International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I10P105
Volume 72 Issue 10, 46-54, October 2024

Research Article

Transfer Learning for Object Classification in Video Frames: An Analysis of DenseNets

Sara Bouraya[1], Abdessamad Belangour[2]

[1,2]Laboratory of Information Technology and Modeling, Hassan II University, Faculty of Sciences Ben M'sik, Casablanca, Morocco.

[1]Corresponding Author : sarabouraya95@gmail.com

Received: 24 May 2024 Revised: 08 August 2024 Accepted: 29 September 2024 Published: 25 October 2024

Abstract *- This work investigates the enhanced capabilities of transfer learning in computer vision, specifically focusing on object detection. We employ Dense Convolutional Network (DenseNet) architectures, renowned for their efficacy in extracting and reusing features for the purpose of object identification in video frames. These advanced networks can circumvent the generally substantial requirements for computational resources and time-consuming training associated with deep learning tasks by utilizing transfer learning techniques. Our research aims to optimize the performance of DenseNet121, DenseNet169, and DenseNet201 models by making minor adjustments using a meticulously selected dataset. The objective is to assess their efficacy in identifying and categorizing items. The results indicate that DenseNet121 and DenseNet201 attain remarkable validation accuracies of 0.9605, while DenseNet169 closely matches with an accuracy of 0.9585. Results showcase the versatility of DenseNet models in tackling various object detection tasks and their comparable performance levels. This means that any of the three models might work for real-world picture recognition tasks. Moreover, our discoveries establish a basis for future investigations into enhancing the reliability of models for real-world use in monitoring and autonomous car systems, capitalizing on the models' proven accuracy.*

Keywords *- CNN, Transfer learning, Object detection, DenseNet, Car object detection.*

1. Introduction

Object detection in computer vision is a specialized technology that aims to accurately recognize and determine the precise location of things in digital images or videos, as shown in many research [1]–[4]. Computer vision is essential in various applications, such as surveillance systems, autonomous cars, picture retrieval, and machine inspection. Object detection goes beyond object recognition by classifying the entire image and identifying the specific location of each object inside the image. It categorizes various items inside an image and accurately identifies their precise positions, usually by employing bounding boxes. This feature enables systems to comprehend an image in detail by accurately identifying the present things and their exact locations. The method combines image categorization and object localization, making it a complicated operation that often depends on deep learning techniques, such as Convolutional Neural Networks (CNNs), to analyze visual input. Deep learning approaches have revolutionized object detection, surpassing standard image processing methods. Transfer learning has become an important technique that allows pre-trained models to be used on fresh datasets. This approach dramatically improves the efficiency and accuracy of object detection tasks. Our prior research has thoroughly investigated this method, analyzing multiple facets of video and object detection. In particular research [5], we examined the process of classifying video recordings, emphasizing the significance of carefully selected data in developing strong models for monitoring one or several objects. A separate investigation [6] examined the progression of deep learning models, assessing their efficacy in identifying objects within intricate visual environments.

The survey [7] thoroughly examines real-time applications, explicitly addressing the difficulties and progress in multi-object tracking algorithms applied to video streams. Finally, in [8], we evaluated the intermediate components, often known as 'neck' models, of deep learning architectures, which gave us valuable information about how they affect the performance of object detection systems. Expanding upon the knowledge gained from earlier investigations, the current research seeks to enhance the comprehension of transfer learning as a fundamental aspect of effective object detection. This research aims to develop approaches that improve the accuracy and efficiency of object detection in diverse and changing contexts by investigating the transfer of knowledge from generalized models to specialized detection tasks.

2. Literature Review

Vehicle recognition inside video streams is a prominent area of research in the computer vision community. Researchers are tackling this topic from different angles and using diverse approaches. This literature review focuses on influential research that has made significant contributions to identifying vehicles, namely those that have applied transfer learning principles. Object detection models are often classified according to the number of phases in the detection process. Single-stage detectors [9]–[11] are highly regarded for their rapid processing speeds, which makes them well-suited for real-time applications. On the other hand, two-stage detectors [12]–[14] are recognized for their high level of accuracy since they utilize a two-step procedure to enhance the precision of object recognition.

The feature extraction layer is crucial in any object detection framework since it significantly influences the system's overall effectiveness. Architectures like AlexNet, ResNet18, GoogleNet, and the DenseNet family, including DenseNet121, DenseNet169, and DenseNet201 [15]–[19], are standard models for effective feature extraction. Frequently trained on extensive datasets, these structures excel in recognizing intricate patterns necessary for precisely determining the location and categorization of objects. Transfer learning uses knowledge acquired from one domain to improve performance on related tasks. Researchers in the automobile industry have shown a preference for DenseNet121, DenseNet169, and DenseNet201 due to their deep architecture and strong ability to extract features.

