EE 5907 Pattern Recognition
CA2 Report

by

Wanry Lin
A0000001X

Department of Electronic and Computing Engineering

National University of Singapore

10/11/2022

Summary

This report shows the result and discussion of the Course Assignment 2 of the EE 5907 Pattern Recognition. The assignment is based on Python 3.10 and MacOS. There is 7 parts of this assignments, including:

- Images data reading and store
- PCA based data distribution visualization
- PCA plus nearest neighbour classification results
- LDA based data distribution visualization
- LDA plus nearest neighbour classification results
- SVM classification results with different parameter values
- CNN classification results

1.Part 0

This part is built for pre-processing the images data. Store them into .json file for acerating following part loading data speed.

1.1 Warning

This part is built in MacOS environment. So it may not work in Windows environment.

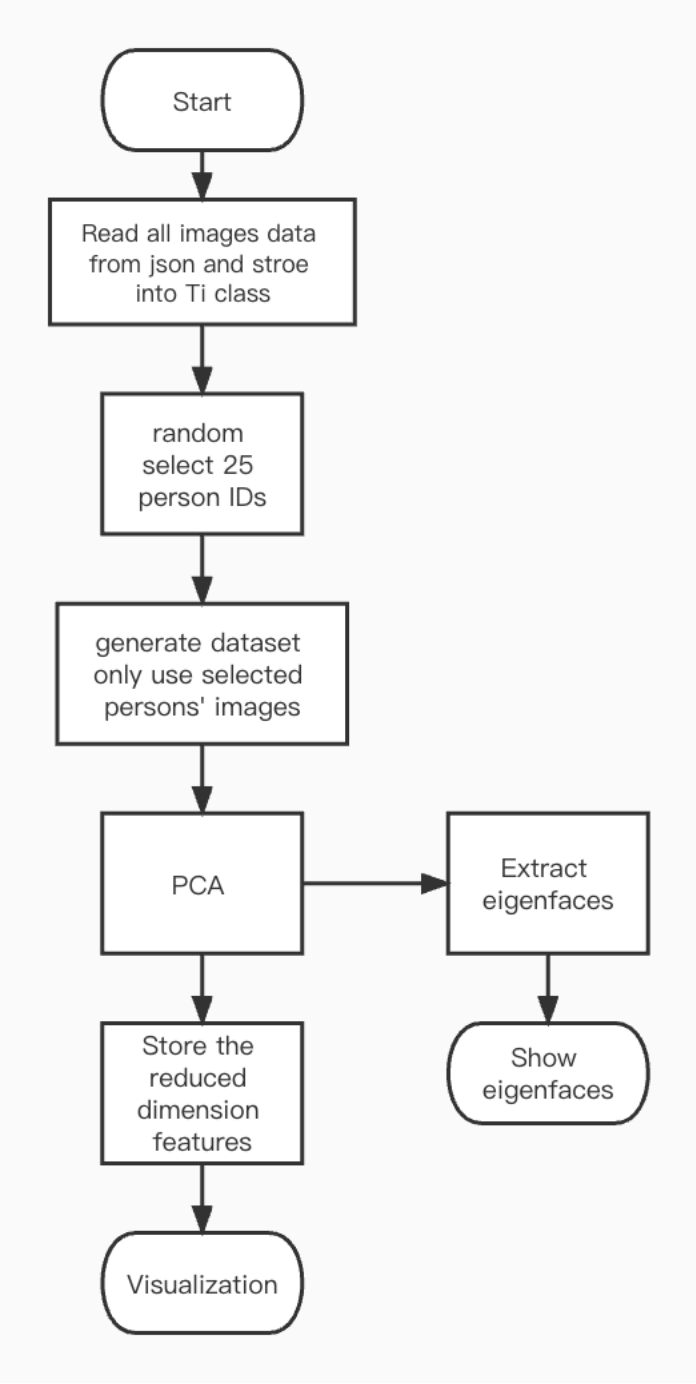
1.2 My photos

I take 20 photos and store them in Me folder.

2.Part 1

This part is using PCA to reduce the dimension of the images data and plotting them in 2D and 3D figures.

2.1 Process flow



2.2 Result

The sample number of dataset is 510, which consists of 500 images randomly from all images and 10 images of myself randomly.

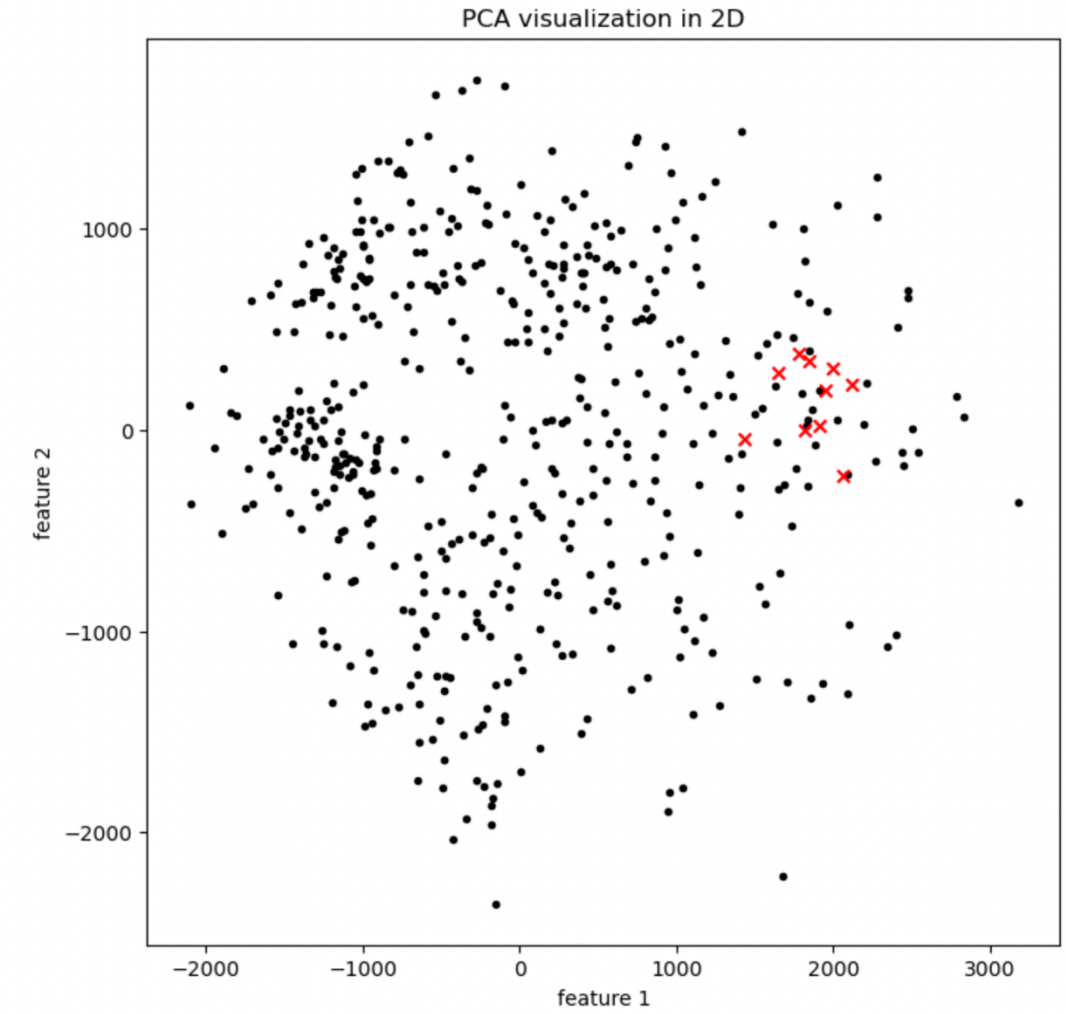
2.2.1 Eigenfaces

eigenface 1

eigenface 2

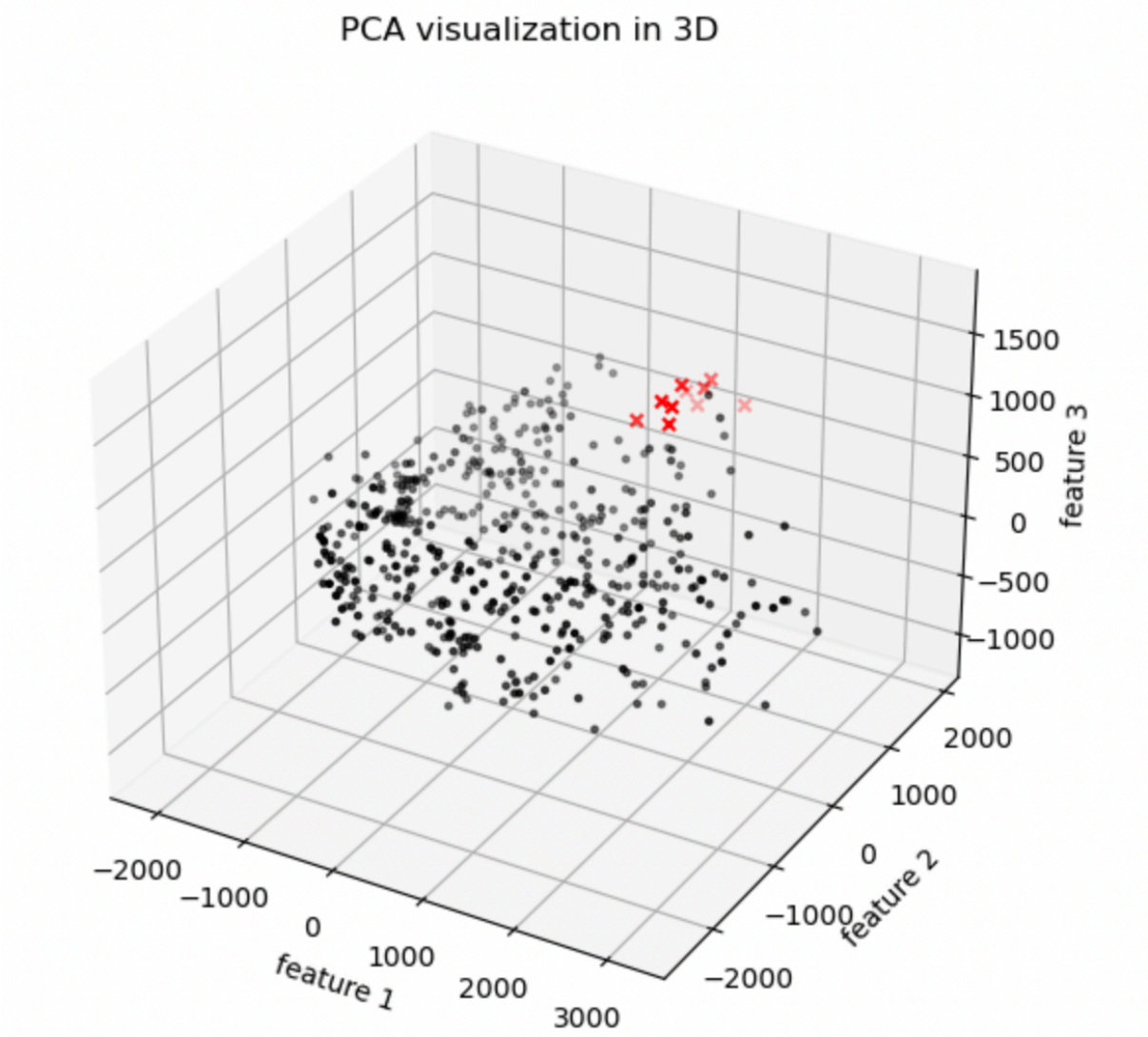
eigenface 3

2.2.2 2D Visualization



The red x points refer to my images.

2.2.3 3D Visualization



The red x points refer to my images.

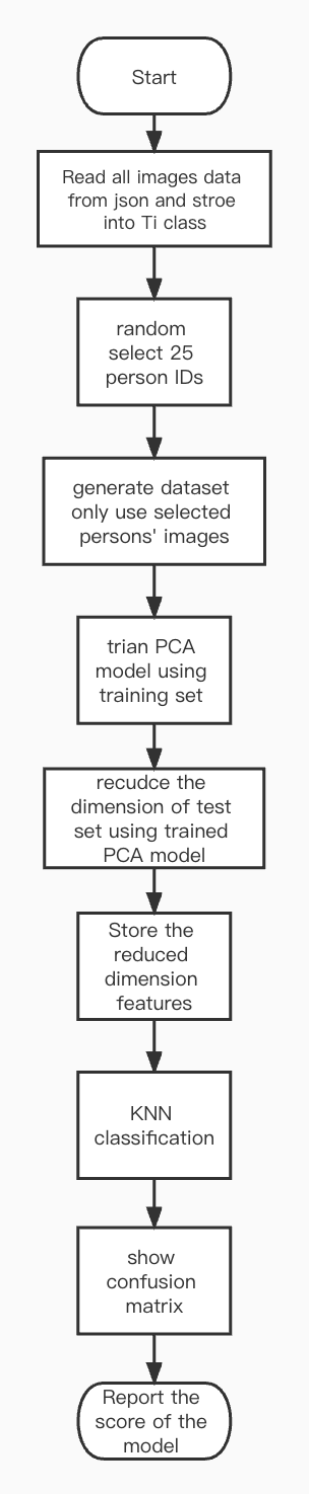
2.3 Discussion

It is obvious that the points of mine can be easily distinguished from the others. The points of mine close to each other but away from points belongs to other people. Because the variance of my images is different from the other images.

3.Part 2

This part is using PCA to reduce the dimension of the images and then using KNN to classify.

