DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING
Course Code: CS-324
Course Title: Machine Learning
Complex Engineering Problem
TE Batch 2020, Spring Semester 2023
Grading Rubric
TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1	Anas Hassan Ansari	CS-20065
S2	Farrukh Ahmed Siddiqui	CS-20090
S3	Syed Muhammad Hamza Naveed	CS-20119

CRITERIA AND SCALES				Marks Obtained		
				S1	S2	S3
Criterion 1: Does the application meet the desired specifications and produce the desired outputs? (CPA-1, CPA-2, CPA-3) **[8 marks]**						
1	2	3	4			
The application does not meet the desired specifications and is producing incorrect outputs.	The application partially meets the desired specifications and is producing incorrect or partially correct outputs.	The application meets the desired specifications but is producing incorrect or partially correct outputs.	The application meets all the desired specifications and is producing correct outputs.			
Criterion 2: How well is the code organization? **[2 marks]**						
1	2	3	4			
The code is poorly organized and very difficult to read.	The code is readable only to someone who knows what it is supposed to be doing.	Some part of the code is well organized, while some part is difficult to follow.	The code is well organized and very easy to follow.			
Criterion 3: Does the report adhere to the given format and requirements? **[6 marks]**						
1	2	3	4			
The report does not contain the required information and is formatted poorly.	The report contains the required information only partially but is formatted well.	The report contains all the required information but is formatted poorly.	The report contains all the required information and completely adheres to the given format.			
Criterion 4: How does the student performed individually and as a team member? (CPA-1, CPA-2, CPA-3) **[4 marks]**						
1	2	3	4			
The student did not work on the assigned task.	The student worked on the assigned task, and accomplished goals partially.	The student worked on the assigned task, and accomplished goals satisfactorily.	The student worked on the assigned task, and accomplished goals beyond expectations.			

Final Score = (Criterial_1_score x 2) + (Criteria_2_score / 2) + (Criteria_3_score x (3/2)) + (Criteria_4_score)
= ________________________

Contents

1. Introduction

This report presents a detailed analysis of the implementation and evaluation of different machine learning models for image classification. The objective is to classify images into two categories: animals and vehicles. The dataset used for training and testing the models is the CIFAR-10 dataset, consisting of 50,000 training images and 10,000 testing images.

2. Data Preprocessing

2.1 Loading the dataset

The CIFAR-10 dataset was downloaded from provided link and loaded into x_train, x_test, y_train and y_test by splitting image pixels data (input) and labels (output).

As the loaded data is already flattened for each image, so we won't be needing to manually flatten the image.

2.2 Applying preprocessing steps

As the CIFAR-10 dataset consists of 10 classes of different images, these classes were converted into 2 classes i.e. animals and vehicles and Binary Encoding was applied to them making them converted into 0 and 1. (0 for animal & 1 for vehicles).

2.3 Splitting the dataset

The training sets from the dataset is further divided into train and validation sets to evaluate the algorithm with different hyperparameters and find the well suited hyperparameters for our algorithms

2.4 Normalizing the dataset

We can apply normalization on the image by normalizing RGB values of every pixels. The formula for normalization is:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

For every pixel, the minimum value could be 0 and the maximum value could be 255, therefore, on putting maximum and minimum values on the formula, we can say we simply need to divide each pixel with 255 to normalize the image.

3. Logistics Regression

Logistic regression is one of the most popular Parametric Machine Learning Algorithm. It is used for predicting the categorical dependent variable using a given set of independent variables. The outcome is categorical or discrete values and not continuous. The Logistic predicts the categorical data on the basis of probabilities, the predicted probabilities lie between 0 and 1.

- Model1_1 is logistics first model with maximum iterations=1000, random state=0 and solver=liblinear.
- Model1_2 is logistics second model with max iteration=5000, random state=0 and solver=lbfgs.
- Model1_3 is logistics third model with max iteration=5000, random state=0 and solver=sag.

By fitting these models, on training set and evaluating metrics on validation set, we found out that the best model among different models of logistic regression is 2nd model i.e., model1_2 therefore we fitted that model on training + validation sets and evaluated the metrics on test set for this model.

4. Extreme Gradient Boost (XGBoost)

XGBoost works by combining a number of weak learners to form a strong learner. A weak learner is a machine learning model that is slightly better than random guessing. However, when weak learners are combined, they can form a strong learner that is much more accurate.