The convolutional layers in these networks are highly proficient at extracting complex characteristics from images, which is crucial for accurately identifying automobiles in different situations. Our approach incorporates these sophisticated methods by utilizing transfer learning to leverage the capabilities of DenseNet121, DenseNet169, and DenseNet201 as the fundamental frameworks of our vehicle identification system. By combining classic and new methods, we aim to find the perfect equilibrium between precision and computational speed. This research makes a valuable contribution to improving object identification models, guaranteeing that they fulfil the challenging demands of real-time operations in the fast-advancing field of computer vision.

3. Methodology

3.1. Dataset

The dataset utilized in this study for car detection, as shown in Figure 1, is meticulously partitioned into training and testing subsets to facilitate a comprehensive assessment of the model's detection capabilities. The training set consists of 1,001 carefully chosen photos encompassing various vehicle kinds displayed in different contexts, orientations, and lighting conditions. This intentional inclusion of diverse elements guarantees that the model is exposed to a wide range of characteristics essential for the robust identification and precise positioning of vehicles. Conversely, the testing set comprises 175 photos carefully selected to evaluate the model's capacity to apply its acquired features to unfamiliar data. This method offers a dependable standard for assessing the model's effectiveness in practical situations. The dataset comprises cars observed in diverse environments, including urban streets and rural locations, and throughout various weather conditions, encompassing rain and fog. Additionally, it encompasses several periods of the day, from sunrise to sunset, guaranteeing a thorough examination of the model's ability to adjust and its precision under a wide range of circumstances.

3.2. Model Architecture Based DenseNet169

The study introduces a complex convolutional neural network based on the DenseNet169 architecture, known for its effective feature propagation and reuse. This feature makes it especially well-suited for intricate jobs like object detection. The model's input layer is designed to handle images with 224x224 pixels and three-color channels (RGB) dimensions. These images are then passed via the DenseNet169 functional block for further processing. This block is distinguished by its high level of interconnectivity, which aids in retaining features over the entire network.

Subsequently, a global average pooling layer is applied to decrease the dimension of the feature maps from 7x7x1664 to a single vector with a dimension of 1664. By implementing this step, the number of parameters is successfully reduced, decreasing the likelihood of overfitting. Next, a dropout layer is implemented, which stochastically deactivates a portion of the neurons throughout the training process. This reduces the possibility of overfitting and enhances the model's capacity to generalize to unfamiliar data. The architecture is finalized with a compact layer comprising a solitary neuron that employs a sigmoid activation function, enabling binary and multi-class classification. This architecture is beneficial for identifying individual items in various picture datasets, like cars, because it can learn intricate feature representations necessary for precise identification and localization.

Fig. 1 An overview of car object detection

3.3. Model Architecture Based DenseNet121

The model in Figure 3 is constructed using the DenseNet121 framework, a modified version of the Dense Convolutional Network known for its efficient computation and ability to capture intricate features with fewer parameters. The model starts with an input layer intended to handle images with dimensions of 224x224 pixels and three-color channels (RGB), which allows it to analyse standard colour images effectively. The images undergo processing by the DenseNet121 core as they go through the network, which is responsible for extracting features. The core consists of interconnected dense blocks that enable the smooth transmission of information and gradients, leading to a comprehensive set of features expressed as a 7x7x1024 tensor. To handle these features with many dimensions, the model utilizes a global average pooling layer, which compresses the features into a 1024-dimensional flat vector. This stage decreases the complexity of the model and improves its efficiency in learning.