3.1 Process Flow

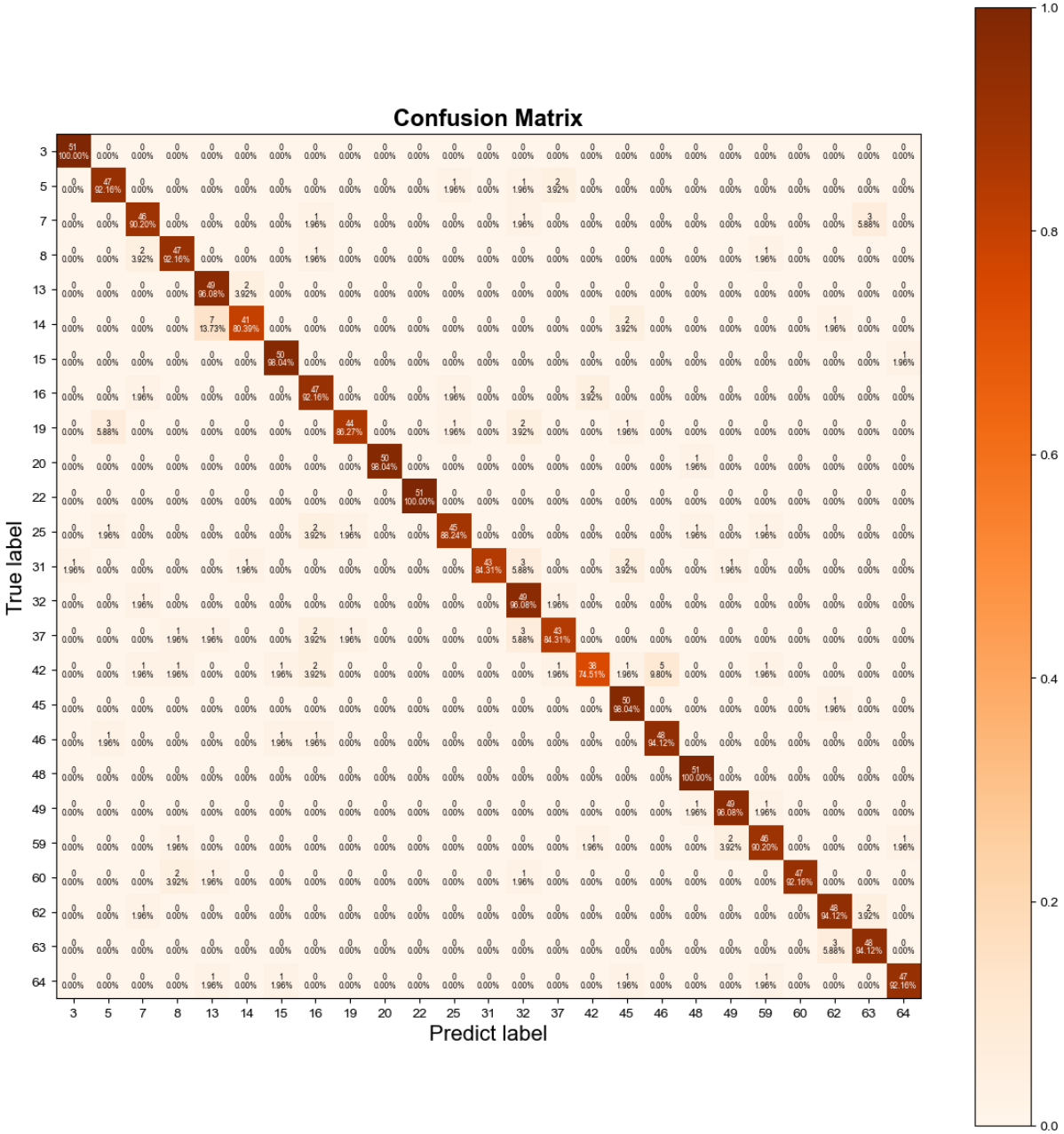


3.2 Result

3.2.1 CMU PIE Test Images Only

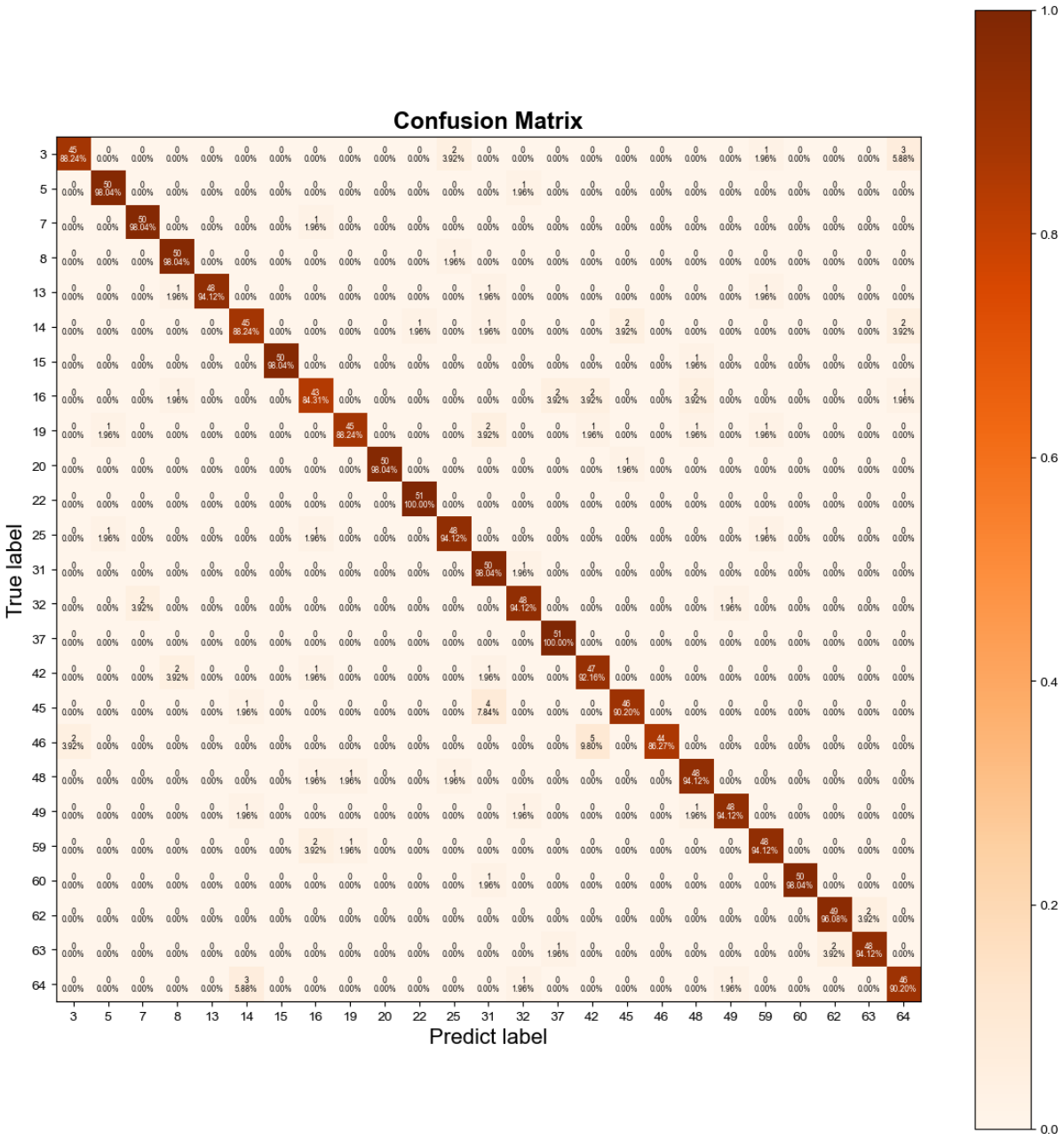
Randomly selected id is : [3, 5, 7, 8, 13, 14, 15, 16, 19, 20, 22, 25, 31, 32, 37, 42, 45, 46, 48, 49, 59, 60, 62, 63, 64]

3.2.1.1 40 Dimensions



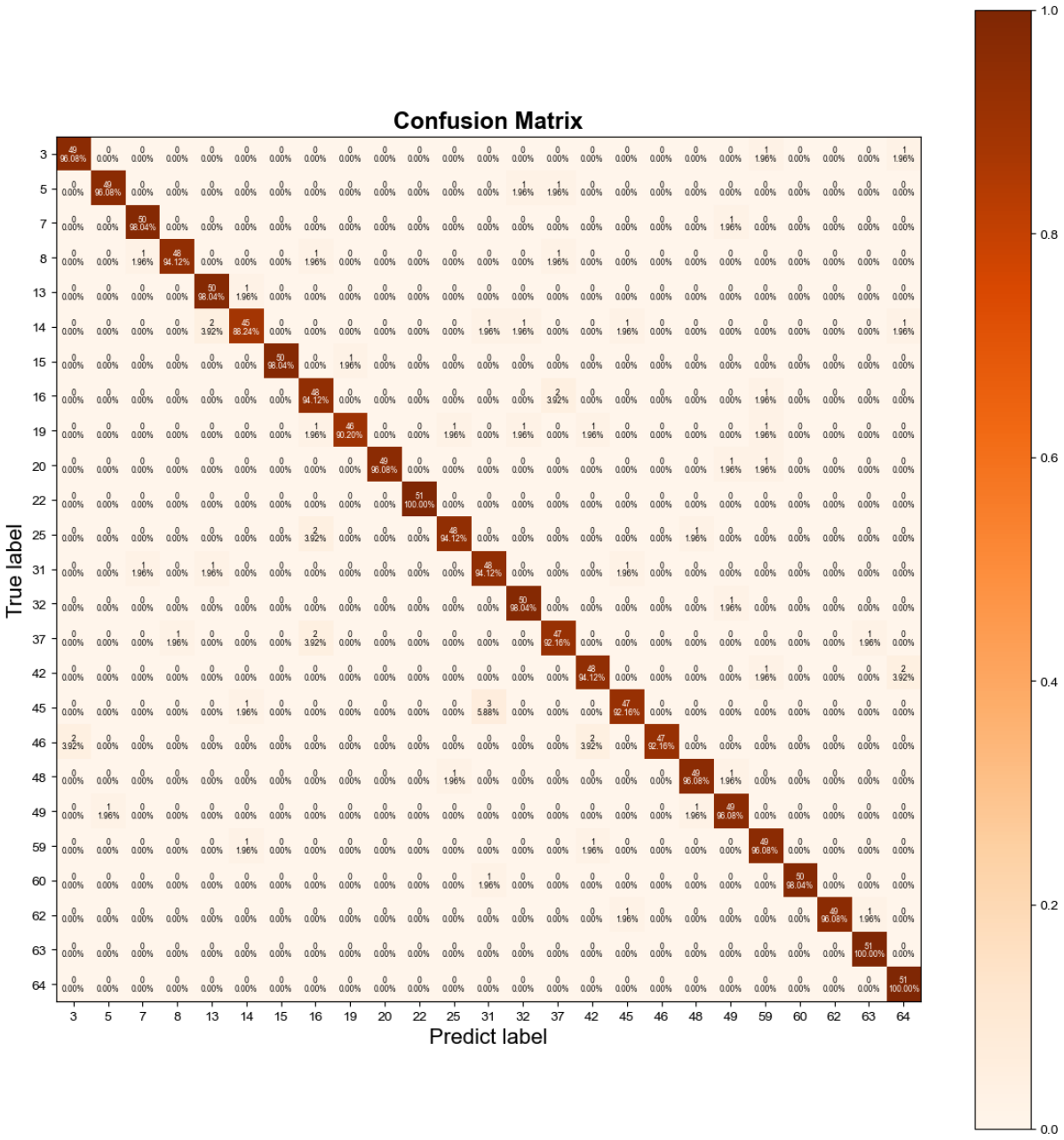
The accuracy of the model is 92.16%
The recall of the model is 96.23%
The precision of the model is 95.61%
The F1 of the model is 95.92%

3.2.1.2 80 Dimensions



The accuracy of the model is 93.96%
The recall of the model is 97.08%
The precision of the model is 96.69%
The F1 of the model is 96.89%

3.2.1.3 200 Dimensions

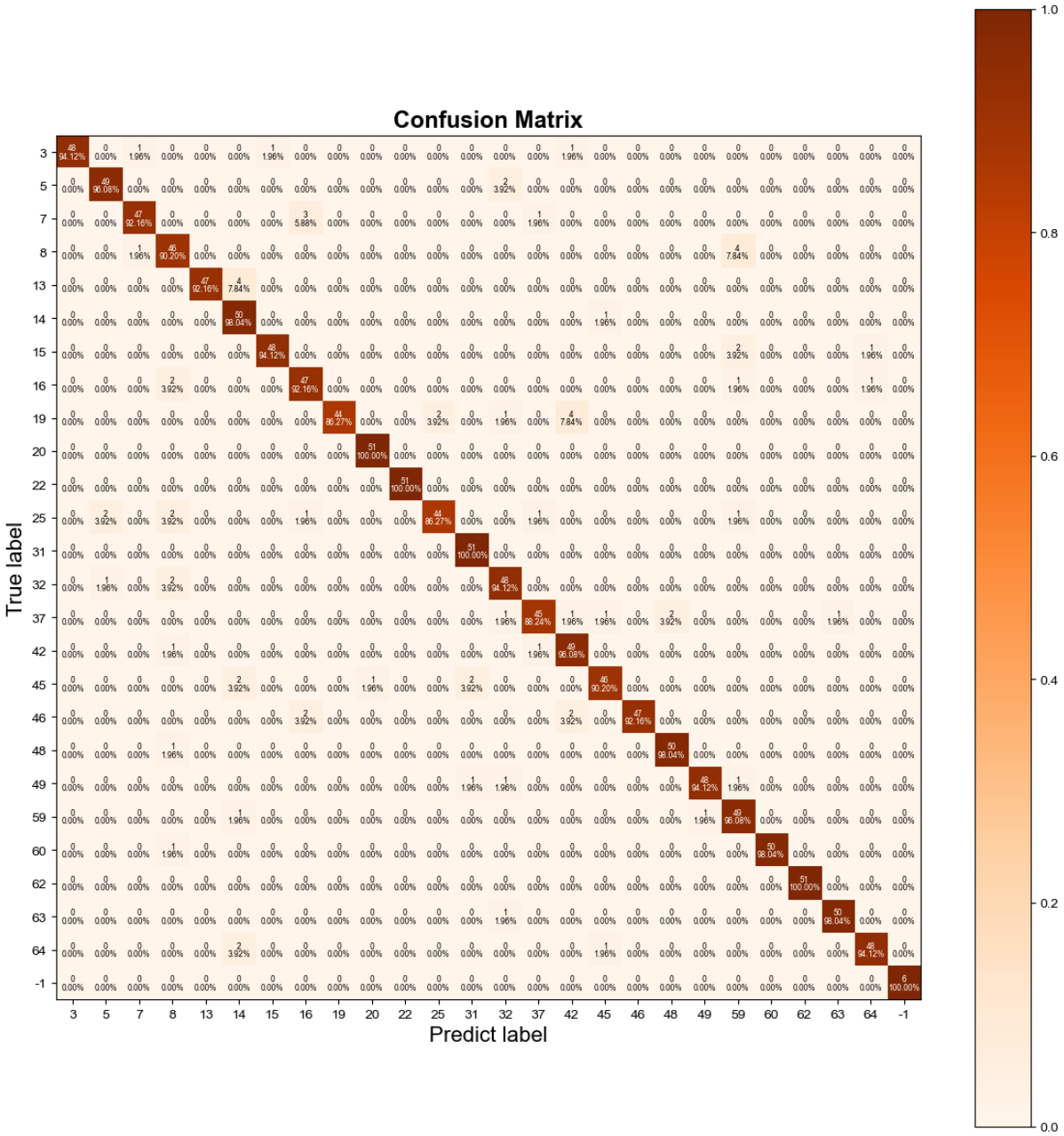


The accuracy of the model is 95.53%
The recall of the model is 97.52%
The precision of the model is 97.91%
The F1 of the model is 97.71%