XGBoost works by training a number of decision trees. Each tree is trained on a subset of the data, and the predictions from each tree are combined to form the final prediction.

Since our data is imbalanced, it has 6 different classes for animals and 4 different classes for vehicles. Therefore, a class ratio has been defined to properly balance the data has been passed to scale_pos_weight argument of every model

- Model2_1 is xgboost's first model with maximum depth = 5 and learning rate = 0.01.
- Model2_2 is xgboost's second model with maximum depth = 7 and learning rate = 0.001.
- Model2_3 is xgboost's third model with maximum depth = 3 and learning rate = 0.1.

By fitting these models, on training set and evaluating metrics on validation set, we found out that the best model among different models of xgboost classification is 3rd model i.e., model2_3 therefore we fitted that model on training + validation sets and evaluated the metrics on test set for this model.

5. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. It is designed for processing structured arrays of data such as images. CNNs use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. In a fully-connected feedforward neural network, every node in the input is tied to every node in the first layer, and so on. In contrast, a convolutional neural network is a special kind of feedforward neural network with fewer weights than a fully-connected network

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are: Convolutional layer, Pooling layer, Fully-connected (FC) layer. The convolutional layer is the first layer of a convolutional network. Convolution is a mathematical operation that allows the merging of two sets of information. In the case of CNN, convolution is applied to the input data to filter the information and produce a feature map. This filter is also called a kernel, or feature detector, and its dimensions can be, for example, 3x3. The convolution operation involves sliding a filter over the input image and computing the dot product between the filter and the input at each location. The output of this operation is called a feature map.

- Model3_1 is CNN's first model with convolution layer kernel size=3, pooling layer kernel size=2, stride=1, padding=1, learning rate=0.001, weight decay=0.0001, number of epochs=10, batch size=32.
- Model3_2 is CNN's second model with convolution layer kernel size=5, pooling layer kernel size=2, stride=1, padding=2, learning rate=0.01, weight decay=0.001, number of epochs=7, batch size=128.
- Model3_3 is CNN's third model with convolution layer kernel size=7, pooling layer kernel size=2, stride=1, padding=3, learning rate=0.1, weight decay=0.01, number of epochs=12, batch size=256.

By fitting these models, on training set and evaluating metrics on validation set, we found out that the best model among different models of CNN is 1st model i.e., model3_1 therefore we fitted that model on training + validation sets and evaluated the metrics on test set for this model.

6. Comparison of Results

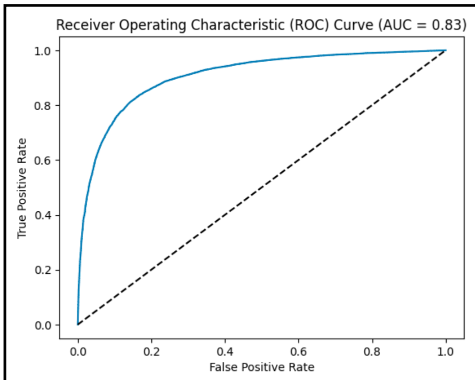
6.1 Comparison of 9 models of different hyperparameters

Following are the training and validation metrics for 9 models of different hyperparamters:

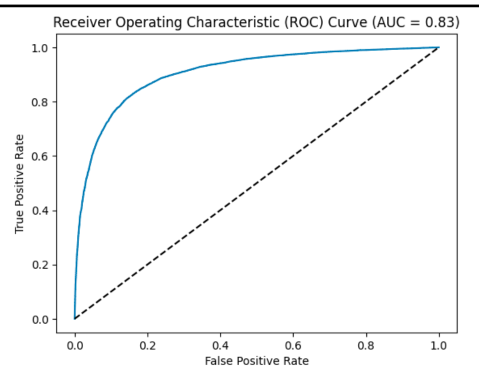
6.1.1 Training Metrics

	Accuracy	Precision	Recall	F1-score	Confusion Matrix
Logistic-1	0.840275	0.820285	0.769053	0.793843	[[21310, 2695], [3694, 12301]]
Logistic-2	0.840375	0.820504	0.769053	0.793946	[[21314, 2691], [3694, 12301]]
Logistic-3	0.840350	0.820364	0.769178	0.793947	[[21311, 2694], [3692, 12303]]
XGBoost-1	0.830700	0.778422	0.806064	0.792002	[[20335, 3670], [3102, 12893]]
XGBoost-2	0.843025	0.793251	0.821569	0.807162	[[20580, 3425], [2854, 13141]]
XGBoost-3	0.864075	0.813405	0.856580	0.834435	[[20862, 3143], [2294, 13701]]
ConvNN-1	0.972184	0.977550	0.952330	0.964775	[[23650, 350], [763, 15237]]
ConvNN-2	0.913161	0.929136	0.847543	0.886466	[[22966, 1034], [2439, 13561]]
ConvNN-3	0.729474	0.812228	0.421017	0.554572	[[22443, 1557], [9264, 6736]]

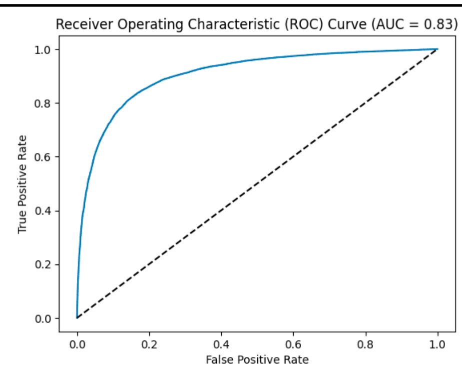
Following are the training set, ROC curves for Logistics Regression algorithm



Training ROC curve model1_1

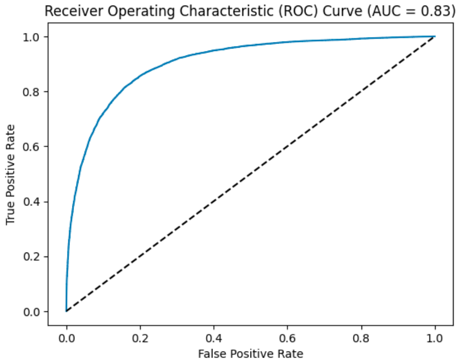


Training ROC curve model1_2

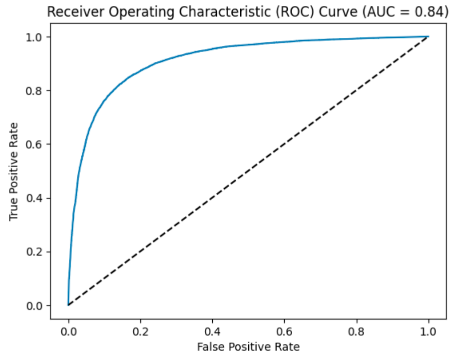


Training ROC curve of model1_3

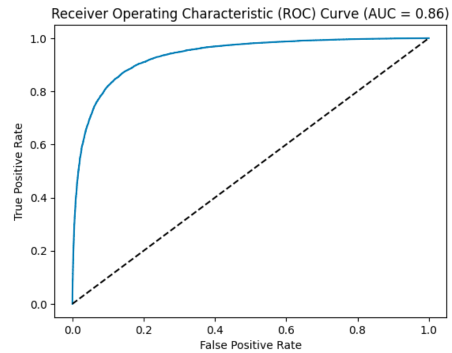
Following are the training set, ROC-Curves for XGBoost algorithm.