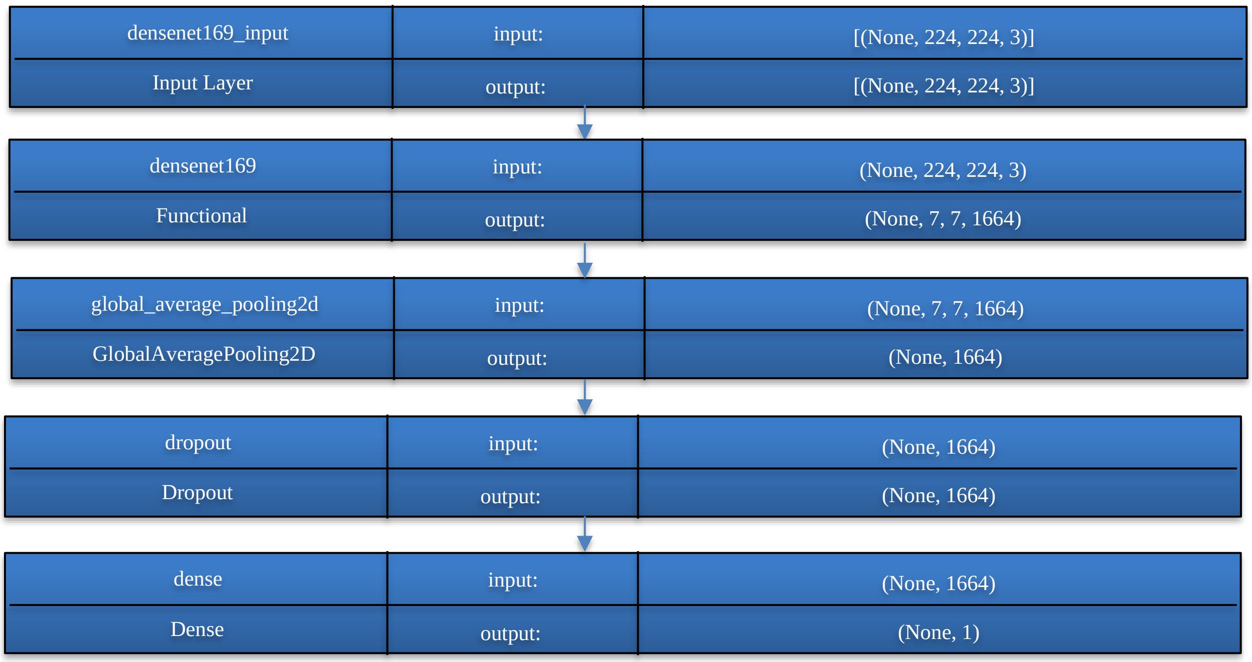

Fig. 2 An overview of the proposed model architecture based DenseNet169

Fig. 3 An overview of the proposed model architecture based DenseNet121

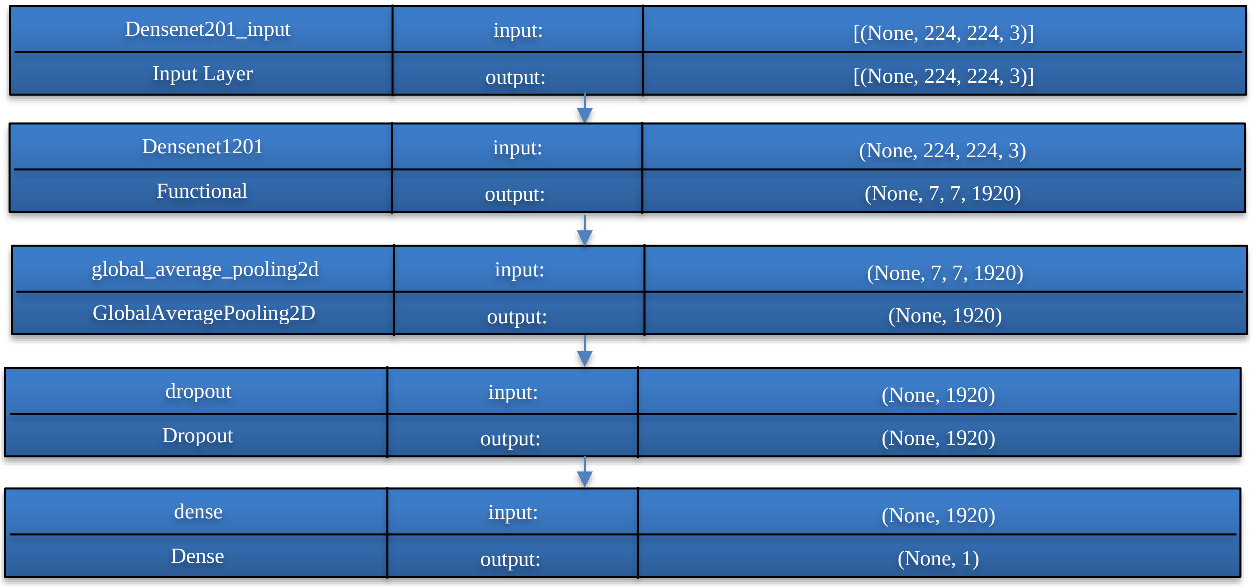

Fig. 4 An overview of the proposed model architecture based DenseNet201

Following the pooling operation, a dropout layer enhances regularization by randomly excluding a subset of neurons throughout the training process. This aids in mitigating overfitting and guarantees the model's resilience. The architecture is finalized with a compact output layer of a solitary neuron, which functions as the unit responsible for making decisions. This layer most likely utilizes a sigmoid activation function specifically designed for binary classification tasks, ultimately generating the final prediction. This approach is especially suitable for complicated object identification scenarios where recognizing subtle aspects is essential for precise item classification.

3.4. Model Architecture Based DenseNet201

The model shown in Figure 4 is built on the DenseNet201 architecture, a type of Densely Connected Convolutional Networks. DenseNet201 is noted for its deep structure and effective transmission of features between layers. The model commences with an input layer specifically engineered to handle images with a size of 224x224 pixels and three-color channels (RGB), enabling efficient processing of popular image formats. Subsequently, the input data is sent into the DenseNet201 block, which consists of a sequence of convolutional and pooling procedures.

These methods process the input data to produce feature maps with dimensions of 7x7x1920. The feature maps undergo processing through a global average pooling 2D layer to decrease the computational burden and the number of parameters. The purpose of this layer is to condense the necessary information into a vector with 1920 dimensions. This allows for the extraction of global characteristics while still keeping the model straightforward. Subsequently, a dropout layer is implemented on this vector. The primary function of this layer is to mitigate overfitting by introducing random deactivation of a subset of neurons during the training process, hence improving the model's capacity to generalize to unseen data. The last element of the model consists of a highly coupled layer containing only one neuron, which functions as the output layer. Depending on the job at hand, this neuron can utilize a sigmoid activation function for binary classification or a softmax activation function for multi-class classification.

