Image Classification and Verification using CNNs on the CompCars Dataset

Qiqi Zhang†, Wageesha Widuranga Waththe Liyanage†, Fateme Baghaei Saryazdi†

***Abstract*—The field of computer vision, particularly in automotive applications, has experienced rapid advancements, making it highly relevant for developing intelligent transportation systems and autonomous vehicles. In this report, we utilize the CompCars [1] dataset to conduct three primary tasks: car make classification, car part classification, and car make verification using state-of-the-art deep learning techniques. These tasks are crucial for improving vehicle recognition systems, contributing significantly to both academic research and practical applications in the automotive industry. Our approach involved training convolutional neural networks (CNNs) including ResNet and GoogLeNet with different types of loss to increase accuracy in car make classification and car part classification, enabling identification of different car manufacturers. In the car make verification task, we employed a Siamese network architecture to measure the similarity between pairs of images, achieving reliable verification of car makes. Our models demonstrated high accuracy in some of the tasks, with significant improvements over baseline methods. These results provide a robust benchmark for future research and can be integrated into commercial products, such as automated vehicle inspection systems and intelligent traffic monitoring, offering substantial benefits in terms of efficiency and accuracy.**

***Index Terms*—Convolutional Neural Networks, Residual Network, GoogLeNet, Siamese Network**

I. Introduction

Cars revolutionize mobility and convenience, offering flexibility in transportation. They are essential in modern life, reflecting economic status and societal stratification. Aside from the importance of cars in the general public, cars present unique challenges and opportunities for computer vision research due to their diverse models, designs, and attributes. This diversity fosters the development of sophisticated and robust vision models and algorithms.

Fine-grained car classification and verification have numerous applications, including intelligent transportation systems, automated toll collection, video surveillance, and personal car consumption. These applications enhance efficiency, security, and user convenience. For our study, we used the CompCars dataset as it provides a comprehensive platform for validating computer vision algorithms, promoting further research and practical applications in car-related tasks. This dataset provides us with 163 car makes expanding into 1716 car models. In total it presents us 136,726 images capturing the entire car and 27,618 images capturing different car parts, interior and exterior. In this study we use multiple CNN architecture to classify and verify car makes and parts.

In car make classification task, we train ResNet18 and GoogLeNet with cross-entropy and focal loss with different parameters.

In car part classification, we aim to classify headlights and taillights with the help of ResNet18 which we train with cross-entropy and focal loss for 75 different classes.

In car make verification we practiced the similarity learning technique to verify the car makers. Similarity learning is a technique whose goal is to make the model learn, which is a similarity function that measures how similar two objects are and returns a similarity value. If the similarity score is higher than a certain threshold (which needs to be decided based on the training performance of the model), then the check is accepted, and if the similarity score is low, it is not accepted as being in the being in the same class.

II. Related Work

Previous research on car model recognition has primarily focused on classification. Zhang et al. [2] developed an evolutionary computing framework to fit a wireframe model to car images for recognition. Hsiao et al. [3] used 3D space curves from 2D images to match and align car models. Lin et al. [4] combined 3D model fitting with fine-grained classification. Krause et al. [5] classified 196 car models using 3D representations, marking the largest scale experiment in this area.

Car model classification is a fine-grained categorization task, distinguishing subcategories within a single object class. Various datasets have been created for fine-grained categorization, including birds [6], dogs [7], cars [5], and flowers [8]. However, these datasets are limited in scale and subcategory diversity.

Car model verification, unlike classification, has not been extensively explored. Face verification, a related field, has seen significant advancements with deep learning algorithms [[9], [10], [11], [12]]. Joint Bayesian [13] is a common verification model used in face recognition, which is adopted in the paper that we used as reference[] as a baseline for car model verification.

Attribute prediction for humans has been a popular research topic [14], [15] , [10], [16], but most datasets often suffer from annotation ambiguities [14] . In contrast, the attributes in the CompCars dataset, such as maximum speed and door

†Department of Physics and Astronomy "Galileo Galilei", University of Padova,
email: {qiqi.zhang, wageeshawiduranga.waththeliyanage, fateme.baghaeisaryazdi}@studenti.unipd.it

number, are clearly defined by manufacturers, offering more reliable evaluation.