3.2.2 CMU PIE test images and my own photo

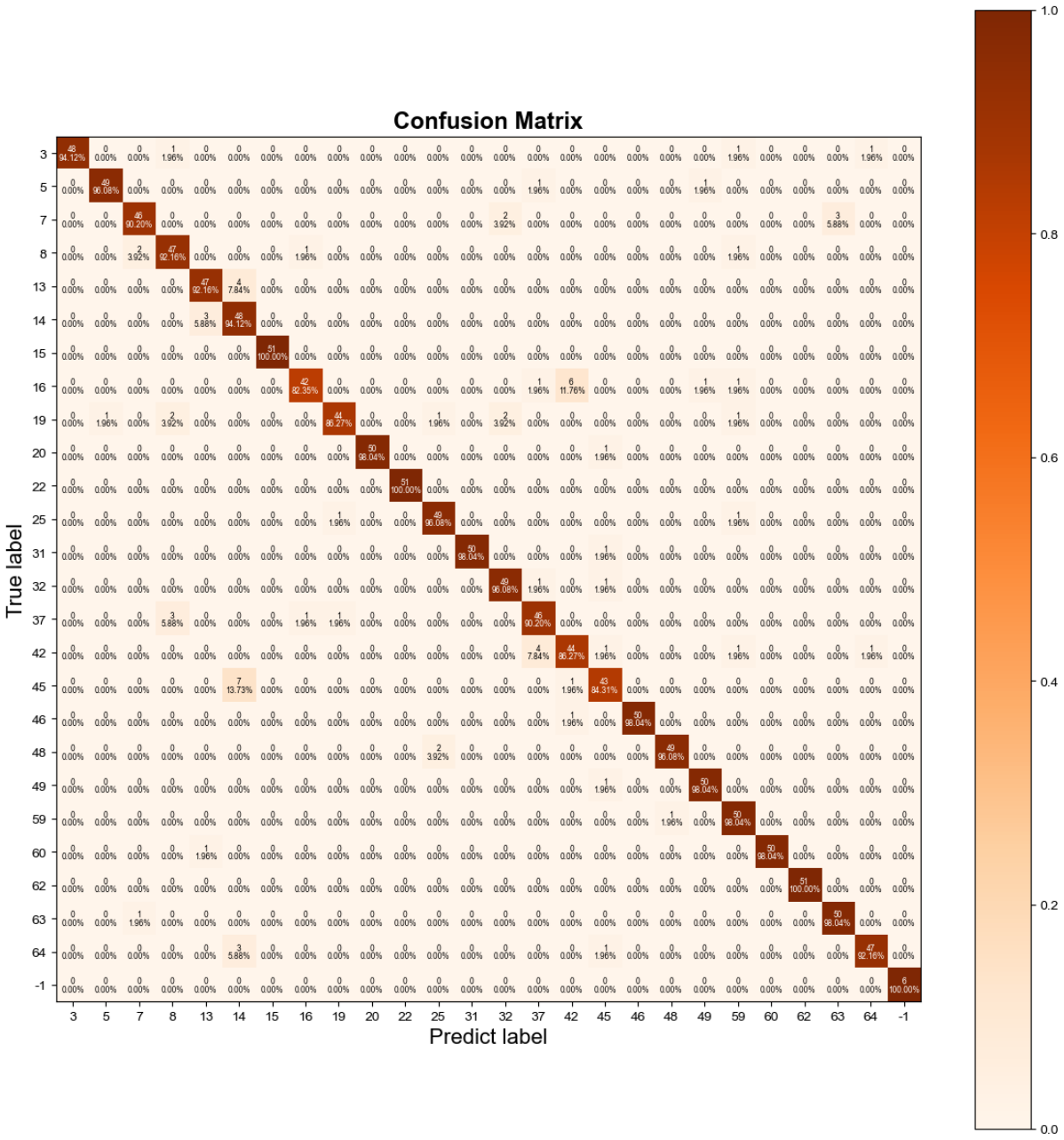
Label -1 refers to my own photos

3.2.2.1 40 Dimensions



The accuracy of my images is 100%
The accuracy of the model is 94.46%
The recall of the model is 96.96%
The precision of the model is 97.35%
The F1 of the model is 97.15%

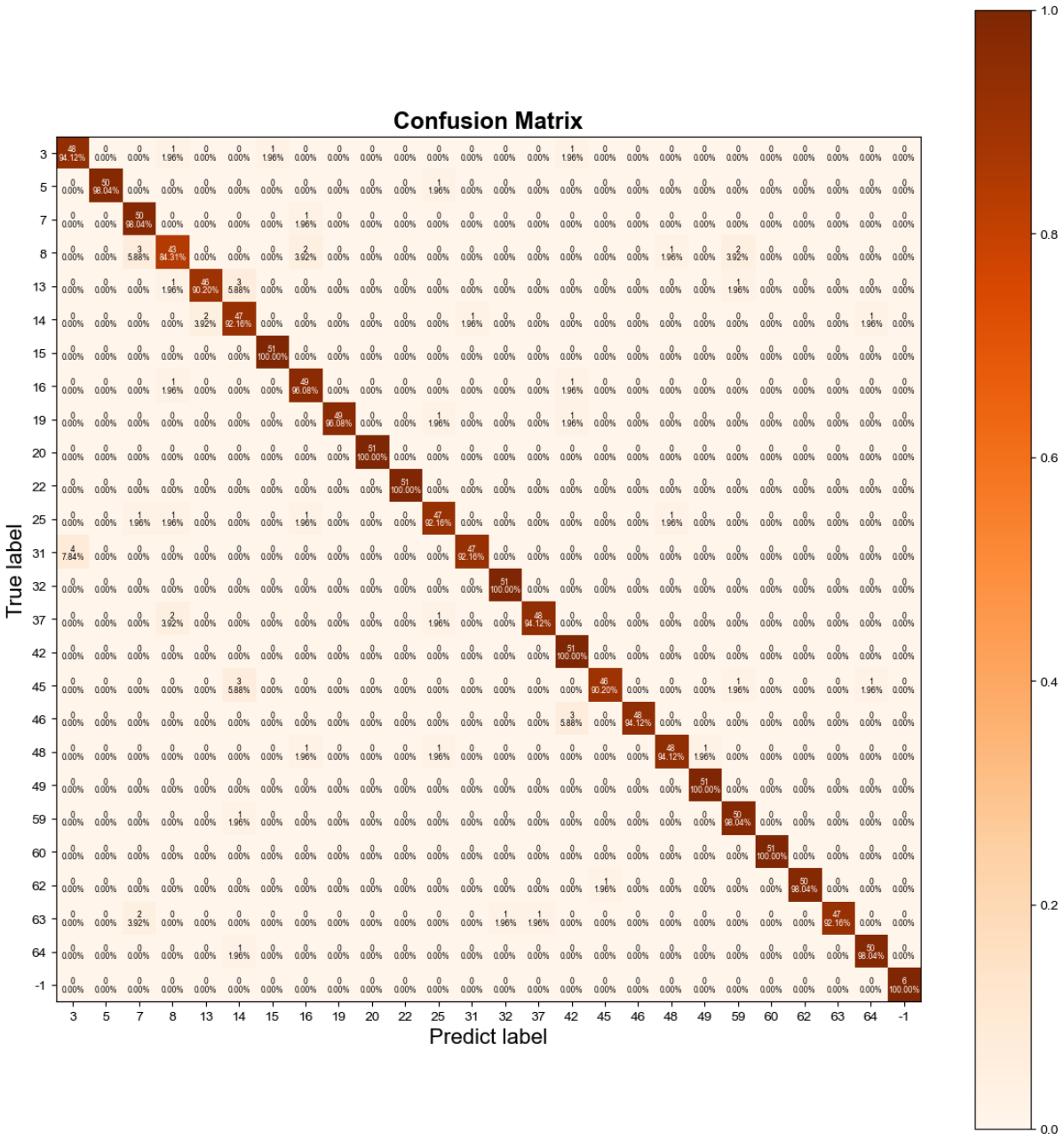
3.2.2.2 80 Dimensions



The accuracy of my images is 100%

The accuracy of the model is 94.22%
The recall of the model is 97.03%
The precision of the model is 97.03%
The F1 of the model is 97.03%

3.2.2.3 200 Dimensions



The accuracy of my images is 100%

The accuracy of the model is 95.71%
The recall of the model is 98.16%
The precision of the model is 97.46%
The F1 of the model is 97.81%

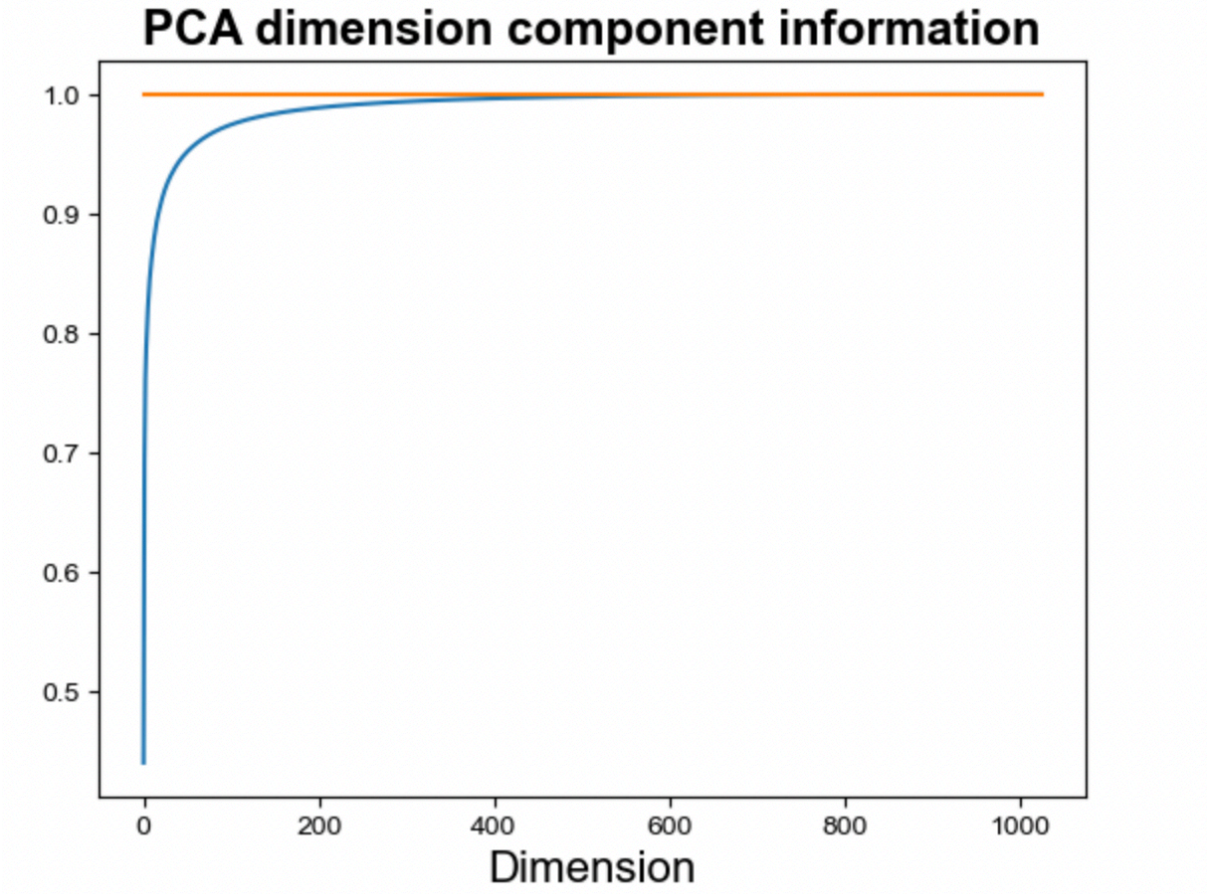
3.3 Discussion

3.3.1 Accuracy and Dimension

In CMU PIE dataset

Dimension	40	80	200
Accuracy	**92.16%**	**93.96%**	**95.53%**
Recall	96.23%	97.08%	97.52%
Precision	95.61%	96.69%	97.91%
F1	95.92%	96.89%	97.71%

As the increase of the dimensions, the accuracy of the model increases as well. Because more dimensions brings more information of the sample.



As the figure shows, the information of the first 400 dimensional component nearly equals to 1. So when the dimension is 40 and 80, there is still many information loss.

3.3.2 Accuracy and Training Set Scale

In CMU PIE dataset

Training set scale	30%	50%	70%
Accuracy	**80.56%**	**90.68%**	**95.53%**
Recall	89.58%	95.75%	97.52%
Precision	88.89%	94.49%	97.91%
F1	89.23%	95.11%	97.71%

It is obvious that the more training set more accurate. But it may also lead to overfitting, if the training set is not very reasonable.

3.3.3 Accuracy and KNN

In CMU PIE dataset, the control group is 40 dimensions with 70% training set using NN.

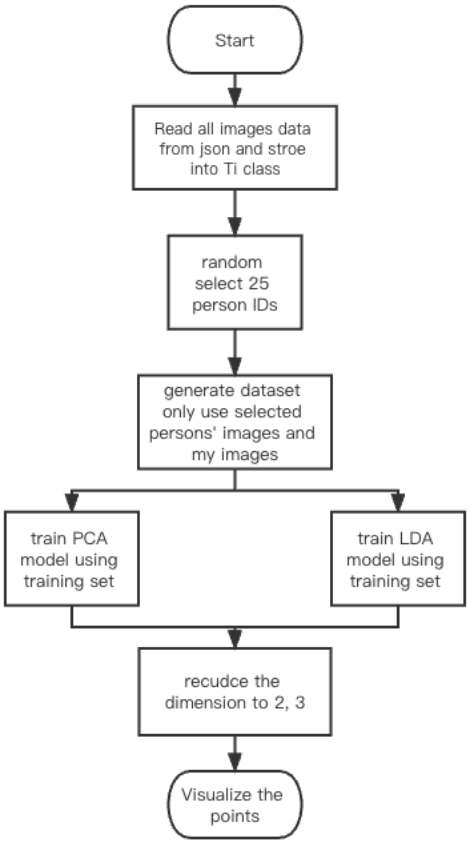
K of KNN	1	3	5	10
Accuracy	**92.16%**	**88.31%**	**84.24%**	**77.96%**
Recall	96.23%	93.83%	91.02%	88.28%
Precision	95.61%	93.76%	91.87%	86.96%
F1	95.92%	93.79%	91.44%	87.62%

As the increase of K, the accuracy of the model decreases. It shows that as the increase of K, the model becomes underfitting. So the best K is 1. (Hopes my KNN algorithm is correct)

4.Part 3

This part is using LDA to reduce the dimension of the images data and plotting them in 2D and 3D figures.

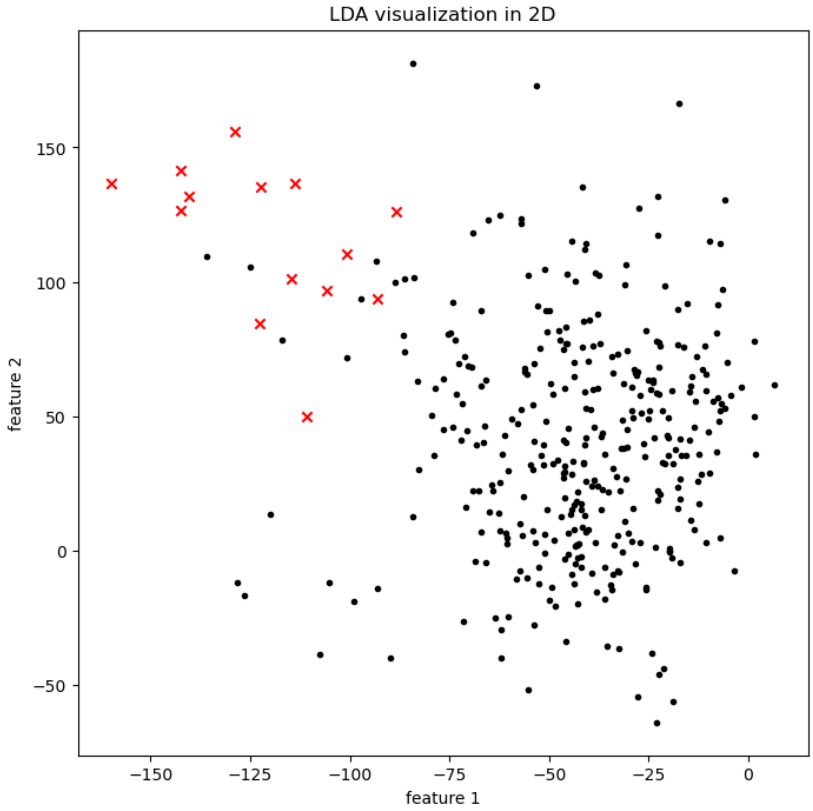
4.1 Process Flow

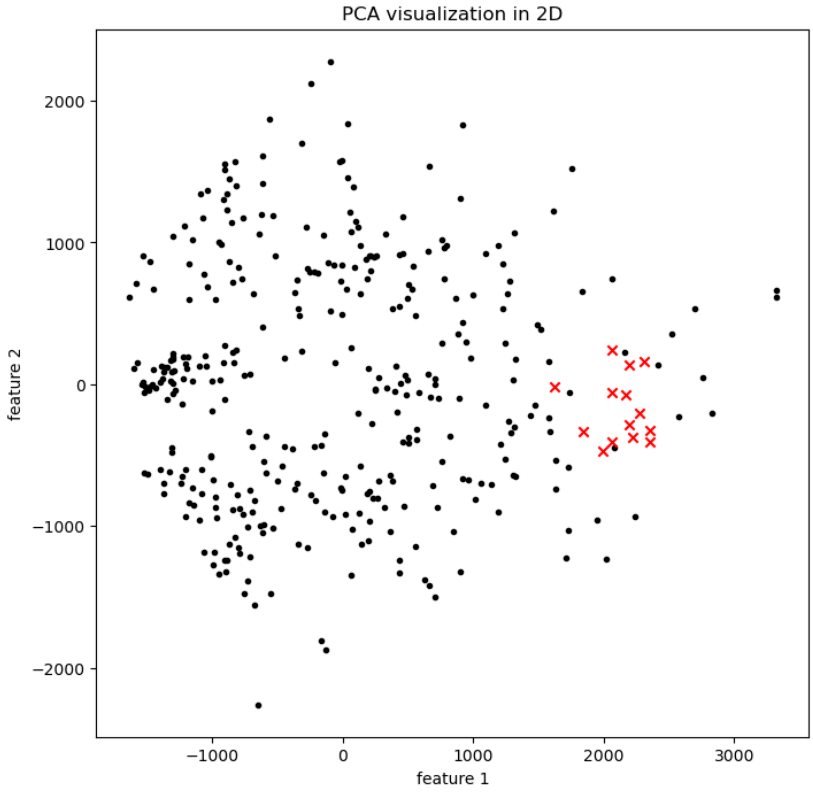


4.2 Result

Randomly selected id is : [3, 5, 6, 9, 12, 13, 14, 17, 21, 22, 24, 25, 30, 31, 32, 36, 37, 40, 41, 42, 51, 53, 57, 64, 65]

