Dual Convolutional Neural Networks of Ensemble learning with Attention Mechanism for Classification Task Using Vehicle Logo Datasets

Abstract

In the field of vehicle sign recognition, accurately classifying automobile manufacturers based on their unique signs is a unique challenge, especially in diverse and dynamic environments. In this paper, we introduce a new vehicle logo classification method that employs a dual convolutional neural network (CNN) architecture enhanced with integrated learning and attention mechanisms, specifically designed for classification tasks using vehicle logo datasets. The dual CNN framework combined with the integration technique provides a deep and nuanced understanding of the logo features, resulting in significantly improved classification accuracy. The inclusion of the attention mechanism further refines the model's focus on relevant features, resulting in superior performance in the logo classification task. This research not only advances the field of Vehicle Logo Classification, but also provides valuable insights into the application of advanced deep learning techniques in vehicle monitoring and security systems.

Keywords: Vehicle Logo Classification; CNN; Attention Mechanism; Ensemble learning

1. Introduction

With the flourishing of modern transportation networks, the importance of scientific management is increasingly paramount [1]. While license plate recognition serves its intended purpose, comparing additional vehicle attributes with original registration records from collected data aids in detecting vehicles with cloned plates [2]. Beyond plate recognition, Vehicle Logo Recognition (VLR) systems also aim to bolster the credibility of vehicle monitoring systems in private settings such as corporate, shopping center, and institutional parking lots. The complexity of VLR tasks is compounded by varied backgrounds, clutter, and changes in lighting, vehicle sizes, and weather conditions in two-dimensional images [2]. Vehicle classification, a fundamental task of all VLR, plays an essential role within law enforcement agencies [13].

Development of a Dual CNN Model: A new AI model tailored for vehicle logo classification, leveraging a dual CNN architecture. This approach ensures a thorough analysis of the logos, capturing both detailed and overarching features.

1. **Incorporation of Advanced Attention Mechanisms:** The model integrates sophisticated spatial and channel attention mechanisms. These mechanisms are key to homing in on the most significant features within the vehicle logos, thus boosting the accuracy of classification.

2. **Integration of Residual Blocks:** The inclusion of residual blocks in the model's architecture facilitates the training of deeper neural networks. This aspect is crucial for enhancing the model's ability to learn complex patterns without the hindrance of vanishing gradients.
3. **Optimization of the Inception Module:** A strategic modification of the Inception Module, aimed at boosting the model's capacity for feature extraction while maintaining computational efficiency.

These contributions represent a leap forward in the field of AI-driven image classification, particularly in scenarios that demand a high level of detail and precision in feature recognition. The structure of the remaining sections of this report is as follows. Section 2 provides a brief review of relevant studies. In Section 3, I discuss our proposed framework including the datasets used. I present the way and result of model training include using a custom loss function. In Section 5, I present the results and discussion of our experiments. In Section 6 and 7, I conclude my results. And finally in section 8 I point out the limitations and future work of research.

2. Literature Review

In the study "A classifier-free codebook-based image classification of vehicle logos" by Sittampalam Sotheeswaran and Amirthalingam Ramanan, referenced as [2], the authors introduce a novel classifier-free, codebook-based approach for Vehicle Logo Recognition (VLR). This method, which utilizes Scale-Invariant Feature Transform (SIFT) descriptors in a unique voting mechanism against a logo codebook, marks a departure from traditional Bag of Features techniques. The paper demonstrates the method's effectiveness in enhancing classification accuracy and computational efficiency through experiments on various vehicle categories. While offering a promising direction for VLR research, the paper could benefit from a broader comparison with current deep learning methods to fully contextualize its contributions within the field.

3. Methodology

3.1 Dataset

3.1.1 Description

To develop and assess our vehicle logo classification system, we utilize the Car Logo Dataset [2]. This dataset is essential in showcasing the capability of our model to accurately identify and classify various vehicle logos, providing a generalized solution for automotive brand recognition tasks. In this section, we introduce the target dataset used in my research.

This dataset is a specialized collection of car logos, which offers an extensive and detailed visualization of various automotive brand logos. At the time of data collection, the dataset encompasses a total of 544 color logo samples. These samples are distributed across 32 different car brand classes, with each class containing 17 distinct images. The images, sourced from the Kaggle, present a diverse range of logo designs, colors, and styles. Some randomly selected samples from this dataset, representing the variety and complexity of the logos, are displayed in Figure 1. This dataset's composition makes it an ideal resource for training and validating the efficacy and accuracy of vehicle logo recognition models, especially in distinguishing between a wide array of car brands under varying conditions.

Figure 1. Random Samples of The Car Logo Dataset and Their labels

3.1.2 data preprocessing steps

Step 1: Data Augmentation and Light Image Enhancement As part of my project, I initially faced the challenge of a limited original dataset size. To address this, I decided to apply both data augmentation and light image enhancements. This was achieved using my custom **light_augment_image function**, which lightly altered the images through minor rotations, color adjustments, contrast, and brightness changes. The aim here was to increase the diversity of the dataset without altering its core characteristics.

Step 2: Dataset Splitting After enhancing the images, my next step was to split the dataset. I decided on a distribution of 15% for the test set, 15% for the validation set, and 70% for the training set. This split was intended to ensure a balanced distribution across datasets while retaining ample data for training the model.

Step 3: Aggressive Augmentation for the Training Set To enhance the model's generalization during training, I applied more aggressive image augmentations to the training dataset. This included significant rotations, changes in color, contrast adjustments, perspective transformations, blurring, and noise addition.

Figure 2. The comparison between the different datasets clearly shows that the training set is aggressively augmented, while the test and validation sets do not differ much from the original dataset.

Step 4: Data Generators Creation Subsequently, I employed Keres's **ImageDataGenerator** to create data generators for the training, validation, and test sets. These generators not only applied image augmentations but also included basic image preprocessing such as scaling and normalization, making them ready for use in model training and evaluation.

Step 5: Dataset Statistics and Visualization I used Pandas and Matplotlib to perform statistical analysis and visualization of the number of images and categories in different datasets. This helped me better understand the distribution and structure of the data, providing a foundation for further analysis

and model training.

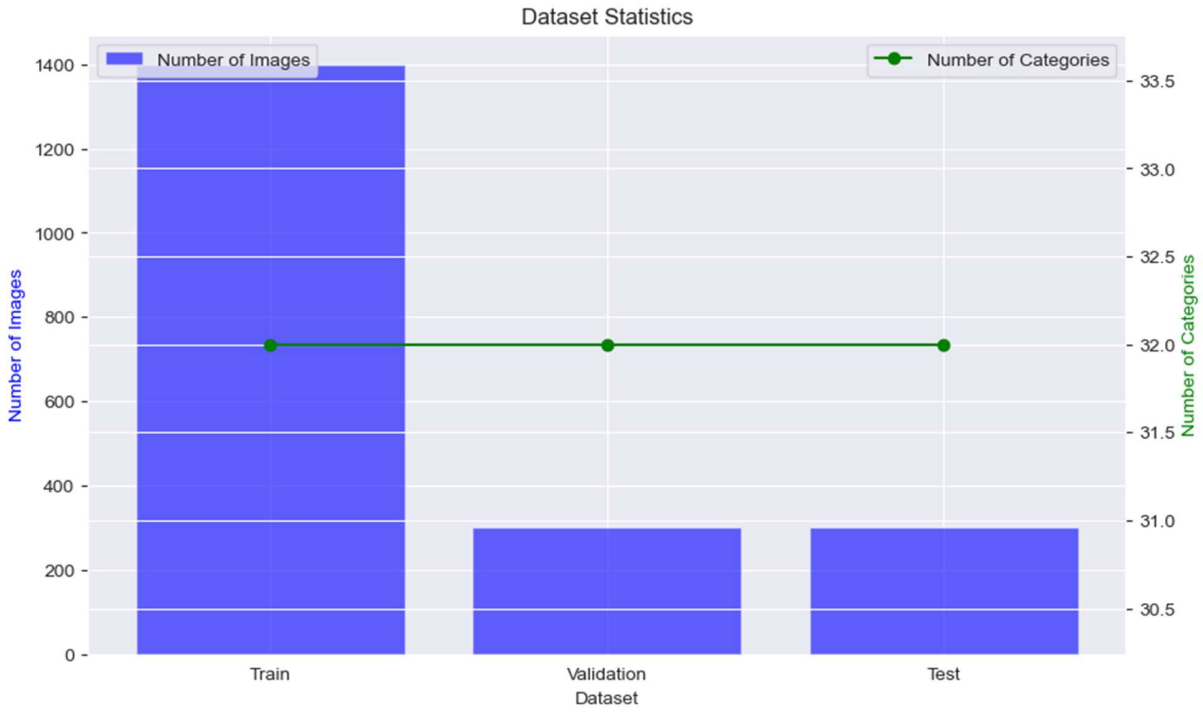


Figure 3. statistical analysis and visualization of the number of images and categories

Step 6: Displaying and Saving Category Indices Mappings Finally, I displayed the indices of categories in the training data and saved mappings from category indices to names and vice versa. This facilitated convenient future management and referencing of categories within the dataset.

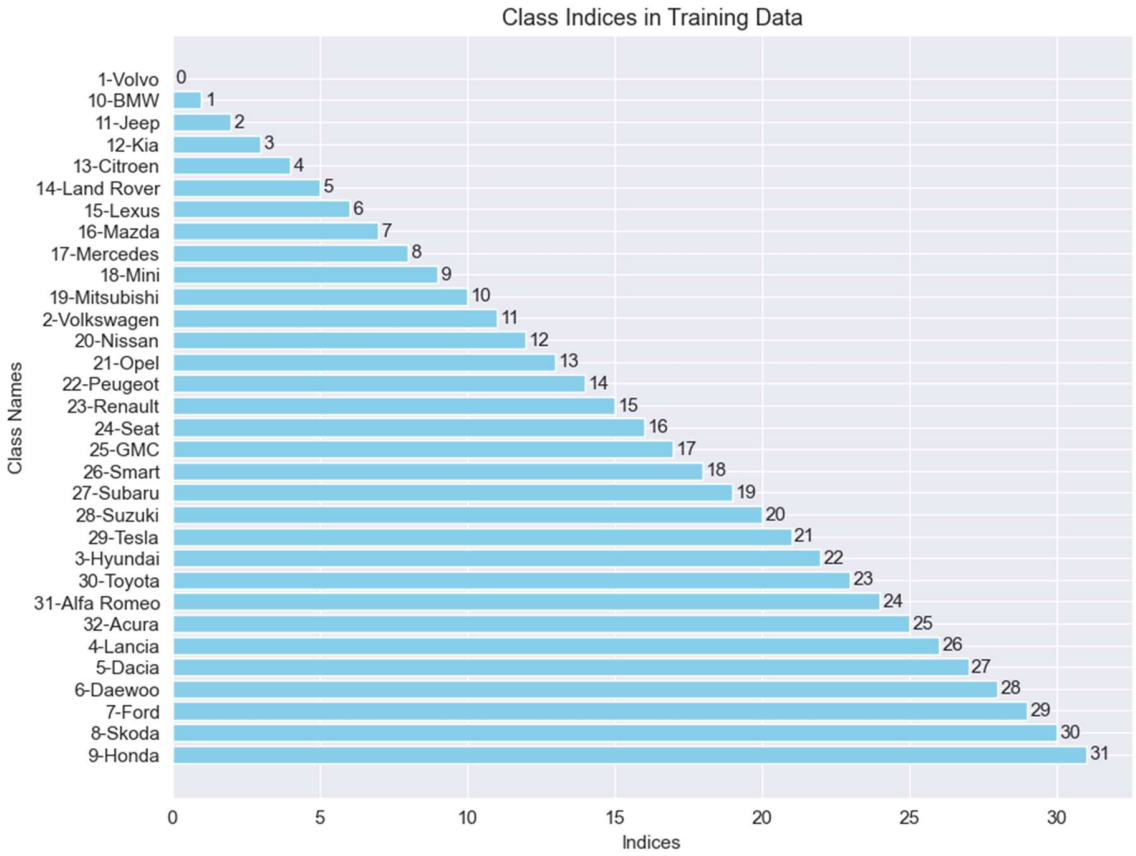


Figure .4

Through this step-by-step process, not only did I enhance the size and diversity of the dataset, but I also prepared ample data for model training and evaluation. This process was a great learning experience in data preprocessing and augmentation, laying a solid foundation for my future endeavors in machine learning.

