 138 214   1  96  25 144  18 232   9  98 266  37 281 239  20   2 212 123
 262 172 280 291 173  10  32 137 174 164 223  75  35 121  62  80   6 254
 275 202  15 160  84  76 100  22 188  46 169  99 299 228 290 190  81 297
 115 135 113 258 132 278  89  33 143   7 114 270  94 286  47 204 145 189
 136 163 126  69  77 166 139 253  67 279 244  42 186]
15

[29.  5.  9.  4.  6.  4.  3.  1.  3.  4.  4.  6.  2.  1.  2.  3.  1.  2.
  1.  0.  1.  0.  2.  2.  2.  3.  2.  0.  1.  1.  4.  6.  1.  4.  0.  1.
  1.  1.  1.  0.  4.  1.  1.  1.  3.  2.  2.  2.  0.  2.  0.  2.  2.  2.
  0.  1.  1.  1.  0.  1.  1.  2.  5.  1.  0.  0.  2.  1.  0.  1.  3.  0.
  1.  4.  2.  1.  0.  1.  2.  1.  1.  1.  1.  1.  1.  2.  1.  0.  2.  2.
  1.  2.  0.  1.  1.  2.  0.  1.  1.  1.  2.  2.  3.  1.  0.  0.  0.  4.
  2.  1.  1.  1.  0.  1.  0.  0.  0.  0.  3.  1.  1.  0.  0.  0.  0.  2.
  0.  1.  1.  0.  0.  2.  0.  1.  0.  1.  0.  2.  1.  0.  0.  1.  0.  1.
  2.  1.  0.  2.  0.  0.  0.  0.  1.  0.  0.  1.  0.  1.  0.  0.  2.  3.
  1.  0.  0.  2.  1.  0.  0.  2.  1.  1.  1.  0.  0.  1.  1.  1.  0.  1.
  1.  0.  0.  0.  0.  1.  0.  1.  0.  0.  1.  1.  2.  0.  1.  0.  1.  2.
  0.  0.  0.  0.  1.  3.  0.  2.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  1.  1.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.
  1.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.
  1.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  1.
  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.]
```

In [20]:

```
print(Feat_1.shape)
```

```
(301, 354)
```

In [21]:

```
print(Feat_2.shape)
```

```
(301, 354)
```

In [22]:

```
cmc = np.zeros(301)
for i in range(301):
    cmc[i] = np.sum(ranked_histogram[:(i + 1)])

print(cmc)
```

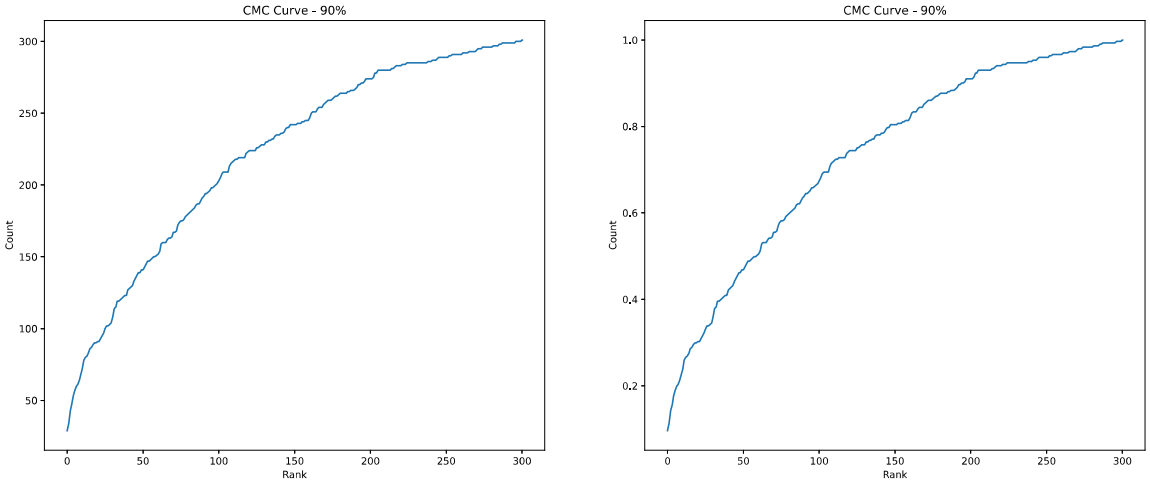
```
[ 29.  34.  43.  47.  53.  57.  60.  61.  64.  68.  72.  78.  80.  81.
  83.  86.  87.  89.  90.  90.  91.  91.  93.  95.  97. 100. 102. 102.
 103. 104. 108. 114. 115. 119. 119. 120. 121. 122. 123. 123. 127. 128.
 129. 130. 133. 135. 137. 139. 139. 141. 141. 143. 145. 147. 147. 148.
 149. 150. 150. 151. 152. 154. 159. 160. 160. 160. 162. 163. 163. 164.
 167. 167. 168. 172. 174. 175. 175. 176. 178. 179. 180. 181. 182. 183.
 184. 186. 187. 187. 189. 191. 192. 194. 194. 195. 196. 198. 198. 199.
 200. 201. 203. 205. 208. 209. 209. 209. 209. 213. 215. 216. 217. 218.
 218. 219. 219. 219. 219. 219. 222. 223. 224. 224. 224. 224. 224. 226.
 226. 227. 228. 228. 228. 230. 230. 231. 231. 232. 232. 234. 235. 235.
 235. 236. 236. 237. 239. 240. 240. 242. 242. 242. 242. 242. 243. 243.
 243. 244. 244. 245. 245. 245. 247. 250. 251. 251. 251. 253. 254. 254.
 254. 256. 257. 258. 259. 259. 259. 260. 261. 262. 262. 263. 264. 264.
 264. 264. 264. 265. 265. 266. 266. 266. 267. 268. 270. 270. 271. 271.
 272. 274. 274. 274. 274. 274. 275. 278. 278. 280. 280. 280. 280. 280.
 280. 280. 280. 280. 281. 281. 282. 283. 283. 283. 283. 284. 284. 284.
 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285.
 286. 286. 286. 287. 287. 287. 288. 289. 289. 289. 289. 289. 289. 289.
 290. 290. 291. 291. 291. 291. 291. 291. 291. 292. 292. 292. 292. 293.
 293. 293. 293. 293. 294. 295. 295. 295. 296. 296. 296. 296. 296. 296.
 296. 297. 297. 297. 297. 298. 298. 299. 299. 299. 299. 299. 299. 299.
 299. 299. 300. 300. 300. 300. 301.]
```

In [23]:

```
fig = plt.figure(figsize=[20, 8])
ax = fig.add_subplot(1, 2, 1)
ax.plot(cmc)
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 90%')
ax = fig.add_subplot(1, 2, 2)
ax.plot(cmc/len(test_gnd))
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 90%')
```

Out[23]:

```
Text(0.5, 1.0, 'CMC Curve - 90%')
```



CMC for PCA-95

In [24]:

```
transformed_test = pca.transform(d2_test_dataset)
transformed_test_95 = transformed_test[:, 0:top95]

transformed_probe = pca.transform(d2_probe_dataset)
transformed_probe_95 = transformed_probe[:, 0:top95]
```

In [25]:

```
# storage for ranked histogram
ranked_histogram = np.zeros(len(test_gnd))

# loop over all IDs in the probe set
for i in range(len(ID_2)):
    # get the true ID of this sample
    true_ID = ID_2[i]
    print('Searching for ID %d' % (true_ID))

    # get the distance between the current probe and the whole gallery, L1 distance her
e. Note that L1
    # may not always be the best choice, so consider your distance metric given your pr
oblem
    dist = np.sqrt(np.sum(((Feat_2[:] - Feat_1[i])**2), axis=1)) #This one needs to cha
nge - Too large?
    print('Distance between source feature and all target features:')
    print(dist)

    # get the sorted order of the distances
    a = np.argsort(dist)
    # apply the order to the gallery IDs, such that the first ID in the list is the clo
sest, the second
    # ID is the second closest, and so on
    ranked = ID_1[a]
    print('Ranked IDs for query:')
    print(a)

    # find the location of the True Match in the ranked list
    ranked_result = np.where(ranked == true_ID)[0][0]
    print(ranked_result)

    # store the ranking result in the histogram
    ranked_histogram[ranked_result] += 1
    print('')

print(ranked_histogram)
```

```
009952 6114.93845201
 6266.9726034  6878.03048626 5789.24522253 8712.48237164 6210.23814123
 6220.2664637  6507.41174164 4770.2778351  7267.97085443 5522.22533897
 5702.19139662 4889.42695119 6642.17449823 4936.06796975 5300.63874506
 5355.49511911 6589.41344298 5854.43812857 5427.66251618 5585.26508857
 8177.25032289 5415.62065228 6188.04224032 6601.65500431 5707.74718017
 6093.11256623 4651.43321237 6561.3571771  6558.41829815 8420.91705063
 6107.52400439 6972.88941827 6471.6148855  5657.88268721 5213.11514206
 6343.68468744 7265.22050182 6662.37638822 5660.13185779 4827.90542267
 6472.73494776 4909.46663359 6441.98589465 6016.63402064 4930.98465906
 4953.82593793 5747.98961603 5826.82492682 5615.4109062  5246.64424561
 5487.32739373]
Ranked IDs for query:
[116 105 171  19 147   3  73 199 245 117 238 276 128 187 180 120 235 161
 119 129 257 151  30 241  91 289  12  82 236 261  95 158 291 176 294 263
  79 295 153  38 208 227 149 111  45  34  48 122  17 103 178  72 181  31
  28 198 209 131  88 168 102 284 154  60 299 127  27 225 148  93 240 205
 264  86 172 196 215 200 243  11  92 265   4  74 101 195  98 242 206 271
 268  53  66  54 300  56 191 183  50 110   0 184 259 140 118 231 222 192
 232  36 269 207 170 298  90 146  70  26 283 288  41 224  87 229 260 274
 237  71  40 137 296  14 218 252 177 175  85 297 150 211  39  55 267   5
 234  59 142  65 230 210 115   2 155  49   8  61 124 293  29 201  52 109
  44 275 159 280 249  23 247  58  68  57  16  63 213 272 254 255  24  21
 141 162 250 104  64  25 239 285  80 194  13 193  75 248  97 106  37 292
 246 160  43 282 290 125 133 203 202  20 256 219 157 156  78 278 277 134
 130  46 266 273 185  32 262 287 221 226 233 144 165  10 166  18  67 204
 152  51 217 197   6 251  35 107 212 223 121 214 281  83 182  84 138 108
  77  96   7 164 190 112 167  33 173 216 286 258 220 179 132  62 169  81
 123  89  22 145 100   9 113 188 163 126   1 174 189  94  99  15 186 143
 228  76 139 135  47 114 270 244  42 136  69 279 253]
137

Searching for ID 1498
Distance between source feature and all target features:
[4769.11920122 6383.03225568 7542.55162014 5461.59939271 4758.93015124
 4525.23800913 5429.73644537 6510.98174194 4878.44176079 5259.67694557
 5522.36831873 7195.01645956 7221.75318468 5505.74935224 5141.7682297
 6212.44689428 5760.98491324 4830.17434072 4449.34857922 6129.68067495
 4554.68470074 4752.6519907  5252.52144723 6137.64790497 4919.25614282
 4737.05308449 6648.96476012 5441.64869411 6912.1853546  5845.60275122
 4998.75634617 6128.17915525 6173.76290707 7098.36144415 5594.474193
 5978.71616344 6439.51317675 7659.76932264 5571.13842284 6173.66602713
 5005.26512689 5090.7520303  6892.41181706 5321.44129064 4756.96836201
 4871.76853047 5280.06595128 7329.54328752 4403.06911347 4367.92722797
 4915.81966796 6341.47163142 5353.26084252 4609.30605631 6156.42346433
 6300.30687496 7022.86511363 4671.86719175 8117.02861366 4967.20962681
 6098.33514561 4883.65994633 5231.3058021  4435.04478967 4429.63659408
 5918.8881855  5002.85382662 7666.12638023 5218.08508425 6175.26821861
 4615.96688006 4888.39406826 5709.67640702 5410.80759363 6289.44542563
 6775.9108819  6714.00213646 6002.6958951  4371.33467376 5873.1655486
 7826.90824666 4982.82743491 5251.93467675 6020.97100476 6105.68249339
 6825.75044207 4276.87460343 4382.8669824  6501.0416463  8249.92558241
 4768.58457039 4806.87799223 4847.93815516 4823.14532494 6727.65173982
 4332.94898755 5233.67214818 4786.48593489 5943.58938505 7465.03070636
 6120.99819815 4953.3938561  5269.03053616 5454.63492838 6254.96249167
 5906.4790874  3856.3973388  4686.79211542 4890.74766188 4719.92691786
 6062.12875865 7126.74511215 5966.66758378 5878.54783306 9283.97403929
 7263.61860526 5639.94121107 5495.46202476 5046.01155724 6169.39199707
 7488.08698069 5032.87917646 4368.95942544 6524.6742911  4392.69767207
 5140.64132749 6359.82402363 6014.03606129 6357.08006148 5003.7024815
 6053.6861966  5388.331544   6391.0800076  5260.81015074 7031.19759127
```

```
 7127.71491923 5867.01956998 6934.43847931 5858.87506503 7438.84907527
 5456.42774007 7246.58355343 4983.83852535 6066.37115359 5713.55784367
 9101.5295083  4495.48468754 5758.55260959 5591.87870131 6737.53493052
 5413.72074548 5963.10972729 5326.16150445 6533.17351584 7261.53747831
 5556.32716403 5968.39640461 6364.09748926 5109.59815883 7510.1963714
 7179.74485527 6525.6015072  5122.26197022 8352.34964643 6920.08289239
 7057.54888164 7373.00226489 5721.45966164 5417.36439416 4832.95386956
 5987.01543958 5484.78622812 7279.39314785 5383.59718935 6483.09860764
 4368.83059221 5462.60703452 5201.8454082  5715.50339692 5747.31795136
 4709.96307102 5683.87494544 5075.40090045 6462.30075999 6253.58098806
 5778.67849951 7821.0445502  5670.48202318 5071.78751246 6666.5307645
 7449.39996534 5250.7076178  5799.37758295 4389.5619478  5434.39962631
 4484.42631591 5259.35368695 6363.19749801 7731.49785904 5730.55076307
 5230.37903837 5518.08891908 5881.92180249 4298.15530109 6218.64804572
 5922.85104972 4919.33988228 4990.70755094 5548.11732313 4771.24938666
 4491.71591924 5647.73284118 5199.76105948 6213.79172612 4925.57270313
 4682.02947152 5627.41712592 4501.75918007 4703.60955814 5398.47561943
 6527.26204644 4912.16199978 6567.22332344 4345.5005912  5601.00556195
 4922.55357284 6033.71318841 5131.65260149 5748.06282082 4414.1506663
 5574.81531146 4656.11615632 8026.14915795 5040.59493778 5712.84352546
 5006.17903968 5765.23575127 5142.81954002 5477.65416907 4668.59803571
 4500.4027123  5072.87927679 6150.39593792 4203.15977983 7470.40016155
 5897.60720012 4938.79921156 3805.44080935 3826.96957468 4642.66738834
 7338.03840216 5789.62132692 4788.78839039 6753.3230159  6922.06803059
 4158.13125729 4878.0251065  4762.45884721 6772.43195684 6077.43528197
 5141.54885976 8139.80659042 5712.91321875 5033.61723766 4468.47849419
 5846.96798831 5562.91410316 5634.46049593 4409.2431651  6006.64908334
 6512.96608824 4340.7631233  5154.18623138 6312.84480328 5149.60689762
 6468.75776234 4218.28554511 5602.47709996 4467.45891035 6577.90719931
 7293.34464693 4891.87314128 4940.30487648 6841.79222534 6778.2882922
 5396.44244158 5393.96528526 4768.29473962 5013.1917369  5072.597718
 7542.52247045 7113.34178441 4581.50912756 4917.91791274 4499.1872816
 6485.69469804 5514.01829803 7565.67912552 4714.17243089 6703.06216482
 4991.678484  ]
Ranked IDs for query:
[247 248 106 255 243 276  86 203  95 271 223  49 175 122  78  87 193 124
  48 268 229  64  63  18 278 264 195 210 146 294 240 217   5  20 292  53
  70 249 231 239  57 215 107 218 180 298 109  25  21  44   4 257 287  90
   0 209  97 252  91  93  17 169  92  45 256   8  61  71 108 281 221  50
 293  24 206 225 214 246 282 101  59  81 142 207 300  30  66 129  40 235
 288 121 263 233 118 188 289 241 182  41 158 162 227 125 260  14 237 274
 272 212 177  68 200  62  96 191  82  22 196   9 133 102  46  43 152  52
 173 131 286 285 219  73 150 168   6 194  27 103 140   3 176 238 171 117
  13 296 201  10 208 155 266  38 230 148  34 224 277 216 267 116 211 187
 181  72 234 262 144 178 167 199 179 228 147  16 236 185 251 192  29 265
 138 136  79 113 202 245 105  65 205  98 151 112 156  35 170  77 269 127
  83 226 130 110 143 259  60  84 100  31  19  23 242  54 119  39  32  69
  15 213 204 184 104  74  55 273  51 128 126 197 157   1 132  36 183 275
 174 295  88   7 270 123 161 220 153 222 279  26 189 299  76  94 149 253
 258  75 284  85 283  42  28 164 254 137  56 134 165  33 291 111 135 160
  11  12 141 154 115 172 280  47 250 166 139 190  99 244 120 159 290   2
 297  37  67 198 186  80 232  58 261  89 163 145 114]
45

Searching for ID 1499
Distance between source feature and all target features:
[4108.16410418 5911.15536052 6588.95472756 4437.79964932 4398.19558477
 5015.29888309 5308.17057637 5255.31578711 4737.16792309 5348.18643421
 5347.82543269 6171.97273503 6046.01142056 5127.62456974 5446.16244525
 6133.20463708 5185.87303461 4474.10434416 5289.20318488 5602.47832398
 4787.51852049 4904.73861536 5003.84250975 6905.70401047 4953.26371972
```

