Multi-Class Object Classification using Deep Learning Models in Automotive Object Detection Scenarios

Soumya A[1], Linga Reddy Cenkeramaddi[2], Chalavadi Vishnu[3], Krishna Mohan C[4]

[1]Department of Computer Science and Engineering, Indian Institute of Technology, Hyderabad, India.
[2]Department of Information and Communication Technology, University of Agder, Grimstad, 4879, Norway.
[3]Department of Computer Science and Engineering, University of Agder, Norway.
[4]Department of Computer Science and Engineering, Indian Institute of Technology, Hyderabad, India.

ABSTRACT

This paper presents two deep learning models using a multi-perspective convolutional neural network (CNN) for classifying objects in the context of intelligent transportation systems (ITS). The proposed model categorizes objects accurately, enabling them to make well-informed decisions in multi-object (such as Persons, Trucks, Motorbikes, Cars, and Cyclists.) detection in complex scenarios for automotive applications. The custom backbone model is designed based on experimentation with the VGG backbone network based on the VGG backbone network, incorporating a multilayer prediction head and custom feature extraction blocks for classifying multiple objects in complex scenes. The model is to extract abstract features and features at multiple scales with a custom-designed feature extraction backbone with multiple blocks. The proposed models are lightweight and require fewer computational resources for high classification performance. The automotive publicly available dataset with 19800 images and labels has been used. Results show that when we experimented with the VGG backbone CNN model, the classification accuracy of 99.64% is achieved, and on the other hand, the classification accuracy of custom backbone CNN is 99.46%. The performance of the proposed custom model is also compared to those of pre-trained benchmark models. The experimental findings presented in this paper show that the proposed models achieve higher accuracy than the pre-trained models.

Keywords: Deep Learning, Convolutional Neural Network, Image Classification, Multi-class Classification, Multi-Object Prediction, Computer Vision.

1. INTRODUCTION

Computer vision tasks such as object detection, localization, segmentation, and classification play an important role in autonomous operations. Among these, image classification can be considered the fundamental issue underlying other computer vision tasks. Classification between different objects by discriminating their features, classifying objects in the images, and finding or counting the number of objects in the scene must be automated for several applications. In recent years, convolutional neural networks (CNNs) have excelled in solving image classification tasks.

CNNs are a unique class of multilayer neural networks that can learn to recognize visual patterns in hierarchical order. Previously, CNNs underperformed in working with high-resolution images due to insufficient regularization methods and computational constraints. But today, CNNs excel in image classification, even with large datasets like Imagenet, thanks to powerful GPUs and improved regularization methods. Numerous pre-trained deep learning models are available in the literature for classifying image datasets. The pre-trained models available are VGGNet [1], ResNet [2], EfficientNet [3], MobileNet [4], InceptionNet [5], and DenseNet [6] etc.

Image classification faces challenges like image scale variation and counting the number of objects present in an image. To address these challenges, the proposed model includes a multilayer prediction head. This lets distinct branches learn task-specific parameters, specializing in classifying objects of different scales. Furthermore, our models outperform traditional image classification by predicting object counts, presenting a complete solution to these challenges in image classification tasks.

This paper presents two deep learning models, namely, a custom backbone convolutional neural network, by experimenting with a VGG backbone convolutional neural network for multi-class classification and multi-object prediction in an image. In our proposed model, we resized the original images to a $224 \times 224 \times 3$ input size and passed them through the feature extraction layer. This layer captures hierarchical representations, encompassing low-level and high-level features crucial for object classification. Afterward, these extracted features are directed into distinct prediction branches, each

Sixteenth International Conference on Machine Vision (ICMV 2023), edited by Wolfgang Osten,
Proc. of SPIE Vol. 13072, 1307206 · © 2024 SPIE · 0277-786X
doi: 10.1117/12.3023463

tasked with classifying objects at various scales. Consequently, the model generates ten predictions, each corresponding to potential objects within an image. If an image contains fewer than ten objects, the remaining prediction boxes are detected but suppressed during the result analysis. We performed classification tests in the following categories: Person, Truck, Motorbike, Car, and Cyclist. This approach enables efficient multi-scale object classification with high accuracy.

The contributions of this paper can be summarized as follows:

- Introduce a multilayer prediction approach, allowing distinct prediction branches to learn dedicated parameters.
- Facilitating precise classification of objects across different scales.
- Designed to classify the objects belonging to diverse classes accurately.
- In addition, we predict the number of objects within each image.

Overall, our contributions to this paper offer valuable insights into image classification encompassing multi-class classification and multi-object prediction, showcasing the potential of our novel model architecture and multilayer prediction methodology.

2. RELATED WORKS

In this section, we provide a summary of the current literature pertaining to multi-class image classification. Numerous studies have introduced approaches to image classification, particularly emphasizing the effectiveness of CNN classifiers. In particular, [7] demonstrated the superiority of CNN classifiers over traditional machine learning methods focusing primarily on feature extraction. In [8], advanced neural networks were used to generate an extensive collection of high-resolution micro-doppler spectrogram images depicting flying birds and drones. In the article [9], a framework comprising ensemble CNNs is formulated to address vehicle-type classification. The primary focus is optimizing accurate predictions for images sourced from multi-view visual traffic surveillance sensors, particularly when dealing with imbalanced data. In the context of [10], image classification techniques are employed for building assessment. The work described in [11] involves the design of a feature extraction-centered convolutional neural network specifically designed for vehicle type and color classification. Convolutional neural networks were employed in [12] to classify the large-scale remote sensing images. Deep convolutional neural networks were used in [13] to classify polarimetric synthetic-aperture radar (SAR) images by extracting spatial information. In [14], A deep convolutional neural network was employed to detect complicated wetland areas using optical remote sensing data. The implementation of automated disaster event classification from aerial scenes is discussed in [15]. Using an ensemble of CNNs, satellite image classification with multiple spectrums is studied in [16]. A CNN-based image classification model is designed to predict aerial image classes by representing the images using extracted feature vectors [17]. A new classification architecture in [18] is to classify building damages. This approach incorporates residual connections and dilated convolutions within a CNN framework. Choosing the unlabeled samples to classify is the most challenging and is experimented on hard classes with scene recognition in [19]. An efficient face detection system in varying illuminations is introduced in [20]. A classification task is performed with CNN model [21], designed by considering the input of multiple parallel CNNs as output for a back propagation neural network.

3. THE PROPOSED WORK

We designed convolutional neural network models for classification to analyze the change in prediction results given by a classifier by modifying the feature extraction layer and the prediction head of the classifier. The proposed CNN models are:

3.1 Model1: Multi-layer prediction head CNN with VGG backbone

Initially, we experimented with a model that utilizes a pre-trained VGG backbone with weights pre-trained on the ImageNet dataset. The VGG backbone is frozen, ensuring the pre-trained weights remain unchanged during training. The architecture consists of several convolutional layers followed by custom feature extraction blocks. The model includes multiple prediction heads for classifying different classes. The model has ten output branches, in each branch, there are five classes covering six types of objects. Using the softmax activation to calculate class probabilities.

3.2 Model2: CNN with multi-layer prediction and custom backbone

The second model has a custom-designed feature extraction backbone with a multi-layer prediction head, named as custom backbone shown in Fig. 1, CNN model for multi-class classification and multi-object prediction. On the other hand, it features a custom-designed feature extraction backbone that consists of multiple blocks. Each block has convolutional layers followed by batch normalization and ReLU activation. Max pooling is applied after each block to downsample the feature maps and reduce spatial dimensions. The blocks progressively increase complexity, capturing more abstract features as the network deepens. The final block includes additional convolutional layers with batch normalization and max pooling.

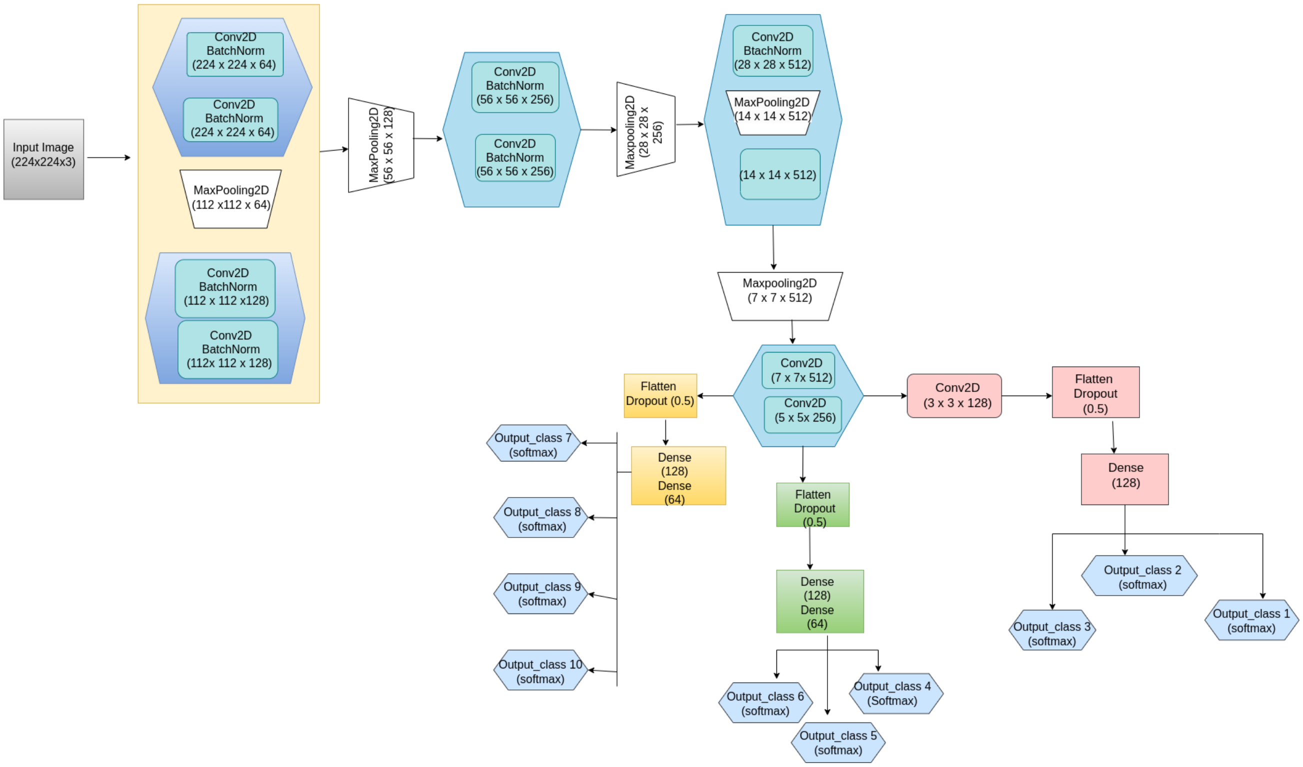


Figure 1. CNN architecture with multi-layer prediction and custom backbone

The feature extraction layer components of the custom model are shown in Fig. 2 which is composed of four distinct blocks of feature extraction layers, denoted as Block 1, Block 2, Block 3, and Block 4. Each block serves a crucial role in the overall learning process.

Block 1: The initial block sets the foundation for feature extraction. Its two consecutive convolutional layers, each comprising 64 filters of size 3x3, work to detect basic patterns and textures in the input images. Batch normalization helps stabilize the learning process, while ReLU activation introduces non-linearity, aiding the model's capacity to capture diverse features. The subsequent max pooling operation reduces the spatial dimensions by a factor of 2, allowing the network to focus on important information while discarding less relevant details. Block 2: Building upon the foundation, the second block adds complexity to capture more abstract features. Consisting of two convolutional layers with 128 filters, it aims to detect higher-level patterns and more complex textures. Block 3: The complexity increases with two convolutional layers equipped with 256 filters in the third block. This block aims to capture even more sophisticated features and visual representations. Block 4: The final block intensifies the feature extraction process. The first convolutional layer employs 512 filters, probing the images for intricate details. Batch normalization continues to ensure smooth optimization, while max pooling reduces dimensionality.

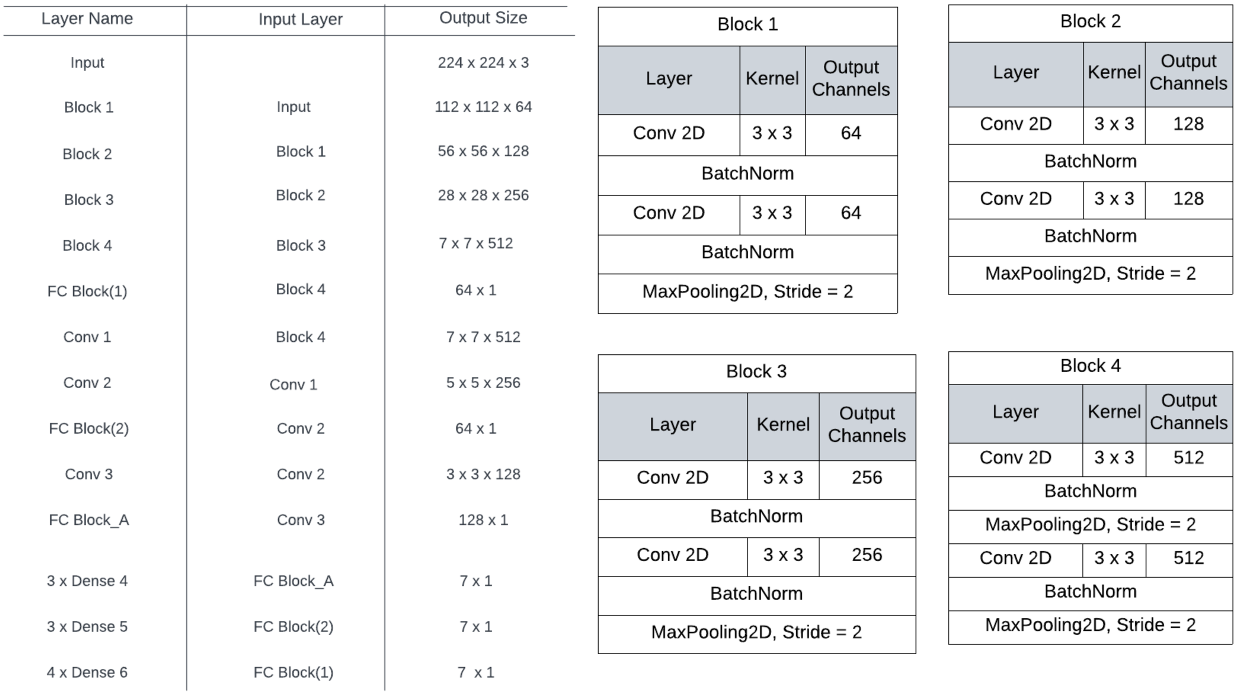
Our custom model forms an efficient hierarchical feature extractor by stacking these four blocks. As we progress through the blocks, the model becomes increasingly adept at capturing complex patterns and features, effectively extracting discriminative information, and making accurate predictions on the given task. Following the feature extraction layers, the custom model's structure is similar to the initial proposed approach, ensuring methodological consistency while utilizing the custom backbone's strengths. The multilayer prediction head remains pivotal, allowing specialization in classifying various object scales, predicting object counts, and excelling in distinct classification tasks.

It is indeed possible for the same object to be detected by multiple output heads. Our approach considers the possibility of distinct objects of the same class within an image. The purpose of employing multiple output heads is to provide a comprehensive understanding of the image, where each head specializes in extracting specific features related to its task. This approach allows us to capture various aspects of objects within the same class, including appearance, scale, and orientation variations.

In summary, the proposed model architecture combines the strengths of the feature extractor with additional layers and multiple output branches, facilitating multi-class classification and multi-object predictions for objects of different scales.

4. DATASET AND IMAGE COLLECTION

The dataset is available in IEEE Dataport [22] and is the considered image collection for the automotive object detection scenario. This dataset comprises camera images corresponding to five classes with varied dimensions. These images were taken under various conditions, such as varying illumination levels, at roads and intersections, during the day and at night, at various object poses, and at various distances from the camera. The dataset contains 19800 camera images and corresponding labels. And the camera image of size $1440 \times 1080 \times 3$. We resized to standard dimensions of $224 \times 224 \times 3$. This resizing was implemented to expedite the training process. There may be one or more objects in one image, so the location of each object is pre-annotated. The dataset categorizes all objects into five distinct categories: person, truck, pedestrian, car, and cyclist. To better estimate the generalization performance, the reported results of the dataset are the averages of 2 independent experiments.



Layer Name	Input Layer	Output Size
Input		224 x 224 x 3
Block 1	Input	112 x 112 x 64
Block 2	Block 1	56 x 56 x 128
Block 3	Block 2	28 x 28 x 256
Block 4	Block 3	7 x 7 x 512
FC Block(1)	Block 4	64 x 1
Conv 1	Block 4	7 x 7 x 512
Conv 2	Conv 1	5 x 5 x 256
FC Block(2)	Conv 2	64 x 1
Conv 3	Conv 2	3 x 3 x 128
FC Block_A	Conv 3	128 x 1
3 x Dense 4	FC Block_A	7 x 1
3 x Dense 5	FC Block(2)	7 x 1
4 x Dense 6	FC Block(1)	7 x 1

Block 1		
Layer	Kernel	Output Channels
Conv 2D	3 x 3	64
BatchNorm		
Conv 2D	3 x 3	64
BatchNorm		
MaxPooling2D, Stride = 2		

Block 2		
Layer	Kernel	Output Channels
Conv 2D	3 x 3	128
BatchNorm		
Conv 2D	3 x 3	128
BatchNorm		
MaxPooling2D, Stride = 2		

Block 3		
Layer	Kernel	Output Channels
Conv 2D	3 x 3	256
BatchNorm		
Conv 2D	3 x 3	256
BatchNorm		
MaxPooling2D, Stride = 2		

Block 4		
Layer	Kernel	Output Channels
Conv 2D	3 x 3	512
BatchNorm		
MaxPooling2D, Stride = 2		
Conv 2D	3 x 3	512
BatchNorm		
MaxPooling2D, Stride = 2		

Figure 2. Multi layer prediction head and feature extraction branch components of custom model

5. EVALUATION OF THE STATE-OF-THE-ART CNNS

In this part, a pre-trained (You Only Look Once) YOLOv8 model is fine-tuned using the images of the dataset [22] to predict the bounding boxes and class probabilities directly from the entire image in a single pass-through single-stage

Table 1. Performance evaluation using YOLOv8 pre-trained model

Class	Images	Precision	Recall	mAP-50	mAP50-95
All	2378	0.811	0.831	0.83	0.68

Table 2. Performance with the state-of-the-art models

Model	Average Accuracy(%)	Average Weighted F1-Score(%)	Trainable Parameters
VGG19	77.85	70.42	8008
Efficientnet_B0	75.68	68.99	10248
Resnet18	79.23	72.61	4104
Densenet121	77.38	70.87	8008
Custom backbone model	99.64	98.10	$1,06,27,526$

detector network. The results of this fine-tuning are presented in Table 1, indicating a mean average precision (mAP) of 81%.

Also, a few CNN-based pre-trained models, EfficientNet, DenseNet, ResNet, and VGG are trained and tested with our automotive dataset camera images available in [22]. Throughout the experimentation process, a consistent trend emerged in training these models. Initially, accuracy demonstrated an upward trajectory as training advanced. Yet, this upward trend flattened around the 75% to 80% accuracy threshold is shown in Table 2. The observations demonstrate the limitations of prevalent pre-trained models for our specific task. The proposed models, marked by its multilayer prediction head, notably overcome these limitations.

6. EXPERIMENTS

6.1 Comparision of VGG backbone and Custom backbone feature extractor:

We also compared a custom backbone model with the VGG-based backbone model. The custom backbone model consisted of multiple convolutional and batch normalization layers, providing a different feature extraction capability than the VGG backbone.

Upon comparing the custom backbone model with the VGG-based model, we observed that both models achieved similar accuracy on the task. Both models, one with a VGG pre-trained backbone and the other with a custom feature extraction backbone, achieved convergence at similar intervals shown in Fig. 3b. Additionally, both models incorporate a multilayer prediction head, allowing for task-specific feature extraction and prediction.

Despite the VGG pre-trained backbone model having the advantage of pre-trained weights, including the multilayer prediction head in both models enables fair comparison. The introduction of this head ensures that each model can capture task-specific information and prevent the tradeoff often observed in models without such a mechanism. However, the loss epoch curve, as shown in Fig. 3a, for the training and validation data of the custom backbone model exhibits some irregularities compared to the model with the VGG pre-trained backbone.

It is worth noting that both models have almost the same number of parameters, further facilitating the comparison. Due to the multilayer prediction head, the difference in accuracies between the models is negligible. This emphasizes the effectiveness of the multilayer prediction head in compensating for the absence of pre-trained weights in the custom feature extraction backbone.

6.2 Custom feature extractor with multilayer prediction head replaced with simple fully connected layers

Now, we observe the results with the model having a multilayer prediction head replaced with several fully connected layers. Upon training the models, we observed notable differences in convergence time and the behaviour of the loss versus epoch curve. The model whose multilayer prediction head was replaced with fully connected layers took longer to converge compared to the VGG pre-trained backbone model.

The models trained for comparison, lacking multilayer prediction heads, exhibit almost twice the parameter count compared to the VGG and a custom backbone model. Despite this parameter disparity, these models attain equivalent performance results, as depicted in Table 4. When we reduced the model parameters without a multilayer prediction head by making fc1 = 128 and fc2 = 64, the accuracy was reduced to 94.30%.

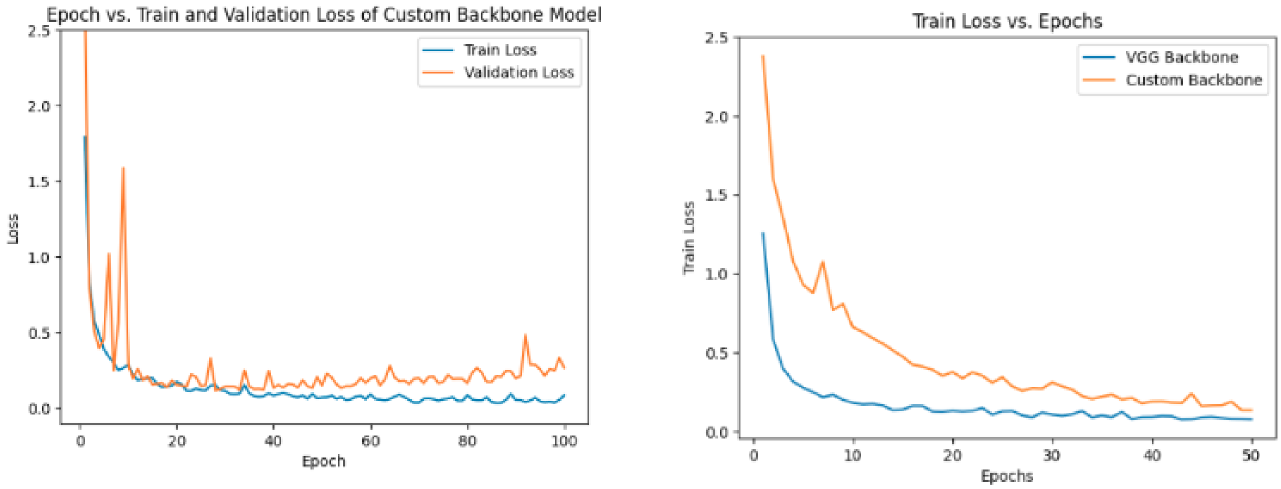


Figure 3. (a). Loss vs Epoch plot for custom backbone model, (b). Comparison of losses for VGG and custom model

Table 3. Class-wise performance evaluation table for custom backbone model

Class	True Positives	False Positives	False Negatives	True Negative	Precision(%)	Recall(%)	F1-Score(%)
Person	1467	6	4	2230	99.59	99.72	99.66
Truck	53	3	2	3649	94.64	96.36	95.49
Car	1177	2	2	2526	99.83	99.83	99.83
Motorbike	14	1	0	3692	93.33	100	96.55
Cyclist	976	8	12	2711	99.18	98.78	98.98

6.3 Evaluation Metrics

Classification evaluation metrics were employed to gauge the efficacy of our classification model's across diverse scenarios.

Accuracy: Accuracy measures the ratio of accurate predictions made by the model to the overall number of predictions. Also, Precision and Recall are calculated with 1. To assess performance, we use the average precision (AP) and average recall (AR), which are derived from the true positive, false positive, and false negative rates in Eq. 1.

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{1}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2}$$

F1-score mathematically shown in Eq.2 reaches its optimum when Precision and Recall are equal. Calculating the precision and recall values of the cumulative TP or FP classifications allows for plotting the precision-recall curve.

6.4 Evaluation Results

We Analysed the performance of our methods on the Automotive dataset with the 15777, 1973, and 1972 images for training, validation, and testing images. Few sample images are with multiple objects belonging to different classes and few images with a single object. The confusion matrix and precision-recall values for each class are depicted, and that shows all five classes classified with 99.46% accuracy of the custom backbone model is discussed in Fig. 4a, and in Table 3 to clarify the impact and function of each component on the performance of the custom model. And the corresponding accuracy plot is shown in Fig. 4b. The predicted objects per class type are 1467, 53, 1177, 14, and 976 for person, truck, car, motorbike, and cyclist classes, respectively. To enhance clarity, we exclude blank predictions for images with fewer than ten objects in our multiclass multilabel classification confusion matrix. This may lead to varying true prediction counts across matrices from the same test dataset. However, accuracy calculations consider the entire dataset for a comprehensive performance evaluation. In addition, We used sparse categorical cross-entropy loss: This function seeks to minimize the

Table 4. Accuracy evaluation of all the experimental models

Model Name	Person	Truck	Car	Motorbike	Cyclist	Total params	Trainable params	Non-trainable params	Accuracy(%)
VGG backbone	99.95	99.9	99.87	99.97	99.66	$2,61,65,254$	$1,14,50,566$	$1,47,14,688$	99.64
Custom backbone	99.73	99.7	99.79	99.97	99.3	$1,06,31,366$	$1,06,27,526$	3840	99.46
$Custom - V2$	99.59	99.81	99.37	99.97	98.6	$3,09,44,902$	$3,09,41,062$	3840	99.24
$Custom - V3$	95.53	99.58	97.75	100	94.26	$79,17,254$	$79,13,414$	3840	94.30
$Custom - V4$	99.8	99.65	99.55	100	99.11	$1,76,87,942$	$1,76,84,102$	3840	99.27

Custom-V2- With fc1=1024,fc2=512, **Custom-V3**- With fc1=128,fc2=64, **Custom-V4**- With fc1=512,fc2=256.

probabilities assigned to other classes while maximizing the output probability associated with the correct class. Sparse categorical cross-entropy loss is defined mathematically as follows:

$$\text{Sparse Categorical Crossentropy} = -\frac{1}{N}\sum_{i=1}^{N}\log(p(y_i))$$

The proposed models can precisely classify object types in challenging situations, such as lighting conditions, invisibility, and viewpoint changes. Consequently, our proposed classification systems achieved higher accuracy than the existing image classification systems.

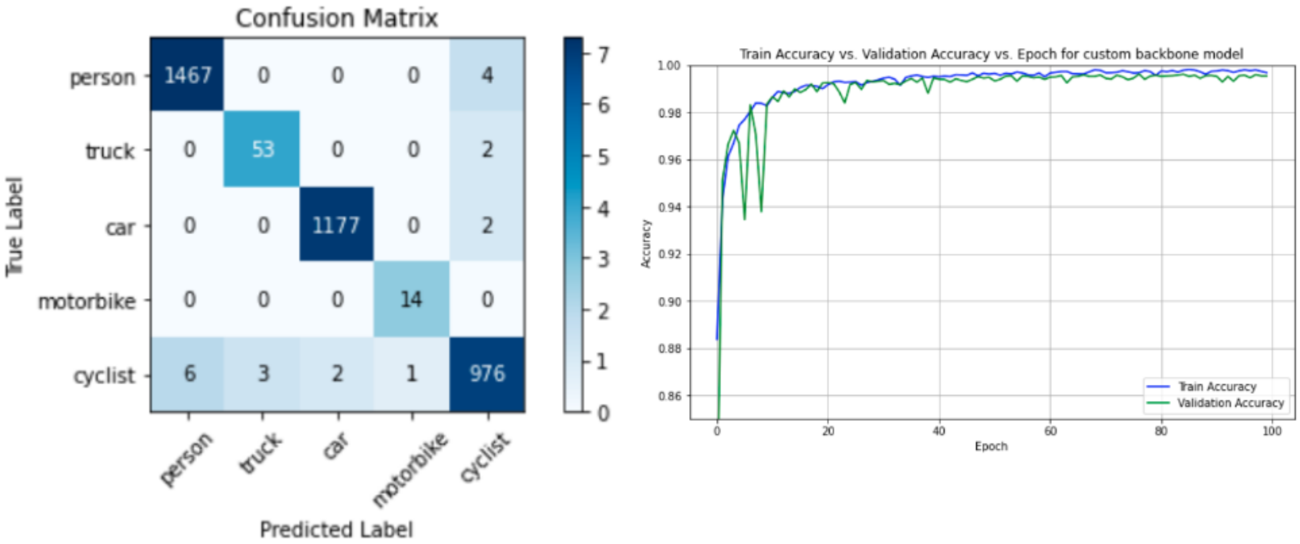


Figure 4. (a)Confusion matrix of custom backbone model with 5x5 class accuracies, (b) Accuracy plot for the custom backbone model

7. CONCLUSION

In this study, we have effectively constructed a neural network model to train the automotive image dataset using experimental data derived from complex scenarios. While pre-trained models achieve an average testing accuracy of 80%, our custom-designed sequential network showcases exceptional testing accuracy rates surpassing 99%. In the case of a model with multilayer feature extraction for prediction, each branch in the model has its separate parameters. Each branch learns and updates its parameters independently, allowing it to specialize in extracting and predicting relevant specific features. In contrast, the model without separate branches lacks this division of parameters. As a result, the parameters are updated collectively for all the predicted outputs. This can introduce a trade-off during training, as the model tries to simultaneously optimize a single set of parameters to cater to all the predicted outputs. The model must find a representation that captures common features shared across different tasks while preserving the unique characteristics required for each task.

Acknowledgment

This work was supported by the Indo-Norwegian Collaboration in Autonomous Cyber-Physical Systems (INCAPS) project: 287918 of the International Partnerships for Excellent Education, Research and Innovation (INTPART) program from the Research Council of Norway.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778 (2016).

[3] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in International conference on machine learning, 10096–10106, PMLR (2021).

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 4510–4520 (2018).

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2818–2826 (2016).

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 4700–4708 (2017).

[7] C. Coman and others, "A deep learning sar target classification experiment on mstar dataset," in 2018 19th international radar symposium (IRS), 1–6, IEEE (2018).

[8] S. Rahman and D. A. Robertson, "Classification of drones and birds using convolutional neural networks applied to radar micro-doppler spectrogram images," IET radar, sonar & navigation **14**(5), 653–661 (2020).

[9] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," IEEE Access **5**, 24417–24425 (2017).

[10] Q. D. Cao and Y. Choe, "Building damage annotation on post-hurricane satellite imagery based on convolutional neural networks," Natural Hazards **103**(3), 3357–3376 (2020).

[11] Z. Tan, "Vehicle classification with deep learning," Master's thesis, Fırat University, Institute of Science, Elazığ, 63p (2019).

[12] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," IEEE Transactions on geoscience and remote sensing **55**(2), 645–657 (2016).

[13] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin, "Polarimetric sar image classification using deep convolutional neural networks," IEEE Geoscience and Remote Sensing Letters **13**(12), 1935–1939 (2016).

[14] M. Rezaee, M. Mahdianpari, Y. Zhang, and B. Salehi, "Deep convolutional neural network for complex wetland classification using optical remote sensing imagery," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(9), 3030–3039 (2018).

[15] C. Kyrkou and T. Theocharides, "Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles.," in CVPR workshops, 517–525 (2019).

[16] M. Pritt and G. Chern, "Satellite image classification with deep learning," in 2017 IEEE applied imagery pattern recognition workshop (AIPR), 1–7, IEEE (2017).

[17] Y. Xu, M. Wei, and M. Kamruzzaman, "Inter/intra-category discriminative features for aerial image classification: A quality-aware selection model," Future Generation Computer Systems **119**, 77–83 (2021).

[18] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Satellite image classification of building damages using airborne and satellite image samples in a deep learning approach," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **4**, 89–96 (2018).

[19] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in 2009 ieee conference on computer vision and pattern recognition, 2372–2379, IEEE (2009).

[20] M. NeelakantappaA. S. G. B. H. K. Srikanth Bethu and P. N. V. S. R. M, "An approach for person detection along with object using machine learning," Advances in Information Technology, Vol. 14, No. 3, pp. 411-417 (2023).

[21] A. Olugboja, Z. Wang, and Y. Sun, "Parallel convolutional neural networks for object detection," Journal of Advances in Information Technology Vol **12**(4) (2021).

[22] X. Gao, Y. Luo, G. Xing, S. Roy, and H. Liu, "Raw adc data of 77ghz mmwave radar for automotive object detection." https://dx.doi.org/10.21227/xm40-jx59 (2022).