University of Information Technology and Sciences (UITS)

Department Of CSE

Report on Vehicle Type Classification using SVM and CNN

Course Title: Machine Learning Lab

Course Code: CSE 432

Submitted By:

Name : Md Robiul Hasan

Student ID : 2125051088

Batch : 50

Section : 7B

Semester : Autumn

Submitted To:

Mrinmoy Biswas Akash

Ratri Datta

Lecturer, Dept. of CSE, UITS

Vehicle Type Classification using SVM and CNN

1. Introduction

The task of image classification involves assigning labels to images based on their visual content. In this project, we aim to classify images of vehicles into four categories: Bus, Car, Truck, and Motorcycle. Two machine learning approaches were employed: Support Vector Machine (SVM) with feature extraction and a Convolutional Neural Network (CNN). This report documents the dataset, methodology, results, and conclusions drawn from the experiments.

2. Dataset Description

The dataset used in this project was obtained from Kaggle and contains images of vehicles categorized into four types: Bus, Car, Truck, and Motorcycle. The dataset was preprocessed to resize images to 64x64 pixels. It was divided into training, validation, and testing sets:

- Training set: 320 images
- Validation set: 80 images
- Testing set: 80 images

Each category contains an equal number of images to ensure balanced classes.

3. Methodology

3.1 Preprocessing

- Images were resized to 64x64 pixels.
- Pixel values were normalized to the range [0, 1].

3.2 Feature Extraction for SVM

Histogram of Oriented Gradients (HOG) was used to extract features from the images. These features were then used to train an SVM classifier.

3.3 Convolutional Neural Network (CNN)

A CNN model was designed with the following architecture:

- **Conv2D (32 filters, kernel size 3x3, ReLU)** followed by **MaxPooling2D**
- **Conv2D (64 filters, kernel size 3x3, ReLU)** followed by **MaxPooling2D** • **Flatten** layer
- **Dense (128 units, ReLU)** with **Dropout (rate=0.5)**
- **Dense (4 units, softmax)** for classification

The model was compiled with the Adam optimizer and categorical cross-entropy loss. It was trained for 10 epochs.

3.4 Evaluation Metrics

- Accuracy
- Precision, Recall, and F1-Score

4. Results and Discussion

4.1 SVM Results

Class	Precision	Recall	F1-Score
Bus	0.00	0.00	0.00
Car	0.23	0.94	0.37
Truck	0.33	0.11	0.17
Motorcycle	0.00	0.00	0.00

Overall Accuracy: 24%

4.2 CNN Results

Class	Precision	Recall	F1-Score
Bus	0.71	0.19	0.30
Car	0.45	0.28	0.34
Truck	0.46	0.67	0.55
Motorcycle	0.47	0.94	0.63

Overall Accuracy: 49%

4.3 Discussion

The CNN outperformed the SVM in terms of accuracy and all other metrics. The ability of the CNN to learn hierarchical features directly from images contributed to its superior performance. SVM, relying on handcrafted features, struggled to capture the complex patterns present in the images. However, the CNN required significantly more computational resources and training time.