```
 5201.64796192 5006.75952073 5401.6738461  6534.32186368 5377.96389735
 4656.89465022 4911.22297281 5387.55546948 5376.77784553 4393.11111696
 4945.36689831 5279.19981884 6848.70598795 4947.75595743 4666.48650384
 4021.8138791  4969.98675473 6203.63468554 6086.30226017 5066.21195315
 5114.80623664 5460.57199638 6319.1300203  4550.53070132 5729.1629992
 5330.20809774 6519.52291546 5560.45327176 5522.47965752 5911.5063926
 5006.09356479 6676.42266763 5280.04374488 7571.14184489 6074.96135603
 5882.36516399 5344.55200912 5082.44913844 5779.33354448 5377.31400593
 6452.81178969 4408.42073427 6461.49512891 5766.2180786  5752.23509379
 4879.60917357 5568.81063984 5269.70167952 4782.33395435 5033.73069091
 5409.30201902 6065.22720692 5508.4873737  4801.12286217 5388.77779546
 6297.56890944 4988.65732989 5395.77129941 6099.89759815 5789.33234301
 5021.31843058 4192.66962287 4768.56001918 5600.13467913 6686.28367628
 5166.85489858 4468.11630436 4806.25139447 5119.69195425 5875.8718801
 4170.14718009 4869.7481447  4397.04119605 5846.18267134 6720.97973217
 5503.78090658 5724.88040927 5043.74985135 5288.65058725 6242.53156306
 5166.78940035 4455.62933986 5172.11763295 5545.46444113 5222.36601704
 5888.09159376 6154.19843623 5287.4022151  5730.31391426 7676.44208045
 6249.45722264 4810.23692748 4773.29098995 5812.80530925 5444.13930433
 6330.09189467 5194.50289133 4440.07661143 6371.33517462 4637.96754429
 4761.87118952 5439.41699828 6018.96767336 5577.50674328 4236.99994485
 6576.51147158 5000.14093976 5706.58876672 6110.63031148 5476.0958187
 7263.0833007  5791.93963792 5674.84887734 5108.09715347 5957.60778458
 4179.17319139 5761.78051622 5162.48008602 6194.09350456 6162.33654347
 7384.74974366 5083.04054941 4962.94202932 5170.20121132 6558.3513186
 5287.51005231 5538.36394109 5927.89044333 5514.96748061 6808.09038217
 4920.18026323 5454.56842393 5774.7372006  4992.52495217 5804.84483528
 6556.39007378 5405.70007164 5756.64086486 6600.92644031 5352.08666973
 6766.15814166 5769.05137498 5349.54890603 5015.89023111 5329.49586061
 5312.9717157  5096.14118614 6034.79276374 4955.40878744 6031.04628326
 5492.38103831 4421.44022406 5107.50627882 5308.97443418 5910.11741887
 4490.880087   4751.28927681 6060.41163971 5181.78798607 5765.06955709
 5021.9742683  8425.92130724 5403.01161569 5479.90628634 5356.77515953
 6073.80868505 4891.98039523 5134.07762324 4561.69552702 4792.94954996
 4436.59379023 5841.49153429 5682.51559107 6192.57415533 5109.2100271
 4135.10477759 5110.80292932 4917.96014791 4882.43108745 5377.73284515
 4972.3748959  5030.01475144 4371.76677221 5491.9925511  4895.18382976
 4893.01776077 5757.52807015 6223.09129595 5372.08055486 5738.93318274
 4511.00369704 5545.18253646 5299.00632493 4828.46999099 5594.70833879
 5628.51491104 4481.41385631 4695.66442343 5111.31275204 5662.20981021
 5142.62708777 4929.56300494 4924.67525652 5425.9043373  4490.38058173
 5565.79437174 4555.63610284 7273.92205684 5897.26382429 5402.35588249
 3952.39603324 5352.7809687  4854.18477353 5088.20841951 5018.60151664
 4958.55178938 5168.91519434 5815.43771182 4356.17362785 6200.88211458
 4730.73232114 5438.10708055 5529.3289125  5289.14454882 5367.66370411
 7320.99982638 5891.75528143 5494.08688647 6492.94660778 6115.15559164
 4518.53611612 4770.24636129 4521.2455312  5561.81564759 5431.33872366
 5851.12653529 6960.98211221 5708.982936   5380.34043209 5232.52509684
 5553.91613062 4703.2298637  6183.36975967 3722.33823568 5073.2336761
 6222.25295203 4804.84889788 5381.66585157 5814.77261616 4685.42502306
 5734.16519708 4267.73810762 4669.51127884 5042.61769553 5917.82635572
 5610.18033802 5327.94493298 6000.29771556 6207.95407515 7221.05709761
 4795.01117647 5310.19461473 4802.41731467 5719.12116891 4702.55710568
 6033.26856609 5774.57867243 5588.76234613 4952.93336998 5165.81266269
 6288.3679527  5861.79640957 5900.20144359 5154.35812719 5964.82743412
 4566.86094325]
Ranked IDs for query:
[268 235  40   0 200  95 140  86 129 276 243 207  34  97   4  66 176 195
   3 122 106  91  17 221 229 180 215 255 257  48 231 193 300 124  30  39
 277 274 222 289 266 245   8 181 125  87 256 117  73  20 194 285  78 287
 271  92 116 218 237  96  70 203 191 210 209  21  31 202 155 227 226  35
```

```
  38 293  24 173 240 147  41 205  81 158 131  22  55  26   5 168 239  85
 185 206  74 278 102  44 269  62 146 238 171 177 138 199 201 223  45  93
  13 192 225 298 142 294 105  90 241 148 107 183  16 121  25 109 264   7
  72  36  57 112 150 103 248  18 217   6 178 286 170 281 169  50  61  10
   9 167 164 236 189 249 213  33  64 204  29 263 272  32  79  82  27 234
 187 161  75 228 259 246 126 119  14 156  46 134 188 208 175 252 100  77
 153  53 247 151 216 108 265  52 258 230  71 128 292 219  88  19 280 220
 224 137 197 132 262 288 101  49 113 275 214  69 162 211 141 184  68 166
 291 157  63  84 136 159 118 273 242 196  98 260 296  94  60 110 251 233
 297 179   1  54 279 152 139 299 282 127 174 290 172  12 182  76 190  59
  43  83 133 254  15 111 144  11 267 198 143 244  42 283 270 212 104 115
 295  80  47 120 123  65  67 253  51  28 160 149 130   2 163  56  89  99
 165 154  37  23 261 284 135 232 250 145  58 114 186]
241

Searching for ID 1500
Distance between source feature and all target features:
[4983.21840189 6065.64286489 6180.20255619 3948.77364065 4641.25489815
 5063.00246578 6344.70300796 6797.53749228 5151.0335208  6123.54155244
 6279.60678962 5984.11325915 4883.08756573 5120.07594596 5289.95123869
 6383.93079907 5478.75713817 5273.44787706 6108.94836956 4559.35778475
 6169.37181362 5261.25536175 6432.83783848 5905.93417918 5451.04454653
 6084.75882089 5702.47644029 5379.43573346 5115.32912    5769.94879121
 4457.76030458 4202.12499442 6282.89025037 6773.80331112 4008.61001221
 6310.60625955 5649.42203408 6151.95806354 4598.55286983 5502.80863489
 4442.98163368 4904.65398985 7775.64325771 5852.8939449  5168.66363438
 4932.77503512 6460.7554949  6919.72857121 4464.91106808 5505.45407356
 4792.04231809 5197.30477439 5085.46732205 4408.60942303 5561.48629127
 6009.76703065 5441.7998477  5064.84609109 5680.24027161 5319.39334376
 4633.14249138 4910.89593698 6332.86806389 5282.74979802 5466.15511961
 4763.507063   5539.99650035 7418.19151969 4860.29164468 7322.75590779
 4924.03120784 4824.87591289 4479.66921427 5571.77387232 5749.61855646
 6296.88486799 6413.00326464 7343.52047498 5319.47146437 4807.48771885
 6338.14493297 6564.91294956 3960.15067061 5928.12909253 6403.11027927
 5660.23761753 5107.62323411 5160.73205975 4213.79884263 6737.50791232
 5014.95907641 4367.57764845 4676.84452031 4703.17606235 6864.87015104
 4829.70513097 6073.9667105  5312.93669716 6135.89903883 6487.92084892
 6422.92198767 4996.91435326 3693.01962046 4389.77595003 4543.51530314
 4269.31210518 5281.11296053 5551.71514062 5645.20016002 5216.18304106
 5376.67712531 5205.4233334  5901.45043052 6659.1026677  6814.9442576
 6608.43297767 4857.49285549 3954.07838886 4909.01344974 4775.9113004
 4735.66794036 6320.97700282 5063.22741926 6185.43243536 5135.06608221
 5345.05047953 7238.81320228 5836.82169788 4641.3225395  4013.66558839
 5176.4281785  4397.6365464  6669.58466148 5002.62310734 5751.49382652
 6610.42687401 7177.11305742 6283.90714284 6032.88632298 7402.51040828
 5506.43545041 5203.64464745 5792.12346495 6782.94620182 6095.09922783
 7075.53485266 5007.23811036 4875.26142904 4004.9274539  4947.64563511
 5157.13072233 4375.56465798 5620.61328565 4988.39964801 5103.5395028
 5494.34204754 4914.81376937 4748.84283566 4773.30040593 5603.45495822
 6401.00230939 5451.99071817 5704.6065425  7185.27366489 6288.87144988
 5677.10223039 7383.81998631 5686.50736072 4051.80808967 6468.5226368
 5023.34829801 4714.44723927 6239.43504815 6250.85446843 6286.37307638
 5056.18816269 4356.10082061 5621.76599639 5672.5256352  5791.52785721
 4213.0987223  5703.25065577 5357.41693641 4881.17771203 4512.53733343
 4859.59140246 7831.49163669 4871.74320706 6433.34817397 7114.4685455
 6544.93747574 4821.03509265 5768.87864134 5302.72793431 5116.74106846
 5423.90074521 5540.9397105  5676.78693348 5955.34729139 4230.37851067
 5209.76501258 5992.48812971 6382.54305048 5175.96508771 7010.82079373
 4841.33256866 4561.29010942 5216.155227   4054.02477657 4298.90878269
 4949.57827115 5168.51238554 6181.0240047  5482.49253199 6054.783188
 5101.85537553 5757.52583707 5654.7845469  4583.66954504 4293.71203129
```

```
 5578.75828661 5850.82560484 5535.71764188 6294.02205561 4044.56344236
 4736.43770061 5745.33549531 3844.47908882 6538.15212    4473.60738574
 4675.16335944 4872.06345788 6122.07249763 5783.98170595 5903.0093815
 4228.83290766 5595.38019378 5476.54749097 4228.8909983  6162.34213118
 4560.36990361 4619.46342232 5824.91558791 4098.65469662 7708.22292545
 4050.21019562 5828.98335951 5167.9784617  5868.41727007 5353.87522765
 5030.94032321 5744.59572863 4844.82050782 7416.38676131 6357.86521246
 5714.71283651 5702.8016211  4688.04010517 6660.57166577 3732.98315374
 5141.8782879  5601.06399802 6199.91574262 4088.71266782 4427.89780879
 5730.77715688 6148.17111163 4644.83505308 5133.48817932 5071.01653245
 6836.38528015 5416.07228908 4579.50318992 4941.82567193 4573.19811181
 6361.057206   4258.95450172 5514.32885354 6678.85051413 7458.76662998
 6241.1711345  6153.97992918 5672.01461986 5214.75103834 5425.1154187
 4975.23269817 6903.6058645  5503.88834342 4495.49498639 5241.79106341
 6539.30530883 6241.82881018 5886.10591284 5375.91763261 4912.44262002
 5193.31916097 4944.34001533 6586.64702927 5852.36501183 6507.78569065
 4191.12405296]
Ranked IDs for query:
[102 259 227   3 117  82 148  34 129 224 245 168 208 263 243 300  31 180
  88 235 238 199 276 105 219 209 176  91 151 103 131  53 264  40  30  48
 229  72 288 184 104  19 240 206 274 272 218  38 241  60   4 128 267 230
  92 257  93 171 120 225 157  65 158 119  50  79 191  71  95 205 252 116
 185  68 187 231 147 183  12  41 118  61 294 156  70  45 273 296 149 210
 285   0 153 101 133 146  90 170 250 175   5 122  57 269  52 215 154  86
  28 194  13 268 124 260   8 150  87 247 211  44 203 130 295  51 141 111
 200 283 207 109 289  21  17 106  63  14 193  97  59  78 125 249 182 293
 110  27 271 195 284  56  24 161  64 237  16 213 155  39 287  49 140 277
 222  66 196 107  54  73 220 236 261 159 152 177 108  36 217  85 282 178
 197 165  58 167  26 256 181 162 255 265 251 226  74 134 216 192  29 233
 179 142 242 246 127 221 298  43 248 292 112 234  23  83 198  11 201  55
 138 214   1  96  25 144  18 232   9  98 266  37 281 239  20   2 212 123
 262 172 280 291 173  10  32 137 174 164 223  75  35 121  62  80   6 254
 275 202  15 160  84  76 100  22 188  46 169  99 299 228 290 190  81 297
 115 135 113 258 132 278  89  33 143   7 114 270  94 286  47 204 145 189
 136 163 126  69  77 166 139 253  67 279 244  42 186]
15

[29.  5.  9.  4.  6.  4.  3.  1.  3.  4.  4.  6.  2.  1.  2.  3.  1.  2.
  1.  0.  1.  0.  2.  2.  2.  3.  2.  0.  1.  1.  4.  6.  1.  4.  0.  1.
  1.  1.  1.  0.  4.  1.  1.  1.  3.  2.  2.  2.  0.  2.  0.  2.  2.  2.
  0.  1.  1.  1.  0.  1.  1.  2.  5.  1.  0.  0.  2.  1.  0.  1.  3.  0.
  1.  4.  2.  1.  0.  1.  2.  1.  1.  1.  1.  1.  1.  2.  1.  0.  2.  2.
  1.  2.  0.  1.  1.  2.  0.  1.  1.  1.  2.  2.  3.  1.  0.  0.  0.  4.
  2.  1.  1.  1.  0.  1.  0.  0.  0.  0.  3.  1.  1.  0.  0.  0.  0.  2.
  0.  1.  1.  0.  0.  2.  0.  1.  0.  1.  0.  2.  1.  0.  0.  1.  0.  1.
  2.  1.  0.  2.  0.  0.  0.  0.  1.  0.  0.  1.  0.  1.  0.  0.  2.  3.
  1.  0.  0.  2.  1.  0.  0.  2.  1.  1.  1.  0.  0.  1.  1.  1.  0.  1.
  1.  0.  0.  0.  0.  1.  0.  1.  0.  0.  1.  1.  2.  0.  1.  0.  1.  2.
  0.  0.  0.  0.  1.  3.  0.  2.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  1.  1.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.
  1.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.
  1.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  1.
  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.]
```

In [26]:

```
cmc = np.zeros(301)
for i in range(301):
    cmc[i] = np.sum(ranked_histogram[:(i + 1)])

print(cmc)
```

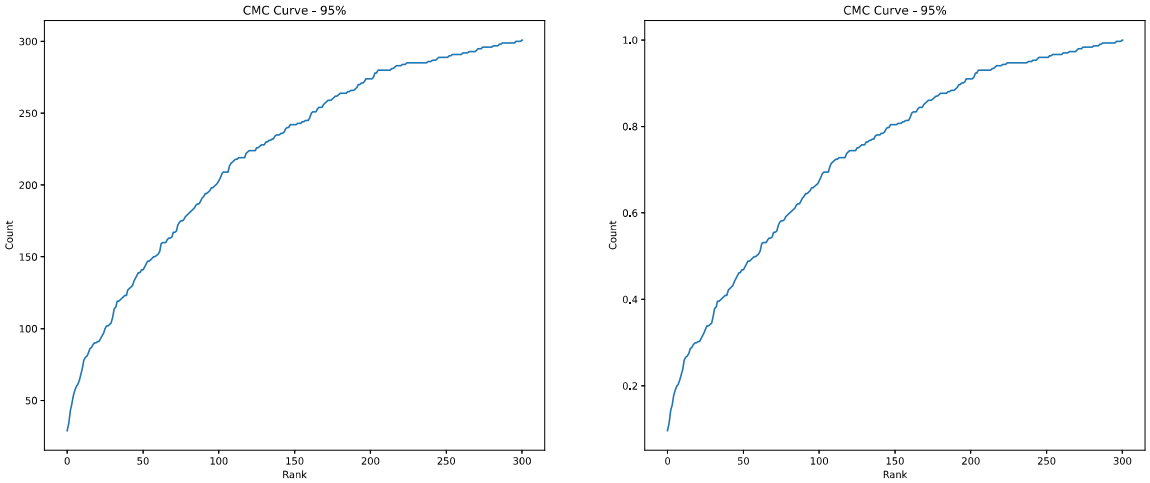
```
[ 29.  34.  43.  47.  53.  57.  60.  61.  64.  68.  72.  78.  80.  81.
  83.  86.  87.  89.  90.  90.  91.  91.  93.  95.  97. 100. 102. 102.
 103. 104. 108. 114. 115. 119. 119. 120. 121. 122. 123. 123. 127. 128.
 129. 130. 133. 135. 137. 139. 139. 141. 141. 143. 145. 147. 147. 148.
 149. 150. 150. 151. 152. 154. 159. 160. 160. 160. 162. 163. 163. 164.
 167. 167. 168. 172. 174. 175. 175. 176. 178. 179. 180. 181. 182. 183.
 184. 186. 187. 187. 189. 191. 192. 194. 194. 195. 196. 198. 198. 199.
 200. 201. 203. 205. 208. 209. 209. 209. 209. 213. 215. 216. 217. 218.
 218. 219. 219. 219. 219. 219. 222. 223. 224. 224. 224. 224. 224. 226.
 226. 227. 228. 228. 228. 230. 230. 231. 231. 232. 232. 234. 235. 235.
 235. 236. 236. 237. 239. 240. 240. 242. 242. 242. 242. 242. 243. 243.
 243. 244. 244. 245. 245. 245. 247. 250. 251. 251. 251. 253. 254. 254.
 254. 256. 257. 258. 259. 259. 259. 260. 261. 262. 262. 263. 264. 264.
 264. 264. 264. 265. 265. 266. 266. 266. 267. 268. 270. 270. 271. 271.
 272. 274. 274. 274. 274. 274. 275. 278. 278. 280. 280. 280. 280. 280.
 280. 280. 280. 280. 281. 281. 282. 283. 283. 283. 283. 284. 284. 284.
 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285.
 286. 286. 286. 287. 287. 287. 288. 289. 289. 289. 289. 289. 289. 289.
 290. 290. 291. 291. 291. 291. 291. 291. 291. 292. 292. 292. 292. 293.
 293. 293. 293. 293. 294. 295. 295. 295. 296. 296. 296. 296. 296. 296.
 296. 297. 297. 297. 297. 298. 298. 299. 299. 299. 299. 299. 299. 299.
 299. 299. 300. 300. 300. 300. 301.]
```

In [27]:

```
fig = plt.figure(figsize=[20, 8])
ax = fig.add_subplot(1, 2, 1)
ax.plot(cmc)
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 95%')
ax = fig.add_subplot(1, 2, 2)
ax.plot(cmc/len(test_gnd))
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 95%')
```

Out[27]:

```
Text(0.5, 1.0, 'CMC Curve - 95%')
```



CMC for PCA - 99

In [28]:

```
transformed_test = pca.transform(d2_test_dataset)
transformed_test_99 = transformed_test[:, 0:top99]

transformed_probe = pca.transform(d2_probe_dataset)
transformed_probe_99 = transformed_probe[:, 0:top99]
```

In [29]:

```
# storage for ranked histogram
ranked_histogram = np.zeros(len(test_gnd))

# loop over all IDs in the probe set
for i in range(len(ID_2)):
    # get the true ID of this sample
    true_ID = ID_2[i]
    print('Searching for ID %d' % (true_ID))

    # get the distance between the current probe and the whole gallery, L1 distance her
e. Note that L1
    # may not always be the best choice, so consider your distance metric given your pr
oblem
    dist = np.sqrt(np.sum(((Feat_2[:] - Feat_1[i])**2), axis=1)) #This one needs to cha
nge - Too large?
    print('Distance between source feature and all target features:')
    print(dist)

    # get the sorted order of the distances
    a = np.argsort(dist)
    # apply the order to the gallery IDs, such that the first ID in the list is the clo
sest, the second
    # ID is the second closest, and so on
    ranked = ID_1[a]
    print('Ranked IDs for query:')
    print(a)

    # find the location of the True Match in the ranked list
    ranked_result = np.where(ranked == true_ID)[0][0]
    print(ranked_result)

    # store the ranking result in the histogram
    ranked_histogram[ranked_result] += 1
    print('')

print(ranked_histogram)
```

```
009952 6114.93845201
 6266.9726034  6878.03048626 5789.24522253 8712.48237164 6210.23814123
 6220.2664637  6507.41174164 4770.2778351  7267.97085443 5522.22533897
 5702.19139662 4889.42695119 6642.17449823 4936.06796975 5300.63874506
 5355.49511911 6589.41344298 5854.43812857 5427.66251618 5585.26508857
 8177.25032289 5415.62065228 6188.04224032 6601.65500431 5707.74718017
 6093.11256623 4651.43321237 6561.3571771  6558.41829815 8420.91705063
 6107.52400439 6972.88941827 6471.6148855  5657.88268721 5213.11514206
 6343.68468744 7265.22050182 6662.37638822 5660.13185779 4827.90542267
 6472.73494776 4909.46663359 6441.98589465 6016.63402064 4930.98465906
 4953.82593793 5747.98961603 5826.82492682 5615.4109062  5246.64424561
 5487.32739373]
Ranked IDs for query:
[116 105 171  19 147   3  73 199 245 117 238 276 128 187 180 120 235 161
 119 129 257 151  30 241  91 289  12  82 236 261  95 158 291 176 294 263
  79 295 153  38 208 227 149 111  45  34  48 122  17 103 178  72 181  31
  28 198 209 131  88 168 102 284 154  60 299 127  27 225 148  93 240 205
 264  86 172 196 215 200 243  11  92 265   4  74 101 195  98 242 206 271
 268  53  66  54 300  56 191 183  50 110   0 184 259 140 118 231 222 192
 232  36 269 207 170 298  90 146  70  26 283 288  41 224  87 229 260 274
 237  71  40 137 296  14 218 252 177 175  85 297 150 211  39  55 267   5
 234  59 142  65 230 210 115   2 155  49   8  61 124 293  29 201  52 109
  44 275 159 280 249  23 247  58  68  57  16  63 213 272 254 255  24  21
 141 162 250 104  64  25 239 285  80 194  13 193  75 248  97 106  37 292
 246 160  43 282 290 125 133 203 202  20 256 219 157 156  78 278 277 134
 130  46 266 273 185  32 262 287 221 226 233 144 165  10 166  18  67 204
 152  51 217 197   6 251  35 107 212 223 121 214 281  83 182  84 138 108
  77  96   7 164 190 112 167  33 173 216 286 258 220 179 132  62 169  81
 123  89  22 145 100   9 113 188 163 126   1 174 189  94  99  15 186 143
 228  76 139 135  47 114 270 244  42 136  69 279 253]
137

Searching for ID 1498
Distance between source feature and all target features:
[4769.11920122 6383.03225568 7542.55162014 5461.59939271 4758.93015124
 4525.23800913 5429.73644537 6510.98174194 4878.44176079 5259.67694557
 5522.36831873 7195.01645956 7221.75318468 5505.74935224 5141.7682297
 6212.44689428 5760.98491324 4830.17434072 4449.34857922 6129.68067495
 4554.68470074 4752.6519907  5252.52144723 6137.64790497 4919.25614282
 4737.05308449 6648.96476012 5441.64869411 6912.1853546  5845.60275122
 4998.75634617 6128.17915525 6173.76290707 7098.36144415 5594.474193
 5978.71616344 6439.51317675 7659.76932264 5571.13842284 6173.66602713
 5005.26512689 5090.7520303  6892.41181706 5321.44129064 4756.96836201
 4871.76853047 5280.06595128 7329.54328752 4403.06911347 4367.92722797
 4915.81966796 6341.47163142 5353.26084252 4609.30605631 6156.42346433
 6300.30687496 7022.86511363 4671.86719175 8117.02861366 4967.20962681
 6098.33514561 4883.65994633 5231.3058021  4435.04478967 4429.63659408
 5918.8881855  5002.85382662 7666.12638023 5218.08508425 6175.26821861
 4615.96688006 4888.39406826 5709.67640702 5410.80759363 6289.44542563
 6775.9108819  6714.00213646 6002.6958951  4371.33467376 5873.1655486
 7826.90824666 4982.82743491 5251.93467675 6020.97100476 6105.68249339
 6825.75044207 4276.87460343 4382.8669824  6501.0416463  8249.92558241
 4768.58457039 4806.87799223 4847.93815516 4823.14532494 6727.65173982
 4332.94898755 5233.67214818 4786.48593489 5943.58938505 7465.03070636
 6120.99819815 4953.3938561  5269.03053616 5454.63492838 6254.96249167
 5906.4790874  3856.3973388  4686.79211542 4890.74766188 4719.92691786
 6062.12875865 7126.74511215 5966.66758378 5878.54783306 9283.97403929
 7263.61860526 5639.94121107 5495.46202476 5046.01155724 6169.39199707
 7488.08698069 5032.87917646 4368.95942544 6524.6742911  4392.69767207
 5140.64132749 6359.82402363 6014.03606129 6357.08006148 5003.7024815
 6053.6861966  5388.331544   6391.0800076  5260.81015074 7031.19759127
```

```
 7127.71491923 5867.01956998 6934.43847931 5858.87506503 7438.84907527
 5456.42774007 7246.58355343 4983.83852535 6066.37115359 5713.55784367
 9101.5295083  4495.48468754 5758.55260959 5591.87870131 6737.53493052
 5413.72074548 5963.10972729 5326.16150445 6533.17351584 7261.53747831
 5556.32716403 5968.39640461 6364.09748926 5109.59815883 7510.1963714
 7179.74485527 6525.6015072  5122.26197022 8352.34964643 6920.08289239
 7057.54888164 7373.00226489 5721.45966164 5417.36439416 4832.95386956
 5987.01543958 5484.78622812 7279.39314785 5383.59718935 6483.09860764
 4368.83059221 5462.60703452 5201.8454082  5715.50339692 5747.31795136
 4709.96307102 5683.87494544 5075.40090045 6462.30075999 6253.58098806
 5778.67849951 7821.0445502  5670.48202318 5071.78751246 6666.5307645
 7449.39996534 5250.7076178  5799.37758295 4389.5619478  5434.39962631
 4484.42631591 5259.35368695 6363.19749801 7731.49785904 5730.55076307
 5230.37903837 5518.08891908 5881.92180249 4298.15530109 6218.64804572
 5922.85104972 4919.33988228 4990.70755094 5548.11732313 4771.24938666
 4491.71591924 5647.73284118 5199.76105948 6213.79172612 4925.57270313
 4682.02947152 5627.41712592 4501.75918007 4703.60955814 5398.47561943
 6527.26204644 4912.16199978 6567.22332344 4345.5005912  5601.00556195
 4922.55357284 6033.71318841 5131.65260149 5748.06282082 4414.1506663
 5574.81531146 4656.11615632 8026.14915795 5040.59493778 5712.84352546
 5006.17903968 5765.23575127 5142.81954002 5477.65416907 4668.59803571
 4500.4027123  5072.87927679 6150.39593792 4203.15977983 7470.40016155
 5897.60720012 4938.79921156 3805.44080935 3826.96957468 4642.66738834
 7338.03840216 5789.62132692 4788.78839039 6753.3230159  6922.06803059
 4158.13125729 4878.0251065  4762.45884721 6772.43195684 6077.43528197
 5141.54885976 8139.80659042 5712.91321875 5033.61723766 4468.47849419
 5846.96798831 5562.91410316 5634.46049593 4409.2431651  6006.64908334
 6512.96608824 4340.7631233  5154.18623138 6312.84480328 5149.60689762
 6468.75776234 4218.28554511 5602.47709996 4467.45891035 6577.90719931
 7293.34464693 4891.87314128 4940.30487648 6841.79222534 6778.2882922
 5396.44244158 5393.96528526 4768.29473962 5013.1917369  5072.597718
 7542.52247045 7113.34178441 4581.50912756 4917.91791274 4499.1872816
 6485.69469804 5514.01829803 7565.67912552 4714.17243089 6703.06216482
 4991.678484  ]
Ranked IDs for query:
[247 248 106 255 243 276  86 203  95 271 223  49 175 122  78  87 193 124
  48 268 229  64  63  18 278 264 195 210 146 294 240 217   5  20 292  53
  70 249 231 239  57 215 107 218 180 298 109  25  21  44   4 257 287  90
   0 209  97 252  91  93  17 169  92  45 256   8  61  71 108 281 221  50
 293  24 206 225 214 246 282 101  59  81 142 207 300  30  66 129  40 235
 288 121 263 233 118 188 289 241 182  41 158 162 227 125 260  14 237 274
 272 212 177  68 200  62  96 191  82  22 196   9 133 102  46  43 152  52
 173 131 286 285 219  73 150 168   6 194  27 103 140   3 176 238 171 117
  13 296 201  10 208 155 266  38 230 148  34 224 277 216 267 116 211 187
 181  72 234 262 144 178 167 199 179 228 147  16 236 185 251 192  29 265
 138 136  79 113 202 245 105  65 205  98 151 112 156  35 170  77 269 127
  83 226 130 110 143 259  60  84 100  31  19  23 242  54 119  39  32  69
  15 213 204 184 104  74  55 273  51 128 126 197 157   1 132  36 183 275
 174 295  88   7 270 123 161 220 153 222 279  26 189 299  76  94 149 253
 258  75 284  85 283  42  28 164 254 137  56 134 165  33 291 111 135 160
  11  12 141 154 115 172 280  47 250 166 139 190  99 244 120 159 290   2
 297  37  67 198 186  80 232  58 261  89 163 145 114]
45

Searching for ID 1499
Distance between source feature and all target features:
[4108.16410418 5911.15536052 6588.95472756 4437.79964932 4398.19558477
 5015.29888309 5308.17057637 5255.31578711 4737.16792309 5348.18643421
 5347.82543269 6171.97273503 6046.01142056 5127.62456974 5446.16244525
 6133.20463708 5185.87303461 4474.10434416 5289.20318488 5602.47832398
 4787.51852049 4904.73861536 5003.84250975 6905.70401047 4953.26371972
```

```
 5201.64796192 5006.75952073 5401.6738461  6534.32186368 5377.96389735
 4656.89465022 4911.22297281 5387.55546948 5376.77784553 4393.11111696
 4945.36689831 5279.19981884 6848.70598795 4947.75595743 4666.48650384
 4021.8138791  4969.98675473 6203.63468554 6086.30226017 5066.21195315
 5114.80623664 5460.57199638 6319.1300203  4550.53070132 5729.1629992
 5330.20809774 6519.52291546 5560.45327176 5522.47965752 5911.5063926
 5006.09356479 6676.42266763 5280.04374488 7571.14184489 6074.96135603
 5882.36516399 5344.55200912 5082.44913844 5779.33354448 5377.31400593
 6452.81178969 4408.42073427 6461.49512891 5766.2180786  5752.23509379
 4879.60917357 5568.81063984 5269.70167952 4782.33395435 5033.73069091
 5409.30201902 6065.22720692 5508.4873737  4801.12286217 5388.77779546
 6297.56890944 4988.65732989 5395.77129941 6099.89759815 5789.33234301
 5021.31843058 4192.66962287 4768.56001918 5600.13467913 6686.28367628
 5166.85489858 4468.11630436 4806.25139447 5119.69195425 5875.8718801
 4170.14718009 4869.7481447  4397.04119605 5846.18267134 6720.97973217
 5503.78090658 5724.88040927 5043.74985135 5288.65058725 6242.53156306
 5166.78940035 4455.62933986 5172.11763295 5545.46444113 5222.36601704
 5888.09159376 6154.19843623 5287.4022151  5730.31391426 7676.44208045
 6249.45722264 4810.23692748 4773.29098995 5812.80530925 5444.13930433
 6330.09189467 5194.50289133 4440.07661143 6371.33517462 4637.96754429
 4761.87118952 5439.41699828 6018.96767336 5577.50674328 4236.99994485
 6576.51147158 5000.14093976 5706.58876672 6110.63031148 5476.0958187
 7263.0833007  5791.93963792 5674.84887734 5108.09715347 5957.60778458
 4179.17319139 5761.78051622 5162.48008602 6194.09350456 6162.33654347
 7384.74974366 5083.04054941 4962.94202932 5170.20121132 6558.3513186
 5287.51005231 5538.36394109 5927.89044333 5514.96748061 6808.09038217
 4920.18026323 5454.56842393 5774.7372006  4992.52495217 5804.84483528
 6556.39007378 5405.70007164 5756.64086486 6600.92644031 5352.08666973
 6766.15814166 5769.05137498 5349.54890603 5015.89023111 5329.49586061
 5312.9717157  5096.14118614 6034.79276374 4955.40878744 6031.04628326
 5492.38103831 4421.44022406 5107.50627882 5308.97443418 5910.11741887
 4490.880087   4751.28927681 6060.41163971 5181.78798607 5765.06955709
 5021.9742683  8425.92130724 5403.01161569 5479.90628634 5356.77515953
 6073.80868505 4891.98039523 5134.07762324 4561.69552702 4792.94954996
 4436.59379023 5841.49153429 5682.51559107 6192.57415533 5109.2100271
 4135.10477759 5110.80292932 4917.96014791 4882.43108745 5377.73284515
 4972.3748959  5030.01475144 4371.76677221 5491.9925511  4895.18382976
 4893.01776077 5757.52807015 6223.09129595 5372.08055486 5738.93318274
 4511.00369704 5545.18253646 5299.00632493 4828.46999099 5594.70833879
 5628.51491104 4481.41385631 4695.66442343 5111.31275204 5662.20981021
 5142.62708777 4929.56300494 4924.67525652 5425.9043373  4490.38058173
 5565.79437174 4555.63610284 7273.92205684 5897.26382429 5402.35588249
 3952.39603324 5352.7809687  4854.18477353 5088.20841951 5018.60151664
 4958.55178938 5168.91519434 5815.43771182 4356.17362785 6200.88211458
 4730.73232114 5438.10708055 5529.3289125  5289.14454882 5367.66370411
 7320.99982638 5891.75528143 5494.08688647 6492.94660778 6115.15559164
 4518.53611612 4770.24636129 4521.2455312  5561.81564759 5431.33872366
 5851.12653529 6960.98211221 5708.982936   5380.34043209 5232.52509684
 5553.91613062 4703.2298637  6183.36975967 3722.33823568 5073.2336761
 6222.25295203 4804.84889788 5381.66585157 5814.77261616 4685.42502306
 5734.16519708 4267.73810762 4669.51127884 5042.61769553 5917.82635572
 5610.18033802 5327.94493298 6000.29771556 6207.95407515 7221.05709761
 4795.01117647 5310.19461473 4802.41731467 5719.12116891 4702.55710568
 6033.26856609 5774.57867243 5588.76234613 4952.93336998 5165.81266269
 6288.3679527  5861.79640957 5900.20144359 5154.35812719 5964.82743412
 4566.86094325]
Ranked IDs for query:
[268 235  40   0 200  95 140  86 129 276 243 207  34  97   4  66 176 195
   3 122 106  91  17 221 229 180 215 255 257  48 231 193 300 124  30  39
 277 274 222 289 266 245   8 181 125  87 256 117  73  20 194 285  78 287
 271  92 116 218 237  96  70 203 191 210 209  21  31 202 155 227 226  35
```