The function provides the probability that the image includes a specific object of interest, such as an automobile. Due to its depth and resilient feature paths, the DenseNet201 model is particularly effective for complicated object identification tasks. It offers great accuracy and reliability in recognizing and classifying objects in varied settings.

4. Results

The convergence trends of the DenseNet169 model during the training process over ten epochs are illustrated in Figure 5, highlighting four key metrics: training loss, validation loss, accuracy, and validation accuracy.

4.1. Training Loss

This line represents the model's training loss, which starts at a relatively high level and shows a sharp decline during the initial epochs, indicating a rapid learning phase. As training continues, the loss gradually levels off, suggesting that the model is nearing its learning capacity with the current dataset.

4.1.1. Accuracy

The accuracy metric, likely reflecting the model's performance on the training data, begins at a moderate level and increases steadily with each epoch. The rate of improvement slows down over time, which may indicate that the model is reaching its maximum potential for correctly classifying the training data.

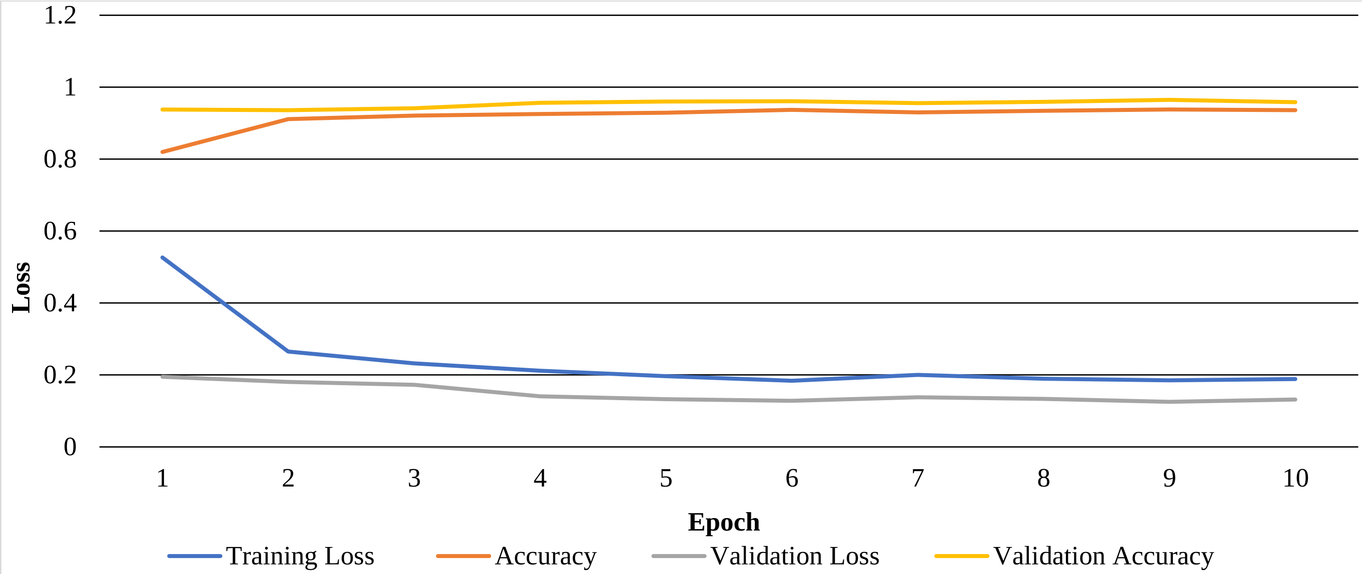

Fig. 5 Convergence Trends of the first model based DenseNet169: Training and validation metrics across epochs