Training ROC curve of model2_1

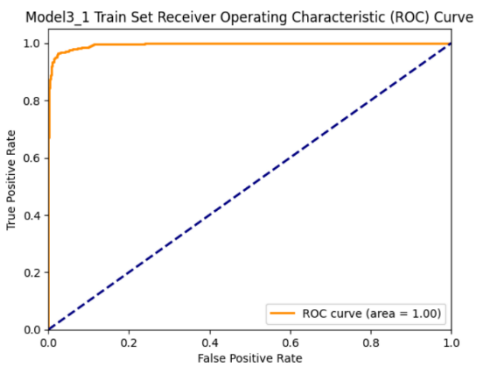


Training ROC curve of model2_2

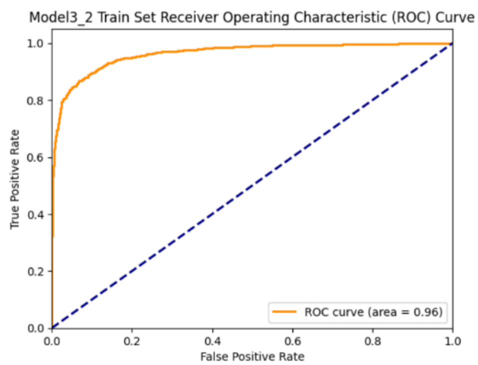


Training ROC curve of model2_3

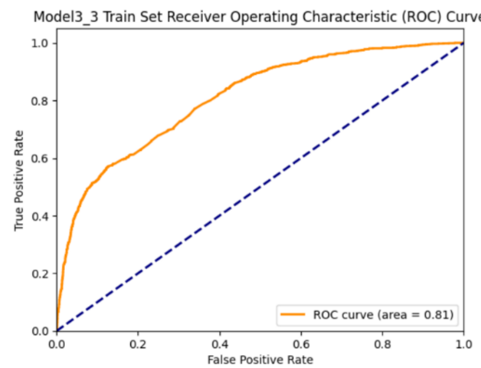
Following are the training set, ROC-Curves for Convolutional Neural Network.



Training ROC curve of model3_1



Training ROC curve of model3_2

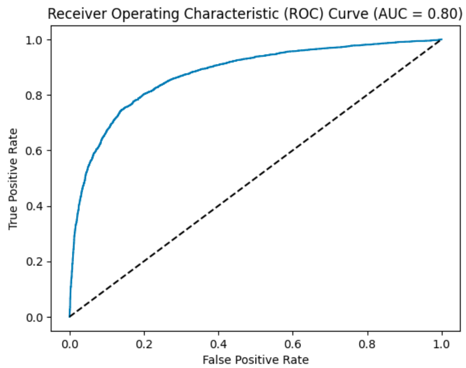


Training ROC curve of model3_3

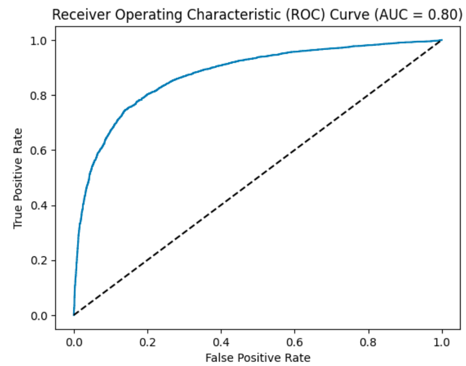
6.1.2 Validation Metrics

	Accuracy	Precision	Recall	F1-score	Confusion Matrix
Logistic-1	0.812800	0.785695	0.732335	0.758077	[[5195, 800], [1072, 2933]]
Logistic-2	0.812900	0.785753	0.732584	0.758237	[[5195, 800], [1071, 2934]]
Logistic-3	0.812600	0.785427	0.732085	0.757819	[[5194, 801], [1073, 2932]]
XGBoost-1	0.809100	0.754369	0.776030	0.765046	[[4983, 1012], [897, 3108]]
XGBoost-2	0.798600	0.737646	0.771536	0.754210	[[4896, 1099], [915, 3090]]
XGBoost-3	0.845500	0.793977	0.829463	0.811332	[[5133, 862], [683, 3322]]
ConvNN-1	0.960808	0.965761	0.935175	0.950222	[[5867, 133], [259, 3741]]
ConvNN-2	0.905960	0.904312	0.855414	0.879184	[[5638, 362], [578, 3422]]
ConvNN-3	0.667355	0.695771	0.299225	0.418478	[[5477, 523], [2803, 1197]]

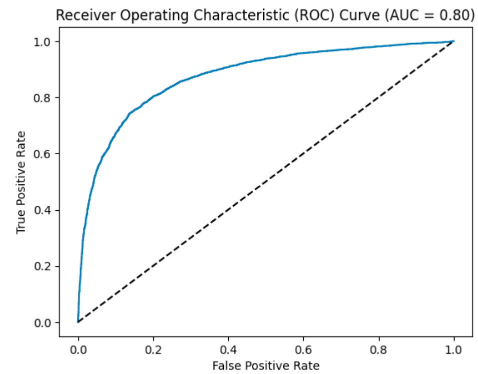
Following are the validation set, ROC curves for Logistic Regression algorithm.