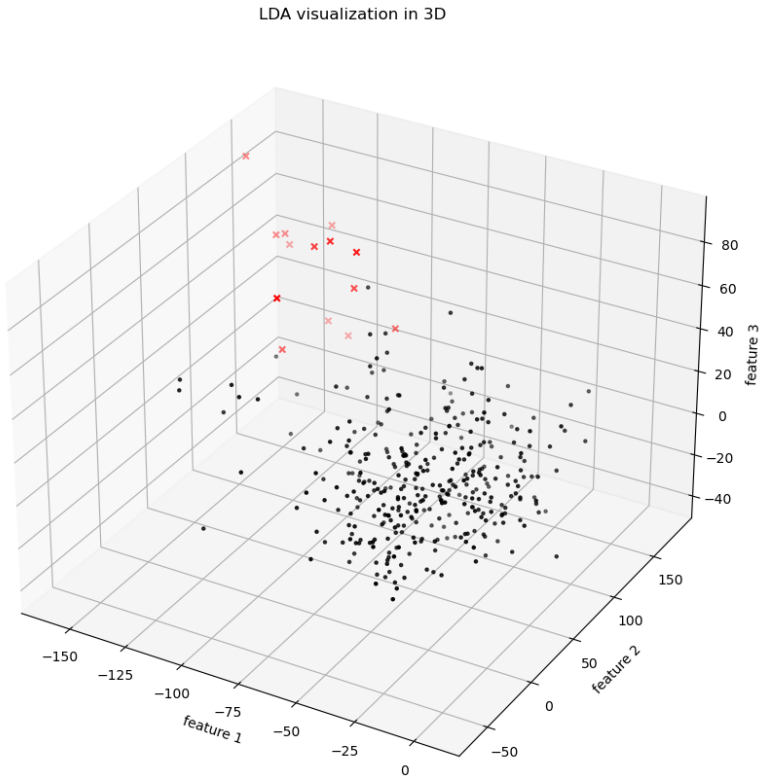
4.2.1 2D Visualization

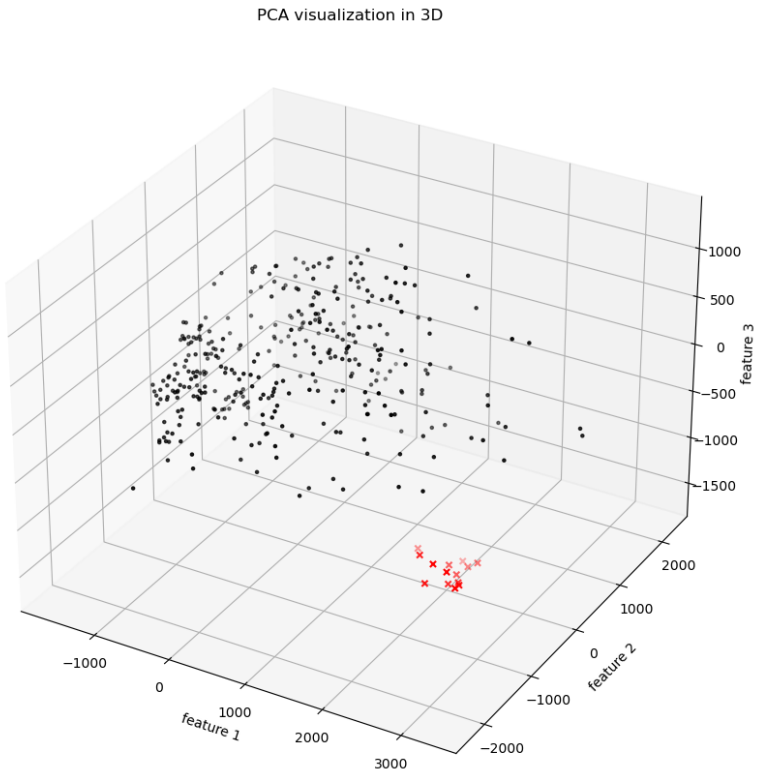




The red x points refer to my images.

4.2.2 3D Visualization





The red x points refer to my images.

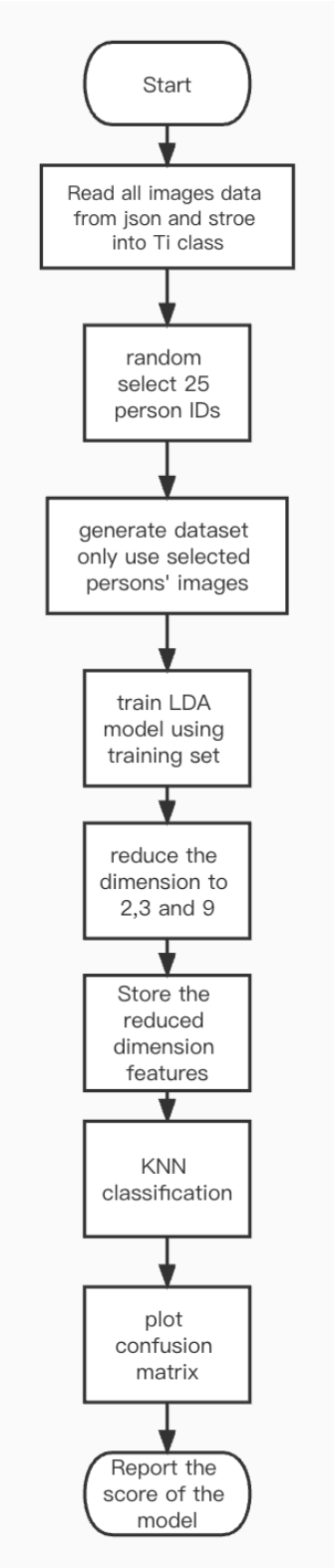
4.3 Discussion

It seems that the feature reduced from LDA is less separatable than from PCA. In my opinion, the reason is that LDA separate samples via average while PCA separate samples via variance. The average value of different person is similar but the variance is much different. So PCA performs better.
the variance of the average value of all samples 971.8932870184626
the variance of the variance value of all samples 1231701.6969136673
It shows that the difference of average value of every image is not very much, but the difference of variance value of every image is very large.

5.Part 4

This part is using LDA to reduce the dimension of the images and then using KNN to classify.

5.1 Process Flow

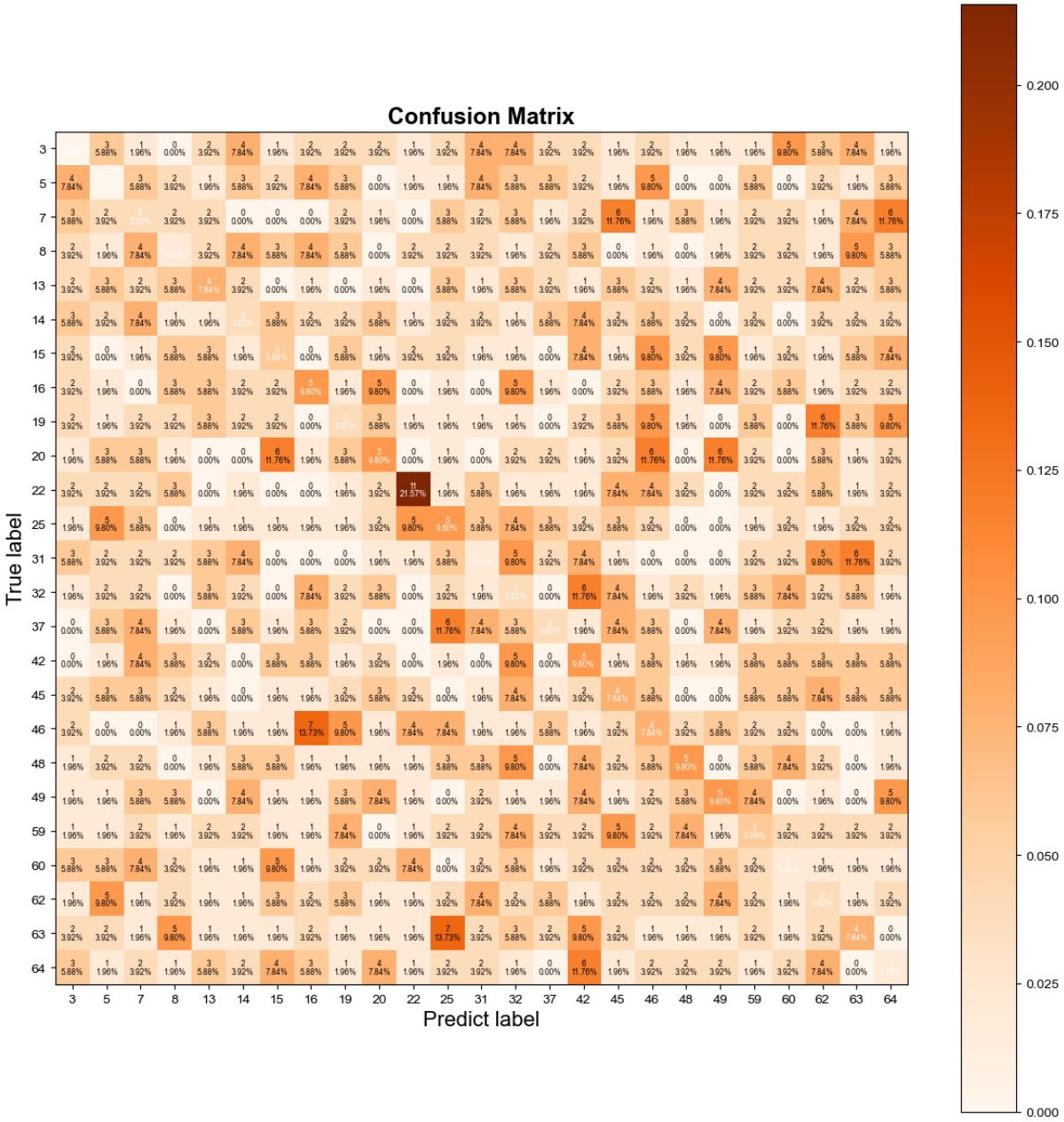


5.2 Result

Randomly selected id is : [3, 4, 8, 12, 15, 16, 18, 20, 21, 24, 25, 28, 31, 36, 40, 41, 43, 45, 46, 47, 53, 54, 58, 61, 64]. Same to Part 2, using PCA to reduce dimensions.