```
  38 293  24 173 240 147  41 205  81 158 131  22  55  26   5 168 239  85
 185 206  74 278 102  44 269  62 146 238 171 177 138 199 201 223  45  93
  13 192 225 298 142 294 105  90 241 148 107 183  16 121  25 109 264   7
  72  36  57 112 150 103 248  18 217   6 178 286 170 281 169  50  61  10
   9 167 164 236 189 249 213  33  64 204  29 263 272  32  79  82  27 234
 187 161  75 228 259 246 126 119  14 156  46 134 188 208 175 252 100  77
 153  53 247 151 216 108 265  52 258 230  71 128 292 219  88  19 280 220
 224 137 197 132 262 288 101  49 113 275 214  69 162 211 141 184  68 166
 291 157  63  84 136 159 118 273 242 196  98 260 296  94  60 110 251 233
 297 179   1  54 279 152 139 299 282 127 174 290 172  12 182  76 190  59
  43  83 133 254  15 111 144  11 267 198 143 244  42 283 270 212 104 115
 295  80  47 120 123  65  67 253  51  28 160 149 130   2 163  56  89  99
 165 154  37  23 261 284 135 232 250 145  58 114 186]
241

Searching for ID 1500
Distance between source feature and all target features:
[4983.21840189 6065.64286489 6180.20255619 3948.77364065 4641.25489815
 5063.00246578 6344.70300796 6797.53749228 5151.0335208  6123.54155244
 6279.60678962 5984.11325915 4883.08756573 5120.07594596 5289.95123869
 6383.93079907 5478.75713817 5273.44787706 6108.94836956 4559.35778475
 6169.37181362 5261.25536175 6432.83783848 5905.93417918 5451.04454653
 6084.75882089 5702.47644029 5379.43573346 5115.32912    5769.94879121
 4457.76030458 4202.12499442 6282.89025037 6773.80331112 4008.61001221
 6310.60625955 5649.42203408 6151.95806354 4598.55286983 5502.80863489
 4442.98163368 4904.65398985 7775.64325771 5852.8939449  5168.66363438
 4932.77503512 6460.7554949  6919.72857121 4464.91106808 5505.45407356
 4792.04231809 5197.30477439 5085.46732205 4408.60942303 5561.48629127
 6009.76703065 5441.7998477  5064.84609109 5680.24027161 5319.39334376
 4633.14249138 4910.89593698 6332.86806389 5282.74979802 5466.15511961
 4763.507063   5539.99650035 7418.19151969 4860.29164468 7322.75590779
 4924.03120784 4824.87591289 4479.66921427 5571.77387232 5749.61855646
 6296.88486799 6413.00326464 7343.52047498 5319.47146437 4807.48771885
 6338.14493297 6564.91294956 3960.15067061 5928.12909253 6403.11027927
 5660.23761753 5107.62323411 5160.73205975 4213.79884263 6737.50791232
 5014.95907641 4367.57764845 4676.84452031 4703.17606235 6864.87015104
 4829.70513097 6073.9667105  5312.93669716 6135.89903883 6487.92084892
 6422.92198767 4996.91435326 3693.01962046 4389.77595003 4543.51530314
 4269.31210518 5281.11296053 5551.71514062 5645.20016002 5216.18304106
 5376.67712531 5205.4233334  5901.45043052 6659.1026677  6814.9442576
 6608.43297767 4857.49285549 3954.07838886 4909.01344974 4775.9113004
 4735.66794036 6320.97700282 5063.22741926 6185.43243536 5135.06608221
 5345.05047953 7238.81320228 5836.82169788 4641.3225395  4013.66558839
 5176.4281785  4397.6365464  6669.58466148 5002.62310734 5751.49382652
 6610.42687401 7177.11305742 6283.90714284 6032.88632298 7402.51040828
 5506.43545041 5203.64464745 5792.12346495 6782.94620182 6095.09922783
 7075.53485266 5007.23811036 4875.26142904 4004.9274539  4947.64563511
 5157.13072233 4375.56465798 5620.61328565 4988.39964801 5103.5395028
 5494.34204754 4914.81376937 4748.84283566 4773.30040593 5603.45495822
 6401.00230939 5451.99071817 5704.6065425  7185.27366489 6288.87144988
 5677.10223039 7383.81998631 5686.50736072 4051.80808967 6468.5226368
 5023.34829801 4714.44723927 6239.43504815 6250.85446843 6286.37307638
 5056.18816269 4356.10082061 5621.76599639 5672.5256352  5791.52785721
 4213.0987223  5703.25065577 5357.41693641 4881.17771203 4512.53733343
 4859.59140246 7831.49163669 4871.74320706 6433.34817397 7114.4685455
 6544.93747574 4821.03509265 5768.87864134 5302.72793431 5116.74106846
 5423.90074521 5540.9397105  5676.78693348 5955.34729139 4230.37851067
 5209.76501258 5992.48812971 6382.54305048 5175.96508771 7010.82079373
 4841.33256866 4561.29010942 5216.155227   4054.02477657 4298.90878269
 4949.57827115 5168.51238554 6181.0240047  5482.49253199 6054.783188
 5101.85537553 5757.52583707 5654.7845469  4583.66954504 4293.71203129
```

```
 5578.75828661 5850.82560484 5535.71764188 6294.02205561 4044.56344236
 4736.43770061 5745.33549531 3844.47908882 6538.15212    4473.60738574
 4675.16335944 4872.06345788 6122.07249763 5783.98170595 5903.0093815
 4228.83290766 5595.38019378 5476.54749097 4228.8909983  6162.34213118
 4560.36990361 4619.46342232 5824.91558791 4098.65469662 7708.22292545
 4050.21019562 5828.98335951 5167.9784617  5868.41727007 5353.87522765
 5030.94032321 5744.59572863 4844.82050782 7416.38676131 6357.86521246
 5714.71283651 5702.8016211  4688.04010517 6660.57166577 3732.98315374
 5141.8782879  5601.06399802 6199.91574262 4088.71266782 4427.89780879
 5730.77715688 6148.17111163 4644.83505308 5133.48817932 5071.01653245
 6836.38528015 5416.07228908 4579.50318992 4941.82567193 4573.19811181
 6361.057206   4258.95450172 5514.32885354 6678.85051413 7458.76662998
 6241.1711345  6153.97992918 5672.01461986 5214.75103834 5425.1154187
 4975.23269817 6903.6058645  5503.88834342 4495.49498639 5241.79106341
 6539.30530883 6241.82881018 5886.10591284 5375.91763261 4912.44262002
 5193.31916097 4944.34001533 6586.64702927 5852.36501183 6507.78569065
 4191.12405296]
Ranked IDs for query:
[102 259 227   3 117  82 148  34 129 224 245 168 208 263 243 300  31 180
  88 235 238 199 276 105 219 209 176  91 151 103 131  53 264  40  30  48
 229  72 288 184 104  19 240 206 274 272 218  38 241  60   4 128 267 230
  92 257  93 171 120 225 157  65 158 119  50  79 191  71  95 205 252 116
 185  68 187 231 147 183  12  41 118  61 294 156  70  45 273 296 149 210
 285   0 153 101 133 146  90 170 250 175   5 122  57 269  52 215 154  86
  28 194  13 268 124 260   8 150  87 247 211  44 203 130 295  51 141 111
 200 283 207 109 289  21  17 106  63  14 193  97  59  78 125 249 182 293
 110  27 271 195 284  56  24 161  64 237  16 213 155  39 287  49 140 277
 222  66 196 107  54  73 220 236 261 159 152 177 108  36 217  85 282 178
 197 165  58 167  26 256 181 162 255 265 251 226  74 134 216 192  29 233
 179 142 242 246 127 221 298  43 248 292 112 234  23  83 198  11 201  55
 138 214   1  96  25 144  18 232   9  98 266  37 281 239  20   2 212 123
 262 172 280 291 173  10  32 137 174 164 223  75  35 121  62  80   6 254
 275 202  15 160  84  76 100  22 188  46 169  99 299 228 290 190  81 297
 115 135 113 258 132 278  89  33 143   7 114 270  94 286  47 204 145 189
 136 163 126  69  77 166 139 253  67 279 244  42 186]
15

[29.  5.  9.  4.  6.  4.  3.  1.  3.  4.  4.  6.  2.  1.  2.  3.  1.  2.
  1.  0.  1.  0.  2.  2.  2.  3.  2.  0.  1.  1.  4.  6.  1.  4.  0.  1.
  1.  1.  1.  0.  4.  1.  1.  1.  3.  2.  2.  2.  0.  2.  0.  2.  2.  2.
  0.  1.  1.  1.  0.  1.  1.  2.  5.  1.  0.  0.  2.  1.  0.  1.  3.  0.
  1.  4.  2.  1.  0.  1.  2.  1.  1.  1.  1.  1.  1.  2.  1.  0.  2.  2.
  1.  2.  0.  1.  1.  2.  0.  1.  1.  1.  2.  2.  3.  1.  0.  0.  0.  4.
  2.  1.  1.  1.  0.  1.  0.  0.  0.  0.  3.  1.  1.  0.  0.  0.  0.  2.
  0.  1.  1.  0.  0.  2.  0.  1.  0.  1.  0.  2.  1.  0.  0.  1.  0.  1.
  2.  1.  0.  2.  0.  0.  0.  0.  1.  0.  0.  1.  0.  1.  0.  0.  2.  3.
  1.  0.  0.  2.  1.  0.  0.  2.  1.  1.  1.  0.  0.  1.  1.  1.  0.  1.
  1.  0.  0.  0.  0.  1.  0.  1.  0.  0.  1.  1.  2.  0.  1.  0.  1.  2.
  0.  0.  0.  0.  1.  3.  0.  2.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  1.  1.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.
  1.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.
  1.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  1.
  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.]
```

In [30]:

```
cmc = np.zeros(301)
for i in range(301):
    cmc[i] = np.sum(ranked_histogram[:(i + 1)])

print(cmc)
```

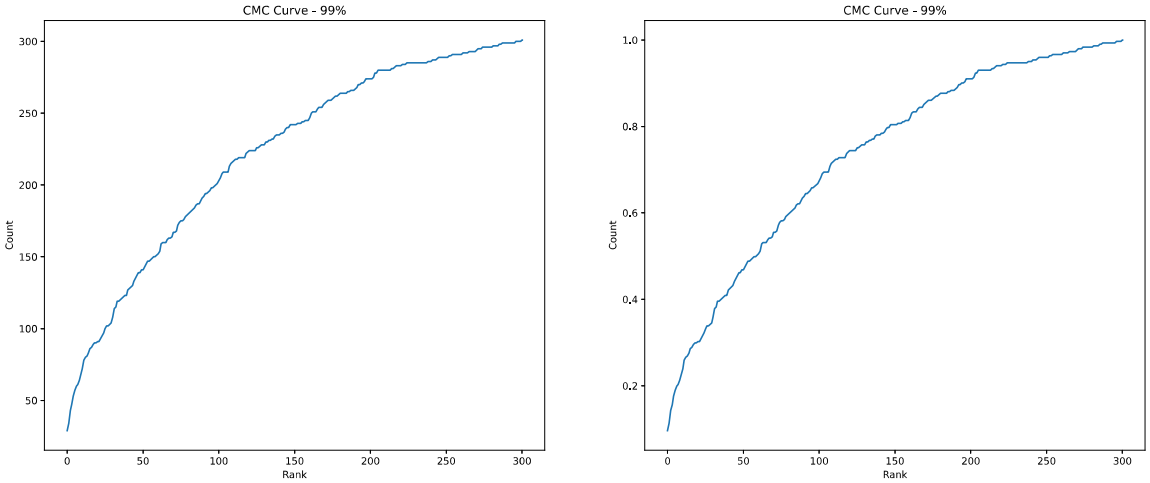
```
[ 29.  34.  43.  47.  53.  57.  60.  61.  64.  68.  72.  78.  80.  81.
  83.  86.  87.  89.  90.  90.  91.  91.  93.  95.  97. 100. 102. 102.
 103. 104. 108. 114. 115. 119. 119. 120. 121. 122. 123. 123. 127. 128.
 129. 130. 133. 135. 137. 139. 139. 141. 141. 143. 145. 147. 147. 148.
 149. 150. 150. 151. 152. 154. 159. 160. 160. 160. 162. 163. 163. 164.
 167. 167. 168. 172. 174. 175. 175. 176. 178. 179. 180. 181. 182. 183.
 184. 186. 187. 187. 189. 191. 192. 194. 194. 195. 196. 198. 198. 199.
 200. 201. 203. 205. 208. 209. 209. 209. 209. 213. 215. 216. 217. 218.
 218. 219. 219. 219. 219. 219. 222. 223. 224. 224. 224. 224. 224. 226.
 226. 227. 228. 228. 228. 230. 230. 231. 231. 232. 232. 234. 235. 235.
 235. 236. 236. 237. 239. 240. 240. 242. 242. 242. 242. 242. 243. 243.
 243. 244. 244. 245. 245. 245. 247. 250. 251. 251. 251. 253. 254. 254.
 254. 256. 257. 258. 259. 259. 259. 260. 261. 262. 262. 263. 264. 264.
 264. 264. 264. 265. 265. 266. 266. 266. 267. 268. 270. 270. 271. 271.
 272. 274. 274. 274. 274. 274. 275. 278. 278. 280. 280. 280. 280. 280.
 280. 280. 280. 280. 281. 281. 282. 283. 283. 283. 283. 284. 284. 284.
 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285. 285.
 286. 286. 286. 287. 287. 287. 288. 289. 289. 289. 289. 289. 289. 289.
 290. 290. 291. 291. 291. 291. 291. 291. 291. 292. 292. 292. 292. 293.
 293. 293. 293. 293. 294. 295. 295. 295. 296. 296. 296. 296. 296. 296.
 296. 297. 297. 297. 297. 298. 298. 299. 299. 299. 299. 299. 299. 299.
 299. 299. 300. 300. 300. 300. 301.]
```

In [31]:

```
fig = plt.figure(figsize=[20, 8])
ax = fig.add_subplot(1, 2, 1)
ax.plot(cmc)
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 99%')
ax = fig.add_subplot(1, 2, 2)
ax.plot(cmc/len(test_gnd))
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve - 99%')
```

Out[31]:

```
Text(0.5, 1.0, 'CMC Curve - 99%')
```



Deep Learning Model - Siamese

Week 7 example 1

In [32]:

```
def GetSiameseData(imgs, labels, batch_size):

    image_a = np.zeros((batch_size, np.shape(imgs)[1], np.shape(imgs)[2], np.shape(imgs
)[3]));
    image_b = np.zeros((batch_size, np.shape(imgs)[1], np.shape(imgs)[2], np.shape(imgs
)[3]));
    label = np.zeros(batch_size);

    for i in range(batch_size):

        if (i % 2 == 0):
            idx1 = random.randint(0, len(imgs) - 1)
            idx2 = random.randint(0, len(imgs) - 1)
            l = 1
            while (labels[idx1] != labels[idx2]):
                idx2 = random.randint(0, len(imgs) - 1)

        else:
            idx1 = random.randint(0, len(imgs) - 1)
            idx2 = random.randint(0, len(imgs) - 1)
            l = 0
            while (labels[idx1] == labels[idx2]):
                idx2 = random.randint(0, len(imgs) - 1)

        image_a[i, :, :, :] = imgs[idx1,:,:,:]
        image_b[i, :, :, :] = imgs[idx2,:,:,:]
        label[i] = l

    return [image_a, image_b], label

def PairGenerator(imgs, labels, batch_size):
    while True:
        [image_a, image_b], label = GetSiameseData(imgs, labels, batch_size)
        yield [image_a, image_b], label
```

In [33]:

```
def conv_block(inputs, filters, spatial_dropout = 0.0, max_pool = True):

    x = layers.Conv2D(filters=filters, kernel_size=(3,3), padding='same', activation='r
elu')(inputs)
    x = layers.Conv2D(filters=filters, kernel_size=(3,3), padding='same', activation=No
ne)(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation('relu')(x)
    if (spatial_dropout > 0.0):
        x = layers.SpatialDropout2D(spatial_dropout)(x)
    if (max_pool == True):
        x = layers.MaxPool2D(pool_size=(2, 2))(x)

    return x

def fc_block(inputs, size, dropout):
    x = layers.Dense(size, activation=None)(inputs)
    x = layers.BatchNormalization()(x)
    x = layers.Activation('relu')(x)
    if (dropout > 0.0):
        x = layers.Dropout(dropout)(x)

    return x

def vgg_net(inputs, filters, fc, spatial_dropout = 0.0, dropout = 0.0):

    x = inputs
    for idx,i in enumerate(filters):
        x = conv_block(x, i, spatial_dropout, not (idx==len(filters) - 1))

    x = layers.Flatten()(x)

    for i in fc:
        x = fc_block(x, i, dropout)

    return x
```

In [34]:

```
embedding_size = 32
dummy_input = keras.Input((128, 64, 1))
base_network = vgg_net(dummy_input, [8, 16, 32], [256], 0.2, 0)
embedding_layer = layers.Dense(embedding_size, activation=None)(base_network)
base_network = keras.Model(dummy_input, embedding_layer, name='SiameseBranch')
base_network.summary()
```