Validation ROC curve of model1_1

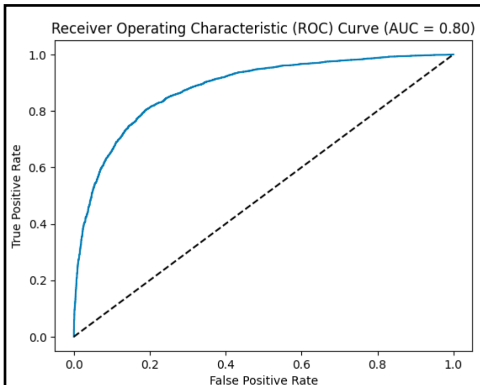


Validation ROC curve of model1_2

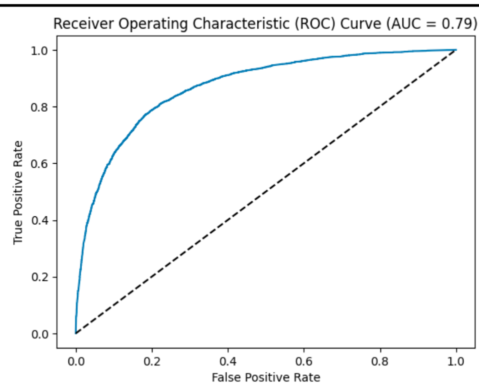


Validation ROC curve of model1_3

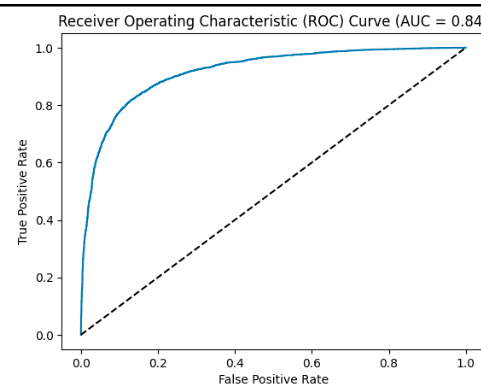
Following are validation set, ROC curves for XGBoost algorithm.



Validation ROC curve of model2_1

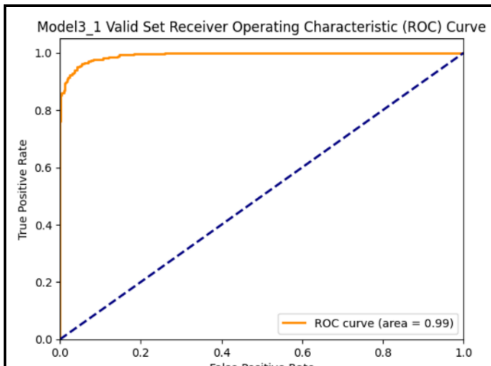


Validation ROC curve of model2_2

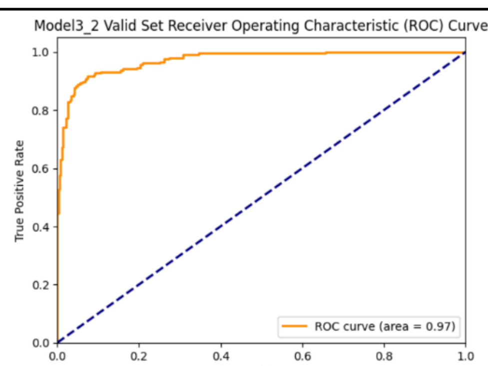


Validation ROC curve of model2_3

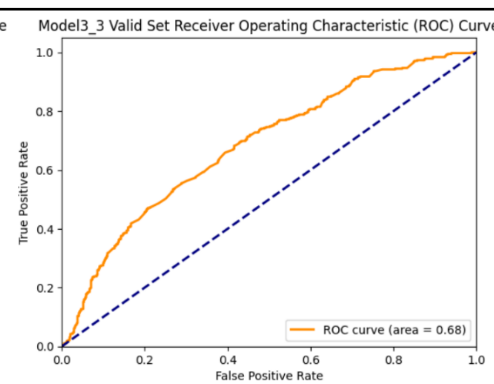
Following are validation set, ROC curves for Convolutional Neural Network.



