National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS370: Artificial Intelligence

Semester Project Hyperparameters Tuning

Traffic Sign Recognition

Name	CMS
Aatir Khan	**321838**
Amal Saqib	**282496**
Arshanullah Tawhidi	**321788**

Baseline trained model 0:

Hyperparameter	Value
Number of Epochs	5
Number of iterations	981
Learning rate	0.001
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	32
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9247 = 92.47%** **Loss: 0.2717** **Accuracy on validation: 0.9765 = 97.65%** **Loss: 0.0863** **Accuracy on test: 0.9403 = 94.03%** **Difference (train - test) = -1.56%** **Difference (validation - test) = 3.62%**

Hyperparameter tuned trained model 1:

Hyperparameter	Value
Number of Epochs	**15**
Number of iterations	981
Learning rate	0.001
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense

Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	32
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9415 = 94.15%** **Loss: 0.2145** **Accuracy on validation: 0.9765 = 97.86%** **Loss: 0.0755** **Accuracy on test: 0.9447 = 94.47%** **Difference (train - test) = -0.32%** **Difference (validation - test) = 3.39%**

Hyperparameter tuned trained model 2:

Hyperparameter	**Value**
Number of Epochs	**10**
Number of iterations	981
Learning rate	**0.0005**
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	32
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9570 = 95.70%** **Loss: 0.1504** **Accuracy on validation: 0.9884 = 98.84%** **Loss: 0.0426** **Accuracy on test: 0.9585 = 95.85%**

	Difference (train - test) = 1.23% **Difference (validation - test) = 2.99%**

Hyperparameter tuned trained model 3:

Hyperparameter	**Value**
Number of Epochs	5
Number of iterations	981
Learning rate	0.001
Activation function	relu
Number of hidden layers in NN	**13**
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	32
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.8208 = 82.08%** **Loss: 0.6133** **Accuracy on validation: 0.9218 = 92.18%** **Loss: 0.2713** **Accuracy on test: 0.8755 = 87.55%** **Difference (train - test) = -5.47%** **Difference (validation - test) = 4.63%**

Hyperparameter tuned trained model 4:

Hyperparameter	**Value**
Number of Epochs	5
Number of iterations	**491**
Learning rate	0.001
Activation function	relu
Number of hidden layers in NN	12
Number of neurons in each hidden layer	81 neurons
Max Depth of the tree	-

Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	**64**
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9213 = 92.13%** **Loss: 0.2659** **Accuracy on validation: 0.9807 = 98.07%** **Loss: 0.0788** **Accuracy on test: 0.9481 = 94.81%** **Difference (train - test) = -2.68%** **Difference (validation - test) = 3.26%**

Hyperparameter tuned trained model 5:

Hyperparameter	**Value**
Number of Epochs	**10**
Number of iterations	**491**
Learning rate	**0.0005**
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	**64**
Regularization penalty	-
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9643 = 96.43%** **Loss: 0.1377** **Accuracy on validation: 0.9889 = 98.89%** **Loss: 0.0422** **Accuracy on test: 0.9624 = 96.24%** **Difference (train - test) = 0.19%**

	Difference (validation - test) = 2.65%

Hyperparameter tuned trained model 6:

Hyperparameter	**Value**
Number of Epochs	5
Number of iterations	981
Learning rate	0.001
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout 256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	32
Regularization penalty	**0.01**
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9123 = 91.23%** **Loss: 0.5185** **Accuracy on validation: 0.9745 = 97.45%** **Loss: 0.3142** **Accuracy on test: 0.9487 = 94.87%** **Difference (train - test) = 3.64%** **Difference (validation - test) = 2.58%**

Hyperparameter tuned trained model 7 (final model):

Hyperparameter	**Value**
Number of Epochs	**10**
Number of iterations	**491**
Learning rate	**0.0005**
Activation function	relu
Number of hidden layers in NN	10
Number of neurons in each hidden layer	2 - Conv2D 1 - Dropout