```
Model: "SiameseBranch"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 64, 1)]      0
_________________________________________________________________
conv2d (Conv2D)              (None, 128, 64, 8)        80
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 64, 8)        584
_________________________________________________________________
batch_normalization (BatchNo (None, 128, 64, 8)        32
_________________________________________________________________
activation (Activation)      (None, 128, 64, 8)        0
_________________________________________________________________
spatial_dropout2d (SpatialDr (None, 128, 64, 8)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 32, 8)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 32, 16)        1168
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 64, 32, 16)        2320
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 32, 16)        64
_________________________________________________________________
activation_1 (Activation)    (None, 64, 32, 16)        0
_________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 64, 32, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 16, 16)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 16, 32)        4640
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 32, 16, 32)        9248
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 16, 32)        128
_________________________________________________________________
activation_2 (Activation)    (None, 32, 16, 32)        0
_________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 32, 16, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 16384)             0
_________________________________________________________________
dense (Dense)                (None, 256)               4194560
_________________________________________________________________
batch_normalization_3 (Batch (None, 256)               1024
_________________________________________________________________
activation_3 (Activation)    (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                8224
=================================================================
Total params: 4,222,072
Trainable params: 4,221,448
Non-trainable params: 624
_________________________________________________________________
```

In [35]:

```
input_a = keras.Input((128, 64, 1), name='InputA')
input_b = keras.Input((128, 64, 1), name='InputB')

embedding_a = base_network(input_a)
embedding_b = base_network(input_b)
```

In [36]:

```
combined = layers.concatenate([embedding_a, embedding_b])
combined = layers.Dense(128, activation='relu')(combined)
output = layers.Dense(1, activation='sigmoid')(combined)

siamese_network = keras.Model([input_a, input_b], output, name='SiameseNetwork')
siamese_network.summary()
```

```
Model: "SiameseNetwork"
__________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connect
ed to
==================================================================================
InputA (InputLayer)             [(None, 128, 64, 1)] 0
__________________________________________________________________________________
InputB (InputLayer)             [(None, 128, 64, 1)] 0
__________________________________________________________________________________
SiameseBranch (Functional)      (None, 32)           4222072     InputA
[0][0]
                                                                 InputB
[0][0]
__________________________________________________________________________________
concatenate (Concatenate)       (None, 64)           0           Siamese
Branch[0][0]
                                                                 Siamese
Branch[1][0]
__________________________________________________________________________________
dense_2 (Dense)                 (None, 128)          8320        concate
nate[0][0]
__________________________________________________________________________________
dense_3 (Dense)                 (None, 1)            129         dense_2
[0][0]
==================================================================================
Total params: 4,230,521
Trainable params: 4,229,897
Non-trainable params: 624
__________________________________________________________________________________
```

In [37]:

```
siamese_network.compile(loss='binary_crossentropy', optimizer=keras.optimizers.RMSprop
(), metrics=['accuracy'])
```

In [38]:

```
test_fea = np.reshape(test_gal.transpose(), (301, 128, 64, 1))
test_pro_fea = np.reshape(test_pro.transpose(), (301, 128, 64, 1))
train_fea = np.reshape(train_fea.transpose(), (5933, 128, 64, 1))
```

In [39]:

```
print(test_fea.shape)
```

```
(301, 128, 64, 1)
```

In [40]:

```
print(test_pro_fea.shape)
```

```
(301, 128, 64, 1)
```

In [41]:

```
print(train_fea.shape)
```

```
(5933, 128, 64, 1)
```

In [42]:

```
batch_size = 32
training_gen = PairGenerator(train_fea, gnd, batch_size)

siamese_test_x, siamese_test_y = GetSiameseData(test_fea, test_gnd, 5933)
siamese_network.fit(training_gen, steps_per_epoch = 256 // batch_size, epochs=100, vali
dation_data = (siamese_test_x, siamese_test_y))
```

```
Epoch 1/100
8/8 [==============================] - 28s 3s/step - loss: 0.7875 - accura
cy: 0.4804 - val_loss: 10.4787 - val_accuracy: 0.5001
Epoch 2/100
8/8 [==============================] - 24s 3s/step - loss: 0.7723 - accura
cy: 0.5037 - val_loss: 1.8938 - val_accuracy: 0.5001
Epoch 3/100
8/8 [==============================] - 25s 3s/step - loss: 0.7476 - accura
cy: 0.4976 - val_loss: 0.9554 - val_accuracy: 0.5001
Epoch 4/100
8/8 [==============================] - 25s 3s/step - loss: 0.7347 - accura
cy: 0.5000 - val_loss: 0.9820 - val_accuracy: 0.5011
Epoch 5/100
8/8 [==============================] - 24s 3s/step - loss: 0.7257 - accura
cy: 0.5710 - val_loss: 1.0308 - val_accuracy: 0.5001
Epoch 6/100
8/8 [==============================] - 24s 3s/step - loss: 0.7041 - accura
cy: 0.5785 - val_loss: 0.8523 - val_accuracy: 0.5001
Epoch 7/100
8/8 [==============================] - 23s 3s/step - loss: 0.6826 - accura
cy: 0.5509 - val_loss: 0.6873 - val_accuracy: 0.5077
Epoch 8/100
8/8 [==============================] - 24s 3s/step - loss: 0.7053 - accura
cy: 0.5675 - val_loss: 0.8533 - val_accuracy: 0.5009
Epoch 9/100
8/8 [==============================] - 25s 4s/step - loss: 0.6812 - accura
cy: 0.5822 - val_loss: 1.7557 - val_accuracy: 0.5001
Epoch 10/100
8/8 [==============================] - 25s 3s/step - loss: 0.7176 - accura
cy: 0.4875 - val_loss: 0.8647 - val_accuracy: 0.5024
Epoch 11/100
8/8 [==============================] - 23s 3s/step - loss: 0.6796 - accura
cy: 0.5398 - val_loss: 0.7114 - val_accuracy: 0.5781
Epoch 12/100
8/8 [==============================] - 22s 3s/step - loss: 0.7295 - accura
cy: 0.5466 - val_loss: 0.6660 - val_accuracy: 0.6794
Epoch 13/100
8/8 [==============================] - 22s 3s/step - loss: 0.7191 - accura
cy: 0.5546 - val_loss: 0.6233 - val_accuracy: 0.6867
Epoch 14/100
8/8 [==============================] - 22s 3s/step - loss: 0.7036 - accura
cy: 0.5311 - val_loss: 0.6818 - val_accuracy: 0.5077
Epoch 15/100
8/8 [==============================] - 22s 3s/step - loss: 0.6601 - accura
cy: 0.5885 - val_loss: 0.6687 - val_accuracy: 0.6000
Epoch 16/100
8/8 [==============================] - 23s 3s/step - loss: 0.6508 - accura
cy: 0.6124 - val_loss: 0.6137 - val_accuracy: 0.7420
Epoch 17/100
8/8 [==============================] - 24s 3s/step - loss: 0.7215 - accura
cy: 0.5823 - val_loss: 0.6625 - val_accuracy: 0.6349
Epoch 18/100
8/8 [==============================] - 23s 3s/step - loss: 0.7004 - accura
cy: 0.5178 - val_loss: 0.6372 - val_accuracy: 0.6460
Epoch 19/100
8/8 [==============================] - 22s 3s/step - loss: 0.6760 - accura
cy: 0.5388 - val_loss: 0.6637 - val_accuracy: 0.6434
Epoch 20/100
8/8 [==============================] - 22s 3s/step - loss: 0.7109 - accura
cy: 0.5755 - val_loss: 0.6379 - val_accuracy: 0.6973
Epoch 21/100
```

```
8/8 [==============================] - 22s 3s/step - loss: 0.6405 - accura
cy: 0.6217 - val_loss: 0.5682 - val_accuracy: 0.7472
Epoch 22/100
8/8 [==============================] - 23s 3s/step - loss: 0.7476 - accura
cy: 0.5370 - val_loss: 0.6855 - val_accuracy: 0.5485
Epoch 23/100
8/8 [==============================] - 23s 3s/step - loss: 0.6919 - accura
cy: 0.5714 - val_loss: 0.7264 - val_accuracy: 0.4900
Epoch 24/100
8/8 [==============================] - 22s 3s/step - loss: 0.6316 - accura
cy: 0.6621 - val_loss: 0.7544 - val_accuracy: 0.5078
Epoch 25/100
8/8 [==============================] - 22s 3s/step - loss: 0.6313 - accura
cy: 0.6370 - val_loss: 0.7373 - val_accuracy: 0.4974
Epoch 26/100
8/8 [==============================] - 22s 3s/step - loss: 0.6514 - accura
cy: 0.6188 - val_loss: 0.7456 - val_accuracy: 0.5100
Epoch 27/100
8/8 [==============================] - 22s 3s/step - loss: 0.6146 - accura
cy: 0.6412 - val_loss: 0.7283 - val_accuracy: 0.5129
Epoch 28/100
8/8 [==============================] - 22s 3s/step - loss: 0.6122 - accura
cy: 0.6630 - val_loss: 0.7421 - val_accuracy: 0.5001
Epoch 29/100
8/8 [==============================] - 22s 3s/step - loss: 0.5903 - accura
cy: 0.6727 - val_loss: 0.6970 - val_accuracy: 0.5016
Epoch 30/100
8/8 [==============================] - 23s 3s/step - loss: 0.6426 - accura
cy: 0.6016 - val_loss: 0.7030 - val_accuracy: 0.5001
Epoch 31/100
8/8 [==============================] - 22s 3s/step - loss: 0.6495 - accura
cy: 0.5932 - val_loss: 0.7039 - val_accuracy: 0.5304
Epoch 32/100
8/8 [==============================] - 22s 3s/step - loss: 0.6238 - accura
cy: 0.6205 - val_loss: 0.6958 - val_accuracy: 0.5001
Epoch 33/100
8/8 [==============================] - 22s 3s/step - loss: 0.6751 - accura
cy: 0.6404 - val_loss: 0.6935 - val_accuracy: 0.4986
Epoch 34/100
8/8 [==============================] - 22s 3s/step - loss: 0.6251 - accura
cy: 0.6211 - val_loss: 0.7495 - val_accuracy: 0.5001
Epoch 35/100
8/8 [==============================] - 22s 3s/step - loss: 0.6741 - accura
cy: 0.5965 - val_loss: 0.6941 - val_accuracy: 0.5001
Epoch 36/100
8/8 [==============================] - 23s 3s/step - loss: 0.6598 - accura
cy: 0.6306 - val_loss: 0.6929 - val_accuracy: 0.5774
Epoch 37/100
8/8 [==============================] - 23s 3s/step - loss: 0.6338 - accura
cy: 0.5831 - val_loss: 0.6906 - val_accuracy: 0.4252
Epoch 38/100
8/8 [==============================] - 23s 3s/step - loss: 0.6707 - accura
cy: 0.6345 - val_loss: 0.7052 - val_accuracy: 0.5001
Epoch 39/100
8/8 [==============================] - 24s 3s/step - loss: 0.6832 - accura
cy: 0.5257 - val_loss: 0.6951 - val_accuracy: 0.5001
Epoch 40/100
8/8 [==============================] - 24s 3s/step - loss: 0.6413 - accura
cy: 0.6362 - val_loss: 0.7630 - val_accuracy: 0.4999
Epoch 41/100
8/8 [==============================] - 24s 3s/step - loss: 0.6885 - accura
```

```
cy: 0.5658 - val_loss: 0.7247 - val_accuracy: 0.5129
Epoch 42/100
8/8 [==============================] - 25s 4s/step - loss: 0.6259 - accura
cy: 0.6269 - val_loss: 0.6888 - val_accuracy: 0.5314
Epoch 43/100
8/8 [==============================] - 28s 4s/step - loss: 0.6276 - accura
cy: 0.6612 - val_loss: 0.8319 - val_accuracy: 0.5001
Epoch 44/100
8/8 [==============================] - 25s 3s/step - loss: 0.6367 - accura
cy: 0.6614 - val_loss: 0.7498 - val_accuracy: 0.5001
Epoch 45/100
8/8 [==============================] - 26s 4s/step - loss: 0.6338 - accura
cy: 0.6592 - val_loss: 0.7956 - val_accuracy: 0.5001
Epoch 46/100
8/8 [==============================] - 26s 4s/step - loss: 0.5883 - accura
cy: 0.6732 - val_loss: 0.7760 - val_accuracy: 0.5001
Epoch 47/100
8/8 [==============================] - 27s 4s/step - loss: 0.6356 - accura
cy: 0.6269 - val_loss: 0.7018 - val_accuracy: 0.5026
Epoch 48/100
8/8 [==============================] - 25s 4s/step - loss: 0.6681 - accura
cy: 0.5654 - val_loss: 0.6953 - val_accuracy: 0.5336
Epoch 49/100
8/8 [==============================] - 26s 4s/step - loss: 0.6671 - accura
cy: 0.6001 - val_loss: 0.7409 - val_accuracy: 0.5001
Epoch 50/100
8/8 [==============================] - 24s 3s/step - loss: 0.5909 - accura
cy: 0.7400 - val_loss: 0.7599 - val_accuracy: 0.5001
Epoch 51/100
8/8 [==============================] - 24s 3s/step - loss: 0.6538 - accura
cy: 0.6243 - val_loss: 0.6650 - val_accuracy: 0.6074
Epoch 52/100
8/8 [==============================] - 26s 4s/step - loss: 0.6454 - accura
cy: 0.5987 - val_loss: 0.6717 - val_accuracy: 0.5879
Epoch 53/100
8/8 [==============================] - 25s 4s/step - loss: 0.6464 - accura
cy: 0.6306 - val_loss: 0.6788 - val_accuracy: 0.5427
Epoch 54/100
8/8 [==============================] - 25s 4s/step - loss: 0.6410 - accura
cy: 0.6237 - val_loss: 0.6767 - val_accuracy: 0.5500
Epoch 55/100
8/8 [==============================] - 23s 3s/step - loss: 0.6579 - accura
cy: 0.6096 - val_loss: 0.6997 - val_accuracy: 0.5350
Epoch 56/100
8/8 [==============================] - 22s 3s/step - loss: 0.6511 - accura
cy: 0.5909 - val_loss: 0.6976 - val_accuracy: 0.5186
Epoch 57/100
8/8 [==============================] - 25s 3s/step - loss: 0.6572 - accura
cy: 0.5883 - val_loss: 0.7618 - val_accuracy: 0.5001
Epoch 58/100
8/8 [==============================] - 24s 3s/step - loss: 0.6411 - accura
cy: 0.6597 - val_loss: 0.6993 - val_accuracy: 0.5001
Epoch 59/100
8/8 [==============================] - 23s 3s/step - loss: 0.6632 - accura
cy: 0.6328 - val_loss: 0.7161 - val_accuracy: 0.5001
Epoch 60/100
8/8 [==============================] - 22s 3s/step - loss: 0.6638 - accura
cy: 0.5993 - val_loss: 0.6915 - val_accuracy: 0.5142
Epoch 61/100
8/8 [==============================] - 22s 3s/step - loss: 0.6204 - accura
cy: 0.6848 - val_loss: 0.7049 - val_accuracy: 0.5011
```

```
Epoch 62/100
8/8 [==============================] - 22s 3s/step - loss: 0.6182 - accura
cy: 0.6644 - val_loss: 0.7410 - val_accuracy: 0.5001
Epoch 63/100
8/8 [==============================] - 22s 3s/step - loss: 0.6530 - accura
cy: 0.6240 - val_loss: 0.6968 - val_accuracy: 0.5609
Epoch 64/100
8/8 [==============================] - 25s 4s/step - loss: 0.6259 - accura
cy: 0.6643 - val_loss: 0.6857 - val_accuracy: 0.5362
Epoch 65/100
8/8 [==============================] - 25s 3s/step - loss: 0.6420 - accura
cy: 0.6672 - val_loss: 0.7402 - val_accuracy: 0.5001
Epoch 66/100
8/8 [==============================] - 25s 4s/step - loss: 0.6507 - accura
cy: 0.6153 - val_loss: 0.7113 - val_accuracy: 0.5004
Epoch 67/100
8/8 [==============================] - 26s 4s/step - loss: 0.6737 - accura
cy: 0.5862 - val_loss: 0.7020 - val_accuracy: 0.5024
Epoch 68/100
8/8 [==============================] - 24s 3s/step - loss: 0.6590 - accura
cy: 0.5640 - val_loss: 0.6967 - val_accuracy: 0.5094
Epoch 69/100
8/8 [==============================] - 24s 3s/step - loss: 0.6317 - accura
cy: 0.6400 - val_loss: 0.7076 - val_accuracy: 0.5055
Epoch 70/100
8/8 [==============================] - 23s 3s/step - loss: 0.6539 - accura
cy: 0.5933 - val_loss: 0.7521 - val_accuracy: 0.5001
Epoch 71/100
8/8 [==============================] - 24s 3s/step - loss: 0.6042 - accura
cy: 0.6562 - val_loss: 0.7492 - val_accuracy: 0.5001
Epoch 72/100
8/8 [==============================] - 25s 3s/step - loss: 0.6398 - accura
cy: 0.6473 - val_loss: 0.7138 - val_accuracy: 0.5001
Epoch 73/100
8/8 [==============================] - 25s 3s/step - loss: 0.6673 - accura
cy: 0.6339 - val_loss: 0.6872 - val_accuracy: 0.5139
Epoch 74/100
8/8 [==============================] - 24s 3s/step - loss: 0.6156 - accura
cy: 0.6523 - val_loss: 0.7286 - val_accuracy: 0.5001
Epoch 75/100
8/8 [==============================] - 24s 3s/step - loss: 0.6611 - accura
cy: 0.6278 - val_loss: 0.6946 - val_accuracy: 0.5147
Epoch 76/100
8/8 [==============================] - 24s 3s/step - loss: 0.6238 - accura
cy: 0.6851 - val_loss: 0.6897 - val_accuracy: 0.5373
Epoch 77/100
8/8 [==============================] - 24s 3s/step - loss: 0.6202 - accura
cy: 0.6708 - val_loss: 0.7294 - val_accuracy: 0.5019
Epoch 78/100
8/8 [==============================] - 24s 3s/step - loss: 0.6386 - accura
cy: 0.6070 - val_loss: 0.7281 - val_accuracy: 0.5048
Epoch 79/100
8/8 [==============================] - 24s 3s/step - loss: 0.6511 - accura
cy: 0.6465 - val_loss: 0.6996 - val_accuracy: 0.5014
Epoch 80/100
8/8 [==============================] - 23s 3s/step - loss: 0.6371 - accura
cy: 0.6493 - val_loss: 0.7321 - val_accuracy: 0.5001
Epoch 81/100
8/8 [==============================] - 24s 3s/step - loss: 0.6413 - accura
cy: 0.6147 - val_loss: 0.6777 - val_accuracy: 0.5110
Epoch 82/100
```