Validation ROC curve of model3_1



Validation ROC curve of model3_2



Validation ROC curve of model3_3

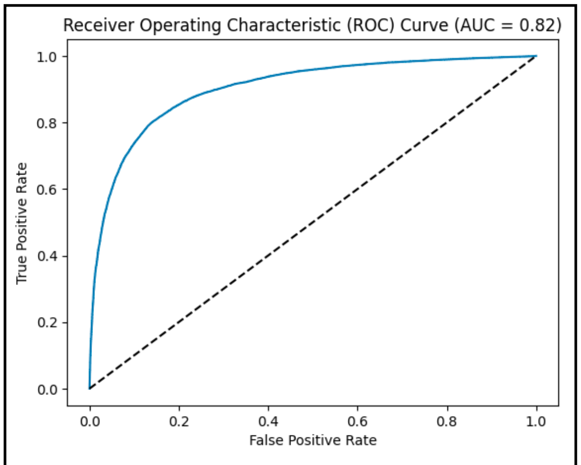
6.2 Comparison of 3 different algorithms

Following are the training and testing metrics for 3 algorithms that have been selected as best for particular algorithm and have been fitted on train + validation sets.

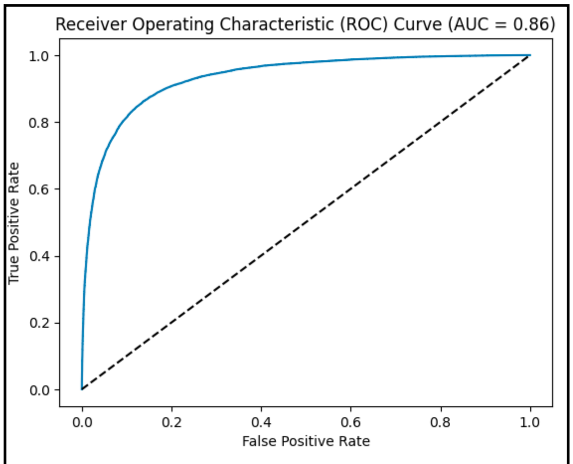
6.2.1 Training Metrics

	Accuracy	Precision	Recall	F1-score	Confusion Matrix
Logistics	0.836380	0.815965	0.763050	0.788621	[[26558, 3442], [4739, 15261]]
XGBoost	0.862780	0.812046	0.854800	0.832875	[[26043, 3957], [2904, 17096]]
ConvNN	0.937846	0.947998	0.893635	0.920014	[[29020, 980], [2127, 17873]]

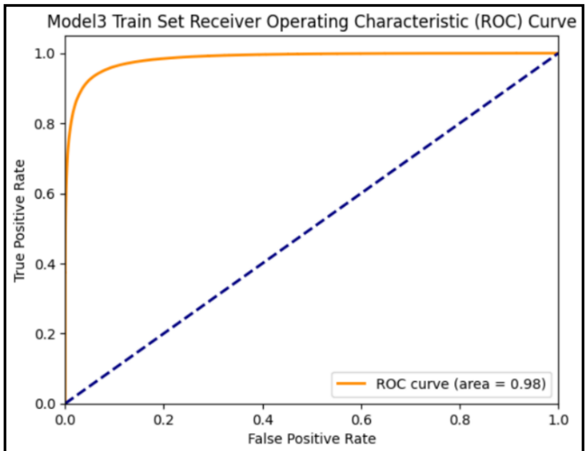
Following are the ROC curves:



Training ROC curve of model1



Training ROC curve of model2

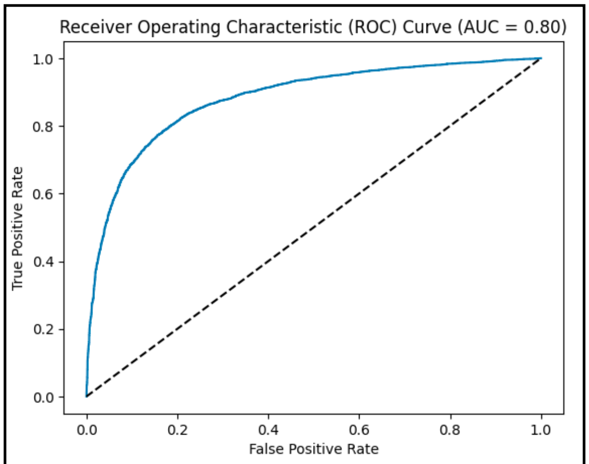


Training ROC curve of model3

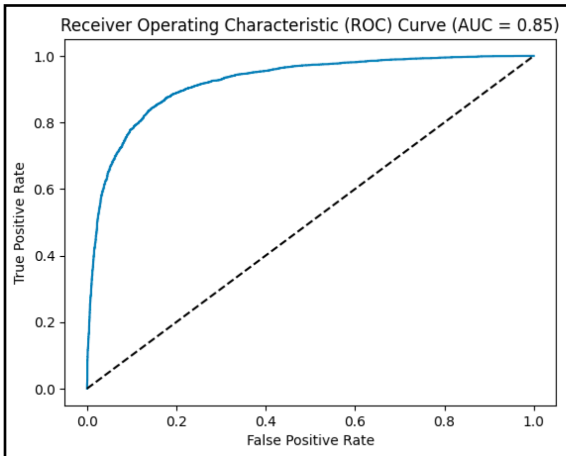
6.2.2 Testing Metrics

	Accuracy	Precision	Recall	F1-score	Confusion Matrix
Logistics	0.81660	0.791599	0.73500	0.762250	[[5226, 774], [1060, 2940]]
XGBoost	0.81660	0.791599	0.73500	0.762250	[[5226, 774], [1060, 2940]]
ConvNN	0.92412	0.946692	0.85865	0.900524	[[5807, 193], [565, 3435]]

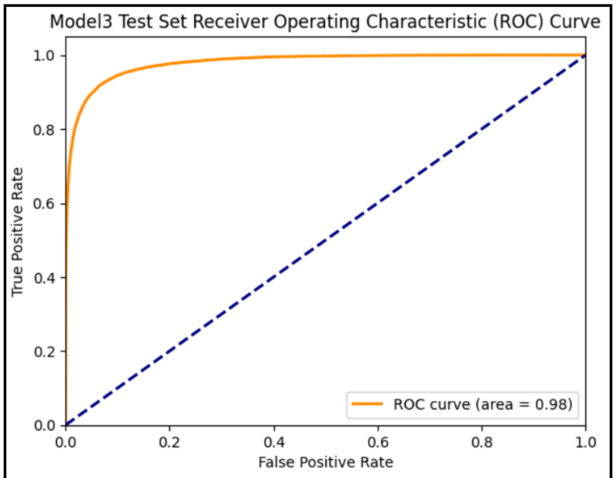
Following are the ROC curves:



Testing ROC curve of model1



Testing ROC curve of model2



Testing ROC curve of model3

7. Underfitting & Overfitting

7.1 Underfitting

A model is said to be underfitted when it fails to fit on training data. Such model gives poor performance on the training data and poor generalization to other data as well. Underfitted models have high bias and high variance.

From section 6 (Comparison of Results), it can be seen clearly that the models' train and validation metrics are providing good scores therefore the models have not been underfitted, except the 3rd model of Convolutional Neural Network (model3_3) due to the large values of learning rate, weight decay and batch size used, which gives an average of 72.9% accuracy.

7.2 Overfitting

A model is said to be overfitted when the data has been fitted too much. Such model gives a good performance on the training data, but poor generalization to other data, i.e., the model performs well in the training data but performs poorly in the test data.

From section 6 (Comparison of Results), it can be seen clearly the models that are not underfitted are able to perform well on validation and test data. Hence no overfitted models.

8. Conclusion

This report presents a detailed analysis of the implementation and evaluation of different machine learning models for image classification. The objective is to classify images into two categories animals and vehicles. The dataset used for training and testing the models is the CIFAR-10 dataset, consisting of 50,000 training images and 10,000 testing images. The CIFAR-10 dataset was downloaded from provided link and loaded into the models as appropriate variables. The 10 classes of the dataset were converted into 2 classes i.e. animals and vehicles by using Binary encoding, using 0 label for animals and 1 label for vehicles. The training set was further split into train and validation sets to evaluate the algorithms with different hyperparameters. Normalization on the images was done for RGB values of every pixels by simply dividing each pixel with 255 to normalize the image. Standardization was also performed on the CNN models by using 0.5 mean and 0.5 standard deviation values for RGB values of each pixel. Logistics Regression, XGBoost Classifier and Convolutional Neural Network were used to create 3 models of each algorithm to produce a total of 9 algorithms based on different hyperparameters values. Then the best model was chosen from each algorithm based on evaluation metrics, which leads us to the conclusion that the CNN model3_1 was the best performing out of all the 9 models.