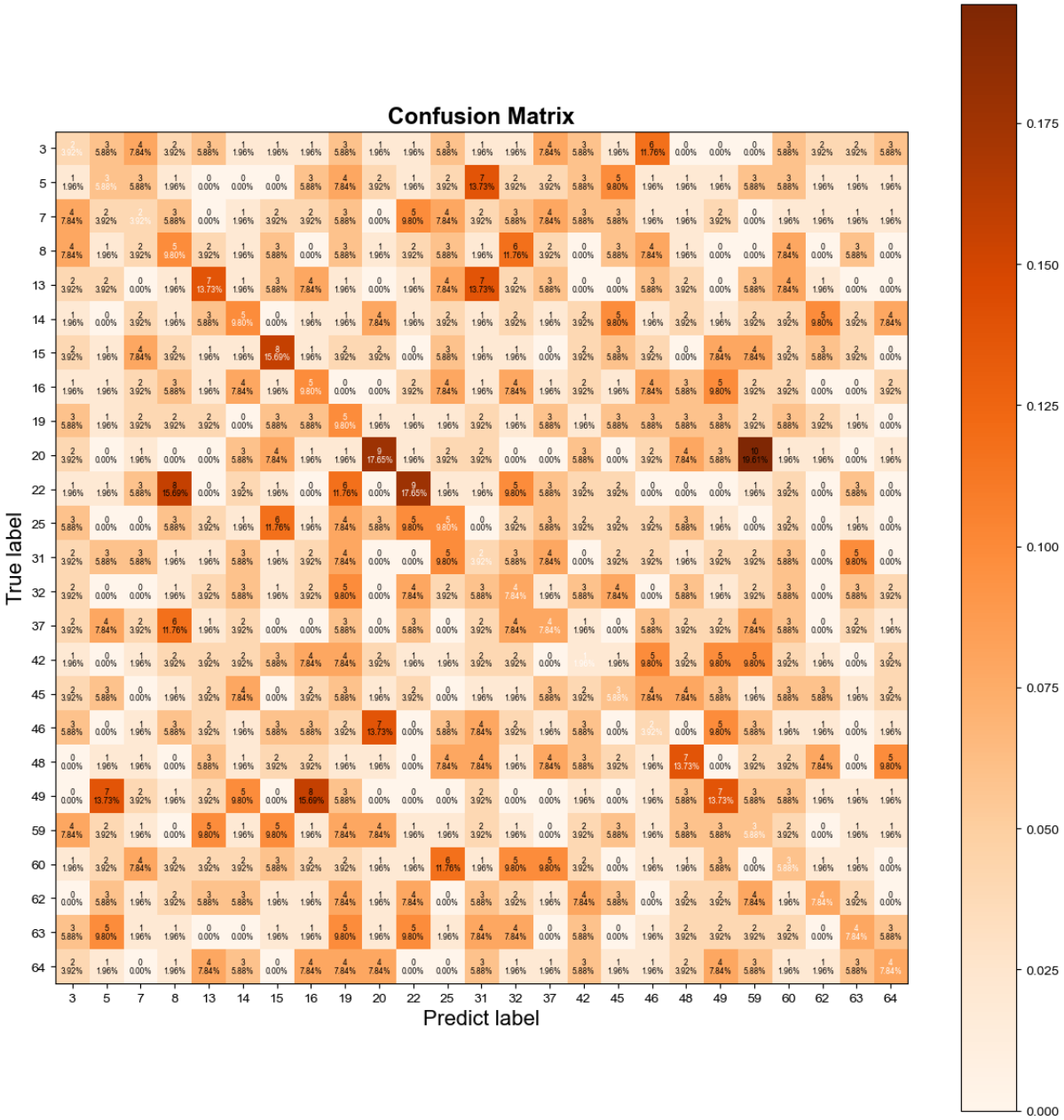
5.2.1 CMU PIE Images Only

5.2.1.1 2 Dimension



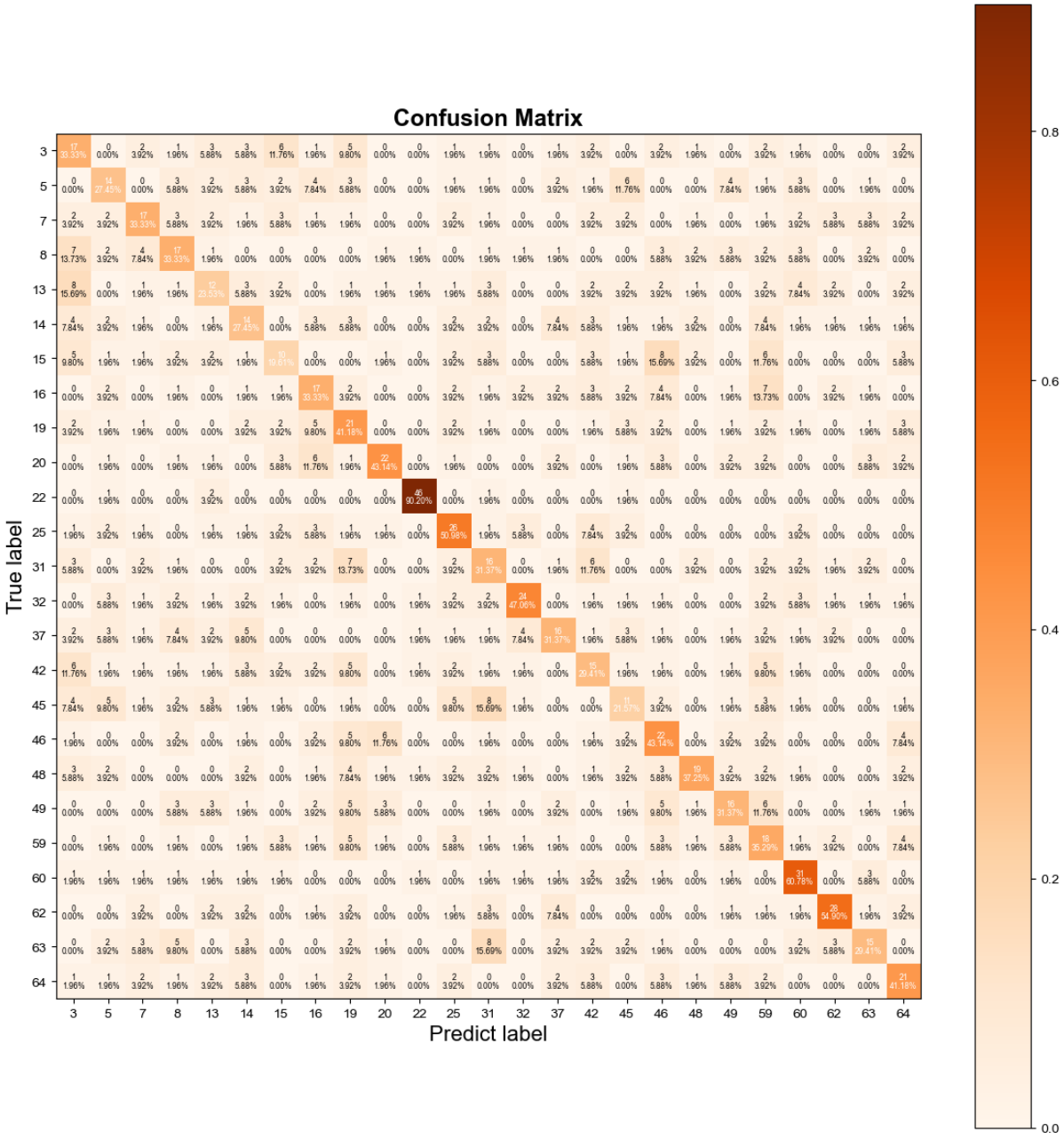
The accuracy of the model is 6.20%
The recall of the model is 11.55%
The precision of the model is 11.79%
The F1 of the model is 11.67%

5.2.1.2 3 Dimensions



The accuracy of the model is 8.86%
The recall of the model is 16.52%
The precision of the model is 16.05%
The F1 of the model is 16.28%

5.2.1.3 9 Dimensions

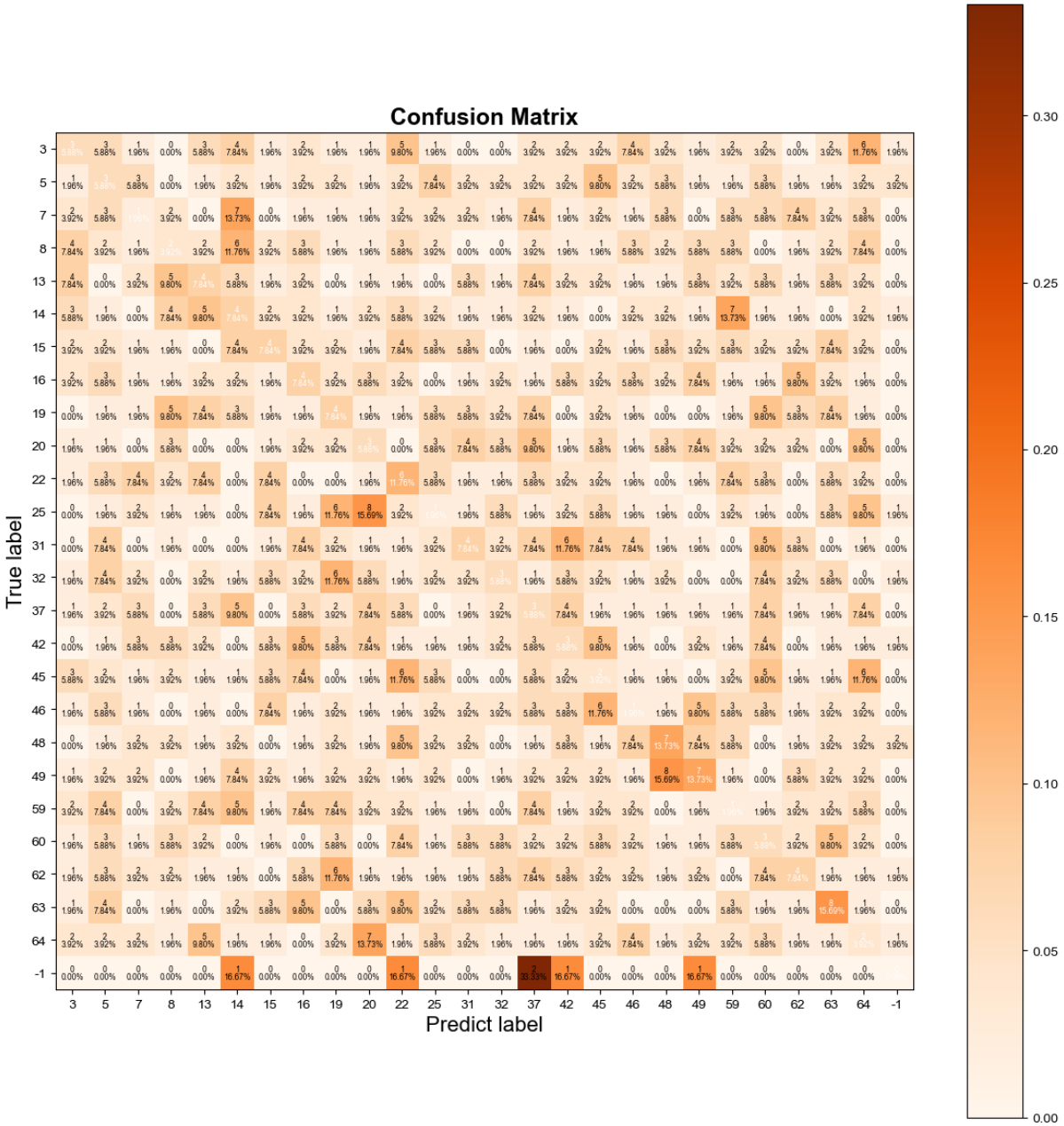


The accuracy of the model is 38.04%
The recall of the model is 56.33%
The precision of the model is 53.95%
The F1 of the model is 55.11%

5.2.2 CMU PIE test images and my own photo

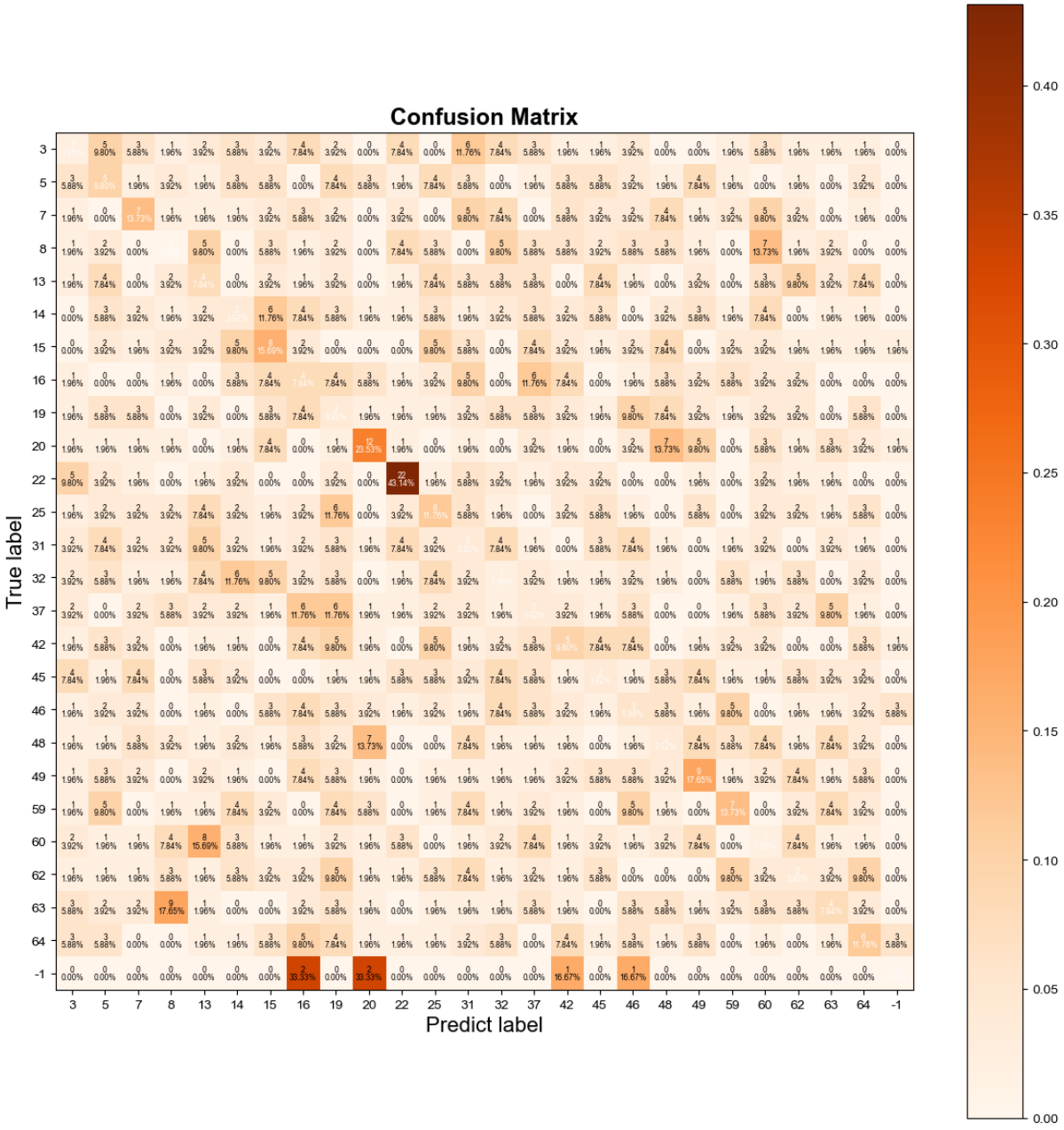
Label -1 refers to my images

5.2.2.1 2 Dimensions



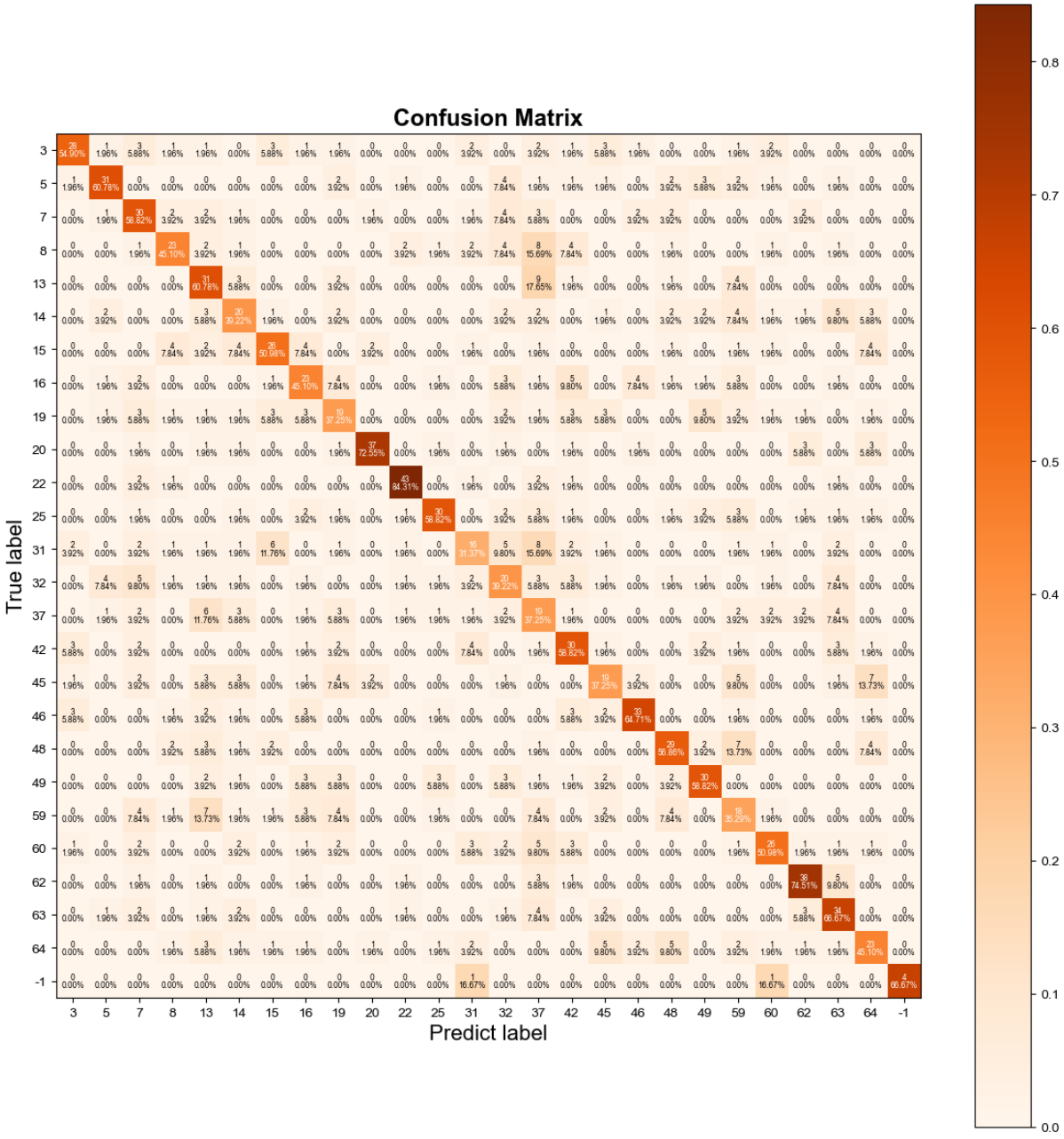
The accuracy of my images is 0%
The accuracy of the model is 6.79%
The recall of the model is 12.55%
The precision of the model is 12.89%
The F1 of the model is 12.72%