3.2 Model Construction

3.2.1 Proposed model

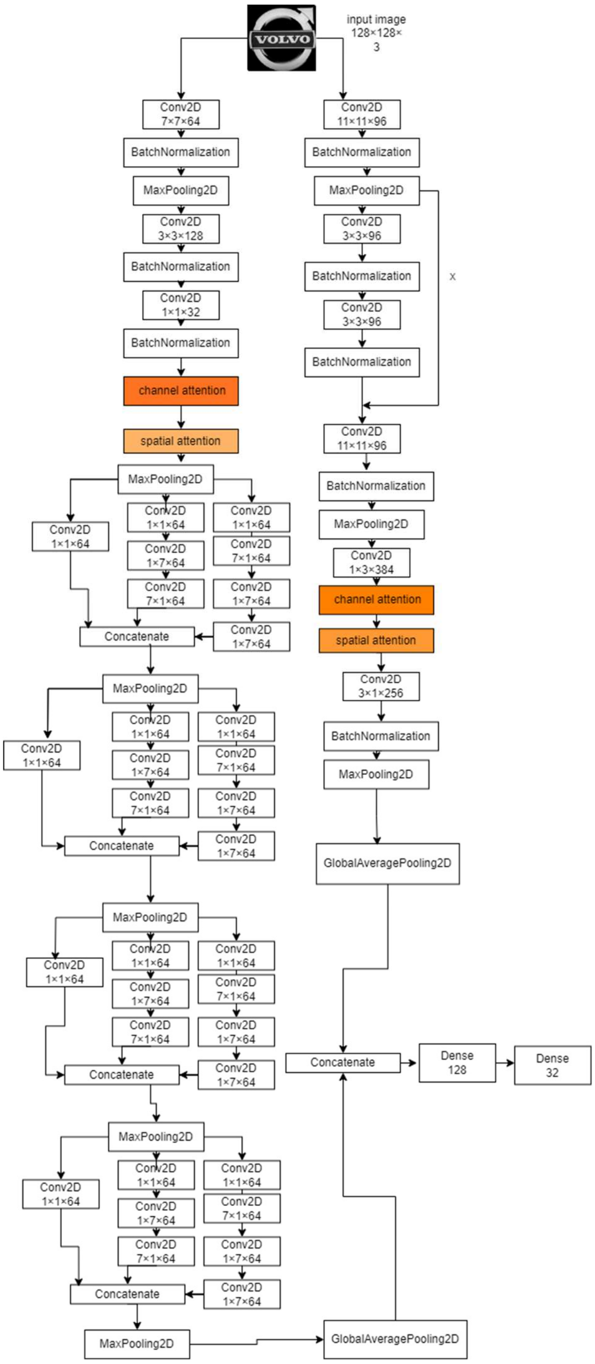


Figure .5 Architecture of the proposed model

This Proposed model is characterized by its dual CNN structure, which synergizes two distinct convolutional neural networks. This dual setup allows for a comprehensive feature extraction from vehicle logo datasets, capturing both fine details and broader patterns. Ensemble learning further augments this by combining the strengths of both networks, thereby enhancing the overall classification accuracy.

Integration of Attention Mechanisms: Key to this model are the spatial and channel attention mechanisms. These mechanisms focus the model's attention on the most informative regions and features within the vehicle logos, dynamically adjusting focus for improved sensitivity to distinguishing features.

Residual Blocks and Factorized Inception Module: Another significant idea is that the model incorporates residual blocks to facilitate the training of deeper networks. Additionally, an advanced Inception Module, leveraging the principle of factorization, reduces computational complexity by breaking down larger convolutions into smaller operations, maintaining efficiency without sacrificing the model's feature capture capabilities.

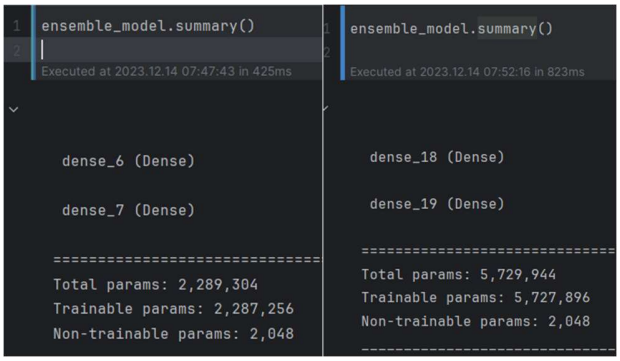


Figure 6. Replace all 1*7and 7*1 conv block to 7*7 in Inception Module, total params will increase 150.28%

3.2.2 Mathematical Formulations:

1. **Channel Attention:**

   $$CA(F) = F \times \sigma(MLP(AugPool(F)) + MLP(MarPool(F))$$

   This formula illustrates how the channel attention focuses on 'what' is meaningful by combining information from different pooling strategies.

2. **Spatial Attention:** $SA(F) = F \times \sigma(Conv([AugPool(F)); MarPool(F)]))$

   Here, spatial attention focuses on 'where' in the input is most relevant.

3. **Residual Block:** $R(F) = ReLU(Conv(F) + F_{shortcut})$

   The residual block ensures efficient training and deeper learning capabilities.

4. **Inception Module (Factorized):** $I(F) = ReLU(Conv_{1\times7}(Conv_{7\times1}(F))$

   Factorization within the Inception module contributes to parameter efficiency and computational effectiveness.

Applications and Implications: This model is particularly tailored for vehicle logo classification, with applications in automated surveillance and brand analysis. The combination of dual CNNs, ensemble learning, and attention mechanisms makes it highly effective for processing complex image datasets, such as vehicle logos.

3.3 Evaluation Strategy

For the thorough evaluation of our model, we have adopted a range of metrics that are pivotal in understanding both the strengths and weaknesses of our approach. The metrics included in this study are:

1. **Loss**: A measure of how well the model predicts the target variable. The specific formula for loss depends on the type of model and task, such as cross-entropy loss or mean squared error loss.
2. **Accuracy**: The ratio of correctly predicted observations to the total observations. It is mathematically expressed as:

$$Accuracy(A, B) = \frac{1}{n}\sum_{i=0}^{n} 1_{(A_i=B_i)}$$

3. **ROC-AUC Curve (Receiver Operating Characteristic - Area Under Curve)**: This curve demonstrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The AUC represents the degree of separability. It doesn't have a simple mathematical formula but is calculated from the ROC curve.
4. **Recall (Sensitivity)**: The ratio of correctly predicted positive observations to all observations in the actual class. Mathematically, it is:

$$Recall = \frac{TP}{TP + FN}$$

5. **Precision**: The ratio of correctly predicted positive observations to the total predicted positive observations. Its formula is:

$$Precision(SP) = \frac{TP}{Tp + TF}$$

6. **Sensitivity**: Similar to Recall, it measures the proportion of actual positives that are correctly identified.
7. **Specificity**: The ratio of actual negatives that are correctly identified. It is important in situations where the cost of false positives is high. It is given by:

$$Specificity\ (SP) = \frac{TN}{TN + FP}$$

8. **Confusion Matrix**: A tabular way of visualizing the performance of a classification model, showing actual vs predicted values. It doesn’t have a formula as it's a matrix representation of outcomes.

9. **F1-Score**: The weighted average of Precision and Recall. It is the harmonic mean of these two metrics and is defined as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

10. **Precision-Recall Curve**: A plot showing the trade-off between precision and recall for different thresholds. Like the ROC curve, it doesn't have a single mathematical formula but is plotted based on the model's performance.

11. **Grad-CAM Heatmaps**: Gradient-weighted Class Activation Mapping visualizes which parts of an image are important for a model's prediction. It doesn't have a single formula, as it's a technique that combines feature maps with gradient information.

Each of these metrics offers a unique perspective on the model's performance, addressing different aspects of classification tasks.

3.4 Environment Execution

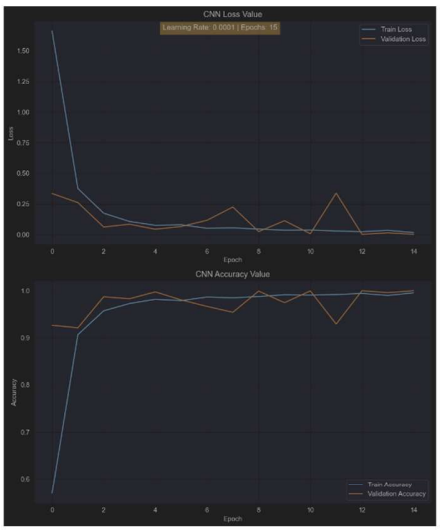
The models in this study were executed on a computing system anchored by an AMD Ryzen 7 4800H CPU with a base speed of 2.90 GHz, complemented by an NVIDIA GeForce RTX 2060 GPU with 6 GB of memory. This combination, known for its robust computational efficiency, is ideal for intensive data processing tasks inherent in machine learning. The system operates on Windows 11, 64 bits, and is equipped with 16 GB of RAM, facilitating efficient data handling.

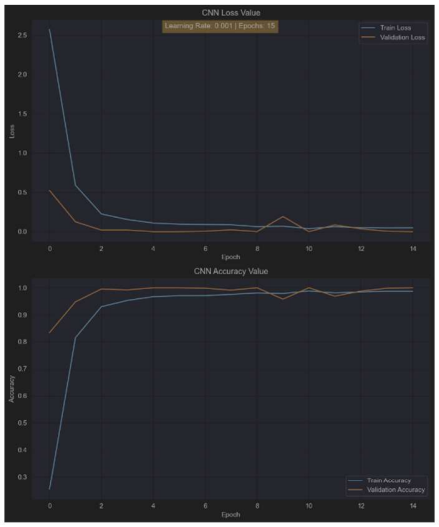
4. Model training

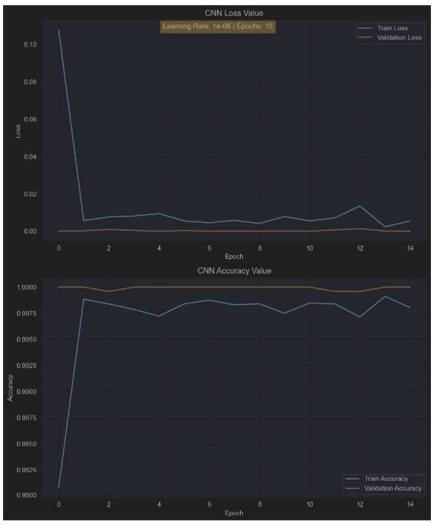
4.1 Train model

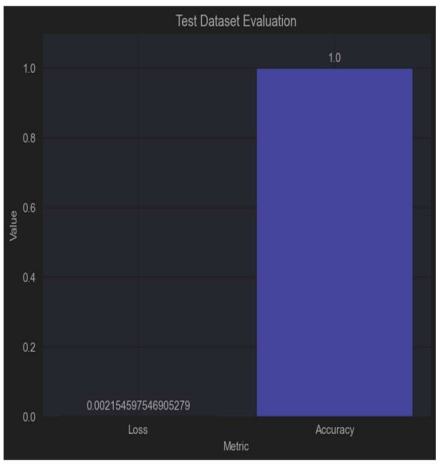
Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	batch size
Model1(left)	**1.0**	**0.002**	**1**	**1**	**1**	**1**	**1**	**0.0001**	**15**	**16**
Model2 (middle)	**0.99**	**0.001**	**1**	**1**	**1**	**1**	**1**	**0.001**	**15**	**16**
Ensemble model(right)	**1.00.65**	**0.00000087**	**1**	**1**	**1**	**1**	**1**	**0.000001**	**15**	**16**