```
8/8 [==============================] - 24s 3s/step - loss: 0.6177 - accura
cy: 0.6317 - val_loss: 0.7218 - val_accuracy: 0.5001
Epoch 83/100
8/8 [==============================] - 24s 3s/step - loss: 0.6084 - accura
cy: 0.6803 - val_loss: 0.7547 - val_accuracy: 0.5001
Epoch 84/100
8/8 [==============================] - 24s 3s/step - loss: 0.6316 - accura
cy: 0.6269 - val_loss: 0.6820 - val_accuracy: 0.5513
Epoch 85/100
8/8 [==============================] - 24s 3s/step - loss: 0.6292 - accura
cy: 0.6427 - val_loss: 0.7667 - val_accuracy: 0.5001
Epoch 86/100
8/8 [==============================] - 24s 3s/step - loss: 0.5858 - accura
cy: 0.6751 - val_loss: 0.7014 - val_accuracy: 0.5035
Epoch 87/100
8/8 [==============================] - 24s 3s/step - loss: 0.6551 - accura
cy: 0.6117 - val_loss: 0.6873 - val_accuracy: 0.5345
Epoch 88/100
8/8 [==============================] - 23s 3s/step - loss: 0.5833 - accura
cy: 0.6474 - val_loss: 0.7590 - val_accuracy: 0.5001
Epoch 89/100
8/8 [==============================] - 24s 3s/step - loss: 0.6543 - accura
cy: 0.6171 - val_loss: 0.7232 - val_accuracy: 0.5006
Epoch 90/100
8/8 [==============================] - 24s 3s/step - loss: 0.6092 - accura
cy: 0.6568 - val_loss: 0.8197 - val_accuracy: 0.5001
Epoch 91/100
8/8 [==============================] - 24s 3s/step - loss: 0.6609 - accura
cy: 0.6099 - val_loss: 0.6991 - val_accuracy: 0.5036
Epoch 92/100
8/8 [==============================] - 24s 3s/step - loss: 0.6250 - accura
cy: 0.6646 - val_loss: 0.7367 - val_accuracy: 0.5004
Epoch 93/100
8/8 [==============================] - 24s 3s/step - loss: 0.6300 - accura
cy: 0.6229 - val_loss: 0.7089 - val_accuracy: 0.5458
Epoch 94/100
8/8 [==============================] - 24s 3s/step - loss: 0.6381 - accura
cy: 0.6497 - val_loss: 0.6773 - val_accuracy: 0.5626
Epoch 95/100
8/8 [==============================] - 24s 3s/step - loss: 0.6322 - accura
cy: 0.6455 - val_loss: 0.7090 - val_accuracy: 0.5326
Epoch 96/100
8/8 [==============================] - 25s 4s/step - loss: 0.6311 - accura
cy: 0.6260 - val_loss: 0.6744 - val_accuracy: 0.5840
Epoch 97/100
8/8 [==============================] - 26s 4s/step - loss: 0.6104 - accura
cy: 0.6881 - val_loss: 0.7293 - val_accuracy: 0.5233
Epoch 98/100
8/8 [==============================] - 27s 4s/step - loss: 0.6583 - accura
cy: 0.6302 - val_loss: 0.6909 - val_accuracy: 0.5788
Epoch 99/100
8/8 [==============================] - 25s 3s/step - loss: 0.6313 - accura
cy: 0.5985 - val_loss: 0.7182 - val_accuracy: 0.5029
Epoch 100/100
8/8 [==============================] - 25s 4s/step - loss: 0.5951 - accura
cy: 0.6626 - val_loss: 0.6781 - val_accuracy: 0.5581
```

Out[42]:

```
<tensorflow.python.keras.callbacks.History at 0x172c8467be0>
```

In [43]:

```
gal_x, gal_y = GetSiameseData(test_fea, test_gnd, 301)
gal_predict = siamese_network.predict(gal_x)

pro_x, pro_y = GetSiameseData(test_pro_fea, test_pro_gnd, 301)
pro_predict = siamese_network.predict(pro_x)
```

CMC for DNN

In [44]:

```
vector = np.vectorize(np.int)

ID_1 = vector(np.array(test_gnd))
ID_2 = vector(np.array(test_pro_gnd))
Feat_1 = gal_predict
Feat_2 = pro_predict
print(Feat_1)
print(Feat_2)
```

```
[[0.623222  ]
 [0.6127546 ]
 [0.63220525]
 [0.55209017]
 [0.59780556]
 [0.597301  ]
 [0.6248723 ]
 [0.61275446]
 [0.604179  ]
 [0.62517965]
 [0.59174716]
 [0.5820576 ]
 [0.6054485 ]
 [0.6319502 ]
 [0.5715554 ]
 [0.43918318]
 [0.63264275]
 [0.59732586]
 [0.61495787]
 [0.61438674]
 [0.59991854]
 [0.5608426 ]
 [0.6163433 ]
 [0.6159894 ]
 [0.6286116 ]
 [0.59386015]
 [0.6120465 ]
 [0.61091113]
 [0.61845   ]
 [0.55908406]
 [0.617306  ]
 [0.54823667]
 [0.5845285 ]
 [0.60981166]
 [0.61752266]
 [0.5181973 ]
 [0.54559916]
 [0.55629516]
 [0.56742764]
 [0.48593268]
 [0.6313373 ]
 [0.60821056]
 [0.54827666]
 [0.58517915]
 [0.54948366]
 [0.54426736]
 [0.59780556]
 [0.61306465]
 [0.5916325 ]
 [0.59015524]
 [0.606838  ]
 [0.60475814]
 [0.5916325 ]
 [0.61164683]
 [0.5769371 ]
 [0.61475545]
 [0.53155637]
 [0.4281074 ]
 [0.6131019 ]
 [0.5204571 ]
 [0.55721825]
```

```
[0.5481301 ]
[0.6056046 ]
[0.5627798 ]
[0.62383467]
[0.5580293 ]
[0.5799424 ]
[0.42279124]
[0.6248733 ]
[0.5656905 ]
[0.6118911 ]
[0.6203215 ]
[0.61030275]
[0.5433955 ]
[0.6303734 ]
[0.6194217 ]
[0.56742764]
[0.6050057 ]
[0.59541625]
[0.6122057 ]
[0.53578436]
[0.49930334]
[0.54559916]
[0.6228722 ]
[0.5993295 ]
[0.61769664]
[0.6014838 ]
[0.42306885]
[0.60493064]
[0.46490735]
[0.60948795]
[0.59526336]
[0.5927208 ]
[0.5918068 ]
[0.57957965]
[0.6189941 ]
[0.5239694 ]
[0.61380935]
[0.6185473 ]
[0.38128874]
[0.56449336]
[0.4486956 ]
[0.61495787]
[0.5840632 ]
[0.6283755 ]
[0.60805726]
[0.58553755]
[0.6102383 ]
[0.59358203]
[0.570052  ]
[0.5455698 ]
[0.6148361 ]
[0.60718864]
[0.5943391 ]
[0.5930613 ]
[0.54438907]
[0.62830305]
[0.5168786 ]
[0.635704  ]
[0.5294502 ]
[0.6014838 ]
[0.6167419 ]
```

```
[0.5974021 ]
[0.5683602 ]
[0.5710916 ]
[0.3741033 ]
[0.5981518 ]
[0.5225342 ]
[0.6171755 ]
[0.61400986]
[0.6314948 ]
[0.5986068 ]
[0.6344853 ]
[0.61605597]
[0.5785248 ]
[0.6231392 ]
[0.5696838 ]
[0.6252456 ]
[0.59780556]
[0.61417425]
[0.5991143 ]
[0.6058596 ]
[0.62160265]
[0.36116055]
[0.5996599 ]
[0.59034514]
[0.61503345]
[0.61676735]
[0.61999035]
[0.54907864]
[0.5975858 ]
[0.41252023]
[0.5455698 ]
[0.4324523 ]
[0.63400185]
[0.51700324]
[0.6209345 ]
[0.62976456]
[0.62539196]
[0.59538513]
[0.57841283]
[0.6163908 ]
[0.6147761 ]
[0.60432065]
[0.5803964 ]
[0.6198493 ]
[0.6017087 ]
[0.5870299 ]
[0.578366  ]
[0.43250605]
[0.61523545]
[0.6134211 ]
[0.63264275]
[0.5869834 ]
[0.61780906]
[0.6078469 ]
[0.5963849 ]
[0.46070415]
[0.6066047 ]
[0.5977168 ]
[0.5441392 ]
[0.6032016 ]
[0.54172426]
```

```
[0.5896876 ]
[0.62032807]
[0.52141714]
[0.55583894]
[0.6136186 ]
[0.5700779 ]
[0.59746826]
[0.6012863 ]
[0.51871705]
[0.60718864]
[0.5199144 ]
[0.62296104]
[0.5435468 ]
[0.54559916]
[0.6237406 ]
[0.56865066]
[0.6184436 ]
[0.5713062 ]
[0.6006454 ]
[0.6216932 ]
[0.5416085 ]
[0.5337528 ]
[0.5519368 ]
[0.5496924 ]
[0.5445054 ]
[0.5996599 ]
[0.5876701 ]
[0.5788606 ]
[0.55437315]
[0.6310703 ]
[0.5276569 ]
[0.53155637]
[0.5360635 ]
[0.609246  ]
[0.58207595]
[0.606838  ]
[0.58120817]
[0.5810708 ]
[0.6235242 ]
[0.5883399 ]
[0.6206294 ]
[0.60948795]
[0.48834682]
[0.5853418 ]
[0.5064946 ]
[0.5940578 ]
[0.6058456 ]
[0.5887346 ]
[0.549826  ]
[0.52440846]
[0.5996587 ]
[0.5553578 ]
[0.49121088]
[0.5226131 ]
[0.5552117 ]
[0.5553578 ]
[0.57231677]
[0.60507816]
[0.4784131 ]
[0.62375003]
[0.5308814 ]
```

```
    [0.6282088 ]
    [0.5330252 ]
    [0.6314948 ]
    [0.53673446]
    [0.62880504]
    [0.6214854 ]
    [0.56449336]
    [0.4907276 ]
    [0.6282088 ]
    [0.56954956]
    [0.6238597 ]
    [0.52154505]
    [0.5496924 ]
    [0.589916  ]
    [0.6236069 ]
    [0.5279873 ]
    [0.571739  ]
    [0.48327717]
    [0.608862  ]
    [0.49966648]
    [0.5829092 ]
    [0.62953174]
    [0.6068983 ]
    [0.553599  ]
    [0.62383467]
    [0.60016924]
    [0.54645336]
    [0.5950879 ]
    [0.57841283]
    [0.6067827 ]
    [0.6162843 ]
    [0.60221684]
    [0.5853116 ]
    [0.5686214 ]
    [0.60677487]
    [0.4423089 ]
    [0.58911294]
    [0.6131077 ]
    [0.6246499 ]
    [0.60971653]
    [0.6069699 ]
    [0.56080157]
    [0.6225814 ]
    [0.624129  ]
    [0.59374934]
    [0.5161699 ]
    [0.58913875]
    [0.4718563 ]
    [0.60718864]
    [0.6122281 ]
    [0.59397405]
    [0.5956184 ]
    [0.61503345]
    [0.6267913 ]
    [0.6190014 ]
    [0.49521598]
    [0.6054485 ]]
   [[0.60019314]
    [0.5038585 ]
    [0.5968591 ]
    [0.44461808]
```

```
    [0.57841414]
    [0.6166612 ]
    [0.5688656 ]
    [0.5958611 ]
    [0.61487937]
    [0.6050659 ]
    [0.61151004]
    [0.4970469 ]
    [0.5871501 ]
    [0.6147537 ]
    [0.6172143 ]
    [0.5734427 ]
    [0.6228882 ]
    [0.6189966 ]
    [0.6277602 ]
    [0.6041345 ]
    [0.6003596 ]
    [0.56110656]
    [0.61487937]
    [0.6110243 ]
    [0.59007657]
    [0.58043575]
    [0.5865774 ]
    [0.60950136]
    [0.5953477 ]
    [0.6135077 ]
    [0.6179588 ]
    [0.617587  ]
    [0.5875997 ]
    [0.6071849 ]
    [0.5879005 ]
    [0.57976997]
    [0.61001074]
    [0.6250188 ]
    [0.6184275 ]
    [0.57679856]
    [0.616014  ]
    [0.52027786]
    [0.6015285 ]
    [0.5441094 ]
    [0.59991086]
    [0.58416   ]
    [0.59194696]
    [0.5173561 ]
    [0.61151004]
    [0.6083166 ]
    [0.6034685 ]
    [0.5403935 ]
    [0.5876865 ]
    [0.46828416]
    [0.6015285 ]
    [0.5717318 ]
    [0.61001074]
    [0.51338816]
    [0.61919856]
    [0.54797256]
    [0.58540773]
    [0.5187919 ]
    [0.5853396 ]
    [0.5370236 ]
    [0.5848881 ]
```

```
[0.57883203]
[0.5940072 ]
[0.57422304]
[0.5848446 ]
[0.522971  ]
[0.6274844 ]
[0.57737166]
[0.5977744 ]
[0.4981686 ]
[0.6002759 ]
[0.5928147 ]
[0.57473063]
[0.54430366]
[0.5615314 ]
[0.52969474]
[0.61151004]
[0.5273002 ]
[0.5626115 ]
[0.6059508 ]
[0.5999398 ]
[0.628571  ]
[0.61097217]
[0.57839704]
[0.62096846]
[0.5504524 ]
[0.6173584 ]
[0.5991562 ]
[0.5921694 ]
[0.50172883]
[0.5615315 ]
[0.6142363 ]
[0.59776425]
[0.5992546 ]
[0.6024476 ]
[0.59021425]
[0.6172143 ]
[0.53818935]
[0.6197484 ]
[0.58197486]
[0.5333098 ]
[0.624354  ]
[0.62588537]
[0.60232425]
[0.59619516]
[0.6112432 ]
[0.5928516 ]
[0.60041106]
[0.53705996]
[0.5747998 ]
[0.57263535]
[0.6273725 ]
[0.621574  ]
[0.54711914]
[0.5883083 ]
[0.6133612 ]
[0.5710161 ]
[0.6104547 ]
[0.5853396 ]
[0.58827674]
[0.59506696]
[0.62405837]
```

```
[0.625149  ]
[0.6049462 ]
[0.62571067]
[0.60916287]
[0.6279048 ]
[0.54422027]
[0.5900297 ]
[0.5247656 ]
[0.62793875]
[0.41046095]
[0.5685727 ]
[0.6232865 ]
[0.5955611 ]
[0.554785  ]
[0.5876865 ]
[0.59054494]
[0.56660116]
[0.5675076 ]
[0.61215854]
[0.5712775 ]
[0.6162062 ]
[0.60999966]
[0.6119644 ]
[0.55958736]
[0.5957698 ]
[0.5656289 ]
[0.6002759 ]
[0.61495477]
[0.62521505]
[0.5901789 ]
[0.6234026 ]
[0.59096646]
[0.6174322 ]
[0.5939378 ]
[0.60542357]
[0.6267166 ]
[0.5905728 ]
[0.54415053]
[0.629219  ]
[0.46478814]
[0.5751163 ]
[0.628976  ]
[0.6255488 ]
[0.6287893 ]
[0.5988028 ]
[0.601537  ]
[0.6173584 ]
[0.408045  ]
[0.6228112 ]
[0.50377464]
[0.61382425]
[0.53347147]
[0.61116105]
[0.5973941 ]
[0.6250311 ]
[0.5950079 ]
[0.60315984]
[0.5469042 ]
[0.61783284]
[0.6115767 ]
[0.61974865]
```

```
[0.6200199 ]
[0.5900297 ]
[0.6137567 ]
[0.6221403 ]
[0.59459734]
[0.59506696]
[0.6042139 ]
[0.6167068 ]
[0.60325193]
[0.6102061 ]
[0.6090869 ]
[0.6211147 ]
[0.56957555]
[0.6162062 ]
[0.5782153 ]
[0.5744315 ]
[0.61980146]
[0.6042267 ]
[0.5336687 ]
[0.6186065 ]
[0.5689033 ]
[0.5927521 ]
[0.5990671 ]
[0.61373043]
[0.5885529 ]
[0.59991086]
[0.61701936]
[0.61820006]
[0.58967495]
[0.6053719 ]
[0.6205022 ]
[0.6042267 ]
[0.61312187]
[0.6234026 ]
[0.6140748 ]
[0.6263478 ]
[0.5379417 ]
[0.5848446 ]
[0.58840674]
[0.614509  ]
[0.6111694 ]
[0.6254258 ]
[0.6223425 ]
[0.5766719 ]
[0.5342174 ]
[0.6126206 ]
[0.58111477]
[0.6113661 ]
[0.60491574]
[0.60450935]
[0.57313734]
[0.56432694]
[0.53067416]
[0.62390655]
[0.55701804]
[0.62810993]
[0.4764328 ]
[0.6212565 ]
[0.46701506]
[0.5921694 ]
[0.49696606]
```