Other car-related research includes detection [17], tracking [18] [19], joint detection and pose estimation [20] [21], and 3D parsing [22]. However, these studies do not focus on fine-grained car models. Research on car parts includes logo recognition [19] and style analysis based on mid-level features [23].

The CompCars dataset offers significant advantages over existing datasets, such as diverse viewpoints, aligned car part images, and rich attribute annotations, making it a valuable resource for advancing car model analysis.

III. Processing Pipeline

In our study, we employed ResNet18, GoogLeNet and Siamese Network as the backbone architectures as our deep learning models for different tasks, leveraging their proven effectiveness in image classification tasks.

Car Make Classification We used both ResNet18 and GoogLeNet to compare their performances and determine the most effective model for this specific task. ResNet18 (Figure 1), with its residual learning framework, effectively mitigates the vanishing gradient problem, allowing for efficient learning with a relatively smaller number of parameters. GoogLeNet(Figure 2), also known as Inception v1, employs inception modules that facilitate multi-scale feature extraction, enhancing the model's ability to capture diverse visual features.

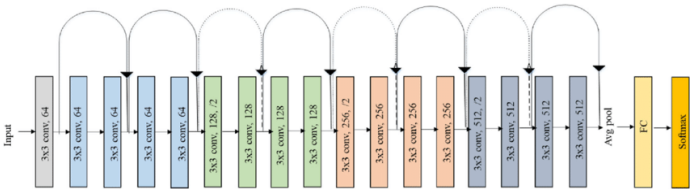

Fig. 1: Architecture of ResNet18.

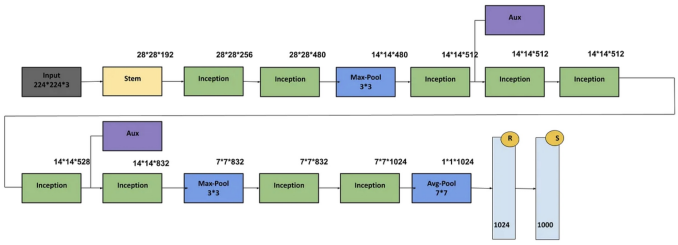

Fig. 2: Architecture of GoogLeNet.

The processing pipeline for car make classification involved several key steps. First, car images were preprocessed by cropping, resizing and normalizing. The images were then fed into both ResNet18 and GoogLeNet models, which were not pre-trained. For fine-tuning, the final fully connected layers of both models were replaced with layers matching the number of car makes in our dataset. The models were trained using both cross-entropy loss function and focal loss function to optimize classification accuracy for comparison. By comparing the performance of ResNet18 and GoogLeNet, we aimed to identify the strengths and weaknesses of each architecture in classifying car makes, providing valuable insights into their applicability in automotive image analysis.

Car Part Classification In this task, we used ResNet18 to classify two different parts of the cars, headlights and taillights. Similar to car make classification task, ResNet18 is helpful due to its residual learning framework and allows for efficient learning. First the images were preprocessed to make them suitable for the learning task and then they were fed into ResNet18 that was not pre-trained. We trained the model using both cross-entropy loss and focal loss to be able to compare the accuracy of the model with different loss functions.

Car Make Verification In this task, we use a Siamese neural network, which is called a one-shot network. The reason behind this one shot is that we require only one training example for each class to train this network. A Siamese network is an artificial neural network that contains two or more identical sub-networks that have the same configuration with the same parameters and weights. In this task, we use the CNN network as the sub-network.

Figure 3 shows the basic architecture of a Siamese network.

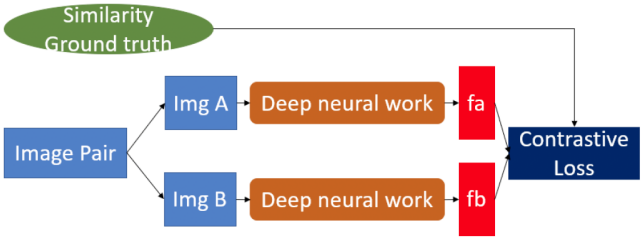

Fig. 3: Architecture of Siamese Network.

Here we have chosen contrastive loss to minimize the distance between similar images (positive pairs) while maximizing the distance between dissimilar images (negative pairs). It works by embedding data points into a space where similar points are closer and dissimilar points are farther apart.

IV. Data Preprocess

In this section, we outline the data preprocessing steps taken to prepare the CompCars dataset for our car make classification, car part classification, and car make verification tasks.