	256 - Dense 43- Dense
Max Depth of the tree	-
Value of k in K-nearest neighbor	-
Value of k in k-means	-
Training, validation, testing split	31367, 7842, 12,631
Batch size	**64**
Regularization penalty	**0.01**
Dropout rate	0.25
Number of features (feature engineering)	7
Accuracy/Loss/Error of the model	**Accuracy on train: 0.9675 = 96.75%** **Loss: 0.1198** **Accuracy on validation: 0.9925 = 99.25%** **Loss: 0.0330** **Accuracy on test: 0.9699 = 96.99%** **Difference (train - test) = -0.24%** **Difference (validation - test) = 2.26%**

Analysis:

Baseline trained model 0: We started with a small number of epochs because it takes about 12 minutes with 5 epochs.
The accuracy of the:
Trained set: 92.47%
Validation set: 97.65%
Test set: 94.03%

The model is neither underfit nor overfit, but we tried more models to improve accuracy.

Hyperparameter tuned trained model 1: We increased number of epochs to 15 and it took about 29 minutes to execute.
The accuracy of the:
Trained set: 94.15%
Validation set: 97.86%
Test set: 94.47%

The model is neither underfit nor overfit. The model took 17 more minutes to execute, but the accuracy improved only slightly.

Hyperparameter tuned trained model 2: We halved the learning rate to improve accuracy. To compensate with the decrease in learning rate, we doubled the number of epochs to 10. This model takes around 22 minutes to run.

The accuracy of the
Trained set: 95.70%
Validation set: 98.84%
Test set: 95.85%

The model is neither underfit nor overfit. However, decreasing the learning rate improved accuracy thereby proving that the previous models were found a suboptimal final set of weight.

Hyperparameter tuned trained model 3: We then added more hidden layers to our baseline model. The model takes around 12 minutes to run.
The accuracy of the
Trained set: 82.08%
Validation set: 92.18%
Test set: 87.55%

The model is underfit. Underfitting occurs when the size of the training dataset used is not enough.

Hyperparameter tuned trained model 4: We increased the batch size to 64 which reduced the number of iterations to 481 and kept the other hyperparameters the same as model 0. The model takes around 7 minutes to run.
The accuracy of the
Trained set: 92.13%
Validation set: 98.07%
Test set: 94.81%

The accuracy of the trained model decreases a bit compared to model 0, but it is more accurate on validation and test sets. This model is much faster than model 0.

Hyperparameter tuned trained model 5: We merged models 2 and 4 by using the hyperparameter changes introduced in both cases. The model takes around 22 minutes to run.
The accuracy of the
Trained set: 96.43%
Validation set: 98.89%
Test set: 96.24%

The model neither underfits nor overfits. This model is more accurate than model 2 and 4 because the lost of accuracy by using a larger batch size can be recovered by increasing the learning rate.

Hyperparameter tuned trained model 6: We used the same hyperparameters as model 0, but added L2 regularization. The model takes around 11 minutes to run.
The accuracy of the
Trained set: 91.23%

Validation set: 97.45%
Test set: 94.87%

The model performs well, but it could have been much better in terms of accuracy. Regularization is used to prevent overfitting. Even though our model performed well on the test dataset, it performs much better on the validation dataset and to decrease the difference between the accuracies of validation and test, we added regularization.

Hyperparameter tuned trained model 7: We merged models 5 and 6 by using the hyperparameter changes introduced in both cases. The model takes around 22 minutes to run.
The accuracy of the
Trained set: 96.75%
Validation set: 99.25%
Test set: 96.99%

This is our final model and has a significantly high accuracy in each of the trained, validation, and test sets. The difference between the trained and test set is quite low, emphasizing our success in tuning the hyperparameters for optimal performance. The loss is also low in this model compared to the rest, hence, model 7 is our final model.