```
    [0.5900297 ]
    [0.5937496 ]
    [0.6073666 ]
    [0.5304697 ]
    [0.60333   ]
    [0.60097444]
    [0.61610615]
    [0.61254317]
    [0.5650012 ]
    [0.54578006]
    [0.59549356]
    [0.52416867]
    [0.5875997 ]
    [0.5883775 ]
    [0.5952631 ]
    [0.60887897]
    [0.6064785 ]
    [0.60895854]
    [0.5871501 ]
    [0.6101955 ]
    [0.62214035]
    [0.57560146]
    [0.6165107 ]
    [0.5777033 ]
    [0.5974221 ]
    [0.5366037 ]
    [0.5929582 ]
    [0.6026788 ]
    [0.6167068 ]
    [0.5813254 ]
    [0.5744315 ]
    [0.5753028 ]
    [0.62923986]
    [0.59551144]
    [0.6104913 ]
    [0.5841702 ]
    [0.6101808 ]
    [0.5979475 ]
    [0.55009055]
    [0.57509905]
    [0.6167068 ]
    [0.4898541 ]
    [0.5626115 ]
    [0.6290062 ]
    [0.6105692 ]
    [0.5207489 ]
    [0.58000094]
    [0.6014904 ]
    [0.6263148 ]
    [0.6177758 ]
    [0.6026247 ]
    [0.47295192]
    [0.59916884]]
```

In [45]:

```
# storage for ranked histogram
ranked_histogram = np.zeros(len(test_gnd))

# loop over all IDs in the probe set
for i in range(len(ID_2)):
    # get the true ID of this sample
    true_ID = ID_2[i]
    print('Searching for ID %d' % (true_ID))

    # get the distance between the current probe and the whole gallery, L1 distance her
e. Note that L1
    # may not always be the best choice, so consider your distance metric given your pr
oblem
    dist = np.sqrt(np.sum(((Feat_2[:] - Feat_1[i])**2), axis=1)) #This one needs to cha
nge - Too large?
    print('Distance between source feature and all target features:')
    print(dist)

    # get the sorted order of the distances
    a = np.argsort(dist)
    # apply the order to the gallery IDs, such that the first ID in the list is the clo
sest, the second
    # ID is the second closest, and so on
    ranked = ID_1[a]
    print('Ranked IDs for query:')
    print(a)

    # find the location of the True Match in the ranked list
    ranked_result = np.where(ranked == true_ID)[0][0]
    print(ranked_result)

    # store the ranking result in the histogram
    ranked_histogram[ranked_result] += 1
    print('')

print(ranked_histogram)
```

```
71163487e-02
 2.14194059e-02 6.28912449e-03 2.25645900e-02 1.36694312e-02
 3.38870287e-03 1.27164721e-02 4.43518162e-04 8.88496041e-02
 4.19467092e-02 3.83845568e-02 1.22823119e-02 1.56219006e-02
 1.36548281e-03 4.44877148e-03 5.01194000e-02 9.25738811e-02
 1.41707063e-02 4.56765294e-02 1.54252052e-02 2.18755603e-02
 2.22819448e-02 5.36539555e-02 6.24643564e-02 9.61171389e-02
 2.88474560e-03 6.97732568e-02 1.31863356e-03 1.50358498e-01
 5.53482771e-03 1.59776241e-01 3.46218944e-02 1.29825234e-01
 3.67615819e-02 3.30417156e-02 1.94246769e-02 9.63215828e-02
 2.34612823e-02 2.58168578e-02 1.06851459e-02 1.42481327e-02
 6.17901087e-02 8.10112357e-02 3.12977433e-02 1.02622628e-01
 3.91916037e-02 3.84138227e-02 3.15281749e-02 1.79123282e-02
 2.03127861e-02 1.78327560e-02 3.96412015e-02 1.65957808e-02
 4.65095043e-03 5.11898398e-02 1.02806091e-02 4.90880013e-02
 2.93691754e-02 9.01876092e-02 3.38330865e-02 2.41125226e-02
 1.00845098e-02 4.54658866e-02 5.23598194e-02 5.14885187e-02
 2.44855881e-03 3.12798619e-02 1.63000226e-02 4.26210761e-02
 1.66105032e-02 2.88438201e-02 7.67007470e-02 5.16922474e-02
 1.00845098e-02 1.36937201e-01 6.41797781e-02 2.21490860e-03
 1.62221193e-02 1.06042385e-01 4.67903614e-02 2.53009200e-02
 4.76479530e-04 9.01550055e-03 2.41665840e-02 1.53839380e-01
 2.76224613e-02]
Ranked IDs for query:
[161 222 296 115  70 106  18 128 130 134 168 242 228 154 126 180  37  85
 169 167 291 164 105 280 125 240 220 156 137  16 174 229 268 190 116 244
 198  88 217 187 203 186 102  58  17 206  38 214  30 184 297  31 158 172
  90  14 100 213 276 288 194   5 270 200 146 254  40 153   8  22  13 226
  95 221 176 189 210  29 119 219 232 255 144 148 185  80  48  10 234 109
 227 178  23  86 292 282 121 196 267 284  36  56 147  27 129 197 265 263
  49 250  33 264  83 160 216   9 127 235 236 204 218 193  19  50 252 195
 182 275 298  98 107 171  42  54 295 253 111  20 152  74   0  84  44 212
  97 300  91 209 170 285  72  96 272 179   2 108   7 150 138 281 258  28
 262 124 192 181 191  66 159 249 274 110  75 208 246  92  46 157 162 141
  99 155  24 248 188 132 215 211 225 261 118 123  34  52 140 260  32  12
 266  26  60 122  62  64  68 224 283  45 103 277 233  25 294  35  65   4
  87 201 271  71  39 230 269 279 166 287 113  76 278 202  67  15 237 114
  55 145 120 199 207   6 136 143 142 151 256 238  82 290  94  78  21 149
 241 139  89 286  59 117 183 257  77 131 163  43  51 101 223 112  63 273
 231 205 177 104 239 251  79  81 133 259  69 293  41  61  47  57   1 175
  93  73  11 247 289 243 299  53 245 165   3 135 173]
50

Searching for ID 1498
Distance between source feature and all target features:
[1.88082457e-02 1.15142882e-01 2.21422911e-02 1.74383312e-01
 4.05872464e-02 2.34019756e-03 5.01357913e-02 2.31403112e-02
 4.12201881e-03 1.39355063e-02 7.49135017e-03 1.21954501e-01
 3.18512917e-02 4.24766541e-03 1.78706646e-03 4.55586910e-02
 3.88681889e-03 4.76837158e-06 8.75878334e-03 1.48668885e-02
 1.86417699e-02 5.78948259e-02 4.12201881e-03 7.97706842e-03
 2.89248228e-02 3.85656357e-02 3.24239731e-02 9.50002670e-03
 2.36536860e-02 5.49370050e-03 1.04260445e-03 1.41441822e-03
 3.14016938e-02 1.18165016e-02 3.11008692e-02 3.92314196e-02
 8.99064541e-03 6.01738691e-03 5.73873520e-04 4.22028303e-02
 2.98738480e-03 9.87235308e-02 1.74728632e-02 7.48919845e-02
 1.90905333e-02 3.48414183e-02 2.70544291e-02 1.01645291e-01
 7.49135017e-03 1.06847882e-02 1.55329108e-02 7.86079168e-02
 3.13149095e-02 1.50717229e-01 1.74728632e-02 4.72695827e-02
 8.99064541e-03 1.05613232e-01 1.97172165e-04 7.10288286e-02
 3.35936546e-02 1.00209475e-01 3.36617827e-02 8.19777846e-02
```

```
  3.41132879e-02 4.01693583e-02 2.49941945e-02 4.47783470e-02
  3.41567993e-02 9.60304141e-02 8.48299265e-03 4.16297317e-02
  2.12270021e-02 1.20832801e-01 1.87255144e-02 2.61867046e-02
  4.42707539e-02 7.46977329e-02 5.74699640e-02 8.93066525e-02
  7.49135017e-03 9.17012095e-02 5.63898683e-02 1.30506158e-02
  1.90615654e-02 9.56958532e-03 8.02922249e-03 4.06043530e-02
  1.96707249e-03 6.85489774e-02 1.64300203e-03 1.98451877e-02
  2.68319845e-02 1.17272556e-01 5.74699044e-02 4.76509333e-03
  2.12371349e-02 1.97467804e-02 1.65537596e-02 2.87871361e-02
  1.78706646e-03 8.08120370e-02 7.47025013e-04 3.70265245e-02
  8.56915712e-02 5.35261631e-03 6.88397884e-03 1.66771412e-02
  2.28062272e-02 7.75820017e-03 2.61498094e-02 1.85903311e-02
  8.19414258e-02 4.42016125e-02 4.63660359e-02 8.37111473e-03
  2.57259607e-03 7.18822479e-02 3.06931138e-02 5.64020872e-03
  4.79853153e-02 8.54671001e-03 3.36617827e-02 3.07246447e-02
  2.39344239e-02 5.05697727e-03 6.14762306e-03 1.40551925e-02
  6.70927763e-03 9.83852148e-03 8.90338421e-03 7.47811198e-02
  2.89716721e-02 9.42357779e-02 8.93735886e-03 2.08540440e-01
  5.04286885e-02 4.28509712e-03 2.34403014e-02 6.42163754e-02
  3.13149095e-02 2.84564495e-02 5.24002314e-02 5.14937639e-02
  6.84285164e-03 4.77238894e-02 2.79515982e-03 9.00173187e-03
  7.03698397e-03 5.94140291e-02 2.32315660e-02 5.33725023e-02
  1.87255144e-02 4.04661894e-03 6.21366501e-03 2.88224816e-02
  4.40120697e-03 2.80349255e-02 1.56921148e-03 2.50635743e-02
  1.35778189e-02 7.71522522e-03 2.84286141e-02 7.48508573e-02
  1.02176070e-02 1.54213250e-01 4.38851118e-02 9.97459888e-03
  6.54739141e-03 9.78791714e-03 2.01985836e-02 1.74643993e-02
  1.64300203e-03 2.10956395e-01 3.80980968e-03 1.15226746e-01
  5.17714024e-03 8.55299234e-02 7.84033537e-03 2.16072798e-02
  6.02972507e-03 2.39934921e-02 1.58415437e-02 7.20971823e-02
  1.16854906e-03 7.42471218e-03 7.47263432e-04 1.01852417e-03
  2.89716721e-02 5.24467230e-03 3.13889980e-03 2.44040489e-02
  2.39344239e-02 1.47874951e-02 2.29460001e-03 1.57494545e-02
  8.79526138e-03 9.91451740e-03 2.11328268e-03 4.94258404e-02
  2.79515982e-03 4.07860875e-02 4.45699096e-02 8.00073147e-04
  1.47746801e-02 8.53326917e-02 3.94880772e-04 5.00980616e-02
  2.62492895e-02 1.99342966e-02 5.27095795e-03 3.04484963e-02
  1.90905333e-02 1.98203325e-03 8.01324844e-04 2.93264389e-02
  1.36294961e-02 1.50078535e-03 1.47746801e-02 5.87952137e-03
  4.40120697e-03 4.92656231e-03 7.34639168e-03 8.10596943e-02
  3.41567993e-02 3.05946469e-02 4.49240208e-03 7.83199072e-03
  6.42442703e-03 3.34113836e-03 4.23294902e-02 8.47839713e-02
  6.38079643e-03 3.78866196e-02 7.63529539e-03 1.40856504e-02
  1.44920349e-02 4.58640456e-02 5.46744466e-02 8.83272290e-02
  4.90516424e-03 6.19833469e-02 9.10854340e-03 1.42568588e-01
  2.25508213e-03 1.51986331e-01 2.68319845e-02 1.22035325e-01
  2.89716721e-02 2.52518058e-02 1.16347671e-02 8.85316730e-02
  1.56713724e-02 1.80269480e-02 2.89523602e-03 6.45822287e-03
  5.40001988e-02 7.32213259e-02 2.35078335e-02 9.48327184e-02
  3.14016938e-02 3.06239128e-02 2.37382650e-02 1.01224184e-02
  1.25228763e-02 1.00428462e-02 3.18512917e-02 8.80587101e-03
  3.13895941e-03 4.33999300e-02 2.49069929e-03 4.12980914e-02
  2.15792656e-02 8.23976994e-02 2.60431767e-02 1.63226128e-02
  2.29460001e-03 3.76759768e-02 4.45699096e-02 4.36986089e-02
  1.02384686e-02 2.34899521e-02 8.51011276e-03 3.48311663e-02
  8.82059336e-03 2.10539103e-02 6.89108372e-02 4.39023376e-02
  2.29460001e-03 1.29147291e-01 5.63898683e-02 1.00048184e-02
  8.43220949e-03 9.82524753e-02 3.90004516e-02 1.75110102e-02
  7.31343031e-03 1.22559071e-03 1.63766742e-02 1.46049470e-01
  1.98325515e-02]
 Ranked IDs for query:
```

```
[ 17  58 206  38 102 186 203 214 187  30 184 297  31 217 158 172  90 100
  14  88 213 198 244 276 288 194   5 270 116 146 200 254  40 190 268 229
 174  16 153  22   8  13 137 220 156 226  95 240 221 125 176 189 210 105
  29 119 219  37 180 126 154 232 228 255 168 128 144 106 148 296 222 185
  10  48  80 234 161 109 227 178  23  86 115 292  70 282 121  18 196 267
 284 130 134  36  56 147 242  27  85 169 129 197 167 291 265 263 164 280
  49 250  33 264  83 160 216   9 127 235 236 204 218 193  19  50 252 195
 182 275 298  98 107 171  42  54 295 253 111  20 152  74   0  84  44 212
  97 300  91 209 170 285  72  96 272 179   2 108   7 150 138 281 258  28
 262 124 192 181 191  66 159 249 274 110  75 208 246  92  46 157 162 141
  99 155  24 248 188 132 215 211 225 261 118 123  34  52 140 260  32  12
 266  26  60 122  62  64  68 224 283  45 103 277 233  25 294  35  65   4
  87 201 271  71  39 230 269 279 166 287 113  76 278 202  67  15 237 114
  55 145 120 199 207   6 136 143 142 151 256 238  82 290  94  78  21 149
 241 139  89 286  59 117 183 257  77 131 163  43  51 101 223 112  63 273
 231 205 177 104 239 251  79  81 133 259  69 293  41  61  47  57   1 175
  93  73  11 247 289 243 299  53 245 165   3 135 173]
128

Searching for ID 1499
Distance between source feature and all target features:
[0.10497716 0.00864252 0.10164312 0.05059791 0.08319816 0.12144521
 0.07364962 0.1006451  0.11966339 0.1098499  0.11629406 0.00183091
 0.09193411 0.11953774 0.12199834 0.07822672 0.12767223 0.12378064
 0.13254419 0.10891852 0.10514364 0.06589058 0.11966339 0.11580834
 0.09486058 0.08521977 0.09136143 0.11428538 0.10013172 0.11829171
 0.1227428  0.12237099 0.09238371 0.1119689  0.09268454 0.08455399
 0.11479476 0.1298028  0.12321153 0.08158258 0.12079802 0.02506188
 0.10631254 0.04889342 0.10469487 0.08894399 0.09673098 0.02214012
 0.11629406 0.11310062 0.1082525  0.04517749 0.0924705  0.02693182
 0.10631254 0.07651582 0.11479476 0.01817217 0.12398258 0.05275658
 0.09019175 0.02357593 0.09012362 0.04180762 0.08967212 0.08361605
 0.09879121 0.07900706 0.08962861 0.02775499 0.1322684  0.08215567
 0.1025584  0.00295261 0.10505989 0.0975987  0.07951465 0.04908767
 0.06631544 0.03447875 0.11629406 0.0320842  0.06739554 0.11073479
 0.10472384 0.13335499 0.11575618 0.08318105 0.12575248 0.05523643
 0.1221424  0.10394022 0.09695342 0.00651285 0.0663155  0.11902031
 0.10254827 0.10403863 0.10723165 0.09499827 0.12199834 0.04297337
 0.12453243 0.08675888 0.03809384 0.12913802 0.13066939 0.10710827
 0.10097918 0.11602721 0.0976356  0.10519508 0.04184398 0.07958379
 0.07741937 0.13215652 0.126358   0.05190316 0.09309229 0.1181452
 0.07580009 0.1152387  0.09012362 0.09306076 0.09985098 0.12884238
 0.12993303 0.10973021 0.13049468 0.11394688 0.13268879 0.04900429
 0.09481373 0.02954963 0.13272277 0.08475503 0.07335672 0.1280705
 0.1003451  0.05956903 0.0924705  0.09532896 0.07138517 0.07229164
 0.11694255 0.07606152 0.12099025 0.11478367 0.11674842 0.06437138
 0.10055384 0.0704129  0.10505989 0.11973879 0.12999907 0.09496292
 0.12818661 0.09575048 0.12221619 0.09872183 0.11020759 0.13150063
 0.09535679 0.04893455 0.13400301 0.03042784 0.07990029 0.13376
 0.1303328  0.13357332 0.10358682 0.10632101 0.1221424  0.08717099
 0.12759522 0.00855866 0.11860827 0.03825548 0.11594507 0.10217813
 0.12981513 0.09979191 0.10794386 0.05168822 0.12261686 0.11636069
 0.12453267 0.12480393 0.09481373 0.11854073 0.1269243  0.09938136
 0.09985098 0.10899791 0.12149081 0.10803595 0.11499014 0.11387089
 0.12589869 0.07435957 0.12099025 0.08299932 0.0792155  0.12458548
 0.10901073 0.03845271 0.12339053 0.07368734 0.09753612 0.10385111
 0.11851445 0.09333691 0.10469487 0.12180337 0.12298408 0.09445897
 0.11015591 0.12528619 0.10901073 0.11790588 0.12818661 0.11885884
 0.1311318  0.04272571 0.08962861 0.09319076 0.119293   0.11595342
 0.13020983 0.12712654 0.08145592 0.03900144 0.11740461 0.08589879
 0.11615011 0.10969976 0.10929337 0.07792136 0.06911096 0.03545818
```