5.2.2.2 3 Dimensions



The accuracy of my images is 0%
The accuracy of the model is 9.29%
The recall of the model is 16.76%
The precision of the model is 17.25%
The F1 of the model is 17.00%

5.2.2.3 9 Dimensions



The accuracy of my images is 66.67%

The accuracy of the model is 53.08%

The recall of the model is 68.34%

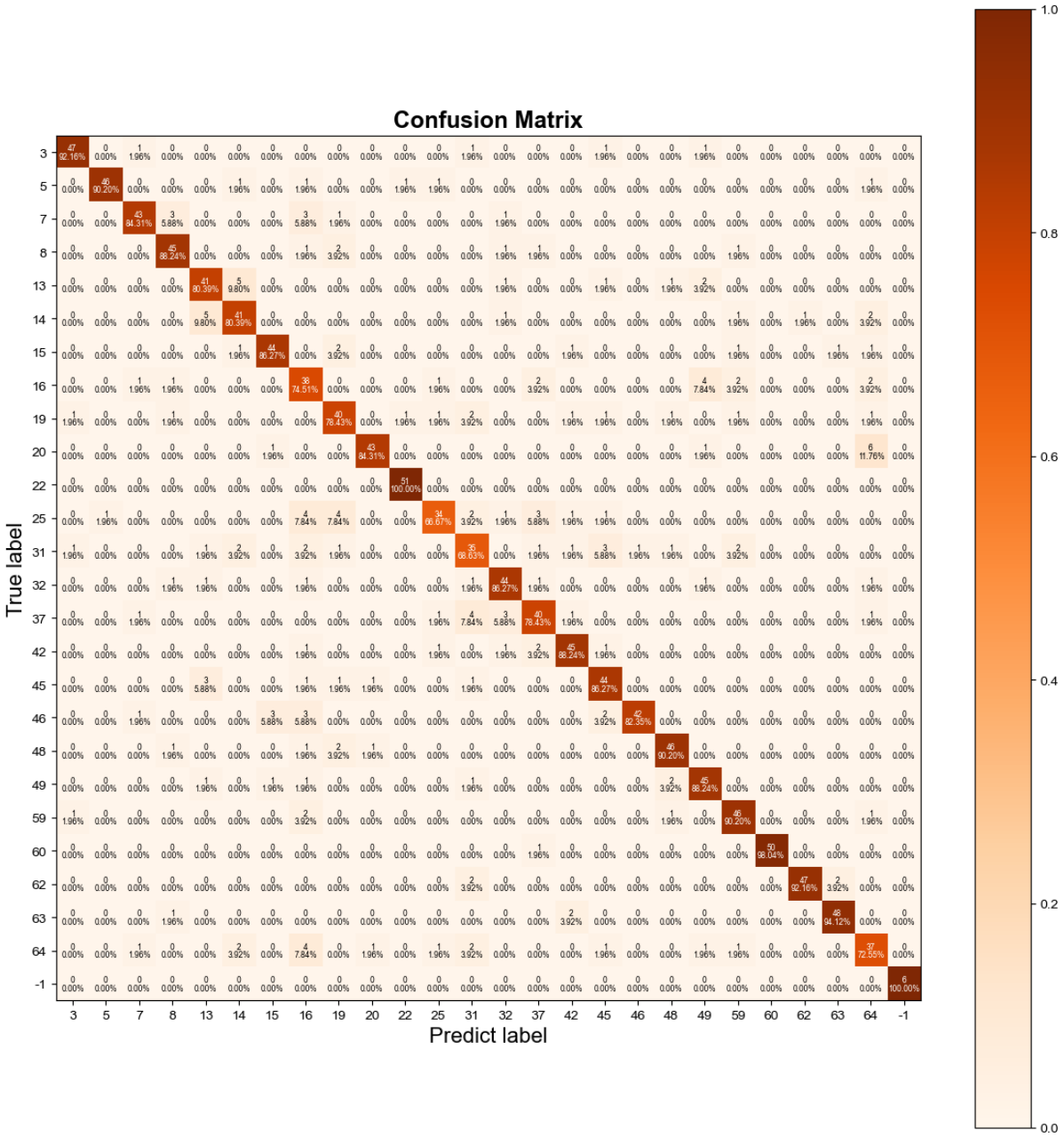
The precision of the model is 70.39%

The F1 of the model is 69.35%

5.3 Discussion

5.3.1 Accuracy and Dimension

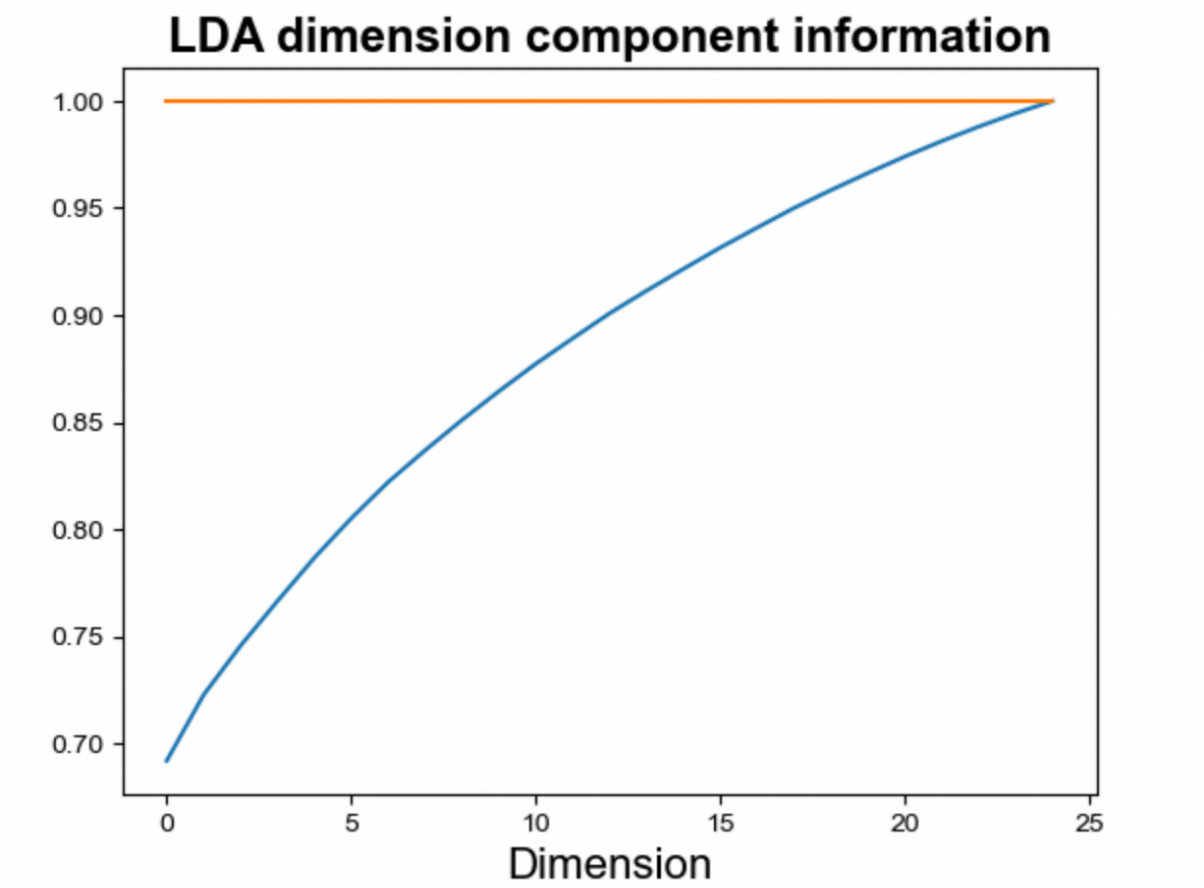
Add the dimension to maximum, 25. Based on CMU PIE and me data set, which contains 26 person in total.



The accuracy of my images is 100%
The accuracy of the model is 84.93%
The recall of the model is 91.81%
The precision of the model is 91.89%
The F1 of the model is 91.85%

In CMU PIE dataset + my own photos

Dimension	2	3	9	25
Accuracy	**6.79%**	**9.29%**	**53.08%**	**84.93%**
Recall	12.55%	16.76%	68.34%	91.81%
Precision	12.89%	17.25%	70.39%	91.89%
F1	12.72%	17.00%	69.35%	91.85%



This figure shows the component information of LDA dimensionality reduction. The first 2 or 3 dime nsions only contain 70% information of the image. So the performance is bad. But if using all 25 dime nsions, it will perform better.

5.3.2 Accuracy and KNN

In CMU PIE dataset + my own photos, using dimensions of feature reduced to 9 data set

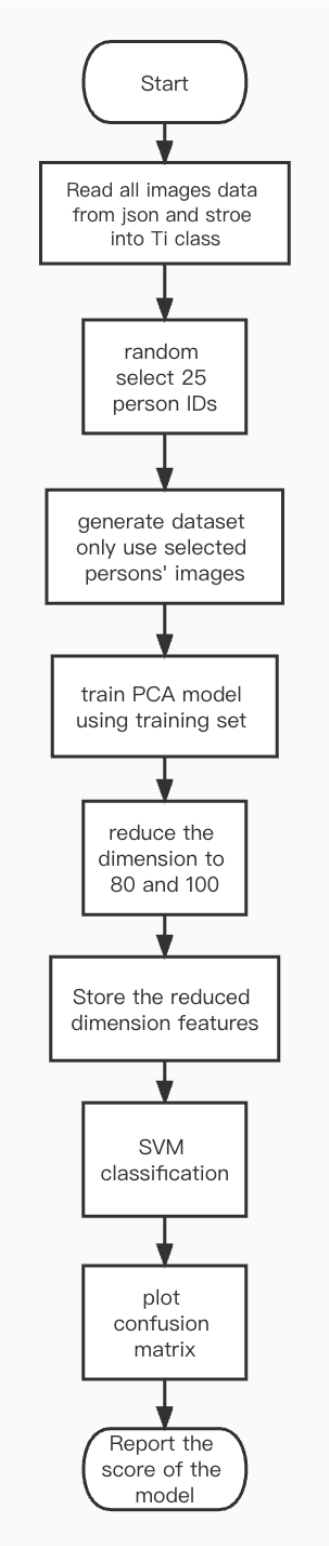
K of KNN	1	3	5	10
Accuracy	**53.08%**	**45.12%**	**36.69%**	**37.16%**
Recall	68.34%	62.15%	54.71%	54.65%
Precision	70.39%	62.22%	52.69%	53.72%
F1	69.35%	62.18%	53.68%	54.18%

The model is increasingly underfitting as K grows.

6.Part 5

This part is using PCA to reduce the dimensionality of the images and using SVM to classify.

6.1 Process Flow

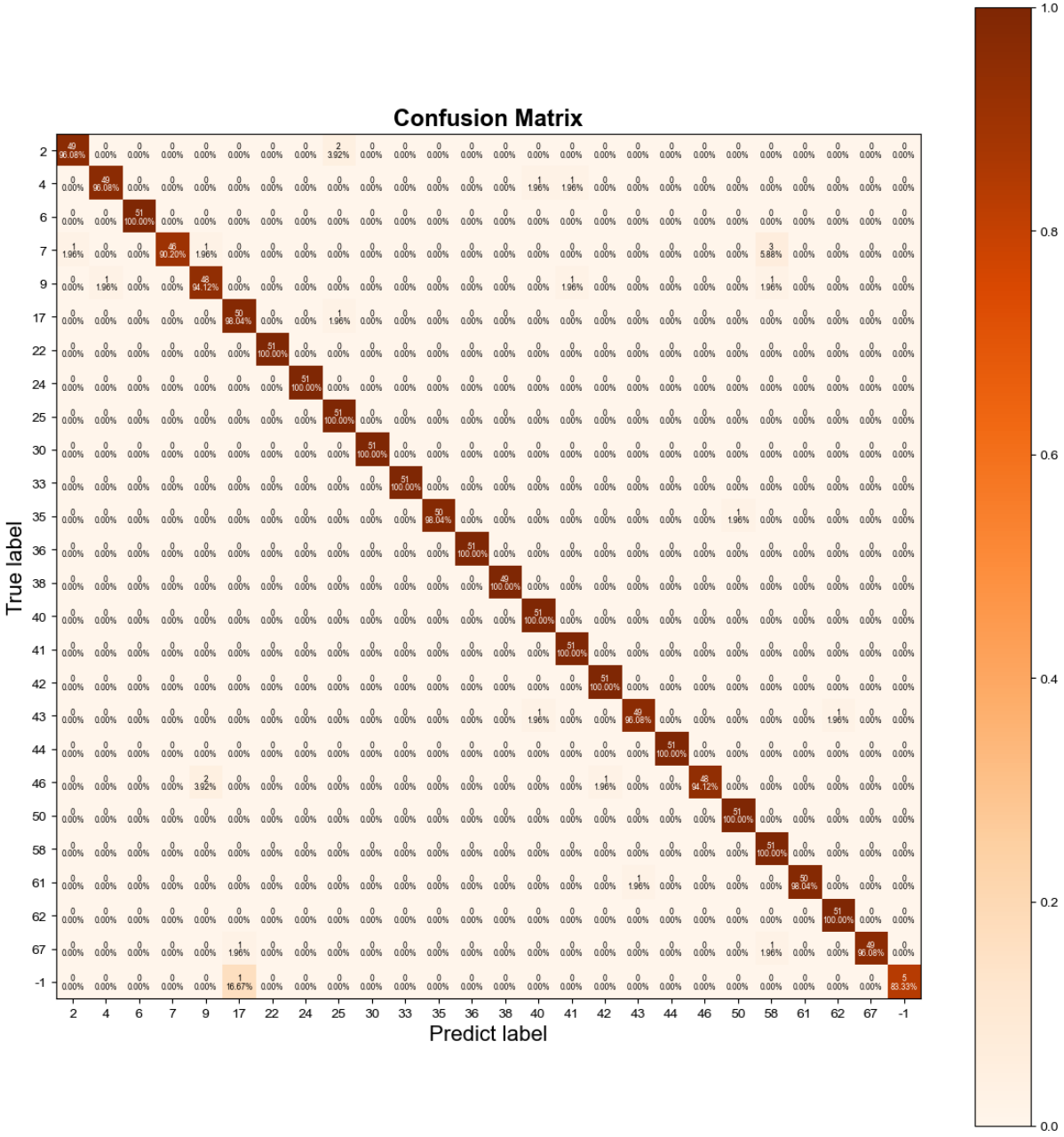


6.2 Result

Random selected id is : [2, 7, 9, 11, 14, 15, 16, 17, 19, 20, 23, 27, 28, 33, 35, 37, 38, 39, 43, 44, 48, 65, 66, 67, 68]

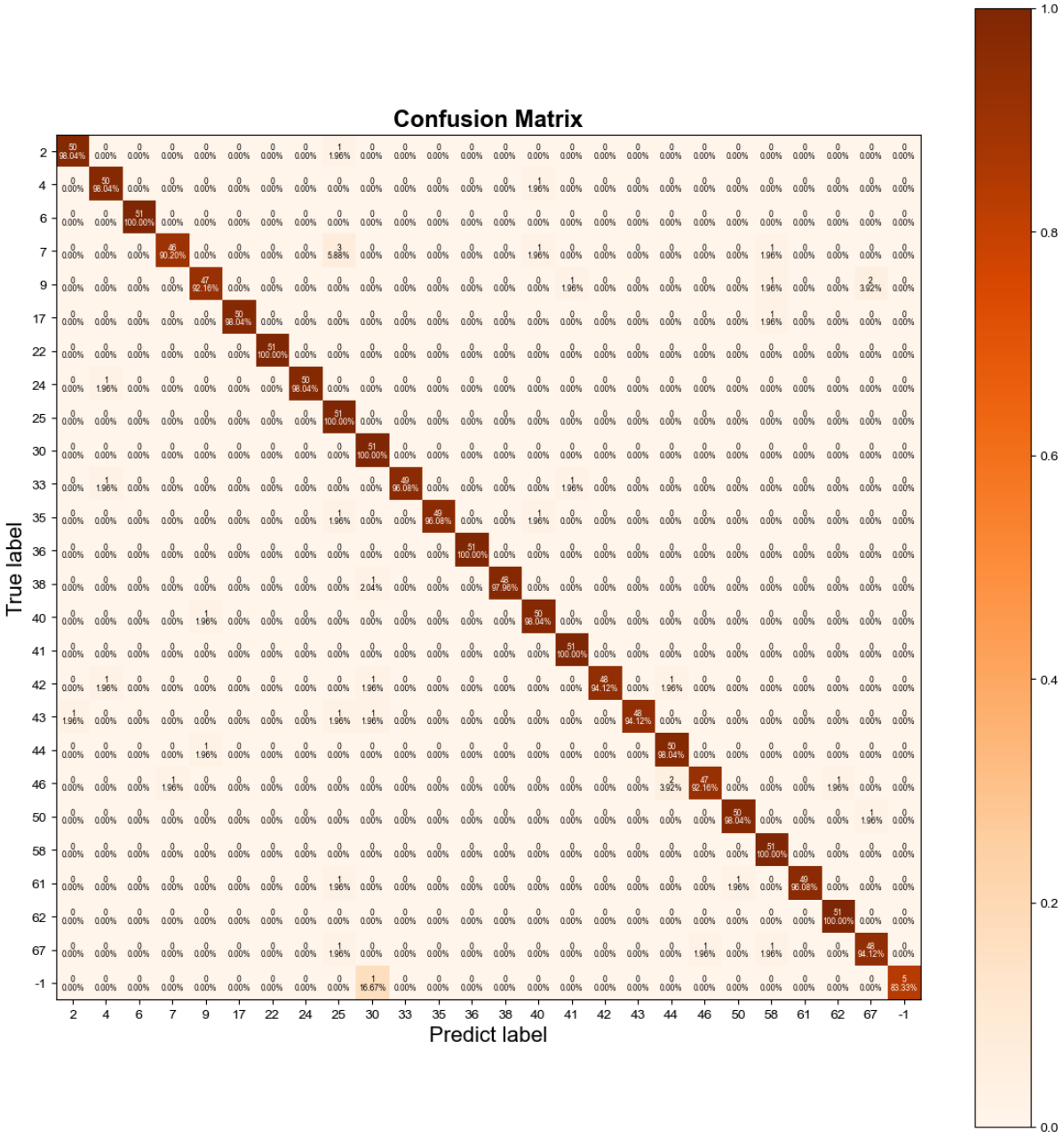
6.2.1 80 Dimensions

6.2.1.1 C = 1



The accuracy of my images is 83.3%
The accuracy of the model is 98.20%
The recall of the model is 98.98%
The precision of the model is 99.21%
The F1 of the model is 99.09%

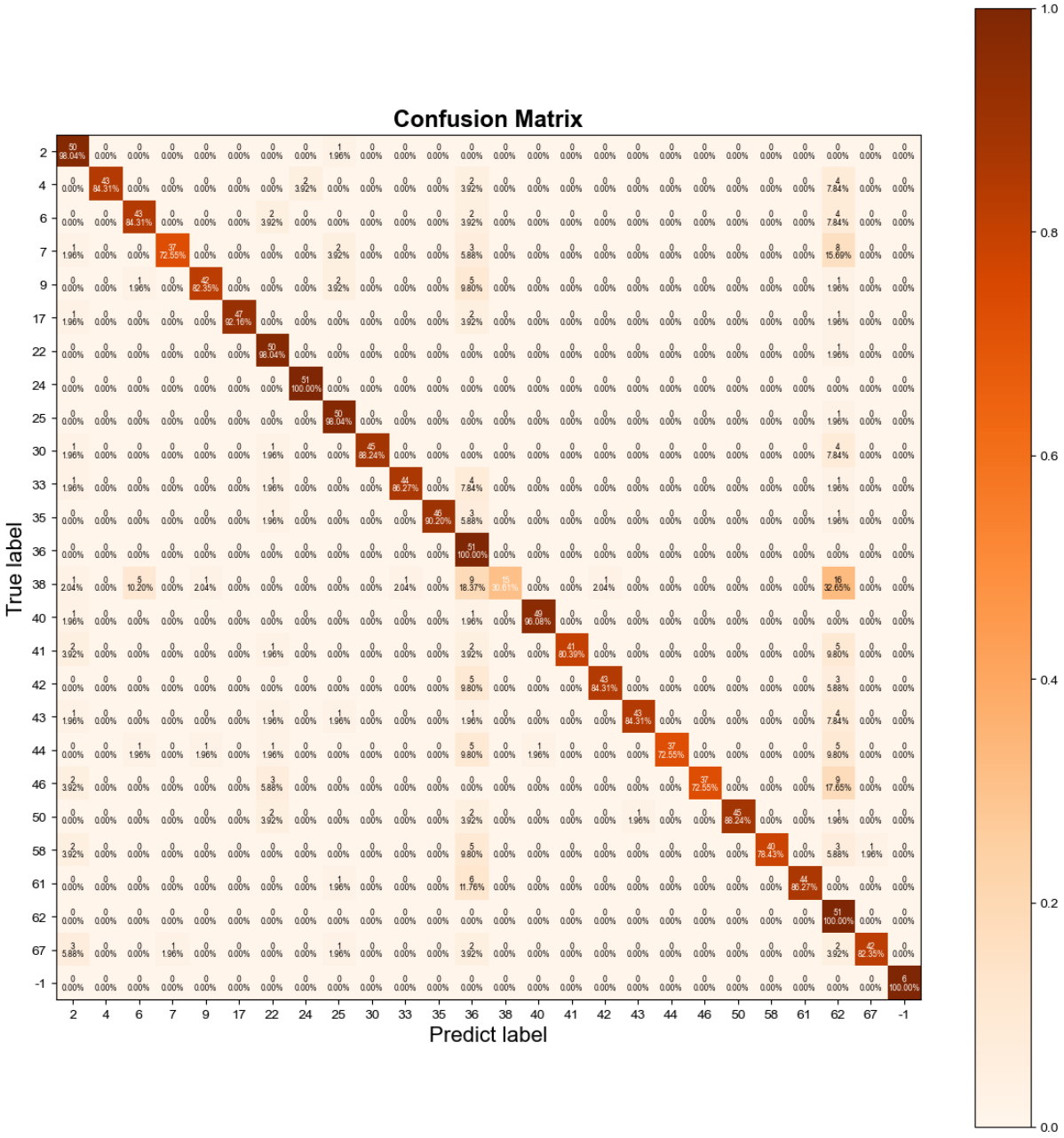
6.2.1.2 C = 0.1



The accuracy of my images is 83.3%

The accuracy of the model is 97.11%
The recall of the model is 98.65%
The precision of the model is 98.42%
The F1 of the model is 98.53%

6.2.1.3 C = 0.01



The accuracy of my images is 100%

The accuracy of the model is 85.38%

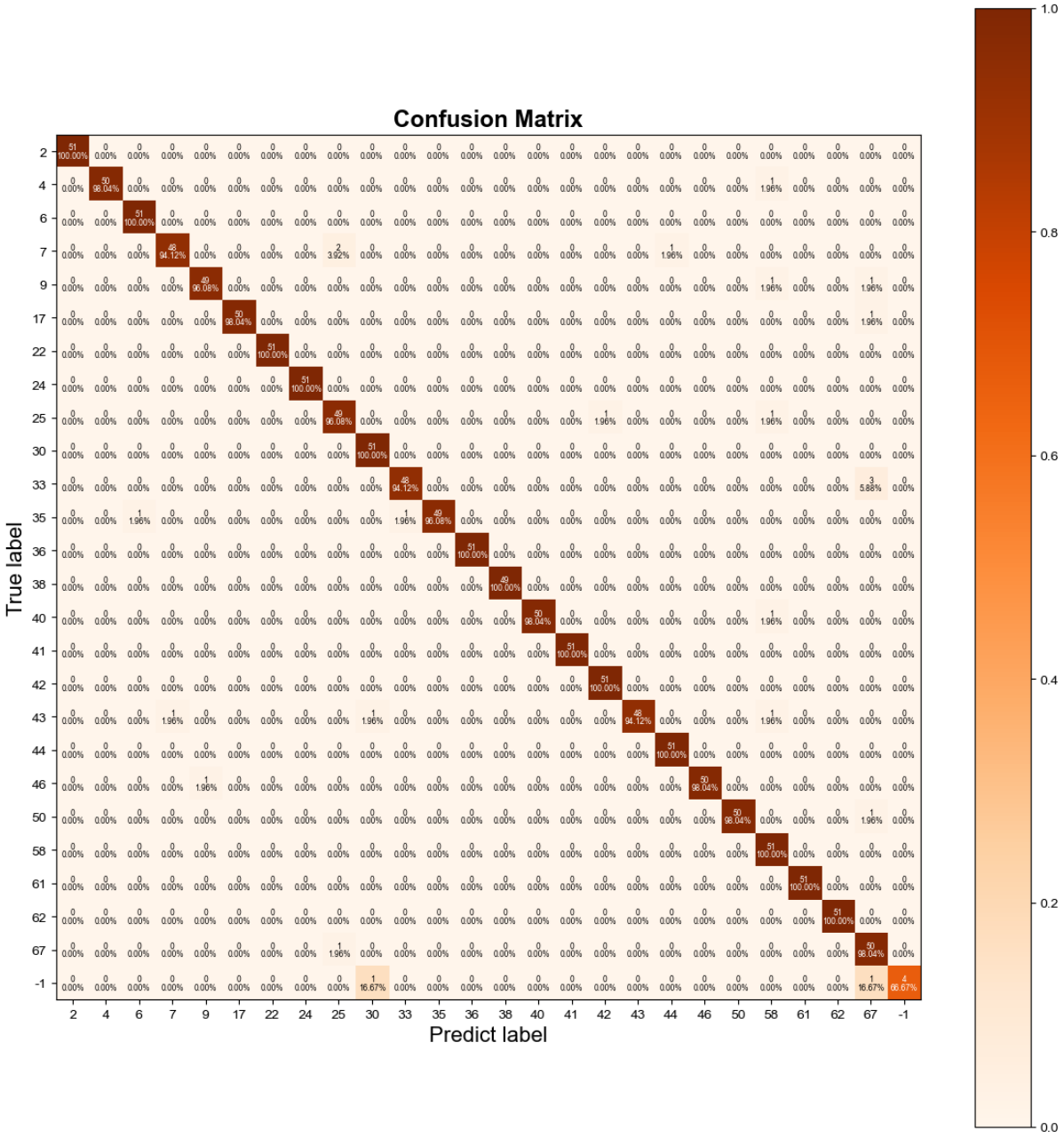
The recall of the model is 91.30%

The precision of the model is 92.94%

The F1 of the model is 92.11%

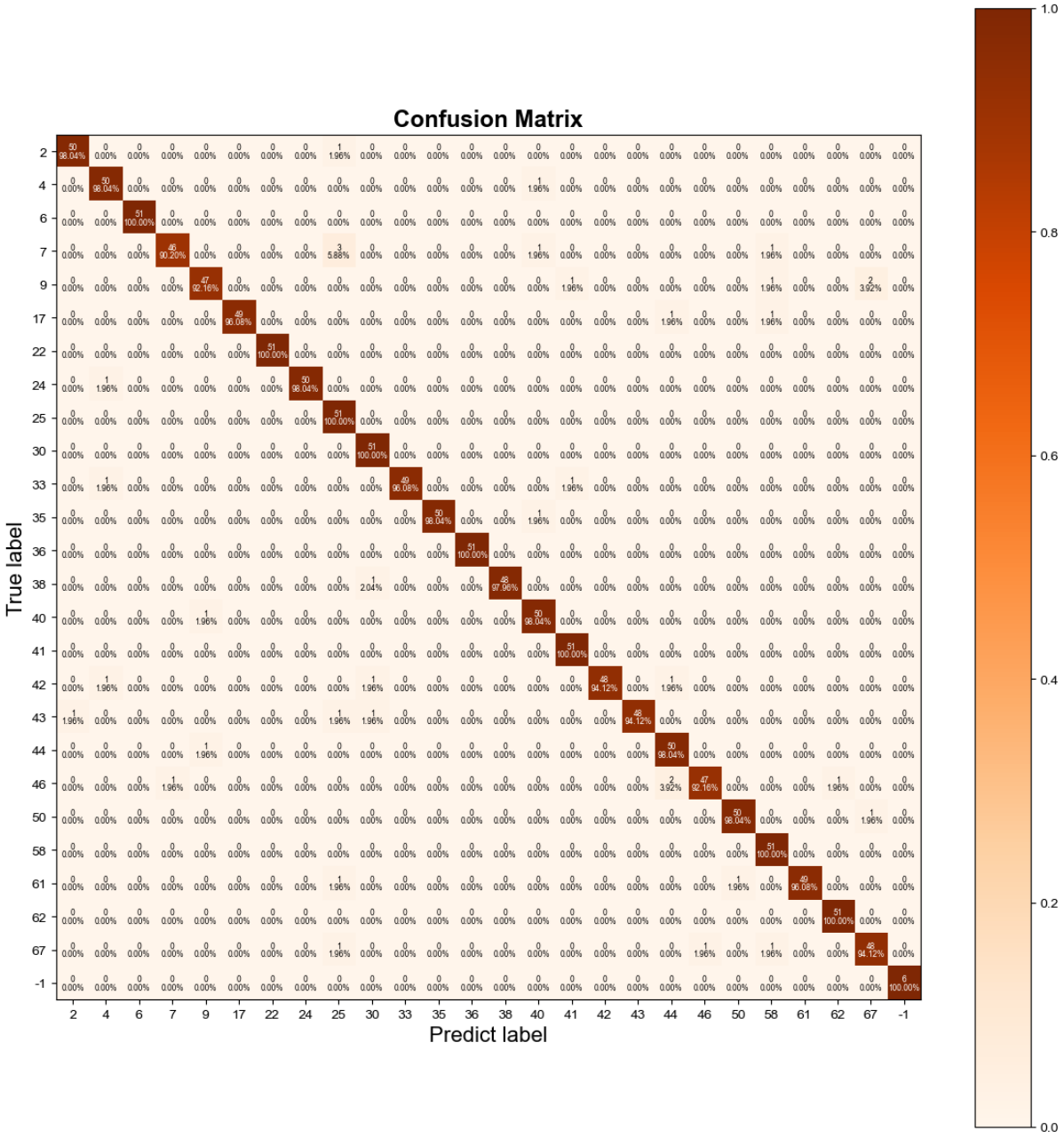
6.2.2 200 Dimensions

6.2.2.1 C = 1



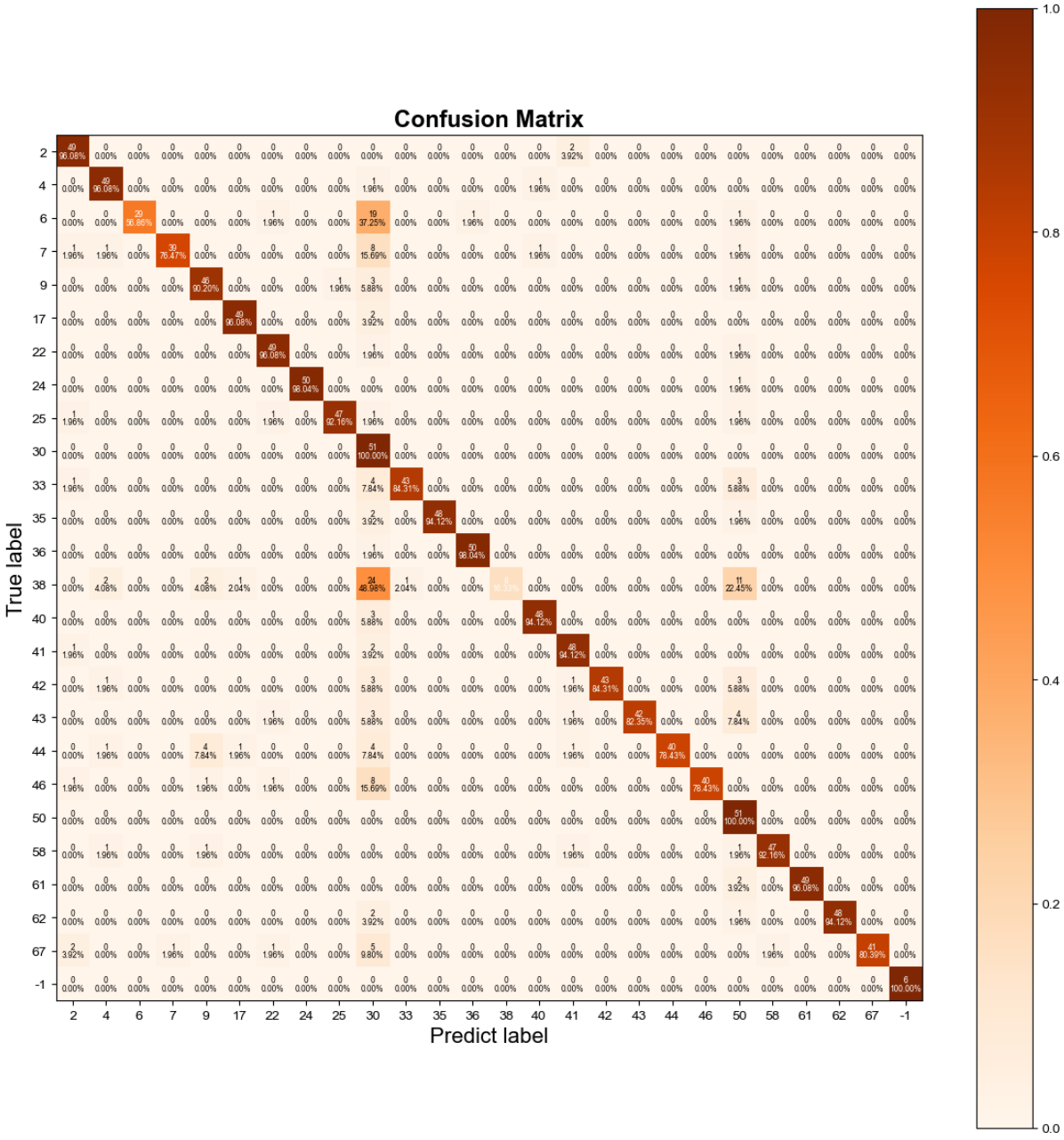
The accuracy of my images is 66.67%
The accuracy of the model is 98.20%
The recall of the model is 98.82%
The precision of the model is 99.37%
The F1 of the model is 99.09%

6.2.2.2 C = 0.1



The accuracy of my images is 66.67%
The accuracy of the model is 97.19%
The recall of the model is 98.57%
The precision of the model is 98.57%
The F1 of the model is 98.57%

6.2.2.3 C = 0.01



The accuracy of my images is 100%
The accuracy of the model is 86.79%
The recall of the model is 94.07%
The precision of the model is 91.81%
The F1 of the model is 92.93%

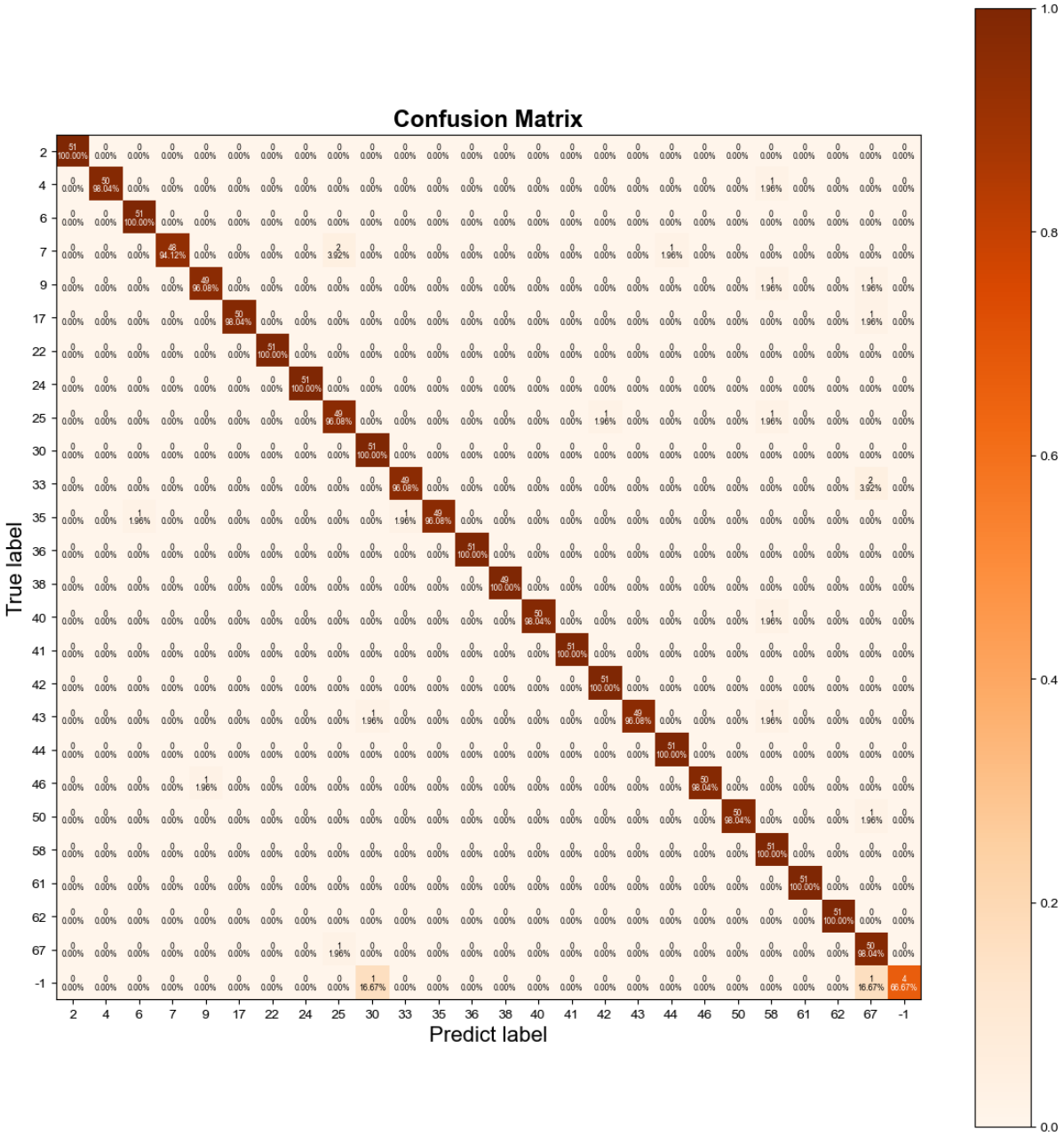
6.3 Discussion

6.3.1 Accuracy and C

C refers to penalty of the misclassification. As C decreases from 1 to 0.01, the accuracy of the model drops as well. Because the penalty of the misclassification drops, so the model will make more mistakes. The effect observed fits the theory.

6.3.2 Normalization

Normalization is usually helpful before using SVM classification. So I uniform the feature before the SVM and the result shows as below: (dimension = 200,C = 1)



The accuracy of the model is 98.36%
The recall of the model is 98.90%
The precision of the model is 99.45%

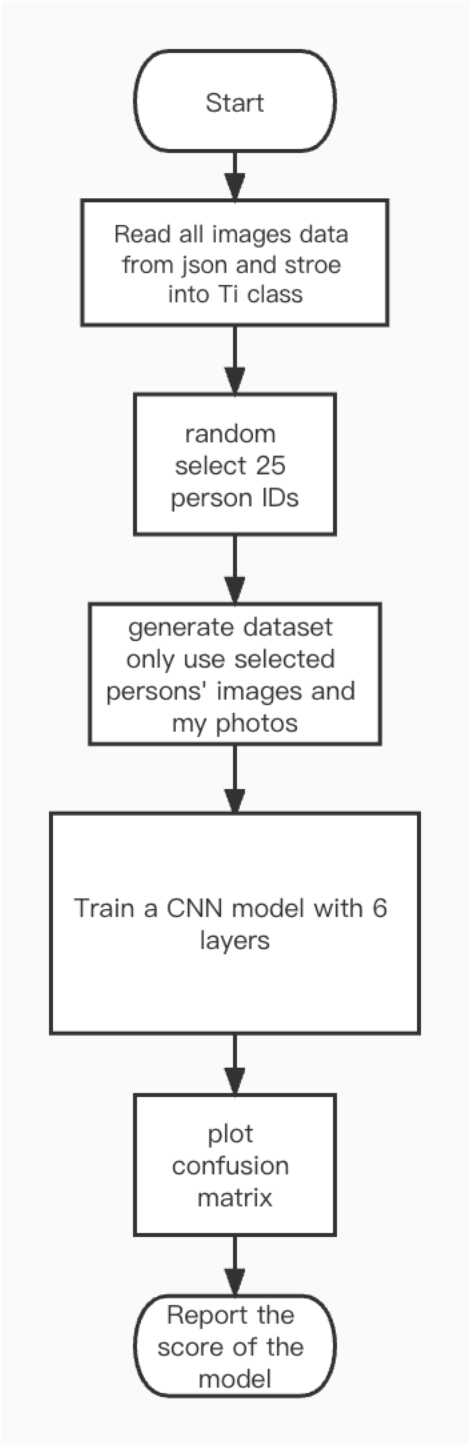
The F1 of the model is 99.17%

The accuracy of normalized input data is similar to that without normalization. So it not works at this data set.

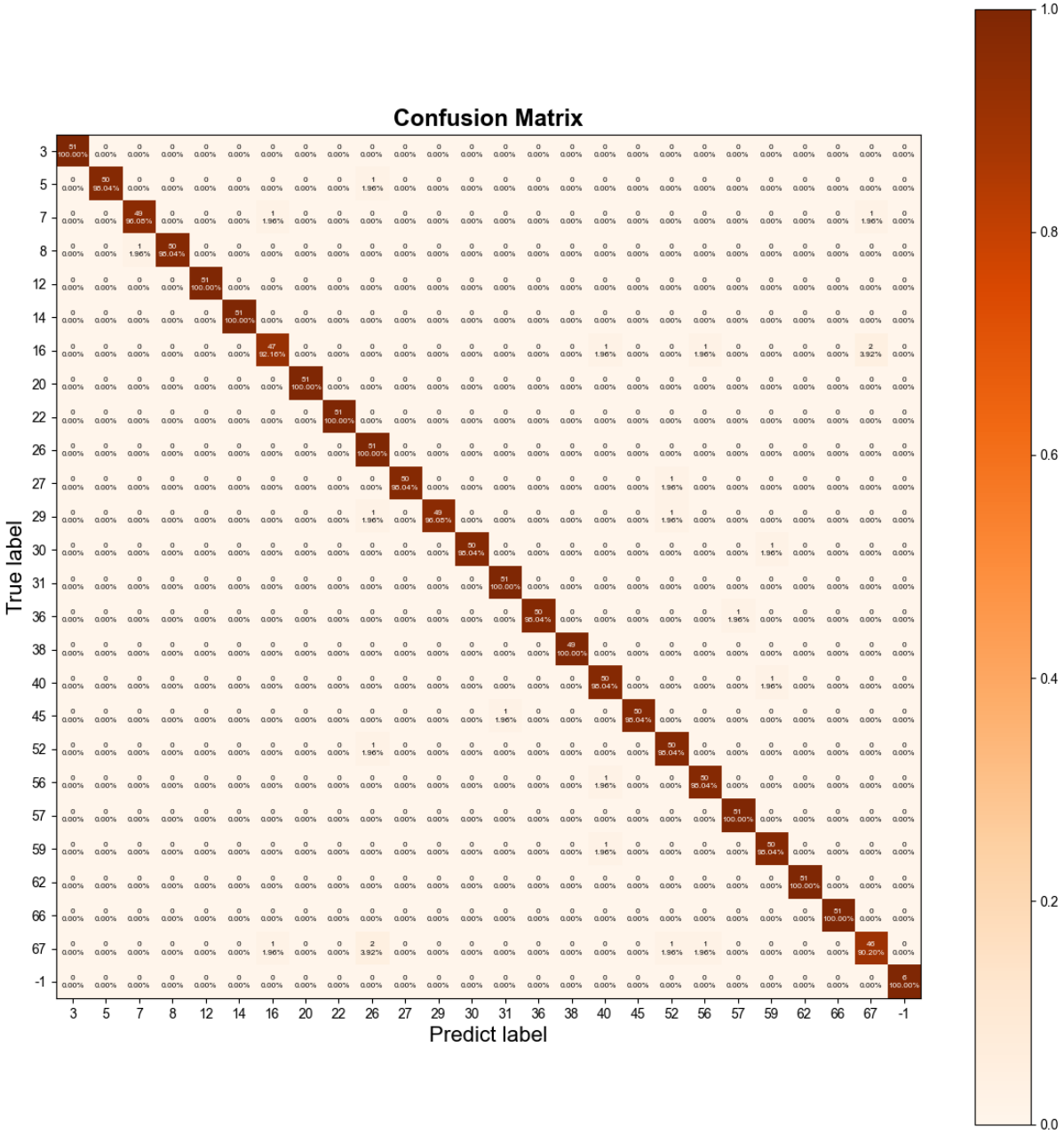
7.Part 6

This part builds a CNN model to classify the images.

7.1 Process Flow



7.2 Result



The accuracy of my images is 100%
The accuracy of the model is 98.20%
The recall of the model is 99.05%
The precision of the model is 99.13%
The F1 of the model is 99.09%

7.3 Discussion

It is my first time to build a CNN model with tensorflow. CNN is powerful and I should learn more about the tensorflow after exam week.

8. Conclusion

From the second half of the EE5907 Pattern Recognition, I learn many basic dimension reduction, clustering and classification algorithms. After writing them Python, I get a deeper understanding of these models. I am carrying on a project in which I plan to use CNN to classification. Now, I have some experience and understanding of building a CNN model. Hope I can successfully apply it in my work soon.