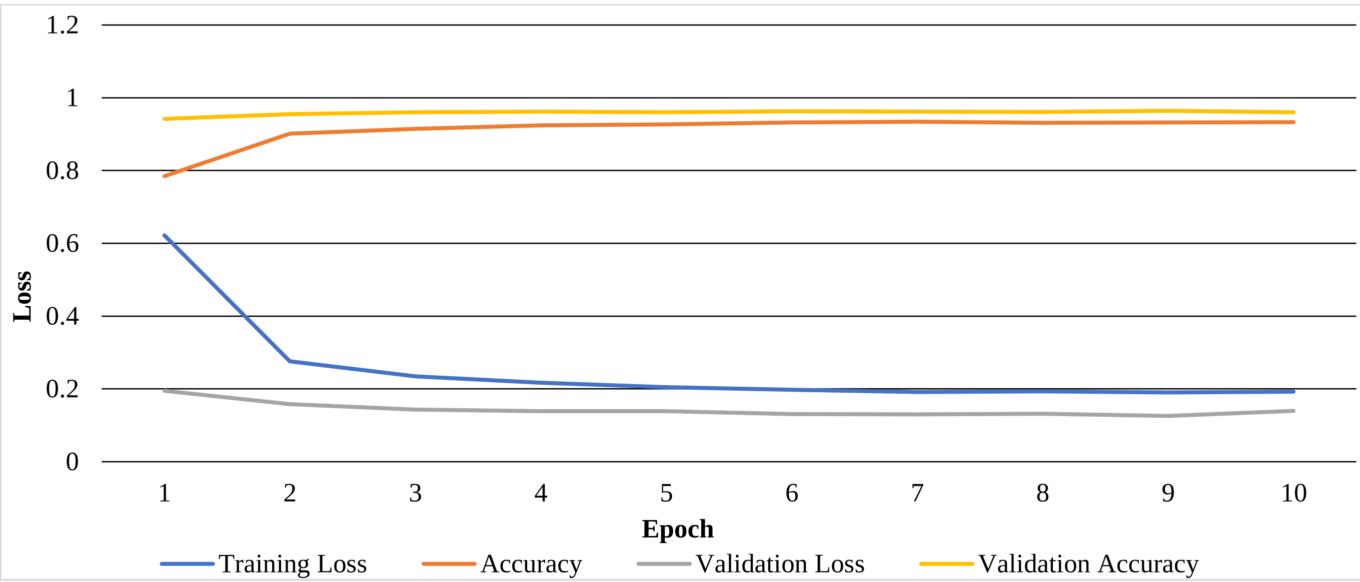

Fig. 6 Convergence Trends of the first model based DenseNet121: Training and validation metrics across epochs

4.1.2. Validation Loss

The validation loss starts slightly higher than the training loss but decreases steadily, eventually converging closely with the training loss as the epochs progress. This convergence is a positive indication of the model's generalisation capabilities, suggesting that it is learning patterns that are applicable not only to the training data but also to unseen validation data.

4.1.3. Validation Accuracy

The validation accuracy starts high and remains relatively stable throughout the training process, with a slight upward trend. The fact that validation accuracy is consistently high and comparable to training accuracy indicates that the model performs well on data that it has not seen before, demonstrating strong generalisation.

4.2. Convergence Trends of DenseNet121

Figure 6 illustrates the convergence trends of the DenseNet121 model over ten training epochs, showing key metrics: loss, accuracy, validation loss, and validation accuracy.

4.2.1. Loss

This curve represents the training loss, which starts at a relatively high value but quickly decreases, indicating that the model is effectively learning from the training data.

The flattening of the curve after the initial epochs suggests that the model is approaching the limit of what it can learn from the provided training set.

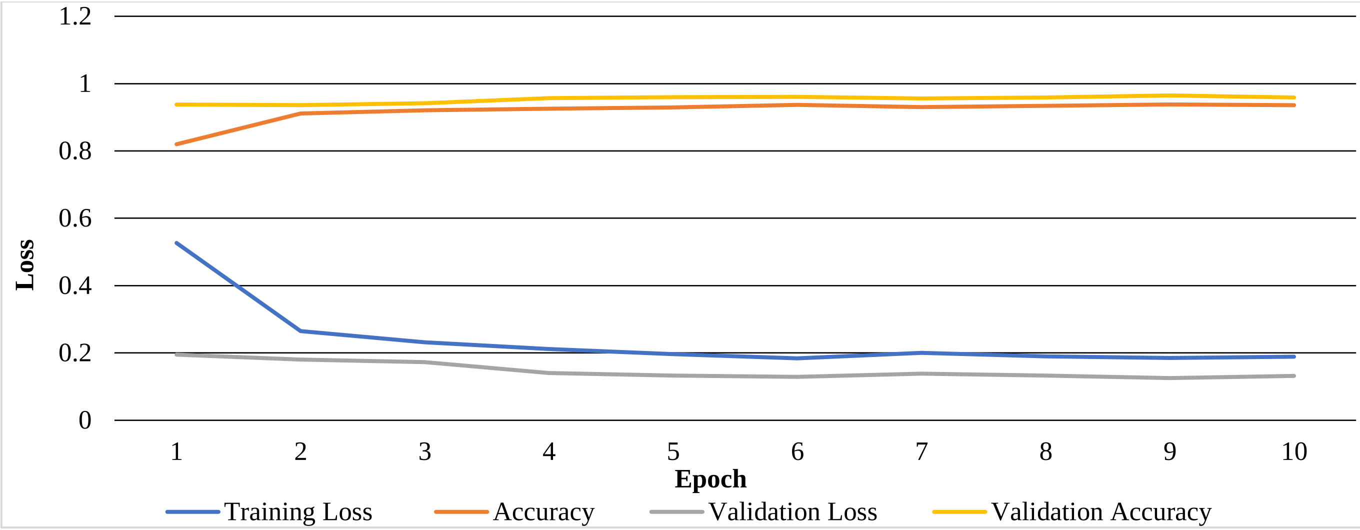

Fig. 7 Convergence Trends of the first model based DenseNet201: Training and validation metrics across epochs

4.2.2. Accuracy

The accuracy, likely reflecting the model's performance on the training data, begins at a lower level but improves as the number of epochs increases. The curve levels off as the model starts to converge, indicating that the predictions are becoming more accurate over time.

4.2.3. Validation Loss

The validation loss starts higher than the training loss but decreases significantly, demonstrating the model's performance on the validation set. Its trend towards a lower value is a positive indication that the model is not overfitting and is learning patterns that generalise well to unseen data.

4.2.4. Validation Accuracy

The validation accuracy starts at a relatively high level, indicating that the model performs well on the validation data from the very first epoch. The line remains stable throughout the training process, reflecting consistent performance when the model is exposed to new, unseen data.

4.3. DenseNet201 Metrics

Figure 7 illustrates the key metrics of the DenseNet201 model throughout ten training epochs, including loss, accuracy, validation loss, and validation accuracy.

4.3.1. Loss

This line shows the loss on the training dataset, which begins at a higher value and rapidly decreases, followed by a more gradual decline. This pattern suggests that the model is effectively learning from the training data initially, but the slowing rate of loss reduction may indicate diminishing returns from further training.

4.3.2. Accuracy

The accuracy of the training set starts at a modest level but increases steadily throughout the epochs, indicating that the model is progressively improving its ability to classify the training data correctly. By the end of the ten epochs, the accuracy approaches a high value, reflecting a good fit to the training data.

4.3.3. Validation Loss

The validation loss starts slightly higher than the training loss but decreases sharply before stabilising at a lower level. This flattening suggests that the model is not overfitting and is effectively generalising to new, unseen data.

4.3.4. Validation Accuracy

The validation accuracy begins at a high level and remains relatively flat across the epochs. Its stability near the upper boundary of the chart indicates that the model consistently maintains high performance on the validation set throughout the training process.The performance measures of the model demonstrate its effective learning and ability to generalize to new data (refer to Figure 8).

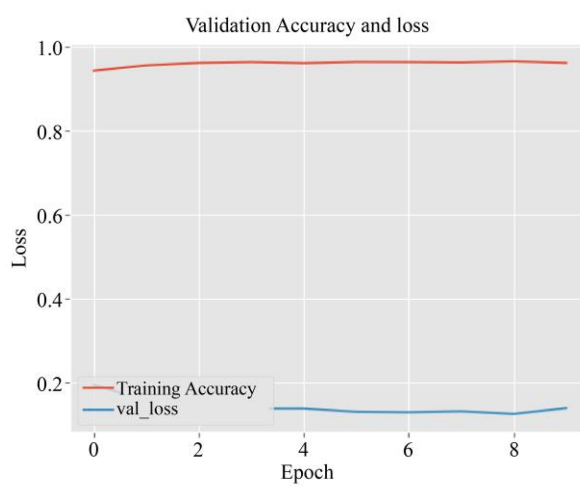

Fig. 8 Validation accuracy and validation loss of the model based DenseNet169

The convergence of both the training and validation loss curves towards the conclusion of the training indicates that further training on this dataset may not yield meaningful improvements to the model. The plateauing of the model suggests that it does not suffer from overfitting and effectively strikes a balance between learning from the training data and achieving good performance on the validation data.

The displayed metrics indicate that the model is effectively learning, as both the validation loss and accuracy demonstrate strong generalizability. The plateauing of both the training and validation loss curves indicates that additional epochs are unlikely to result in substantial enhancements, and the model is approaching an optimal state for its existing architecture and dataset. The model's ability to continually achieve high validation accuracy throughout the training process indicates its strong performance on unknown data (refer to Figure 9). As depicted in Figure 10, the model exhibits exceptional performance from the beginning of training, with minimal fluctuations in loss and accuracy metrics. This may indicate a highly efficient first learning stage or that the model is well-initialized and approaching early convergence. The consistent validation loss and accuracy suggest that the model effectively generalizes to unseen data, and further training is unlikely to yield substantial performance enhancements.

DenseNet121 steadily increases accuracy, beginning at 0.7847 and reaching 0.9327. Nevertheless, DenseNet169 surpasses it somewhat, starting with a higher initial accuracy of 0.8199 and achieving a final accuracy of 0.9364. DenseNet201 exhibits comparable starting accuracy to DenseNet121 and performs closely, ultimately reaching an accuracy of 0.9327. These trends indicate that although all variations train well, DenseNet169 may capture the subtleties of the dataset more effectively. The validation loss for all architectures begins at a low level and remains consistently low. However, DenseNet169 exhibits the most notable drop, reaching a value of 0.1318.

This suggests that DenseNet169 may have a higher generalization ability, as evidenced by its validation accuracy, which reaches a maximum of 0.9648. DenseNet201 closely follows with a peak validation accuracy of 0.9642, while DenseNet121 likewise achieves a peak validation accuracy of 0.9642. These results emphasize the significance of DenseNet's feature propagation in controlling overfitting and improving generalization, which are critical factors in model performance, particularly in object detection tasks. DenseNet169 can strike an ideal equilibrium between depth and parameter efficiency, as it exhibits marginally superior generalization regarding validation accuracy.

Nevertheless, the variations among the models are minimal, indicating that each version can be appropriate based on the application's individual needs, such as processing resources and required level of precision. Although all models exhibit exceptional performance, the decision between them may hinge on finding a balance between computational expense and marginal improvements in accuracy. Subsequent investigations could further examine these structures by integrating supplementary regularization methods or data augmentation tactics to improve performance and stability. Furthermore, augmenting the datasets to incorporate a broader range of diverse and demanding situations could offer a more comprehensive evaluation of the models' resilience and flexibility. Figure 10 demonstrates that DenseNet structures are highly efficient for tasks requiring precise feature differentiation, such as detecting vehicles in intricate surroundings. This showcases their substantial promise for practical object identification applications, as shown in Figure 11.

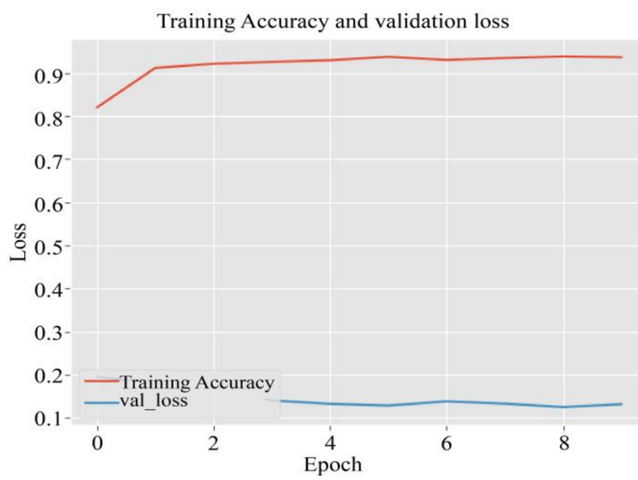

Fig. 9 Validation accuracy and validation loss of the model based DenseNet121

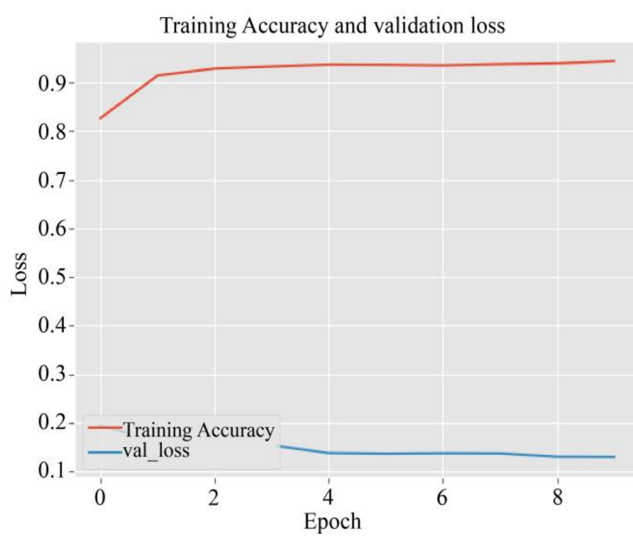

Fig. 10 Validation accuracy and validation loss of the model based DenseNet201

DenseNet121			
Training Loss	Accuracy	Validation Loss	Validation Accuracy
0.6222	0.7847	0.1949	0.9422
0.276	0.9015	0.1586	0.9548
0.2345	0.9144	0.143	0.9605
0.2175	0.9245	0.1387	0.9625
0.2047	0.9268	0.1386	0.9599
0.1978	0.9321	0.1308	0.9628
0.1917	0.9343	0.1297	0.9625
0.1934	0.9312	0.132	0.9616
0.1904	0.9326	0.1258	0.9642
0.1929	0.9327	0.1396	0.9605

DenseNet169			
Training Loss	Accuracy	Validation Loss	Validation Accuracy
0.5267	0.8199	0.195	0.9381
0.2649	0.9116	0.1808	0.9361
0.2319	0.9211	0.1724	0.9416
0.2114	0.9255	0.1405	0.9565
0.1968	0.9293	0.1328	0.9599
0.184	0.9372	0.1287	0.9608
0.2006	0.93	0.1385	0.9559
0.1896	0.9344	0.1332	0.959
0.185	0.9381	0.1254	0.9648
0.1885	0.9364	0.1318	0.9585

DenseNet201			
Training Loss	Accuracy	Validation Loss	Validation Accuracy
0.6222	0.7847	0.1949	0.9422
0.276	0.9015	0.1586	0.9548
0.2345	0.9144	0.143	0.9605
0.2175	0.9245	0.1387	0.9625
0.2047	0.9268	0.1386	0.9599
0.1978	0.9321	0.1308	0.9628
0.1917	0.9343	0.1297	0.9625
0.1934	0.9312	0.132	0.9616
0.1904	0.9326	0.1258	0.9642
0.1929	0.9327	0.1396	0.9605

Fig. 11 Training accuracy and training loss, validation accuracy and validation loss of the model based DenseNet121, DenseNet201, DenseNet169

5. Discussion

When comparing accuracy, DenseNet121 shows steady progress, increasing from 0.7847 to 0.9327. However, DenseNet169 slightly outperforms it, starting with a higher accuracy of 0.8199 and reaching 0.9364. DenseNet201 performs similarly to DenseNet121, starting at the same initial accuracy and concluding with a final accuracy of 0.9327. These trends indicate that while all variants are effective learners, DenseNet169 may capture the dataset's intricacies more effectively. The validation loss remains relatively low across all architectures, with DenseNet169 experiencing the most significant drop to 0.1318. This suggests that DenseNet169 may generalize better, as indicated by its peak validation accuracy of 0.9648. DenseNet201 follows closely with a peak of 0.9642, while DenseNet121 also achieves a peak validation accuracy of 0.9642. These findings highlight DenseNet's ability to manage overfitting and enhance generalization, which is crucial for model performance, especially in object detection tasks. DenseNet169 appears to strike an optimal balance between depth and parameter efficiency, as it slightly outperforms the others in validation accuracy. However, the differences among the models are minimal, suggesting that any of these variants could be appropriate, depending on the application's computational resources and accuracy requirements. While all models perform exceptionally well, the choice between them may hinge on balancing computational cost with marginal accuracy improvements. Future research could explore these architectures with additional regularization techniques or data augmentation strategies to further improve performance and stability.

Additionally, using more diverse and challenging datasets could offer a more rigorous evaluation of the model's robustness. Figure 7 illustrates that the DenseNet architectures excel in tasks requiring fine feature discrimination, such as vehicle detection in complex environments, indicating significant potential for real-world object detection applications.

6. Conclusion

This study demonstrates the effectiveness of DenseNet architectures in vehicle detection within video frames, highlighting the capabilities of deep convolutional networks in this domain. DenseNet121, DenseNet169, and DenseNet201 each show strong potential in learning detailed features necessary for accurate object recognition, as evidenced by the consistent reduction in training loss and the increase in both training and validation accuracy throughout the training process. DenseNet169, in particular, showed slightly better generalization than DenseNet121 and DenseNet201. However, the performance differences were minimal, suggesting that all architectures are competitive options, with the final decision likely dependent on computational efficiency and specific use-case needs. Looking ahead, several areas for future research are evident. Investigating the integration of these models into real-time detection systems could optimize both speed and accuracy. Exploring the impact of using more extensive datasets, especially those with a broader range of environmental conditions and vehicle types, could enhance the models' robustness and applicability. Additionally, combining DenseNet architectures with other models in an ensemble approach may improve detection performance and address potential weaknesses of individual models.

In conclusion, this investigation into DenseNet-based models for vehicle detection underscores the significant potential of deep learning in computer vision. The findings support the ongoing development and refinement of such models, reinforcing their role as vital tools in the growing field of automated detection and surveillance technologies.

References

[1] Prince Kumar et al., "A Comparative Study of Object Detection Algorithms in A Scene," *International Journal of Engineering Research & Technology*, vol. 8, no. 5, pp. 438-440, 2019. [Google Scholar] [Publisher Link]

[2] Kai Kang et al., "Object Detection from Video Tubelets with Convolutional Neural Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 817-825, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Bilel Benjdira et al., "Car Detection Using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," *2019 1st International Conference on Unmanned Vehicle Systems-Oman*, Muscat, Oman, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Nikhil Yadav, and Utkarsh Binay, "Comparative Study of Object Detection Algorithms," *International Research Journal of Engineering and Technology*, vol. 4, no. 11, pp. 586-591, 2017. [Google Scholar] [Publisher Link]

[5] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Joseph Redmon, and Ali Farhadi, "YOLO9000: Better, Faster, Stronger," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263-7271, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xiang Long et al., "PP-YOLO: An Effective and Efficient Implementation of Object Detector," *Arxiv*, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Shaoqing Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Xin Lu et al., "Grid R-CNN," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7363-7372, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ross Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision*, Santiago, Chile, pp. 1440-1448, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012. [Google Scholar] [Publisher Link]

[12] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Arxiv*, pp. 1-14, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[13] Kaiming He et al., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[14] Christian Szegedy et al., "Going Deeper with Convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1-9, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Gao Huang et al., "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 2261-2269, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sinno Jialin Pan, and Qiang Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] Asmaul Hosna et al., "Transfer Learning: A Friendly Introduction," *Journal of Big Data*, vol. 9, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Barret Zoph et al., "Learning Transferable Architectures for Scalable Image Recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8697-8710, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hana' Abd Razak et al., "Anomalous Behaviour Detection Using Transfer Learning Algorithm of Series and DAG Network," *2019 IEEE 9th International Conference on System Engineering and Technology*, Shah Alam, Malaysia, pp. 505-509, 2019. [CrossRef] [Google Scholar] [Publisher Link]