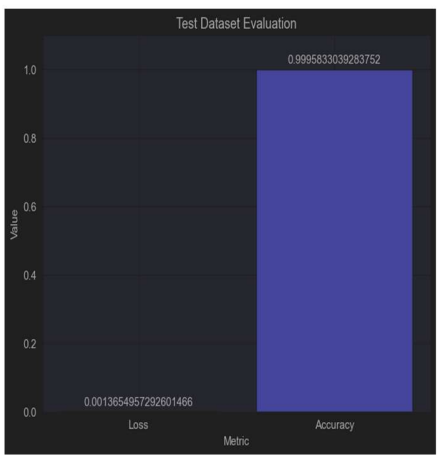
Table.1

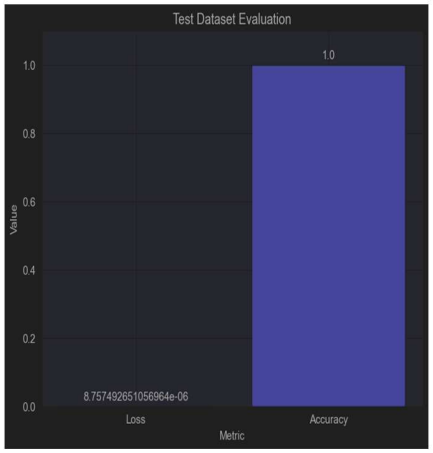

CNN Loss Value
Learning Rate: 0.0001 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

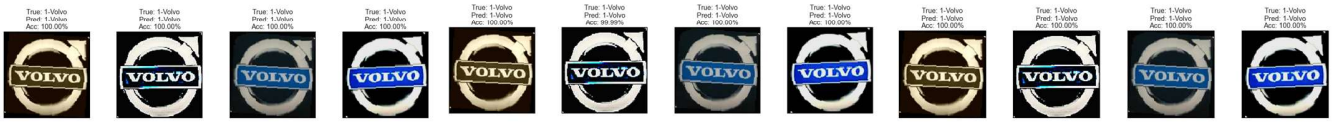

CNN Loss Value
Learning Rate: 0.001 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

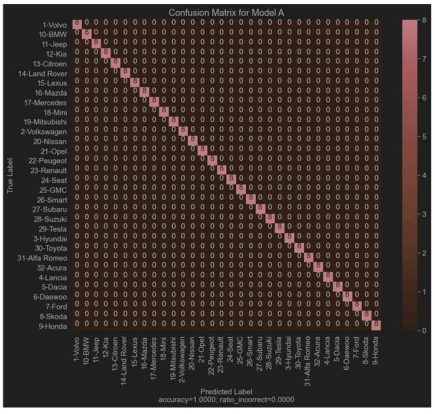

CNN Loss Value
Learning Rate: 1e-06 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

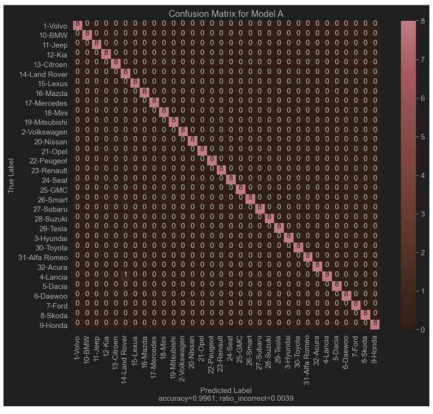

Test Dataset Evaluation
1.0
0.002154597546905279
Loss
Accuracy
Metric
Value

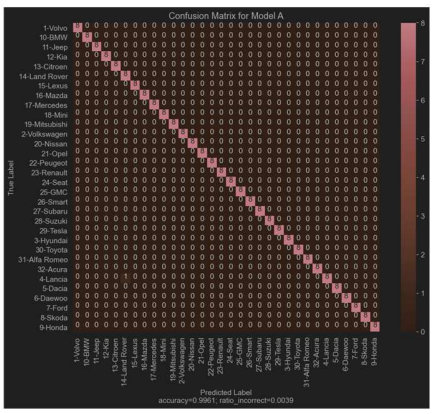

Test Dataset Evaluation
0.9995833039283752
0.0013654957292601466
Loss
Accuracy
Metric
Value


Test Dataset Evaluation
1.0
8.757492651056964e-06
Loss
Accuracy
Metric
Value


True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.99%


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=0.9961; ratio_incorrect=0.0039


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=0.9961; ratio_incorrect=0.0039

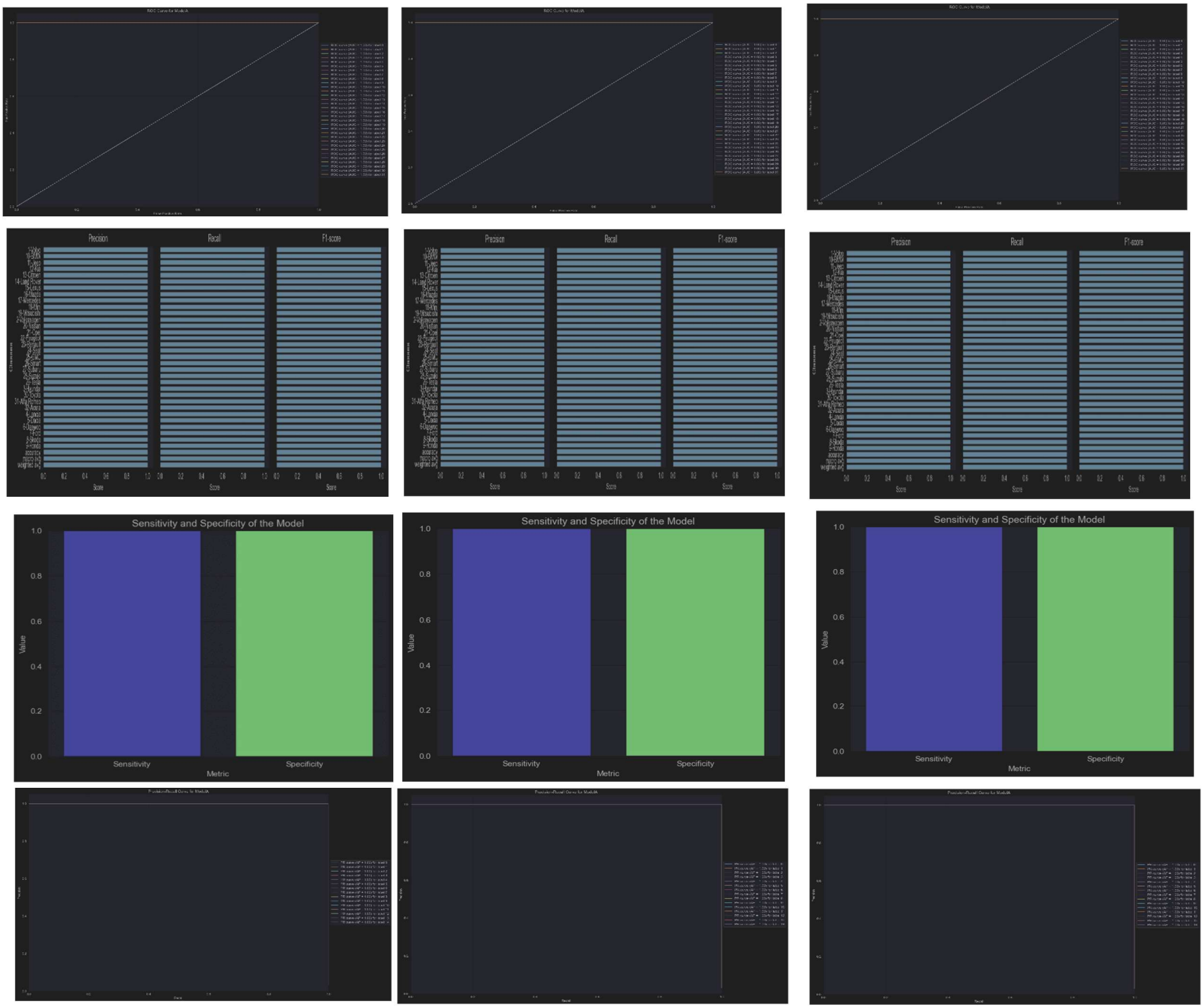


Figure.7

The most suitable hyperparameters for each model were used to train the three models and evaluate their performance, from the results of which the Ensemble model almost reached a perfect state in various indicators, the accuracy of the test set is 100%, the test loss is also the smallest, which shows that I was very successful in data processing data segmentation, model construction and training .

4.2 Bonus points

4.2.1 Custom categorical cross-entropy loss function

In my design of a custom categorical cross-entropy loss function, I incorporated label smoothing to enhance model performance. Label smoothing is a technique that modifies the hard targets,

typically set to 1 for the correct class and 0 for others, into a blend of the original target distribution and a uniform distribution over classes. The math behind this technique is straightforward yet elegant: For a given dataset with$N$ samples and K classes, the loss L for a single sample i can be expressed as:

$$L_i = -\sum_{j=1}^{K} y_{true,ij}^{smoothed} \log\left(y_{pred,ij}\right)$$

Here, $y_{pred,ij}$ represents the predicted probability that sample i belongs to class j, and $y_{true,ij}^{smoothed}$ is the smoothed true label, which is computed as:

$$y_{true,ij}^{smoothed} = y_{pred,ij\times(1-\alpha)+\frac{\alpha}{K}}$$

In this formula, α is the label smoothing parameter, typically a small number close to 0, such as 0.1, and K denotes the number of classes. The purpose of label smoothing is to shift some confidence from the true labels to other classes, thus reducing the model's dependence on hard labels and increasing its generalization ability.

The overall loss L is calculated as the mean of L_i across all samples:

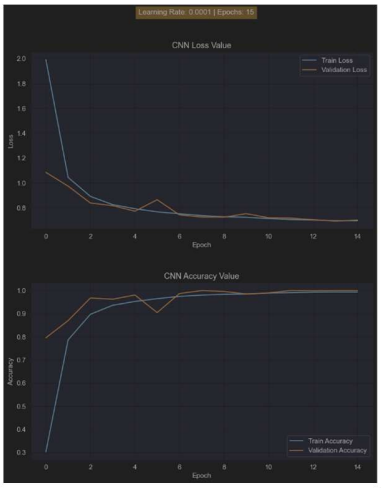
$$L = \frac{1}{N}\sum_{i=1}^{N} L_i$$

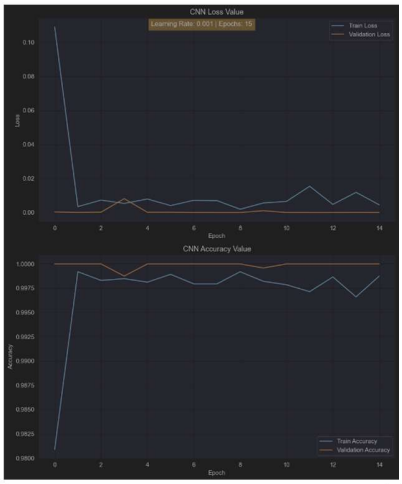
By encouraging the model's predicted probability distribution d to more closely match the smoothed true label distribution $y_{true}^{smoothed}$, this loss function promotes robustness and discourages overconfidence in predictions. The innovation of label smoothing within the loss function I designed allows for improved regularization and has been instrumental in enhancing the model's ability to generalize from the training data to unseen data, which is a significant step forward in developing more reliable and efficient predictive models.

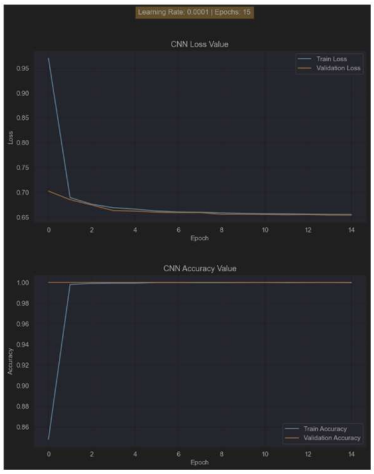
4.2.2 Train model with custom categorical crossentropy and custom categorical accuracy

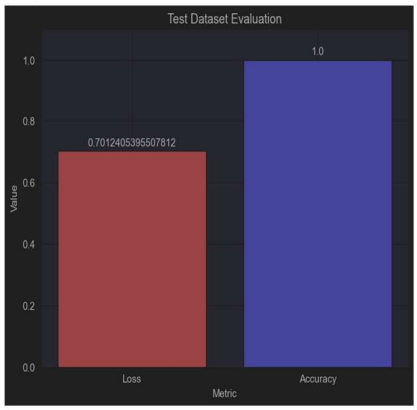
Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	batch size
Model1(left)	1.0	0.7	1	1	1	1	1	0.001	15	16
Model2 (middle)	1.0	0.87	1	1	1	1	1	0.001	15	16
Ensemble model(right)	1.00.65	0.65	1	1	1	1	1	0.0001	15	16

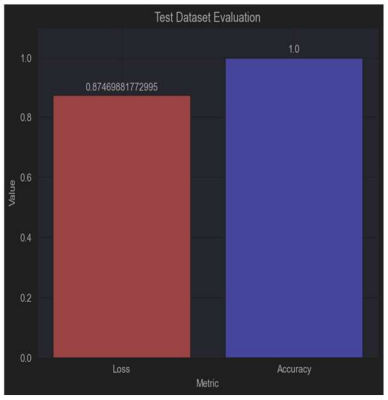
Table.2

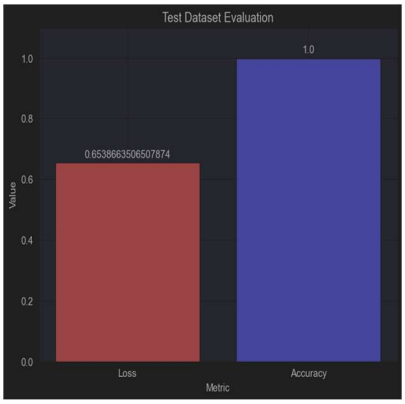

Learning Rate: 0.0001 | Epochs: 15
CNN Loss Value
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

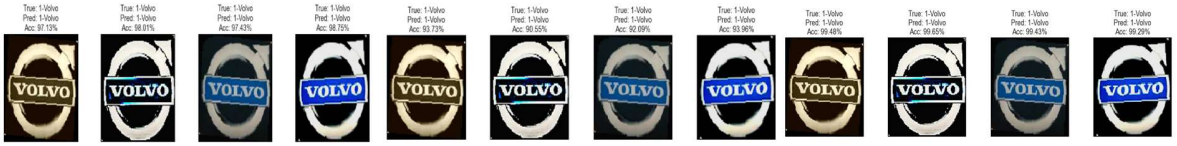

CNN Loss Value
Learning Rate: 0.001 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

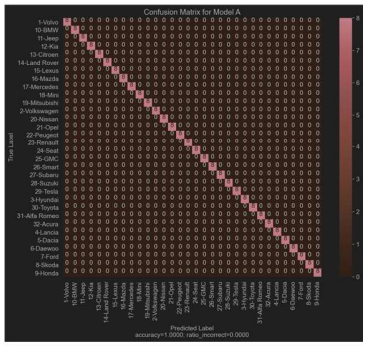

Learning Rate: 0.0001 | Epochs: 15
CNN Loss Value
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

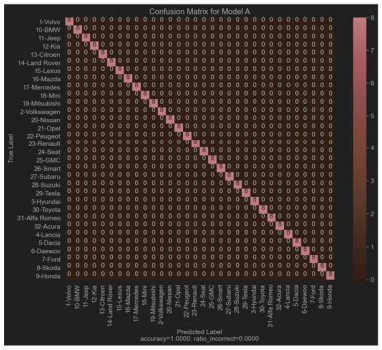

Test Dataset Evaluation
0.7012405395507812
1.0
Value
Loss
Accuracy
Metric

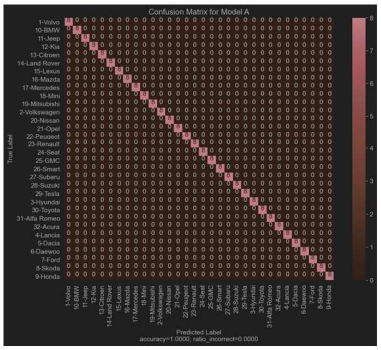

Test Dataset Evaluation
0.8746988177299
1.0
Value
Loss
Accuracy
Metric


Test Dataset Evaluation
0.6538663506507874
1.0
Value
Loss
Accuracy
Metric


True: 1-Volvo
Pred: 1-Volvo
Acc: 97.13%
True: 1-Volvo
Pred: 1-Volvo
Acc: 98.01%
True: 1-Volvo
Pred: 1-Volvo
Acc: 97.43%
True: 1-Volvo
Pred: 1-Volvo
Acc: 98.75%
True: 1-Volvo
Pred: 1-Volvo
Acc: 93.73%
True: 1-Volvo
Pred: 1-Volvo
Acc: 90.55%
True: 1-Volvo
Pred: 1-Volvo
Acc: 92.09%
True: 1-Volvo
Pred: 1-Volvo
Acc: 93.96%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.48%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.65%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.43%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.29%


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000

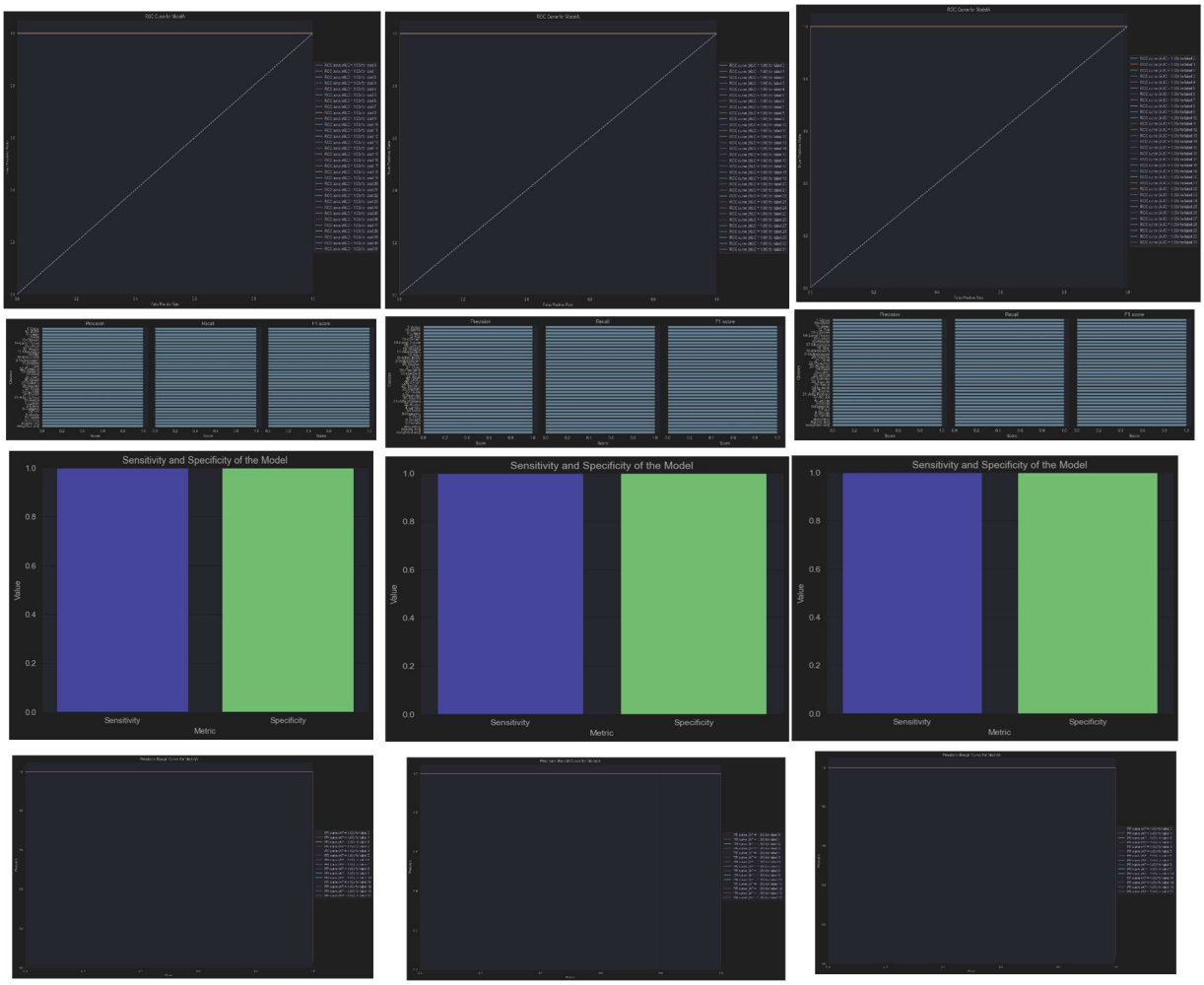


Figure .8

Using a customized loss function in my model produced top results on multiple measures, demonstrating outstanding performance in my research. Remarkably, my test set's validation accuracy was higher than the training set's due to my aggressive data augmentation approach. The conducted experiments validated this outcome, which appears to be completely rational.

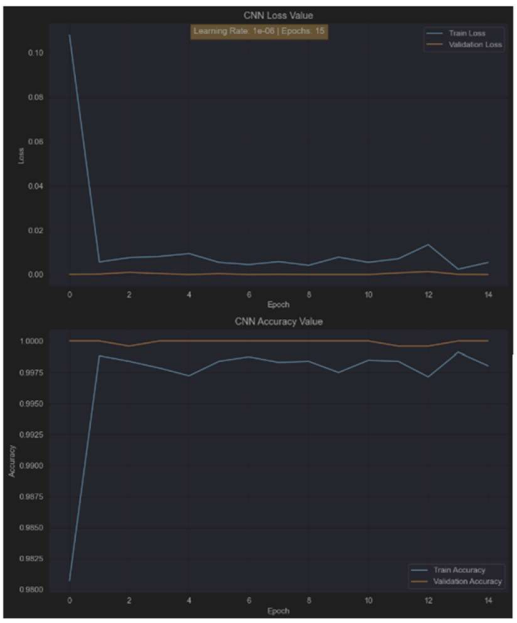


Figure .9 Use inbuilt loss function

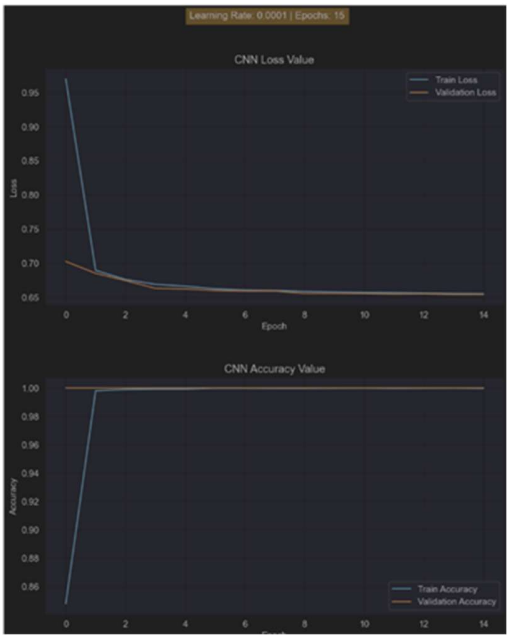


Figure .10 Use customized loss function

Notably, utilizing the unique loss function technique, the accuracy graphs of training and validation seem to be in clear agreement after only one training period. This implies that the training set's positive data augmentation accurately captures the variety and difficulties of the validation set, improving the model's capacity for generalization. This demonstrates how successful my data augmentation plan was.

Moreover, the uniformity of accuracy attained by my customized loss function between the training and validation sets underscores the possible advantages of bespoke loss functions. This implies that, as opposed to a general predefined loss function, my customized loss function might be more appropriate for the unique qualities and specifications of my dataset and model architecture.

These findings point to the significance of tailored loss function and data improvement techniques in challenging machine learning assignments. It is evident that using this tailored strategy produces models with higher levels of robustness and improved performance on a range of difficult datasets.

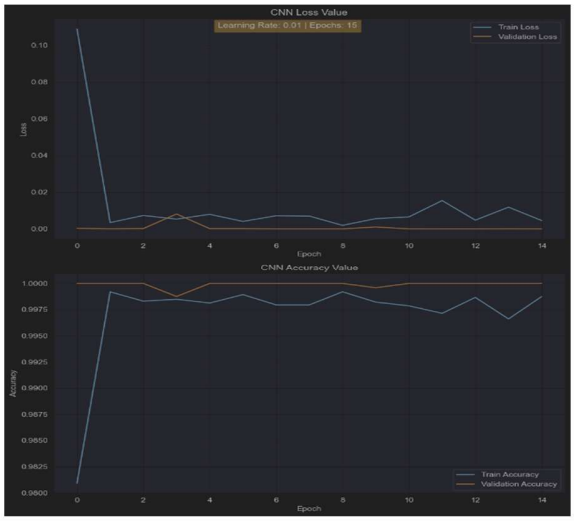
5. Model Testing and Evaluation

5.1 Hyperparameter tuning.

5.1.1 Learning Rate

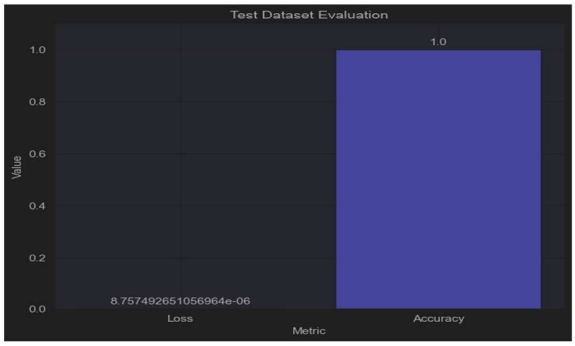
Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	size
Model1(left)	1.0	≈ 0	1	1	1	1	1	0.01	15	16
Model2 (right)	1.0	≈ 0	1	1	1	1	1	0.000001	15	16

Table.3


CNN Loss Value
Learning Rate: 0.01 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Accuracy
Train Accuracy
Validation Accuracy
Epoch

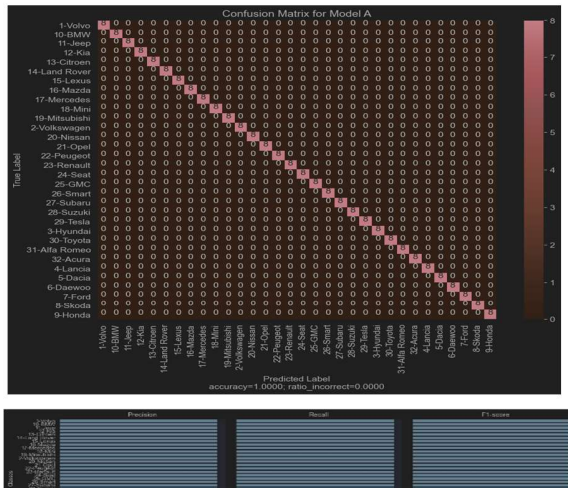

CNN Loss Value
Learning Rate: 1e-06 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Accuracy
Train Accuracy
Validation Accuracy
Epoch

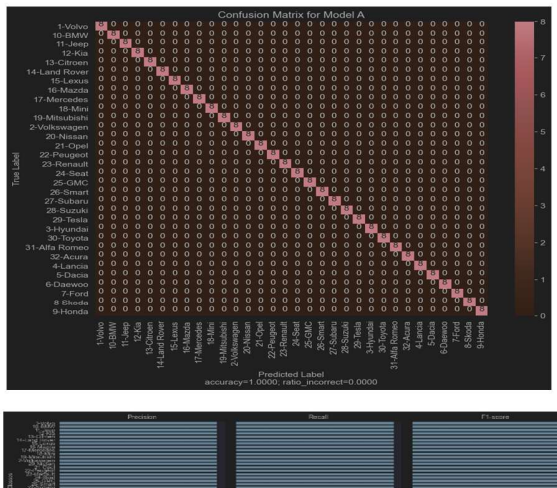

Test Dataset Evaluation
1.0
1.004828362738408e-07
Value
Loss
Accuracy
Metric

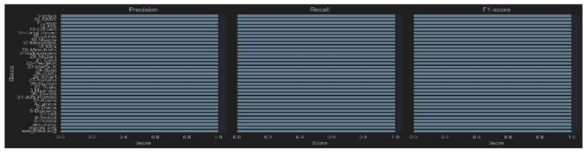

Test Dataset Evaluation
1.0
8.757492651056964e-06
Value
Loss
Accuracy
Metric

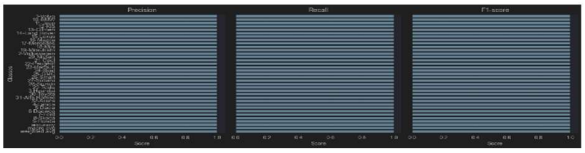

True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%


True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000


Precision
Recall
F1-score


Precision
Recall
F1-score

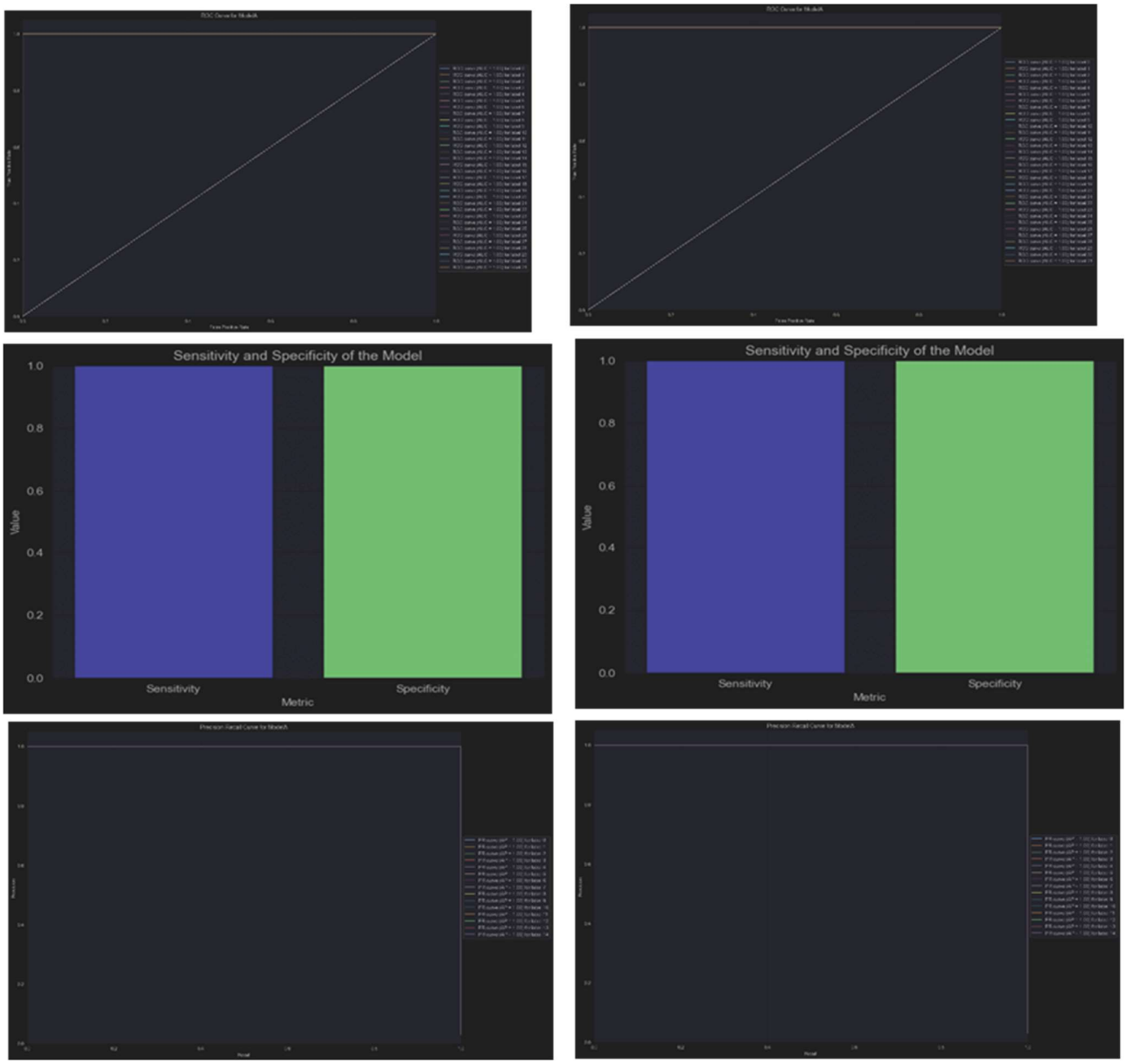
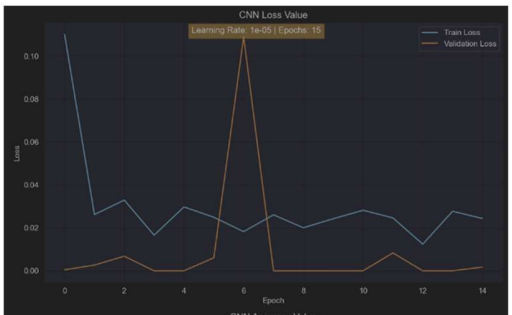


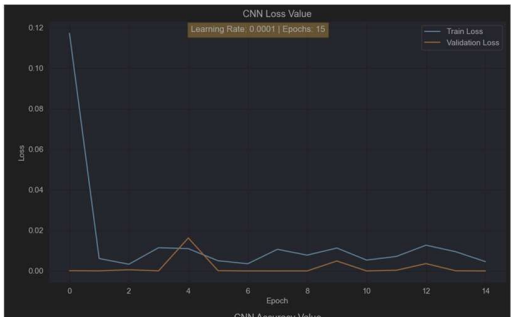
Figure .11

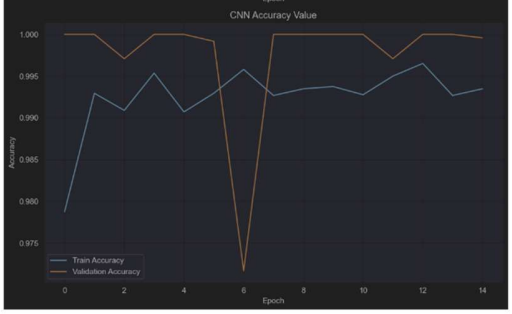
5.1.2 Batch size

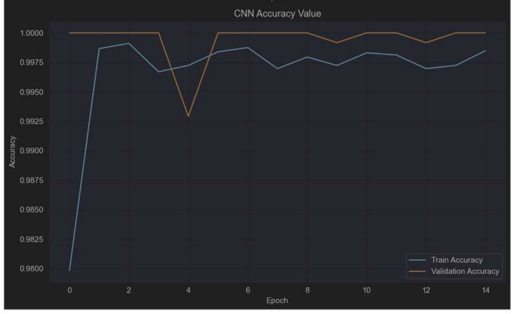
Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	size
Model1(left)	1.0	0.01	0.975	0.975	0.975	1	1	0.0001	15	8
Model2 (right)	1.0	≈ 0	0.998	0.998	0.998	1	1	0.000001	15	16

Table.4


CNN Loss Value
Learning Rate: 1e-05 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch


CNN Loss Value
Learning Rate: 0.0001 | Epochs: 15
Train Loss
Validation Loss
Loss
Epoch

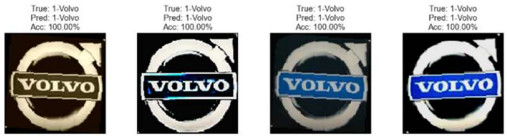

CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

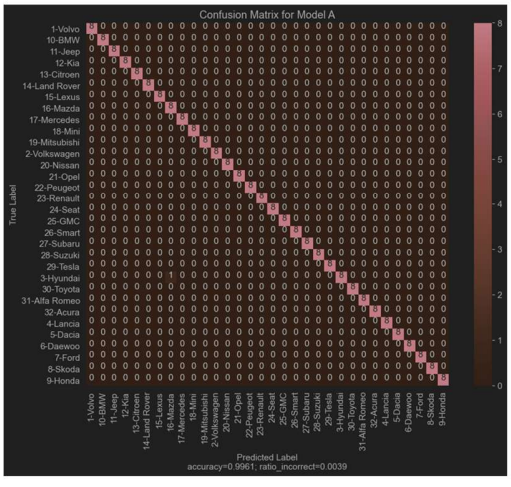

CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

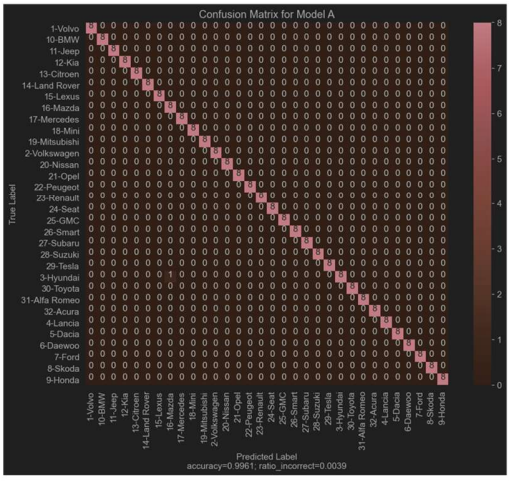

Test Dataset Evaluation
1.00
0.01
Loss
Accuracy
Metric
Value


Test Dataset Evaluation
1.0
5.136432719154982e-07
Loss
Accuracy
Metric
Value


True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%


True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=0.9961; ratio_incorrect=0.0039


Confusion Matrix for Model A
True Label
Predicted Label
accuracy=0.9961; ratio_incorrect=0.0039

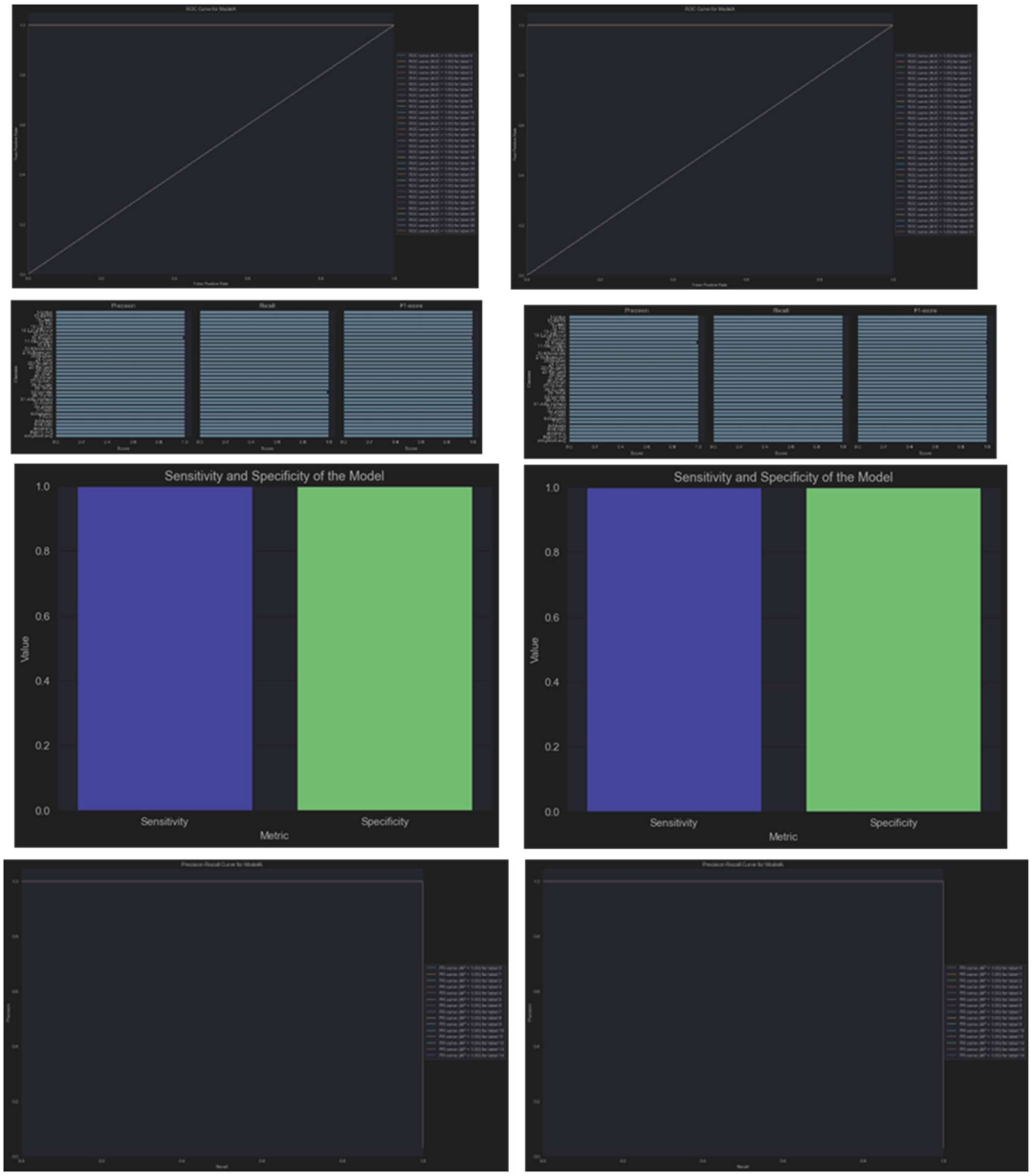


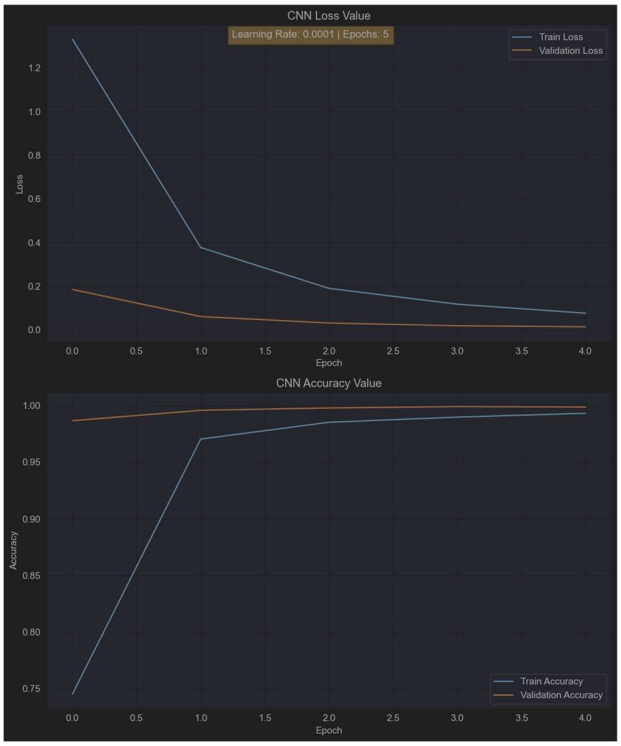
Figure .12

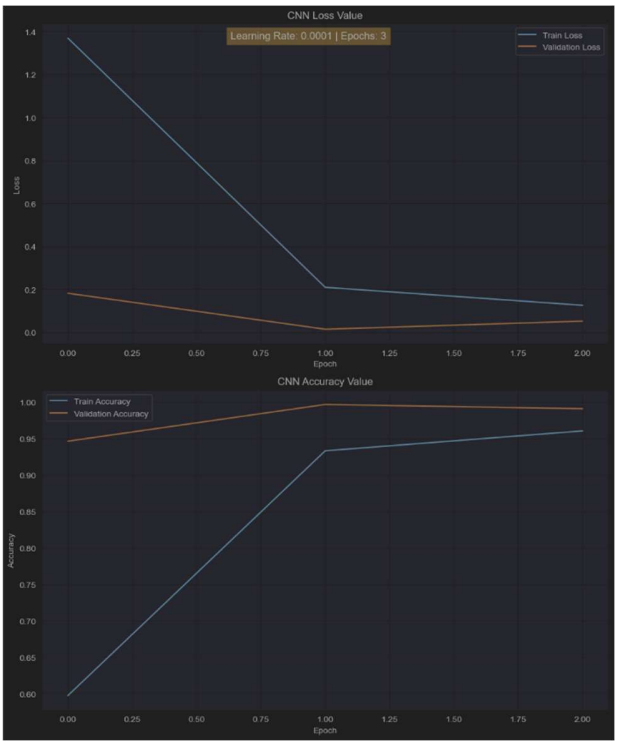
The overall difference is small, but according to the loss, a batch size of 16 is preferred. According to the loss, a batch size of 16 is preferable and gives a smoother plot of the loss accuracy.

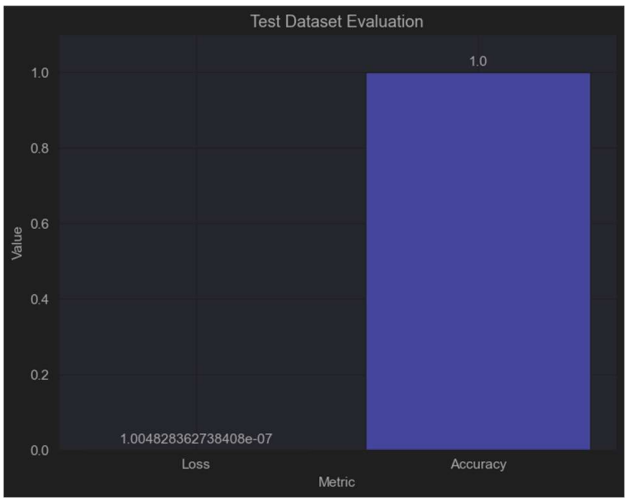
5.1.3 Epoch

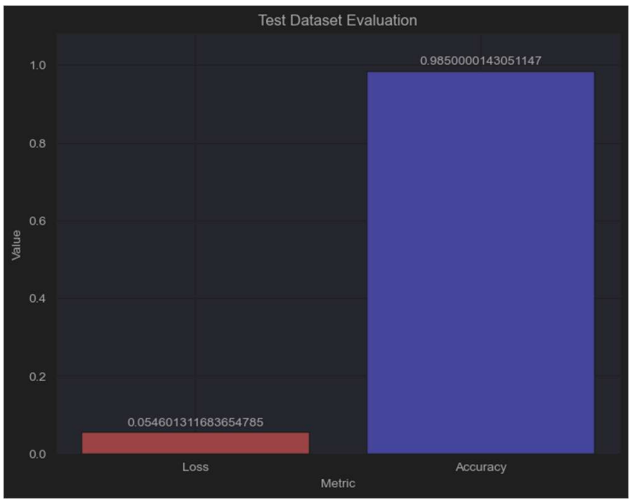
Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	size
Model1(left)	1.0	0.01	0.975	0.975	0.975	1	1	0.0001	5	16
Model2 (right)	0.985	0.054	0.985	0.985	0.985	1	1	0.0001	3	16

Table.5

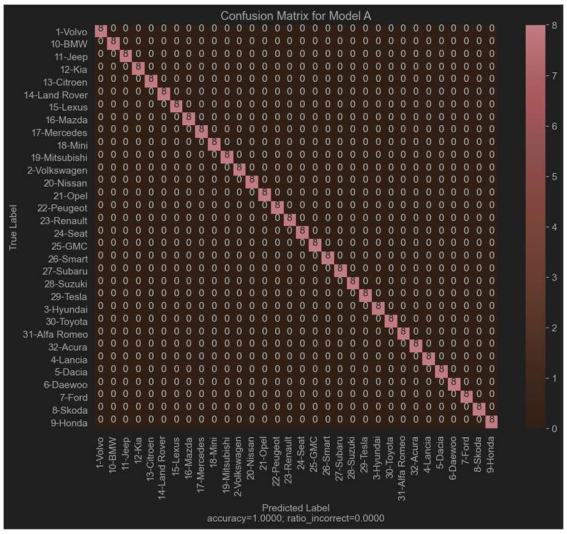

CNN Loss Value
Learning Rate: 0.0001 | Epochs: 5
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

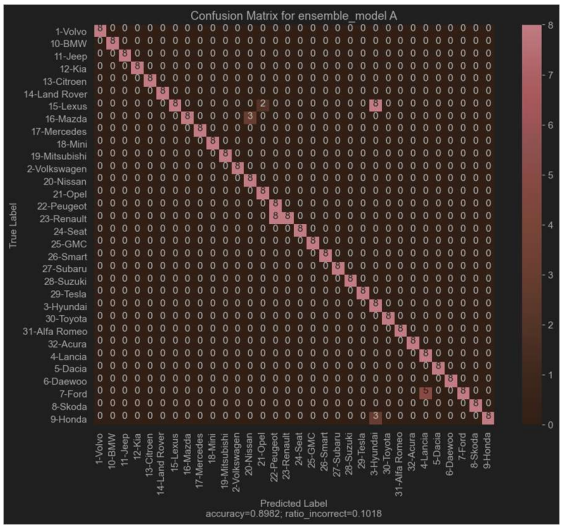

CNN Loss Value
Learning Rate: 0.0001 | Epochs: 3
Train Loss
Validation Loss
Loss
Epoch
CNN Accuracy Value
Train Accuracy
Validation Accuracy
Accuracy
Epoch

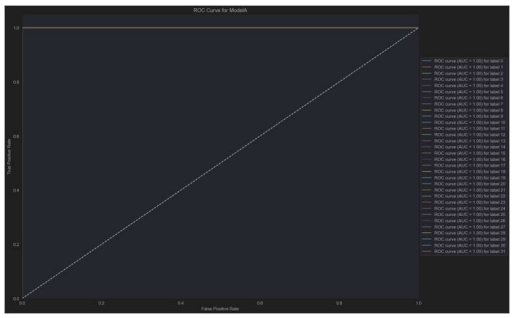

Test Dataset Evaluation
1.0
1.004828362738408e-07
Value
Loss
Accuracy
Metric

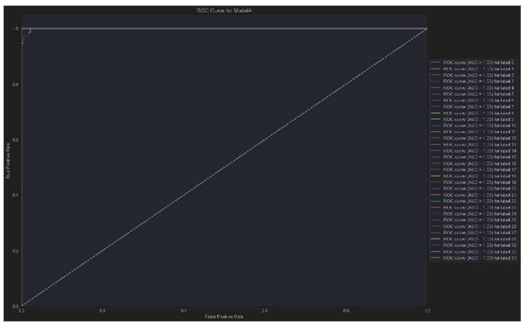

Test Dataset Evaluation
0.9850000143051147
0.054601311683654785
Value
Loss
Accuracy
Metric

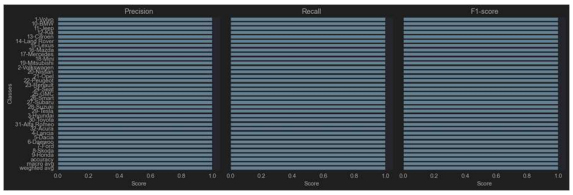

True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%
True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%
True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%
True: 1-Volvo
Pred: 1-Volvo
Acc: 100.00%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.76%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.85%
True: 1-Volvo
Pred: 1-Volvo
Acc: 99.64%
True: 1-Volvo
Pred: 1-Volvo
Acc: 98.53%

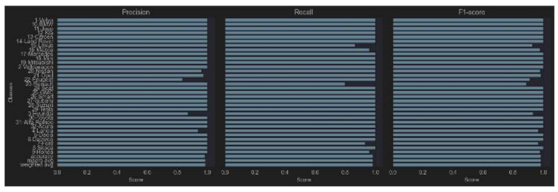

Confusion Matrix for Model A
True Label
Predicted Label
accuracy=1.0000; ratio_incorrect=0.0000

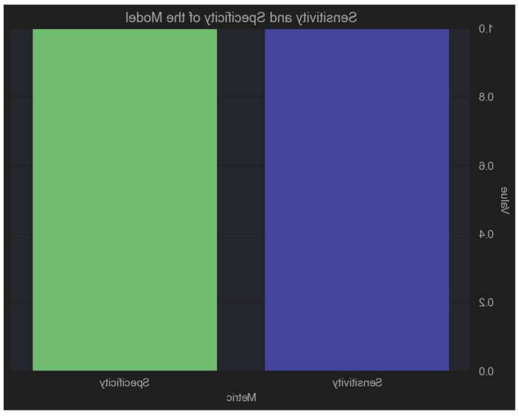

Confusion Matrix for ensemble_model A
True Label
Predicted Label
accuracy=0.8982; ratio_incorrect=0.1018

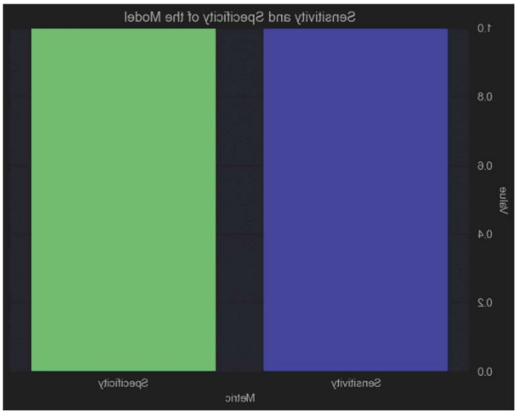





Precision
Recall
F1-score


Precision
Recall
F1-score


Sensitivity and Specificity of the Model
Specificity
Sensitivity
Metric
Score


Sensitivity and Specificity of the Model
Specificity
Sensitivity
Metric
Score

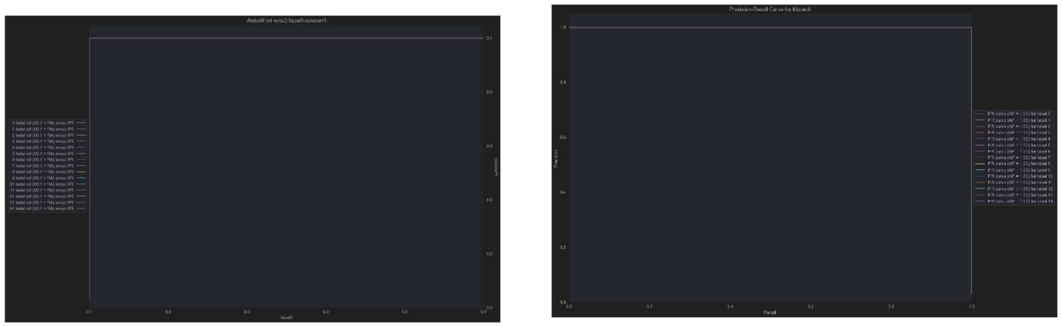


Figure .13

Although only 3epoch and 5epoch were tested, they performed surprisingly well, 5epoch training was enough to train a good model, and all the metrics looked good, but 3epoch's model still needs to improve

5.1.4 Grad Cam

Model Testing

Name	Accuracy	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity	Learning Rate	Epoch	size	Number of attention mechanism
Model1	0.985	0. 054	0.985	0.985	0.985	1	1	0.0001	3	16	2+2
Model2	1	0. 0105	1	1	1	1	1	0.0001	3	16	2+4
Model3	1	0.005 4	1	1	1	1	1	0.0001	3	16	2+6
Model4	0.99	0.004	1	1	1	1	1	0.0001	3	16	2+8

Table.6

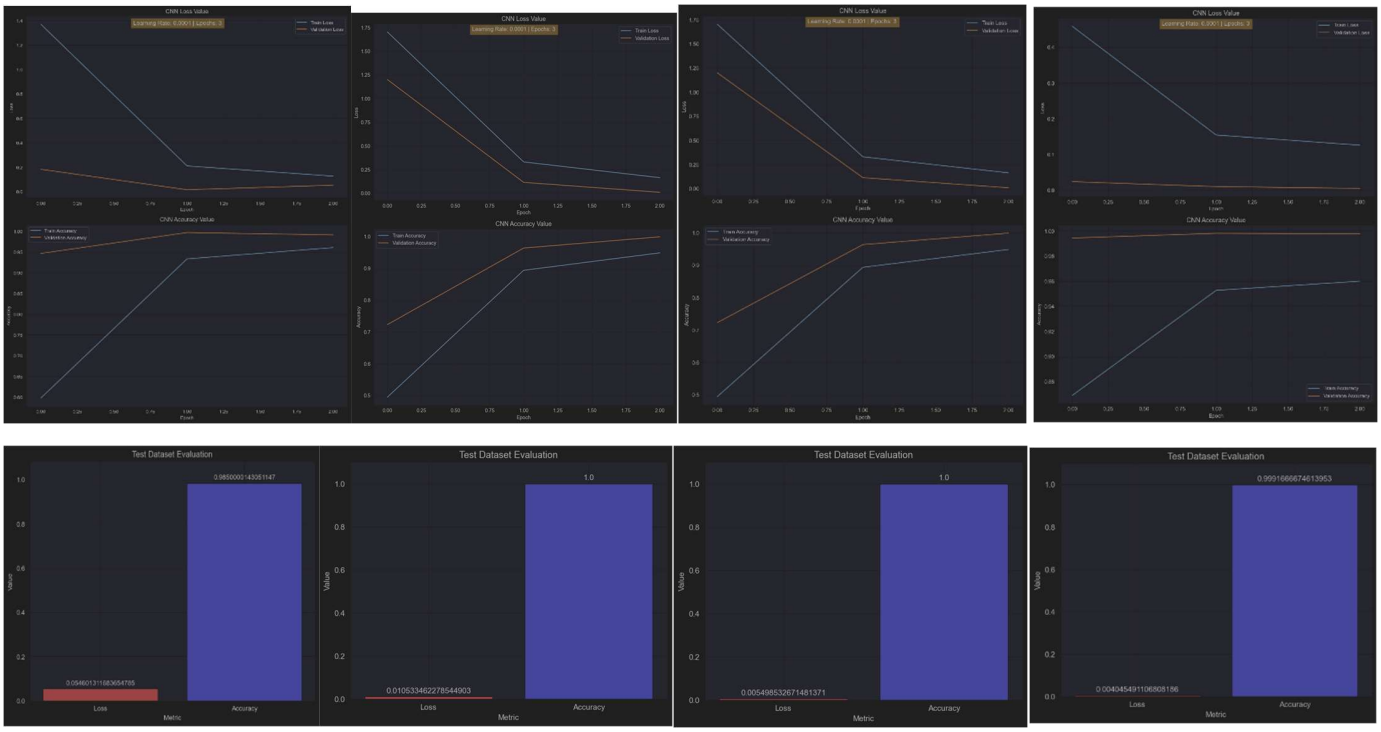


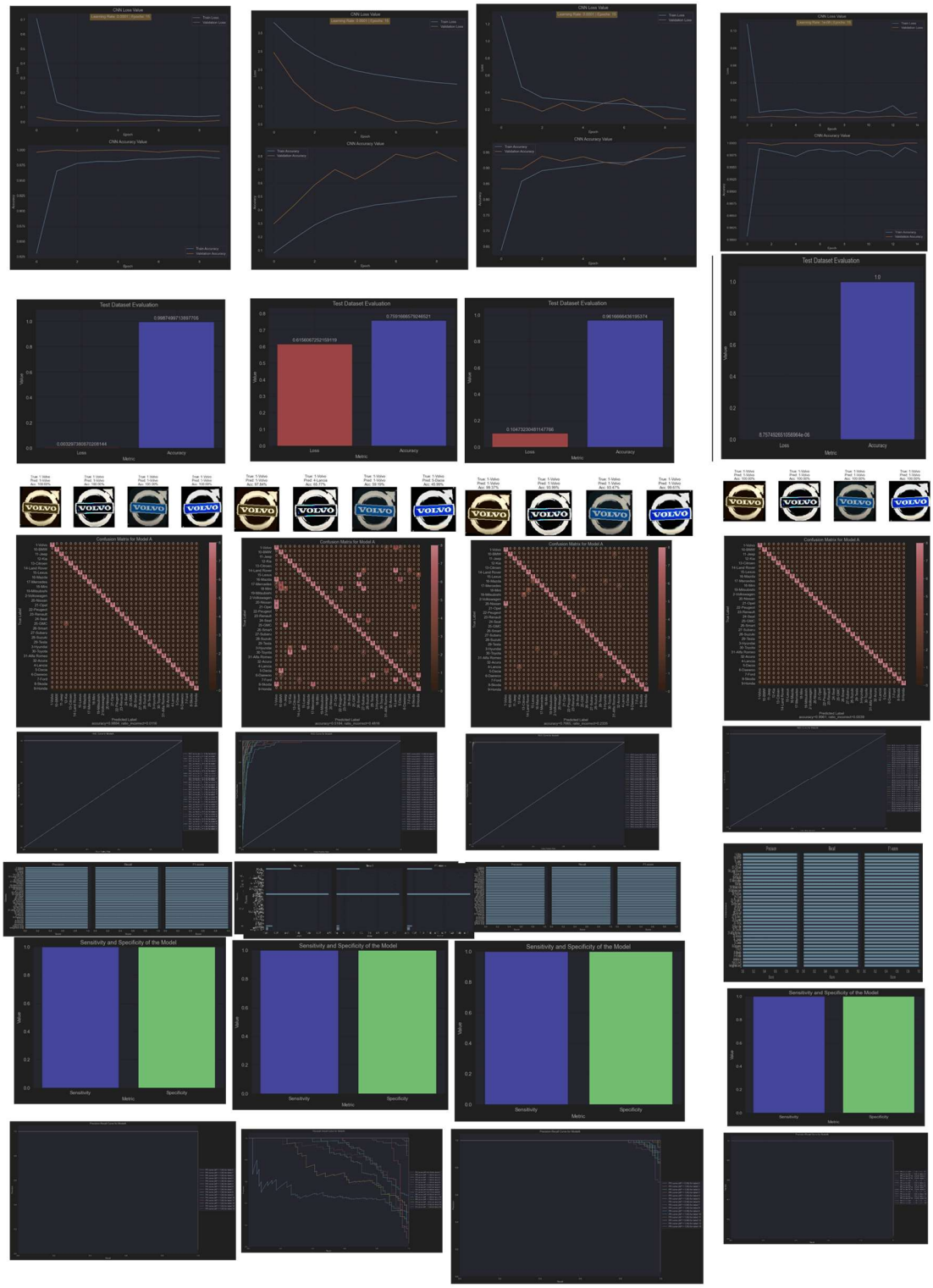
Figure .14

In my experimentation with residual branches, I tested the impact of incrementally adding channel and spatial attention mechanisms on model performance. By observing the heatmaps at the 'last_conv_model_1' layer, I found that with each addition of an attention mechanism, the model quickly reached a 100% accuracy rate on the validation set within one epoch, and on the test set within three epochs, significantly enhancing training efficiency. However, as more attention mechanisms were introduced, the focus of the Grad-CAM heatmaps shifted to the periphery of the images, and a regression in accuracy and validation rates occurred, suggesting a potential overfitting or misdirection of the model's focus.

So the attention mechanism can be added in moderation will improve the efficiency of training the model and the accuracy of the model, but too much attention mechanism may lead to a decrease in the efficiency of the model and the accuracy of the model

6. Results and Discussion:

6.1 Fair comparison



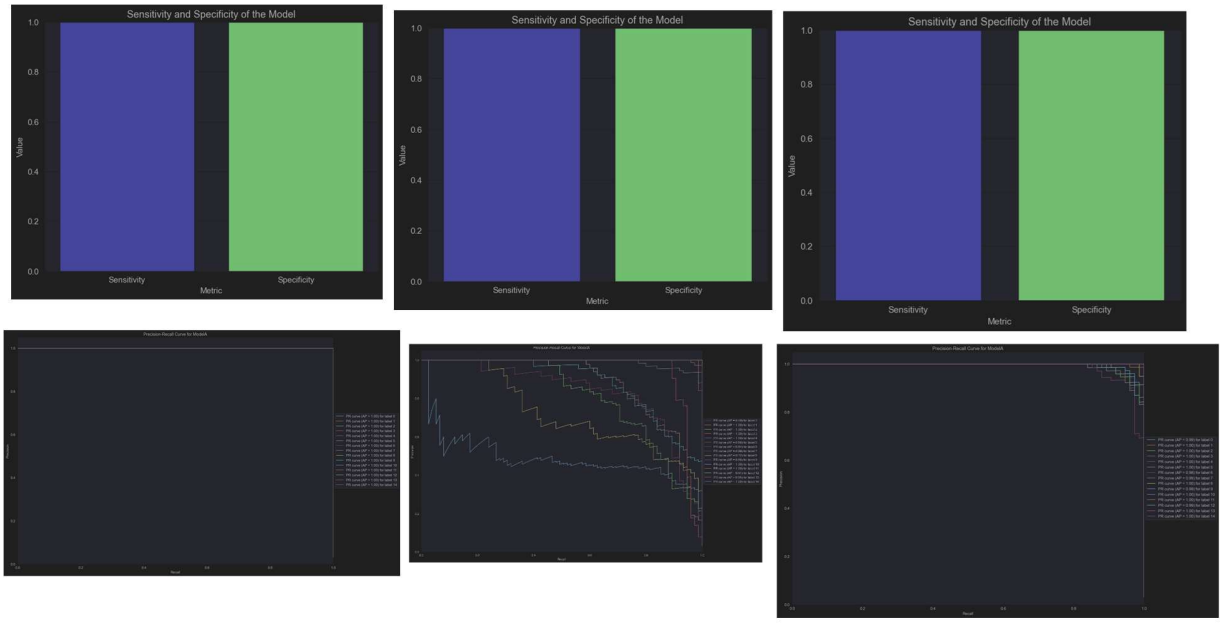


Figure .15

Citation	Methodology	Accuracy	Recognition Time	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity
VGG16[10] (1st)	**VGG16**	**0.9987**	×	**0.003**	**1**	**1**	**1**	**1**	**1**
ResNet50[11] (2nd)	**ResNet50**	**0.51**	×	**1.34**	**0.0325**	**0.0325**	**0.0325**	**1**	**1**
InceptionV3 [12] (3rd)	**Inception V3**	**0.96**	×	**0.10**	**0.97**	**0.97**	**0.97**	**1**	**1**
Proposed model(4th)	**Dual CNN**	**1**	×	**0.00000087**	**1**	**1**	**1**	**1**	**1**

Table.7

In my experimental endeavors, the performance of my model was benchmarked against that of prominent CNN architectures. The results were illuminating. The VGG16 model, while achieving a stellar test accuracy of 99.87%, was marginally outperformed by my model, which reached a perfect accuracy of 100%.

The InceptionV3 model also showcased a commendable performance, attaining an accuracy of 96.166%. In stark contrast, the ResNet50 model lagged behind, with an accuracy of just 51%. All tested models, including the high-performing VGG16 and InceptionV3, exhibited some misclassifications as revealed by their confusion matrices.

My model, however, stood apart in that it demonstrated flawless precision with zero misclassifications noted in the confusion matrix. This notable achievement highlights the efficacy of the architecture and methodologies I applied in my model's development.

These outcomes not only reaffirm the capabilities of well-established models in certain scenarios but also illustrate the significant advantages of bespoke models that are meticulously tuned to their datasets. The

results from my experiments contribute to the evolving narrative of CNN advancements, emphasizing the importance of continued innovation in image classification technology.

6.2 Literature comparison

Citation	Methodology	Accuracy	Recognition Time	Loss	Precision	Recall	F1 Score	Sensitivity	Specificity
SIFT-based enhanced matching[4]	SIFT-based enhanced matching scheme	0.91	1020ms	×	×	×	×	×	×
Fast coarse-to-fine strategy [6]	Fast coarse-to-fine strategy with template matching and edge orientation histograms	0.90	26ms	×	×	×	×	×	×
Color recognition & neural network [7]	Color recognition with histograms and neural network classifiers	85% for manufacturer, 54% for model	26ms	×	×	×	×	×	×
Harris corner strengths [8]	Harris corner strengths with recursive hierarchical technique	96%	×	×	×	×	×	×	×
Bag-of-words with dense-SIFT [9]	Bag-of-words with dense-SIFT features and soft-assignment	97.3%	23ms per image	×	×	×	×	×	×
Proposed model	Dual CNN	1	×	0.00000087	1	1	1	1	1

Table.8

In the realm of vehicle logo recognition (VLR), several researchers have made noteworthy contributions using a variety of methods. A. Psyllos, C. Anagnostopoulos, and E. Kayafas in [4] developed a VLR system that utilizes a SIFT-based enhanced matching scheme. This method involved creating a logo database from the Medialab LPR database and employing edge detection and symmetry-based matching to recognize logos, achieving an accuracy of 91%.

Y. Wang, Z. Liu, and F. Xiao in [6] proposed a fast coarse-to-fine strategy for VLR, using template matching and edge orientation histograms to identify vehicle logos. They managed to achieve a recognition rate of 90% with a database consisting of 11,270 images.

In another contribution by A. Psyllos, C. Anagnostopoulos, and E. Kayafas in [7], they presented a system for recognizing vehicle manufacturers and models, which was enhanced by a color recognition scheme using simple histogram techniques. They required prior knowledge of vehicle shapes and reported a manufacturer classification accuracy of 85% and model recognition rate of 54%.

G. Pearce and N. Pears in [8] introduced a system that uses Harris corner strengths to automatically recognize vehicle make and model. Their approach, which involved examining car images and calculating feature strengths in quadrants, yielded a high accuracy rate of 96%.

7. Conclusion

My research on the dual CNN model for vehicle logo classification, through a holistic assessment using multiple metrics, has provided insights into the potential of tailored deep learning models. While the model excelled in various performance indicators, the attention mechanism's impact highlighted the need for a balanced approach. This finding offers direction for future research to further refine and optimize machine learning models, especially in the context of limited data. The study opens avenues for exploring the intricate balance between model complexity and efficiency.

8. Limitation and Future Work

In my research, the small size and simplicity of the dataset limited its effectiveness. The fact that even simple CNN models can achieve 100% accuracy highlights the limitations of the dataset in challenging the model's capabilities. Despite aggressive data augmentation techniques, it was not enough to realize the full potential of the model.

In future work, I aim to apply the proposed model to more complex datasets, providing more challenging scenarios and better assessing the true potential of the model. Specifically, for models that are difficult to obtain high accuracy on test datasets, I plan to create synthetic datasets that simulate real-world degradation using the REAL-ESRGAN higher-order degradation model [14]. This involves pre-processing the images using a super-resolution technique to produce a dataset that more closely resembles real-world conditions.

By engineering features on the images before and after super-resolution and associating these features with their respective labels, I plan to feed this enhanced data into the proposed model. This approach promises to significantly improve accuracy and provide a robust training environment that reflects real-world conditions. In order to address the possible loss of texture features due to super-resolution using REAL-ESRGAN, I consider exploring an improved REAL-ESRGAN model [15] to prevent the loss of critical features. Additionally, I intend to modify the higher-order degradation function to better model image degradation relevant to my research area, thereby improving the model's ability to accurately identify and classify data.

References

[1] S. Sotheeswaran and A. Ramanan, “A classifierfree codebookbased image classification of vehicle logos,” in *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–6. doi: https://doi.org/10.1109/ICIINFS.2014.7036486.

[2] M. A. RAJAB, “Car_Logo_Dataset,” *www.kaggle.com*, May 22, 2022. https://www.kaggle.com/datasets/mahaarajab/car-logo-dataset (accessed Dec. 13, 2023).

[3] A. P. Psyllos, C. N. E. Anagnostopoulos, and E. Kayafas, “Vehicle Logo Recognition Using a SIFTBased Enhanced Matching Scheme,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322–328, doi: https://doi.org/10.1109/TITS.2010.2042714.

[4] medialab, “New Page 1,” *www.medialab.ntua.gr*, 2013. http://www.medialab.ntua.gr/research/LPRdatabase.html (accessed Dec. 14, 2023).

[5] Y. Wang, Z. Liu, and F. Xiao, “A fast coarsetofine vehicle logo detection and recognition method,” in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 691–696. doi: https://doi.org/10.1109/ROBIO.2007.4522246.
[6] Y. Wang, Z. Liu, and F. Xiao, “A fast coarsetofine vehicle logo detection and recognition method,” in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 691–696. doi: https://doi.org/10.1109/ROBIO.2007.4522246.
[7] G. Pearce and N. Pears, “Automatic make and model recognition from frontal images of cars,” in *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 373–378. doi: https://doi.org/10.1109/AVSS.2011.6027353.
[8] S. Yu, S. Zheng, H. Yang, and L. Liang, “Vehicle logo recognition based on BagofWords,” in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 353–358. doi: https://doi.org/10.1109/AVSS.2013.6636665.
[9] X. Wang, L. Xie, C. Dong, and Y. Shan, “RealESRGAN: Training RealWorld Blind SuperResolution with Pure Synthetic Data,” in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1905–1914. doi: https://doi.org/10.1109/ICCVW54120.2021.00217.
[10] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv.org*, Apr. 10, 2015. https://arxiv.org/abs/1409.1556
[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Jun. 2016, doi: https://doi.org/10.1109/cvpr.2016.90.
[12]J. R. Pangilinan, J. Legaspi, and N. Linsangan, “InceptionV3, ResNet50, and VGG19 Performance Comparison on Tomato Ripeness Classification,” in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 619–624. doi: https://doi.org/10.1109/ISRITI56927.2022.10052920.
[13] Z. Sun, G. Bebis, and R. Miller, “Onroad vehicle detection: a review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, doi: https://doi.org/10.1109/TPAMI.2006.104.
[14] Y. Wang, Z. Liu, and F. Xiao, “A fast coarsetofine vehicle logo detection and recognition method,” in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 691–696. doi: https://doi.org/10.1109/ROBIO.2007.4522246.
[15]Z. Zhu, Y. Lei, Y. Qin, C. Zhu, and Y. Zhu, “IRE: Improved Image SuperResolution Based on RealESRGAN,” *IEEE Access*, vol. 11, pp. 45334–45348, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3256086.