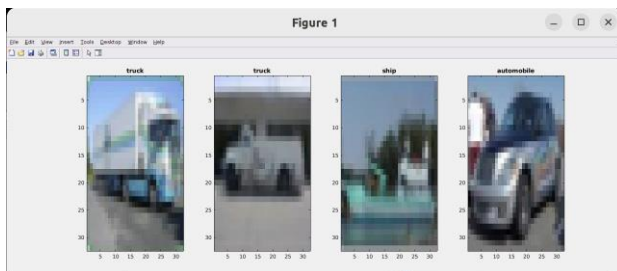


Introduction: This project aims to find out which algorithm has the best accuracy and shortest time.

Data and Preparation: there is a matrix with image, an array with the corresponding number, and another array of labels corresponding words. I generated an array called classes_hor and format it to classes, which fits the submission requirements. And then get all the related image number and labels in extract[] and nLabel[]. Then get the corresponding image data and copy it to the matrix nData, split the nData into trainingData, testingData, flatten it to flatTrainingData and flatTestingData for machine learning usage, and their corresponding labels to trainingLabel and testingLabel. Then got every accuracy, timetaken and confusion_matrix from each model.

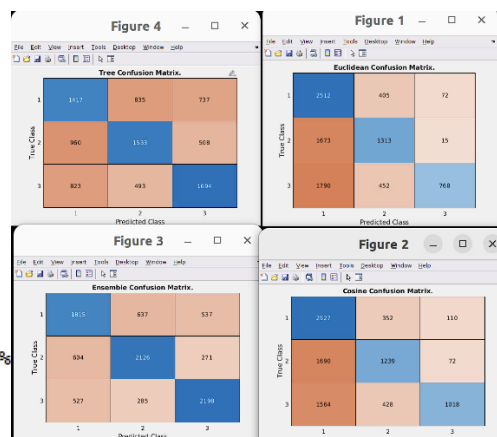


Methodology: All methods that was coded is vectorized instead of for loops for shorter time. For the KNNmodels, I used knnL2 model because I am required to, and I used 5 nearest neighbours since it seems like a good number to choose between accuracy and efficiency. I then used the cosine distance model since it is a formula that seems to be easier to be coded and it takes relatively less time to run.

For the MATLAB included models, I used Ensemble and Decision Tree models to process the data, since SVM is slow. I also used bag and ovabyclass to speed up the operation but with probably lower accuracy.

Results:

Euclidean formula time: 10.2101 s, Accuracy: 51.03%
Cosine formula time: 9.0863 s, Accuracy: 53.16%
Ensemble time: 58.6519 s, Accuracy: 68.21%
Tree time: 23.1032 s, Accuracy: 51.60%



Conclusion: If you are looking for lowest processing time but still relatively high accuracy rate, choose KNN cosine distance but if you are looking for highest accuracy, use ensemble.