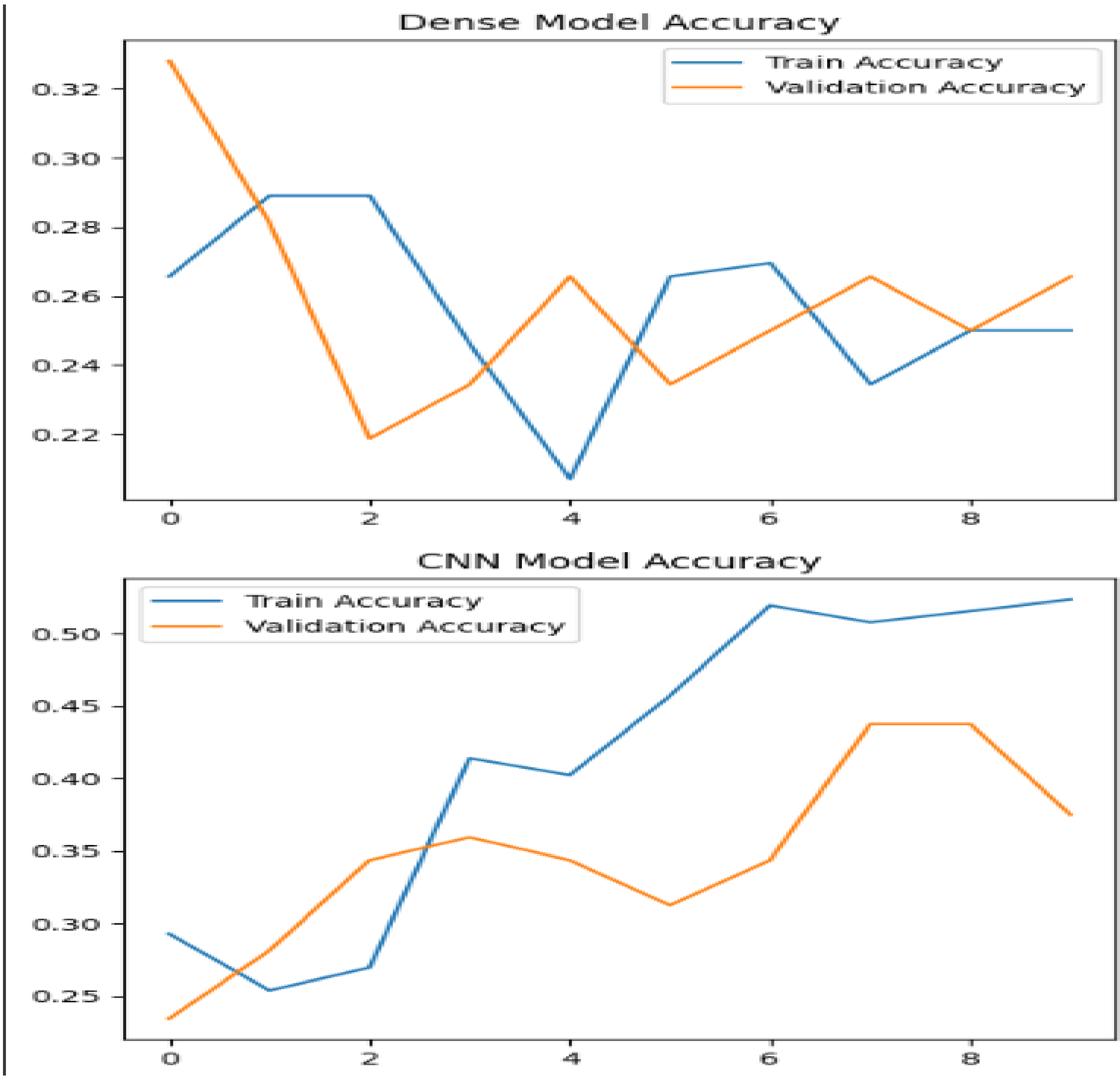
5. Conclusion

This project demonstrates the application of both traditional and deep learning approaches to image classification. While the SVM provided a baseline performance, the CNN significantly improved accuracy and other metrics. For future work, experimenting with more advanced architectures like ResNet or augmenting the dataset could further enhance performance.

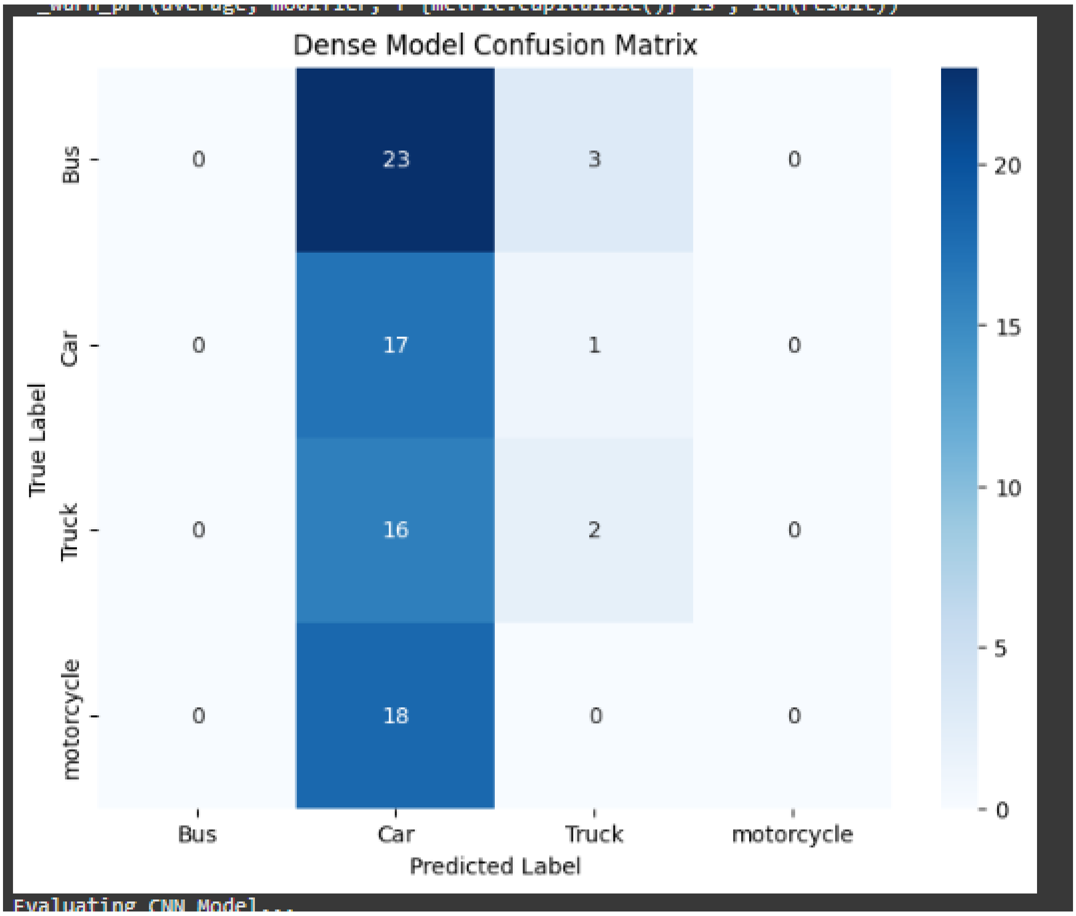
Key Takeaways:

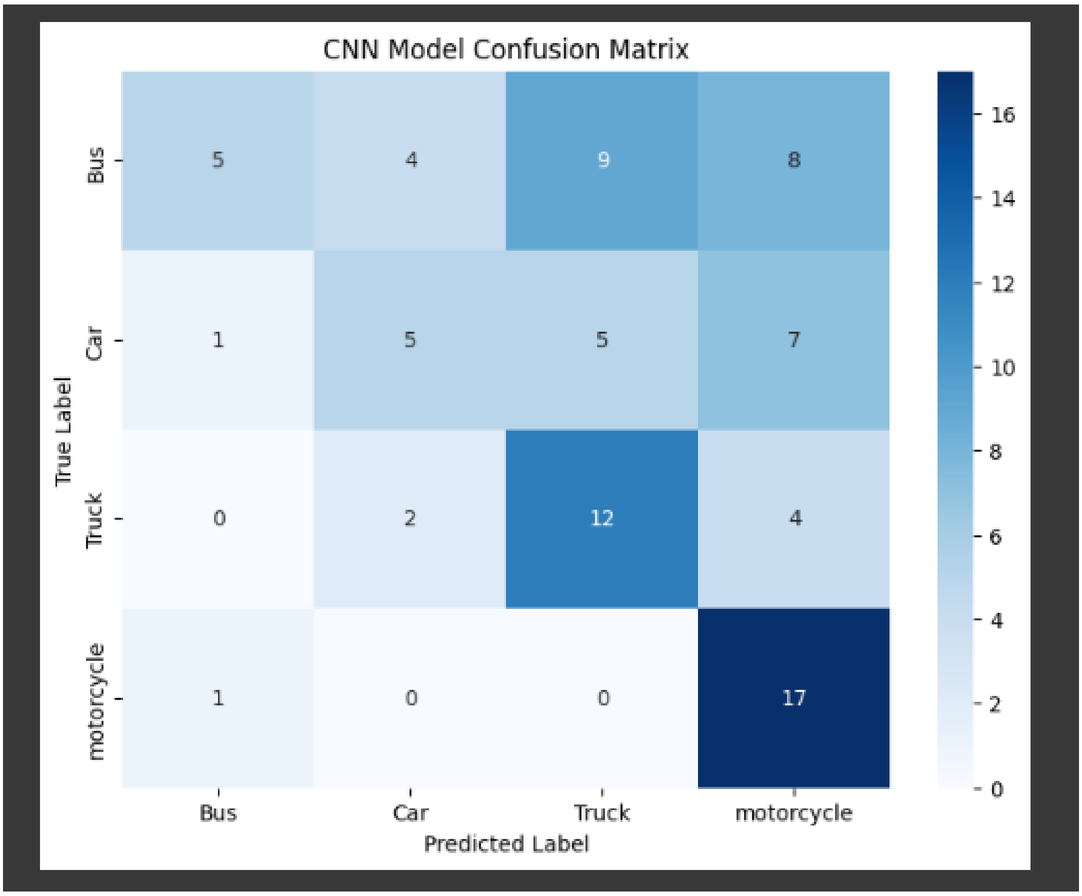
- Deep learning approaches like CNNs are more effective for image classification tasks.
- Preprocessing and feature engineering play a critical role in traditional machine learning methods.
- Model selection depends on the task requirements, computational resources, and available data.

6. Training and Validation Accuracy



7. Confusion Matrix


Dense Model Confusion Matrix
True Label
Predicted Label
Bus
Car
Truck
motorcycle
0
23
3
0
0
17
1
0
0
16
2
0
0
18
0
0
Evaluating CNN Model...


CNN Model Confusion Matrix
True Label
Predicted Label
Bus
Car
Truck
motorcycle
5
4
9
8
1
5
5
7
0
2
12
4
1
0
0
17

Colab Link:

https://colab.research.google.com/drive/1UutVwsnXBYUViokBCEgWXs-OUKk_yTC-#scrollTo=CRl96sVtTsYb

GitHub Link:

https://github.com/robiul-hasan/Machine-Learning