```
 0.12869057 0.06180206 0.13289395 0.01878318 0.12604049 0.02820092
 0.09695342 0.00175008 0.09481373 0.0985336  0.11215064 0.03525373
 0.10811403 0.10575846 0.12089017 0.11732718 0.06978521 0.05056408
 0.10027757 0.02895269 0.09238371 0.09316149 0.10004714 0.11366299
 0.11126253 0.11374256 0.09193411 0.11497954 0.12692437 0.08038548
 0.12129471 0.08248731 0.10220614 0.04138771 0.09774223 0.10746279
 0.12149081 0.08610943 0.0792155  0.0800868  0.13402387 0.10029545
 0.11527529 0.08895424 0.11496481 0.1027315  0.05487457 0.07988307
 0.12149081 0.00536188 0.06739554 0.13379022 0.1153532  0.02553293
 0.08478495 0.1062744  0.13109884 0.12255982 0.10740873 0.02226406
 0.10395285]
Ranked IDs for query:
[247  11  73 289  93 175   1  57 243  47 299  61  41 293  53  69 245 259
 133 165  81  79 251 239 104 177 205 231 273  63 112 223 101  51  43 163
 131  77 257   3 183 117  59 286  89 139 241 149  21  78  94  82 290 238
 256 151 142 143 136   6 207 199 120 145  55 114 237  15  67 202 278  76
 113 287 166 279 269 230  39  71 271 201  87   4  65  35 135 294  25 233
 277 103 173  45 283  68 224  64  62 122  60  26  12 266  32 260  52 140
  34 123 118 261 225 211 215 248 188 132  24 155  99 141 162 157  46 246
  92 208  75 110 274 249 159  66 191 181 124 192 262  28 258 281 138 150
   7 108   2 179 272  96  72 285 170 209  91 300  97 212  44  84   0 152
  74  20 111 253 295  42  54 171 107  98 298 275 182 195 252  50  19 193
 204 218 236 235 127   9 216 160  83 264  33 250  49 263 265 197 129  27
 147  36  56 284 267 196 121 282 292  86  23 178 227 109 234  10  48  80
 185 148 144 255 232 219 119  29 210 189 176 221  95 226  13   8  22 153
  40 254 200 146 270   5 276 194 288 213  14 100  90 172 158  31 297 184
  30 214  38 206  17  58 102 186 203 187 217  88 198 244 116 190 268 229
 174  16 137 156 220 240 125 105  37 180 126 154 228 168 128 106 296 222
 161 115  70  18 130 134 242  85 169 167 291 164 280]
10

Searching for ID 1500
Distance between source feature and all target features:
[5.25534153e-03 1.01589978e-01 8.58938694e-03 1.60830408e-01
 2.70343423e-02 1.12127066e-02 3.65828872e-02 9.58740711e-03
 9.43088531e-03 3.82602215e-04 6.06155396e-03 1.08401597e-01
 1.82983875e-02 9.30523872e-03 1.17658377e-02 3.20057869e-02
 1.74397230e-02 1.35481358e-02 2.23116875e-02 1.31398439e-03
 5.08886576e-03 4.43419218e-02 9.43088531e-03 5.57583570e-03
 1.53719187e-02 2.50127316e-02 1.88710690e-02 4.05287743e-03
 1.01007819e-02 8.05920362e-03 1.25102997e-02 1.21384859e-02
 1.78487897e-02 1.73640251e-03 1.75479650e-02 2.56785154e-02
 4.56225872e-03 1.95702910e-02 1.29790306e-02 2.86499262e-02
 1.05655193e-02 8.51706266e-02 3.91995907e-03 6.13390803e-02
 5.53762913e-03 2.12885141e-02 1.35015249e-02 8.80923867e-02
 6.06155396e-03 2.86811590e-03 1.98000669e-03 6.50550127e-02
 1.77620053e-02 1.37164325e-01 3.91995907e-03 3.37166786e-02
 4.56225872e-03 9.20603275e-02 1.37500763e-02 5.74759245e-02
 2.00407505e-02 8.66565704e-02 2.01088786e-02 6.84248805e-02
 2.05603838e-02 2.66164541e-02 1.14412904e-02 3.12254429e-02
 2.06038952e-02 8.24775100e-02 2.20358968e-02 2.80768275e-02
 7.67409801e-03 1.07279897e-01 5.17261028e-03 1.26338005e-02
 3.07178497e-02 6.11448288e-02 4.39170599e-02 7.57537484e-02
 6.06155396e-03 7.81483054e-02 4.28369641e-02 5.02288342e-04
 5.50866127e-03 2.31224895e-02 5.52368164e-03 2.70514488e-02
 1.55199766e-02 5.49960732e-02 1.19099021e-02 6.29228354e-03
 1.32790804e-02 1.03719652e-01 4.39170003e-02 8.78781080e-03
 7.68423080e-03 6.19387627e-03 3.00085545e-03 1.52342319e-02
 1.17658377e-02 6.72591329e-02 1.42999291e-02 2.34736204e-02
 7.21386671e-02 1.89055204e-02 2.04368830e-02 3.12423706e-03
 9.25332308e-03 5.79470396e-03 1.25969052e-02 5.03742695e-03
```

```
  6.83885217e-02 3.06487083e-02 3.28131318e-02 2.19240189e-02
  1.61255002e-02 5.83293438e-02 1.71402097e-02 7.91269541e-03
  3.44324112e-02 5.00619411e-03 2.01088786e-02 1.71717405e-02
  1.03815198e-02 1.86098814e-02 1.97005272e-02 5.02288342e-04
  2.02621818e-02 3.71438265e-03 2.24562883e-02 6.12282157e-02
  1.54187679e-02 8.06828737e-02 2.24902630e-02 1.94987535e-01
  3.68757844e-02 1.78380013e-02 9.88739729e-03 5.06634712e-02
  1.77620053e-02 1.49035454e-02 3.88473272e-02 3.79408598e-02
  6.71005249e-03 3.41709852e-02 1.07577443e-02 4.55117226e-03
  6.51592016e-03 4.58611250e-02 9.67866182e-03 3.98195982e-02
  5.17261028e-03 9.50628519e-03 1.97665691e-02 1.52695775e-02
  1.79541111e-02 1.44820213e-02 1.19836926e-02 1.15106702e-02
  2.49147415e-05 2.12681293e-02 1.48757100e-02 6.12979531e-02
  2.37705112e-02 1.40660346e-01 3.03322077e-02 2.35275030e-02
  2.01002955e-02 2.33408213e-02 6.64567947e-03 3.91149521e-03
  1.19099021e-02 1.97403491e-01 1.73627138e-02 1.01673841e-01
  8.37576389e-03 7.19770193e-02 5.71256876e-03 8.05437565e-03
  1.95826292e-02 1.04405880e-02 2.28863955e-03 5.85442781e-02
  1.23843551e-02 6.12819195e-03 1.43001676e-02 1.45714283e-02
  1.54187679e-02 8.30823183e-03 1.66918039e-02 1.08511448e-02
  1.03815198e-02 1.23459101e-03 1.12583041e-02 2.19655037e-03
  4.75764275e-03 3.63838673e-03 1.56661868e-02 3.58729362e-02
  1.07577443e-02 2.72331834e-02 3.10170054e-02 1.43529773e-02
  1.22177601e-03 7.17797875e-02 1.31580234e-02 3.65451574e-02
  1.26963854e-02 6.38139248e-03 8.28194618e-03 1.68955922e-02
  5.53762913e-03 1.15708709e-02 1.27515793e-02 1.57735348e-02
  7.65919685e-05 1.50536895e-02 1.22177601e-03 7.67338276e-03
  1.79541111e-02 8.62634182e-03 2.08992958e-02 6.75067902e-02
  2.06038952e-02 1.70417428e-02 9.06050205e-03 5.72091341e-03
  1.99773312e-02 1.68940425e-02 2.87765861e-02 7.12310672e-02
  7.17210770e-03 2.43337154e-02 5.91760874e-03 5.32746315e-04
  9.39130783e-04 3.23111415e-02 4.11215425e-02 7.47743249e-02
  1.84580684e-02 4.84304428e-02 2.26614475e-02 1.29015684e-01
  1.58079863e-02 1.38433427e-01 1.32790804e-02 1.08482420e-01
  1.54187679e-02 1.16989017e-02 1.91813707e-03 7.49787688e-02
  2.11846828e-03 4.47404385e-03 1.06576681e-02 7.09468126e-03
  4.04472947e-02 5.96684217e-02 9.95492935e-03 8.12798142e-02
  1.78487897e-02 1.70710087e-02 1.01853609e-02 3.43048573e-03
  1.03002787e-03 3.51005793e-03 1.82983875e-02 4.74703312e-03
  1.66918635e-02 2.98470259e-02 1.10622048e-02 2.77451873e-02
  8.02636147e-03 6.88447952e-02 1.24902725e-02 2.76970863e-03
  1.12583041e-02 2.41230726e-02 3.10170054e-02 3.01457047e-02
  2.37913728e-02 9.93704796e-03 5.04279137e-03 2.12782621e-02
  4.73231077e-03 7.50100613e-03 5.53579330e-02 3.03494334e-02
  1.12583041e-02 1.15594387e-01 4.28369641e-02 2.35577226e-02
  5.12069464e-03 8.46995711e-02 2.54475474e-02 3.95810604e-03
  2.08663344e-02 1.23273134e-02 2.82377005e-03 1.32496566e-01
  6.27964735e-03]
Ranked IDs for query:
[160 216   9 127  83 235 236 264 204 218 193  19  33 250  50 252 195 182
 275 298  49  98 107 263 265 197 129 171  54  42 295  27 253 147  36  56
 284 267 196 121 111 282  20 292  74 152   0  84  86 212  44  23 178 227
 109 234  48  80  10 185  97 300  91 209 148 170 144 255 232 285 219  72
  96 119 272 179  29 210 189 176   2 221  95 226 108  13  22   8 153   7
 150 138 281 258  28 262 124 192 181  40 254 146 200 191 270   5 288 276
 194  66 159 213 249 100  14  90 172 158  31 297 184 274  30 110  75 208
 214  38 206 246  92  46  17  58 102 186 203 157 187 162 141 217  99 155
  24 248 132 188  88 198 215 244 116 190 268 229 211 225 261 118 123 174
  16  34  52 140 137  32 260 220 156  12 266 240 125  26 105  37 180 126
 154 228  60 168  62 122 128 106  64  68 224 296 222 161 283  45 115  70
  18 130 134 242  85 169 103 167 291 164 280 277 233  25 294  35  65   4
```

```
  87 201 271  71  39 230 269 279 166 287 113  76 202 278  67  15 237 114
  55 145 120 199 207   6 136 143 142 151 256 238 290  82  94  78  21 149
 241 139  89 286  59 117 183 257  77 131 163  43  51 101 223 112  63 273
 231 205 177 104 239 251  79  81 133 259  69 293  41  61  47  57   1 175
  93  73  11 247 289 243 299  53 245 165   3 135 173]
61

[1. 0. 0. 2. 2. 1. 1. 1. 2. 0. 2. 0. 1. 5. 1. 1. 2. 2. 2. 2. 0. 2. 1. 2.
 0. 1. 2. 0. 4. 2. 1. 0. 0. 0. 4. 4. 2. 0. 0. 1. 0. 0. 0. 1. 1. 0. 2. 0.
 1. 1. 1. 1. 2. 2. 1. 0. 1. 2. 1. 2. 0. 1. 1. 0. 2. 1. 1. 2. 0. 2. 3. 1.
 0. 1. 1. 1. 4. 2. 1. 0. 2. 0. 2. 0. 2. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 2.
 1. 0. 2. 3. 0. 2. 1. 1. 0. 2. 2. 0. 1. 0. 0. 1. 3. 0. 1. 1. 0. 2. 0. 0.
 2. 3. 0. 0. 1. 1. 0. 0. 1. 1. 2. 2. 1. 0. 1. 0. 2. 0. 1. 2. 3. 0. 0. 2.
 1. 1. 2. 0. 3. 1. 0. 0. 2. 1. 1. 2. 2. 0. 2. 1. 1. 1. 0. 1. 0. 0. 1. 1.
 0. 1. 1. 2. 2. 0. 2. 0. 2. 4. 2. 0. 2. 1. 2. 2. 0. 3. 0. 2. 1. 0. 1. 1.
 2. 1. 0. 2. 1. 0. 0. 2. 2. 0. 0. 1. 1. 1. 1. 0. 1. 0. 2. 0. 0. 0. 2. 1.
 0. 1. 2. 2. 0. 1. 0. 0. 0. 2. 0. 1. 0. 2. 1. 1. 0. 1. 1. 0. 1. 1. 2. 1.
 0. 3. 1. 0. 1. 0. 0. 1. 2. 1. 2. 0. 0. 1. 0. 2. 2. 1. 1. 1. 1. 1. 1. 0.
 1. 0. 1. 1. 0. 2. 1. 0. 1. 1. 0. 0. 3. 0. 1. 0. 0. 1. 1. 0. 1. 0. 2. 1.
 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 2. 1. 1.]
```

In [46]:

```
cmc = np.zeros(301)
for i in range(301):
    cmc[i] = np.sum(ranked_histogram[:(i + 1)])

print(cmc)
```

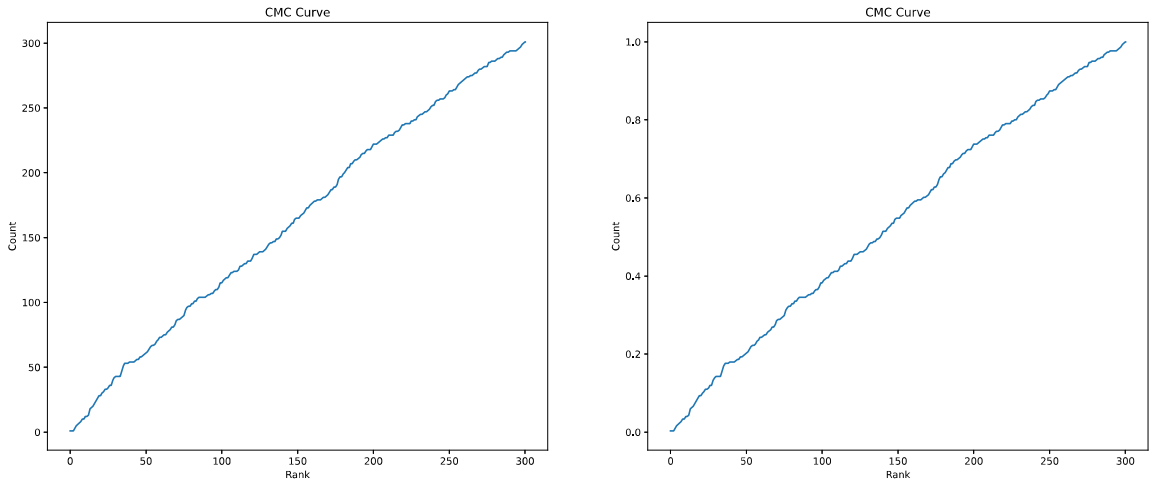
```
[  1.   1.   1.   3.   5.   6.   7.   8.  10.  10.  12.  12.  13.  18.
  19.  20.  22.  24.  26.  28.  28.  30.  31.  33.  33.  34.  36.  36.
  40.  42.  43.  43.  43.  43.  47.  51.  53.  53.  53.  54.  54.  54.
  54.  55.  56.  56.  58.  58.  59.  60.  61.  62.  64.  66.  67.  67.
  68.  70.  71.  73.  73.  74.  75.  75.  77.  78.  79.  81.  81.  83.
  86.  87.  87.  88.  89.  90.  94.  96.  97.  97.  99.  99. 101. 101.
 103. 104. 104. 104. 104. 104. 105. 106. 106. 107. 107. 109. 110. 110.
 112. 115. 115. 117. 118. 119. 119. 121. 123. 123. 124. 124. 124. 125.
 128. 128. 129. 130. 130. 132. 132. 132. 134. 137. 137. 137. 138. 139.
 139. 139. 140. 141. 143. 145. 146. 146. 147. 147. 149. 149. 150. 152.
 155. 155. 155. 157. 158. 159. 161. 161. 164. 165. 165. 165. 167. 168.
 169. 171. 173. 173. 175. 176. 177. 178. 178. 179. 179. 179. 180. 181.
 181. 182. 183. 185. 187. 187. 189. 189. 191. 195. 197. 197. 199. 200.
 202. 204. 204. 207. 207. 209. 210. 210. 211. 212. 214. 215. 215. 217.
 218. 218. 218. 220. 222. 222. 222. 223. 224. 225. 226. 226. 227. 227.
 229. 229. 229. 229. 231. 232. 232. 233. 235. 237. 237. 238. 238. 238.
 238. 240. 240. 241. 241. 243. 244. 245. 245. 246. 247. 247. 248. 249.
 251. 252. 252. 255. 256. 256. 257. 257. 257. 258. 260. 261. 263. 263.
 263. 264. 264. 266. 268. 269. 270. 271. 272. 273. 274. 274. 275. 275.
 276. 277. 277. 279. 280. 280. 281. 282. 282. 282. 285. 285. 286. 286.
 286. 287. 288. 288. 289. 289. 291. 292. 293. 293. 294. 294. 294. 294.
 294. 295. 296. 297. 299. 300. 301.]
```

In [47]:

```
fig = plt.figure(figsize=[20, 8])
ax = fig.add_subplot(1, 2, 1)
ax.plot(cmc)
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve')
ax = fig.add_subplot(1, 2, 2)
ax.plot(cmc/len(test_gnd))
ax.set_xlabel('Rank')
ax.set_ylabel('Count')
ax.set_title('CMC Curve')
```

Out[47]:

```
Text(0.5, 1.0, 'CMC Curve')
```



In [ ]: