CS512 Final Project :

Image Classification using k Nearest Neighbours algorithm

GROUP 2:
ABHISHEK BHATT (ab2083)
HARSH BHATT (hb371)
SIVA HARSHINI DEV BONAM (sdb202)
TIANZHI CAO (tc796)

Project Goals

- Implement KNN algorithm to find nearest neighbours (pre-labeled images in the MNIST training set) of an input image from the MNIST testing set.

- Write an application on top of the KNN implementation that takes an image as input,and returns the digit it contains as an output.

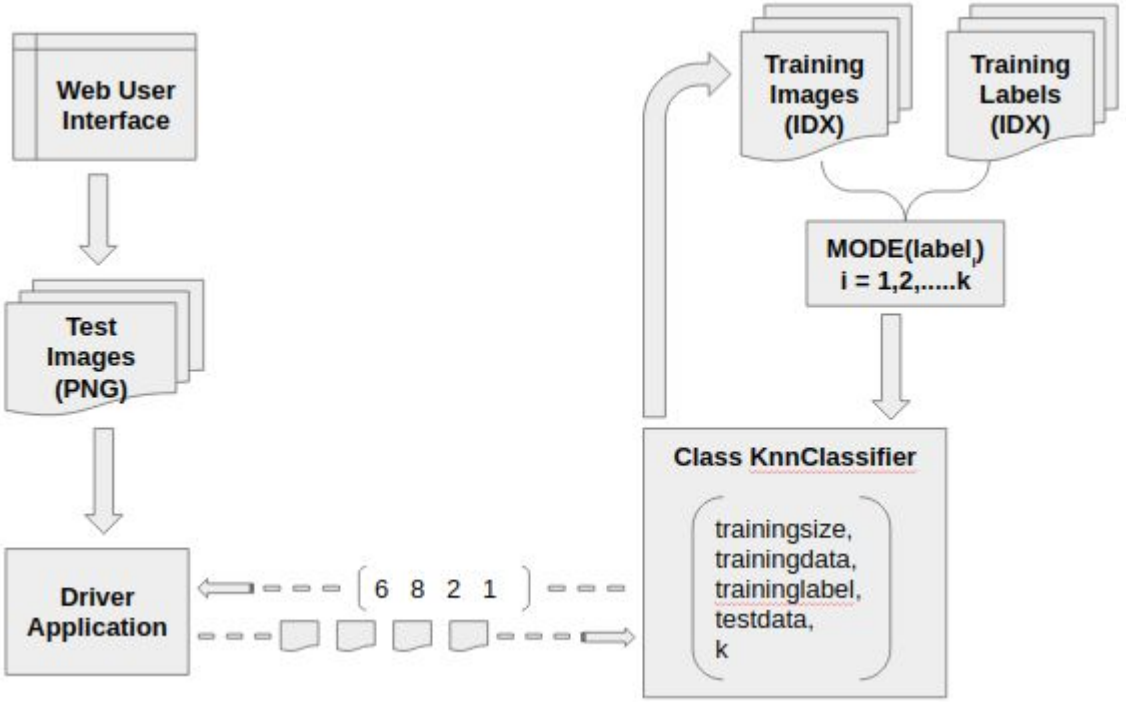
- Summarize the overall performance of the implementation in terms of time and accuracy.

Dataset

- MNIST database of handwritten digits
- Training set : 60,000 examples (IDX file format for implementation)
- Test set : 10,000 examples (PNG file format for implementation)
- Each image is 28x28 pixels in the raw dataset
- Data is at rest and can fit in a typical desktop

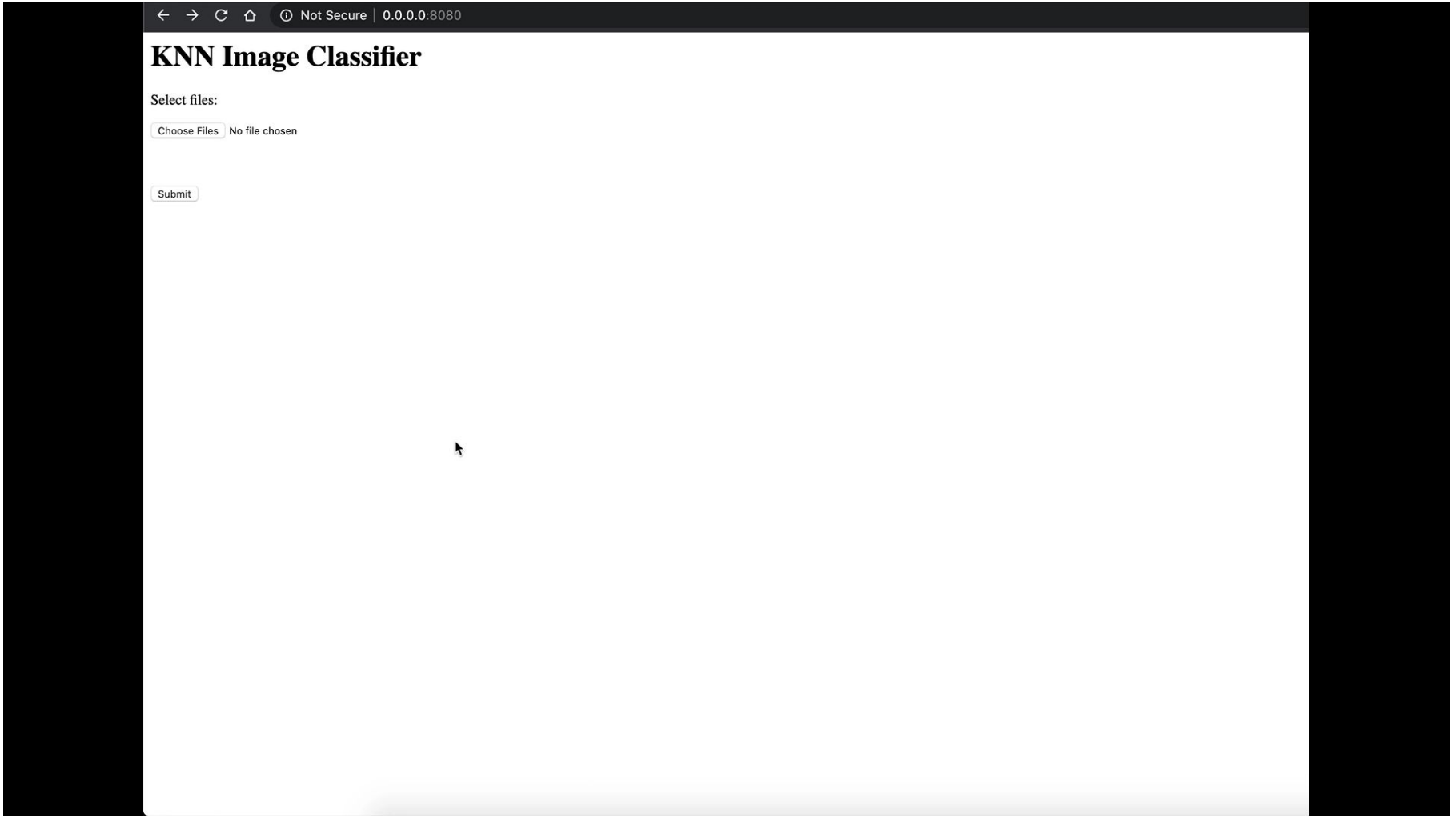
Project Modules

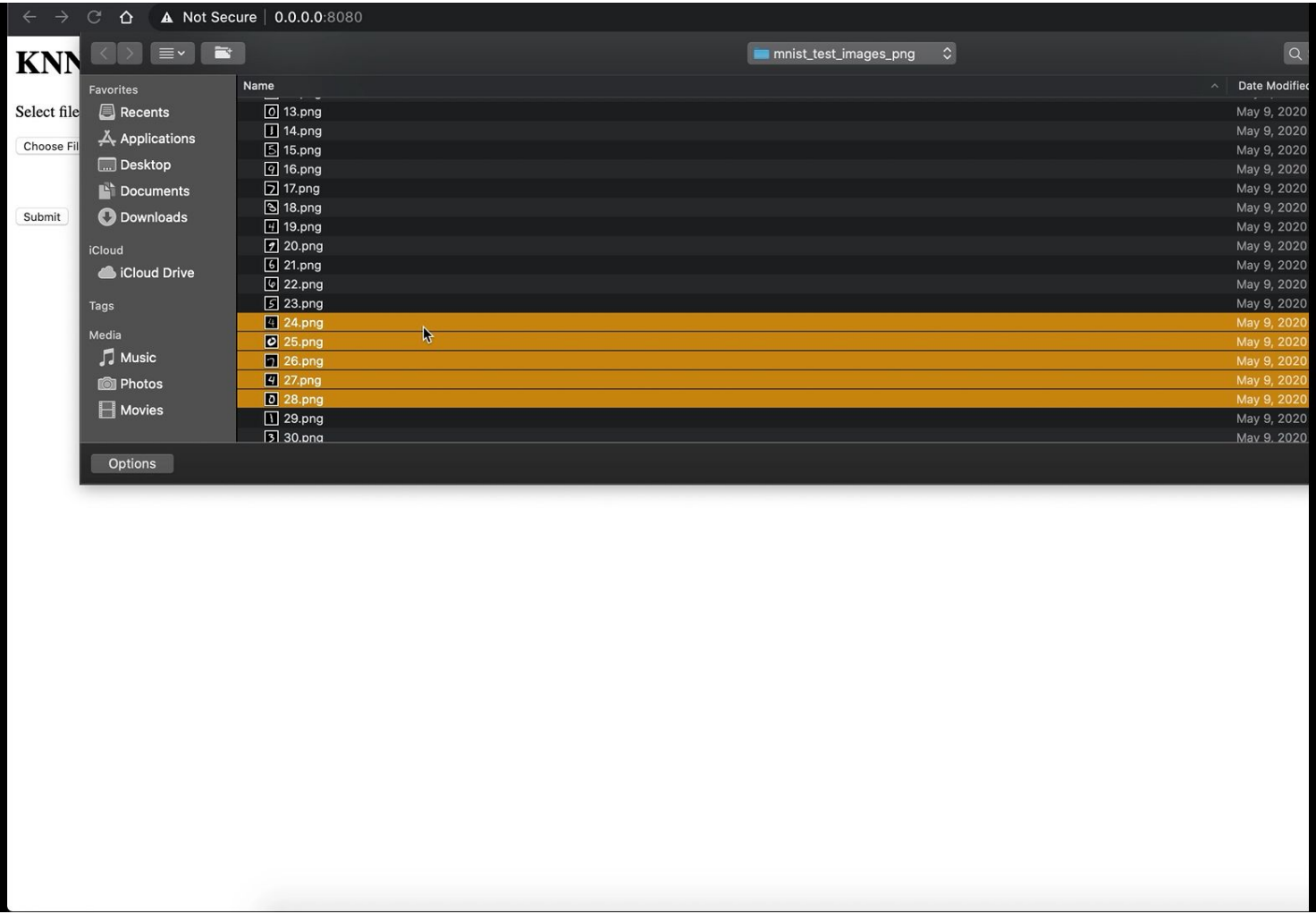
- Algorithm implementation : Python class KnnClassifier
- Image classifier web application : knn_image_classifier



Application Demo

To deploy the web application, we need to run applicationknnimageclassifier.py with Python. The UI is then accessible at port 8080 on the localhost URL.


Not Secure | 0.0.0.0:8080
KNN Image Classifier
Select files:
Choose Files No file chosen
Submit

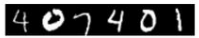

Not Secure | 0.0.0.0:8080
KNN
Select file
Choose Fil
Submit
mnist_test_images_png
Favorites
Recents
Applications
Desktop
Documents
Downloads
iCloud
iCloud Drive
Tags
Media
Music
Photos
Movies
Name
Date Modified
13.png
14.png
15.png
16.png
17.png
18.png
19.png
20.png
21.png
22.png
23.png
24.png
25.png
26.png
27.png
28.png
29.png
30.png
May 9, 2020
Options

Not Secure | 0.0.0.0:8080

KNN Image Classifier

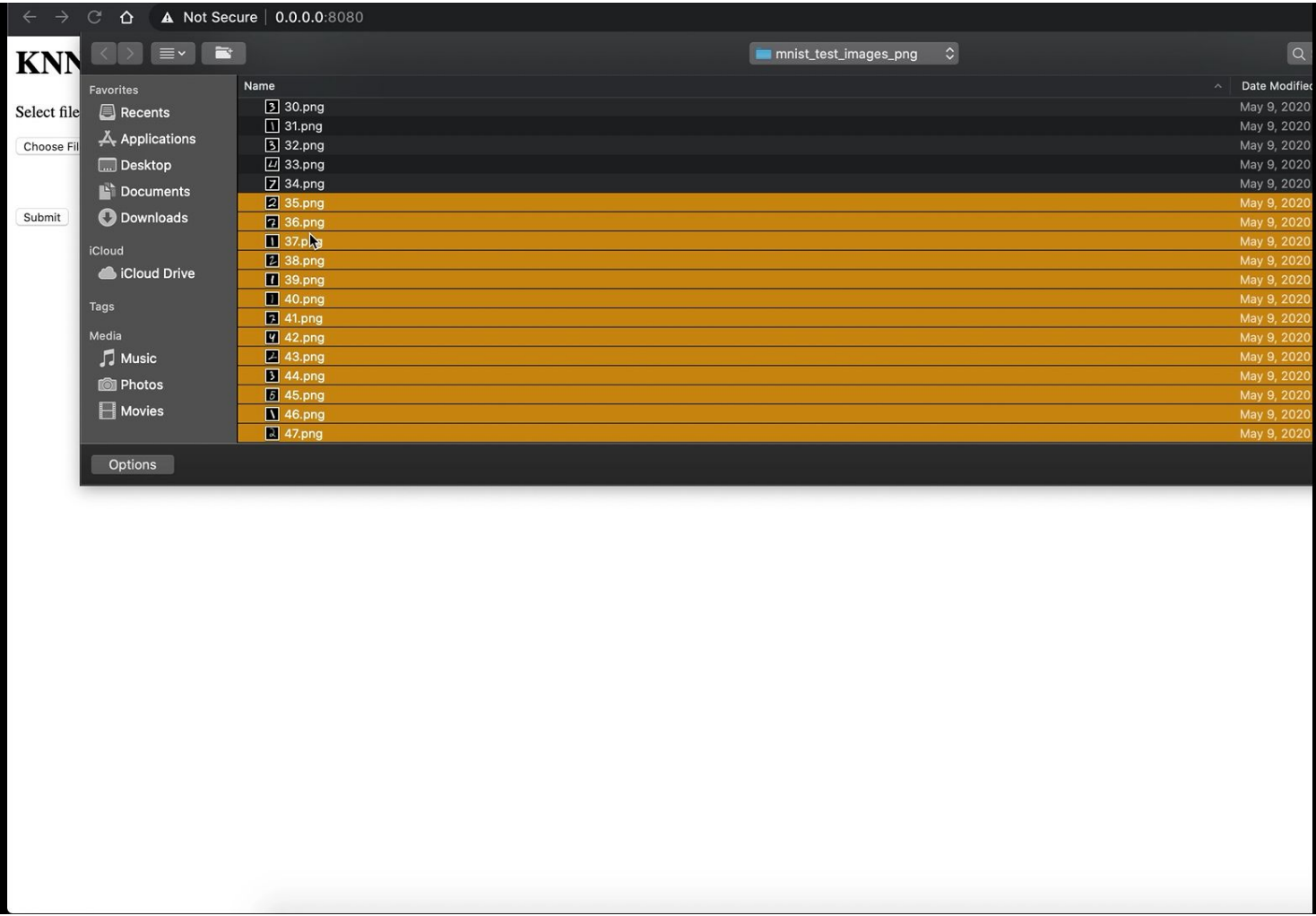
Select files:

Choose Files 6 files



Submit

"407401"


Not Secure
0.0.0.0:8080
KNN
Select file
Choose Fil
Submit
mnist_test_images_png
Favorites
Recents
Applications
Desktop
Documents
Downloads
iCloud
iCloud Drive
Tags
Media
Music
Photos
Movies
Name
Date Modified
30.png
31.png
32.png
33.png
34.png
35.png
36.png
37.png
38.png
39.png
40.png
41.png
42.png
43.png
44.png
45.png
46.png
47.png
May 9, 2020
Options

Not Secure | 0.0.0.0:8080

KNN Image Classifier

Select files:

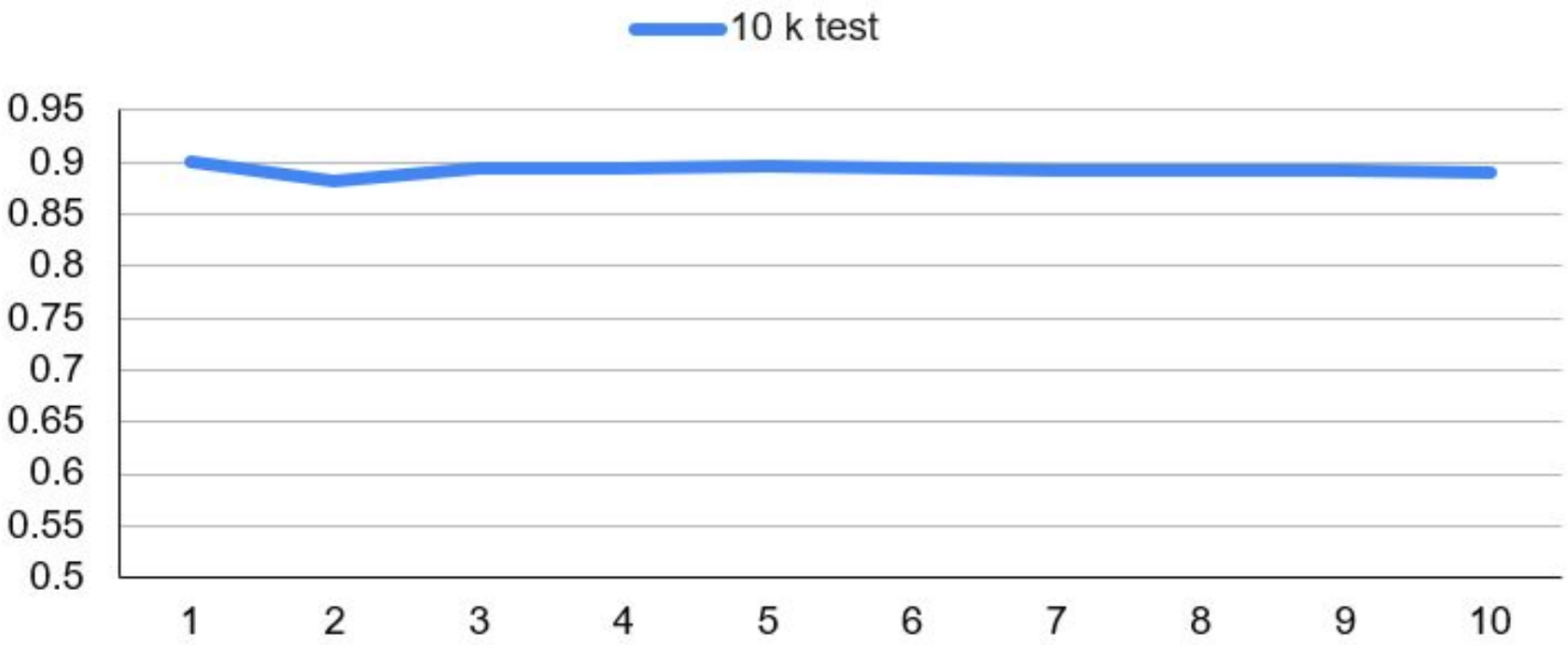
Choose Files 13 files

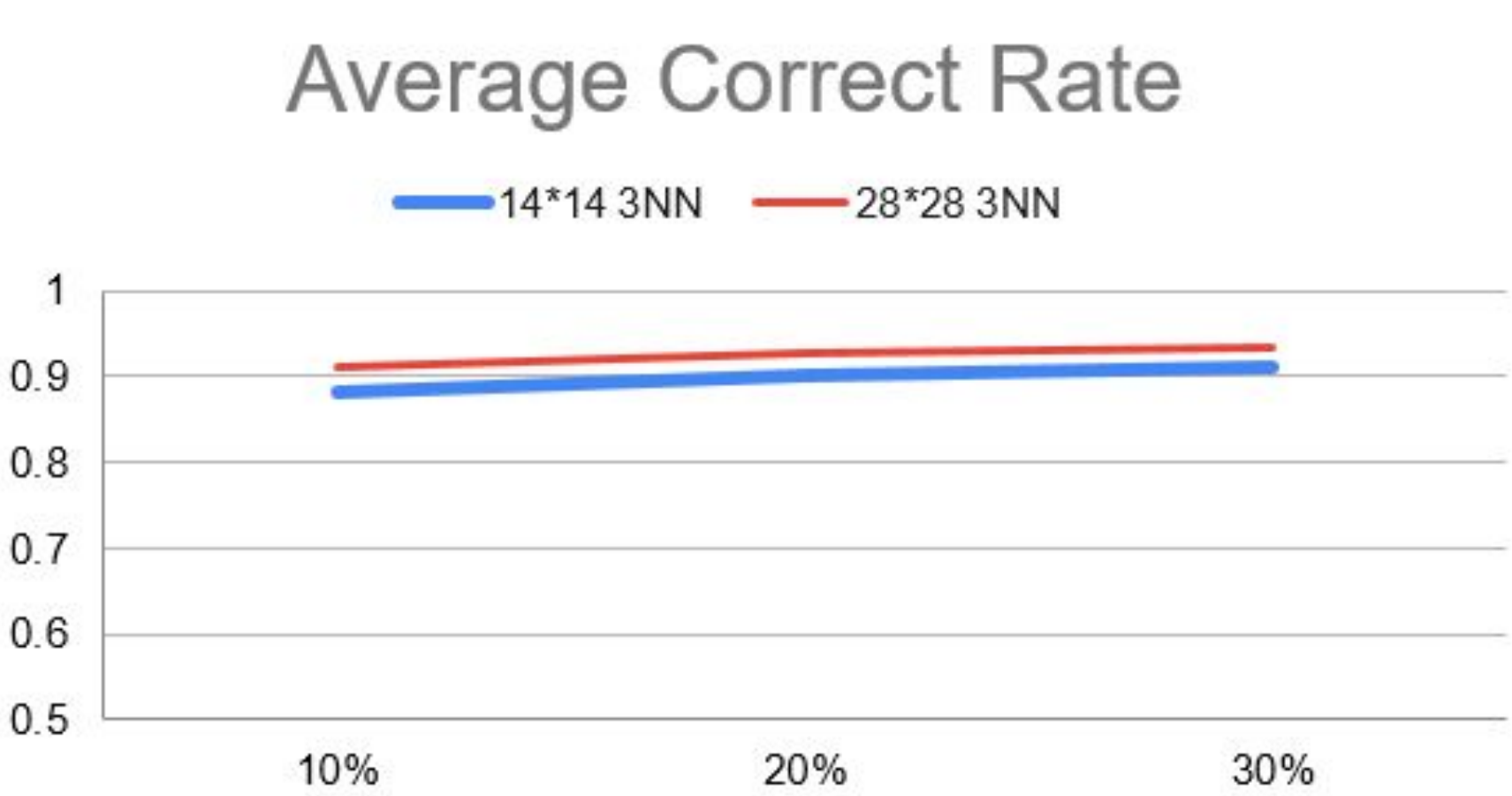
Submit

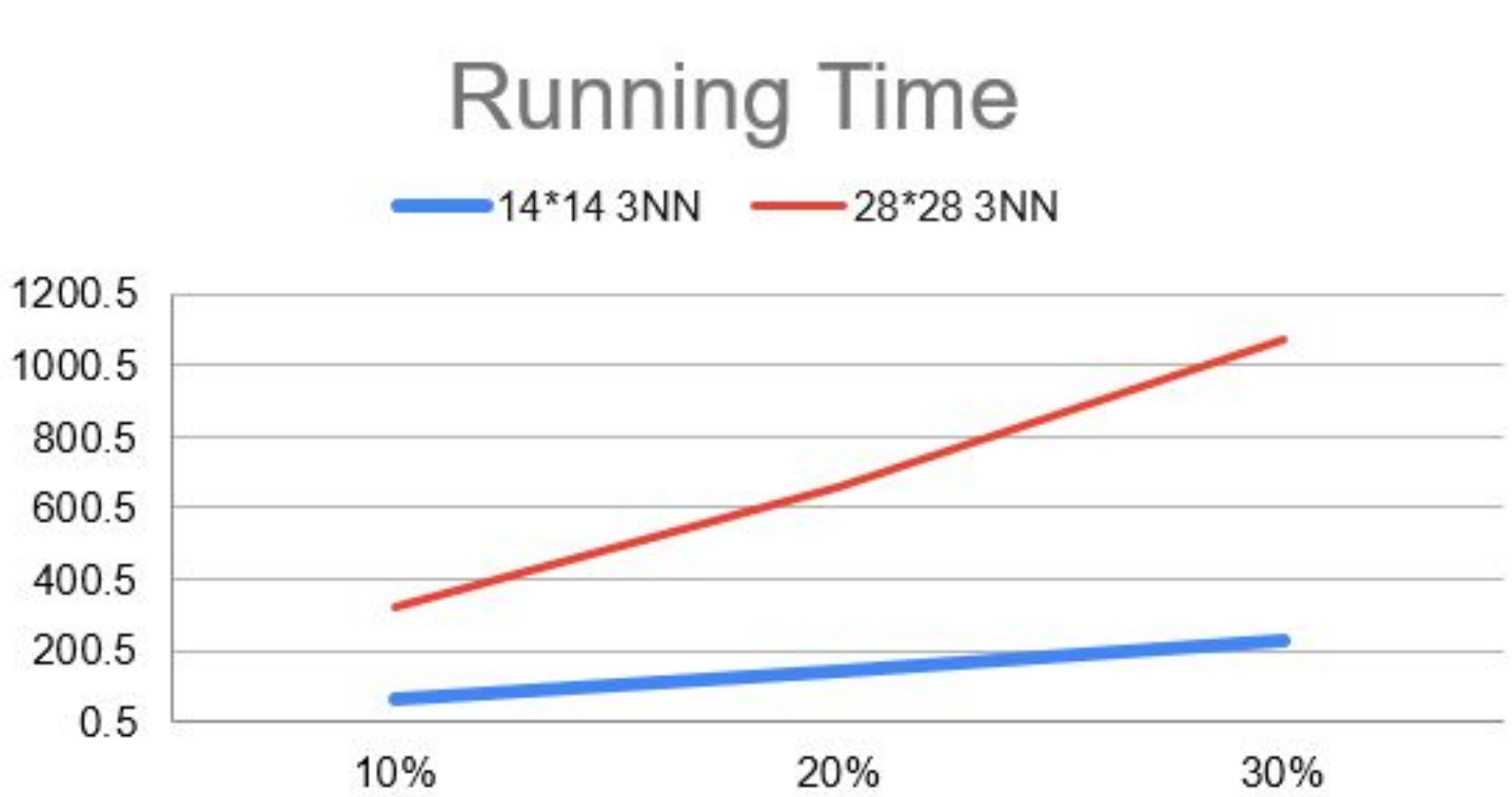
"2712117423512"

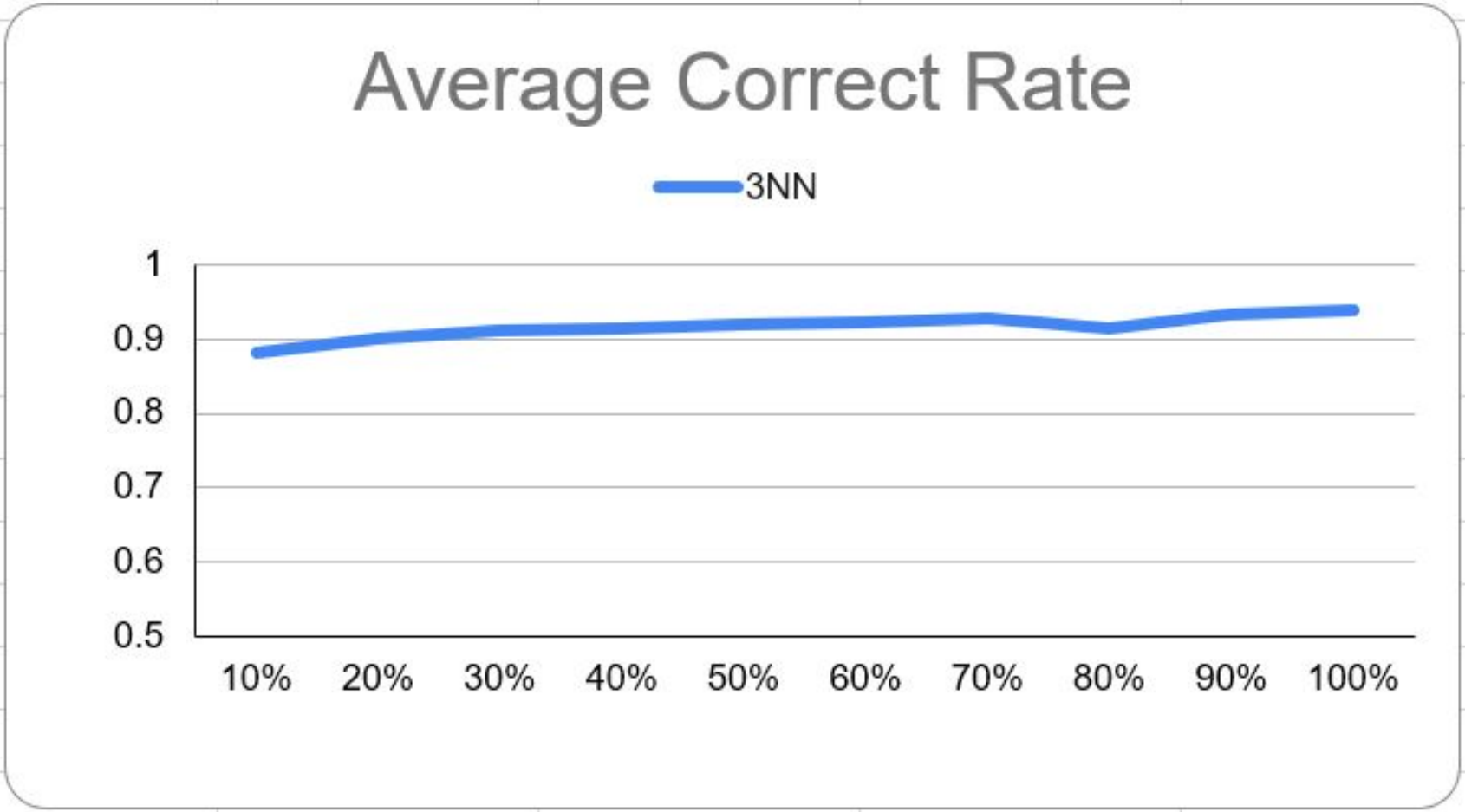
Analysis

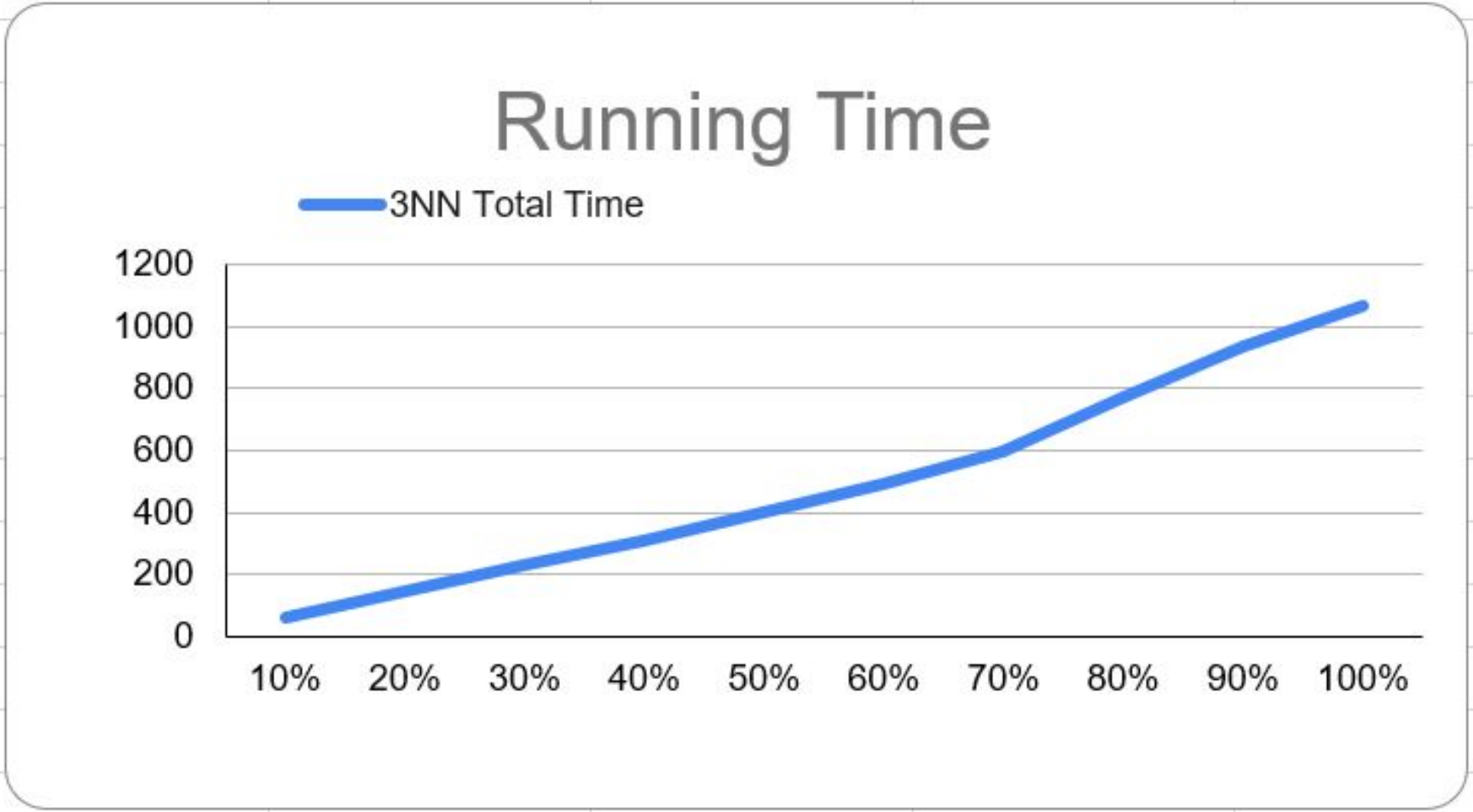
K accuracy rate comparison

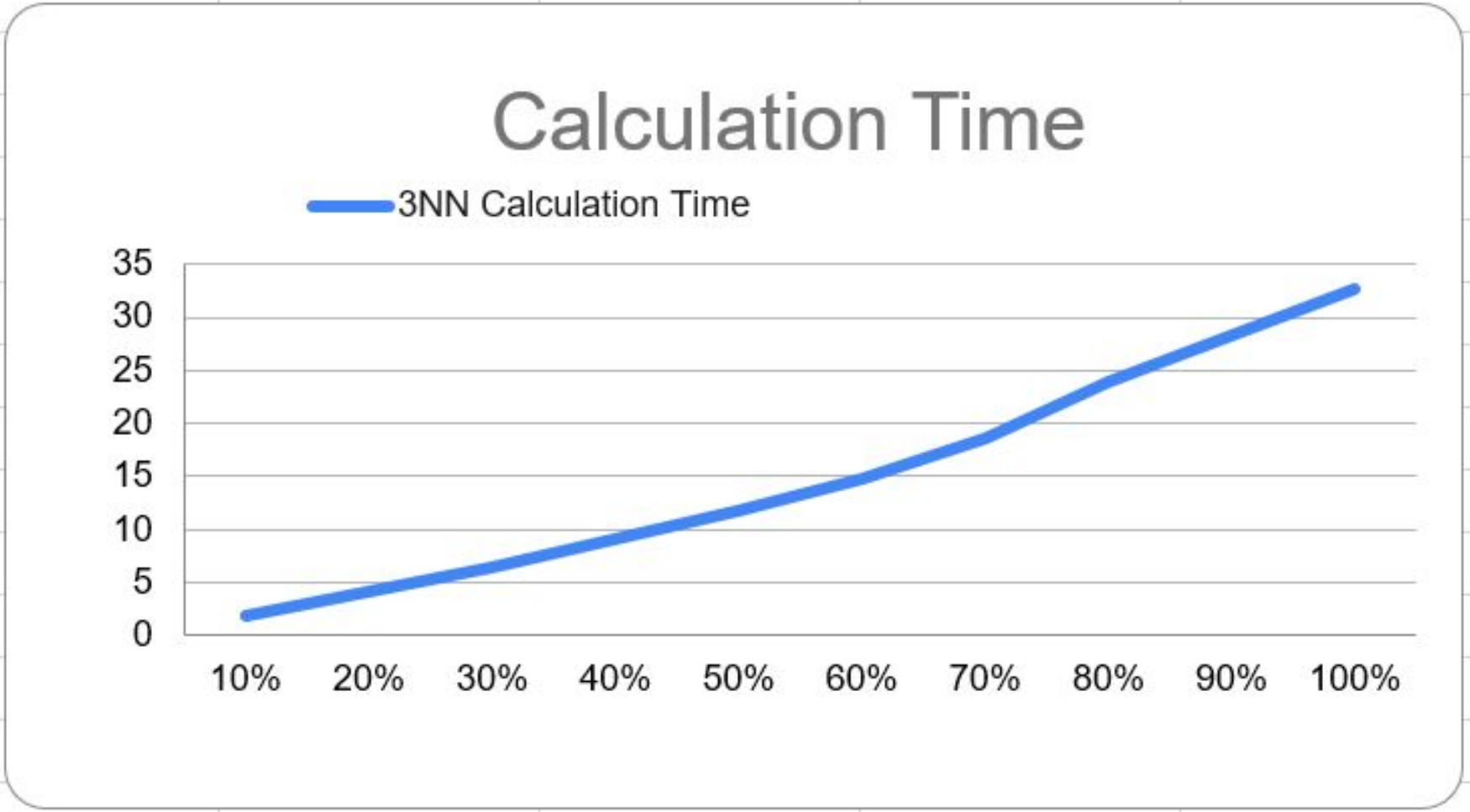



Average Correct Rate
14*14 3NN
28*28 3NN
1
0.9
0.8
0.7
0.6
0.5
10%
20%
30%


Running Time
14*14 3NN
28*28 3NN
1200.5
1000.5
800.5
600.5
400.5
200.5
0.5
10%
20%
30%


Average Correct Rate
3NN
1
0.9
0.8
0.7
0.6
0.5
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%


Running Time
3NN Total Time
1200
1000
800
600
400
200
0
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%


Calculation Time
3NN Calculation Time
35
30
25
20
15
10
5
0
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Observations

- No significant change in accuracy was observed for k = 1 to 10.
- Average correctness of predicted labels decreases after sub-sampling, but the drop is insignificant overall.
- Running time drops sharply after sub-sampling, as we increase size of training set.
- Average correctness of predicted labels is roughly same for all train sizes.
- Running time increases approximately linearly until we take 70% of the training set (10% increments for each observation).
- Beyond 70%, there is a sharp increase in running time per increment.

Conclusion

Through this project we have successfully demonstrated the application of the K-Nearest Neighbours algorithm for image classification and analyzed its performance in terms of the size of the training dataset.

Thank you