Car Make Classification

- **Dataset Selection** We utilized the CompCars dataset, specifically following the predefined training and testing splits provided in the train_test_split directory. This structured approach ensures consistency and reliability in

our experiments, as the splits are designed to represent diverse and balanced data distributions for car classification tasks.

- **Class Selection** To simplify the computational complexity and focus our initial experiments, we selected the first 10 classes from the available car makes in the CompCars dataset. By concentrating on this subset, we could more effectively fine-tune our models and analyze their performance. This selection enables a clear and concise evaluation while maintaining a manageable scope for experimentation.
- **Image Cropping** To enhance the quality and relevance of the input data, we cropped the images based on the provided bounding boxes. Cropping the images ensures that our models focus on the salient features of the vehicles, which is crucial for improving classification accuracy.
- **Relabeling** After selecting the first 10 classes, we relabeled the images from their original class identifiers to a new range of labels from 0 to 9. This relabeling standardizes the class identifiers, facilitating easier interpretation and management during model training and evaluation.
- **Data Splitting** To evaluate the performance of our models, we split the dataset into training and validation sets. The selected images for each of the 10 classes were divided into training and validation sets with a ratio of 7:3, respectively.

In Figure 4 and Figure 5, we plot some examples of the processed dataset.

Fig. 4: Examples for Training set.

Fig. 5: Examples for Testing set.

Car Part Classification

- **Dataset Selection** In the same way as we did in the make classification task, in the part classification task, we use the predefined training and testing splits provided. We use only the dataset for headlights and taillight classification.
- **Class Selection** As we did not have enough data for this task, only 2700 images for part 1 of the cars, we are bound to use all of the 75 classes of the dataset.
- **Relabeling** After selecting the part of the dataset we want to work with, we relabel them to standardize the class identifiers in order to have easier management during training and evaluation.
- **Data Splitting** As the last step of the data preprocessing, we split the dataset into training and validation sets with the ration of 7:3, respectively.

Fig. 6: Examples of the headlights from the training set

Fig. 7: Examples of the headlights from the testing set

Car Make Verification

- **Dataset Selection** We utilized the CompCars dataset, specifically following the predefined verification pairs for training provided in the train test split directory under verification. These pairs were labeled as 0 or 1 in order to obtain the similarity between given image pairs. There were three categories of data files including easy, medium, and hard.
- **Class Selection** To simplify the computational complexity and focus our initial experiments, we selected 20 classes from the available car makes in the CompCars dataset.
- **Image Cropping** As we did in the previous car make classification task, image cropping was done before training the data with a Siamese network.
- **Data Splitting** The selected images for each of the 20 classes were divided into training, validation, and testing.

In Figure 10 ,we plot some examples of the processed dataset for verification task.

V. Learning Framework

In this section, we describe the learning framework employed to tackle the problem of car make classification, car part classification, and car make verification using deep learning models.

Fig. 8: Examples of the taillights from the training set

Fig. 9: Examples of the taillights from the testing set

Fig. 10: Examples of the image pairs from the verification training set

Car Make Classification The learning strategy adopted in this study involves training multiple models with different configurations to evaluate their performance in classifying car makes. We utilized two prominent architectures ResNet18 and GoogLeNet trained separately, each with three distinct models leveraging different loss functions. This multi-faceted approach allows us to systematically analyze the effect of varying loss functions on classification performance.

- **Training Configuration** Model 1 was trained by using Cross-Entropy Loss serves as the baseline. Model 2 utilizes focal loss with parameters $\alpha = 1$ and $\gamma = 2$ to address potential class imbalance. Focal loss down-weights the contribution of easy-to-classify examples, emphasizing hard-to-classify examples to improve model robustness. Model 3 employs focal loss with $\alpha = 0.25$ while maintaining $\gamma = 2$. This adjustment allows for further tuning of the loss function to explore its impact on the learning process, particularly in the context of imbalanced classes.
- **Optimization Process** The optimization of each model was performed using the Adam optimizer. Learning Rate of 0.001 was employed. A batch size of 32 was chosen to balance the memory constraints and convergence stability during training. Each model was trained for 25 epochs.

Car Part Classification The strategy used in this task was to train three models with different configuration to evaluate their performance in car part classification. We worked only with ResNet18 in this task. Each model had the same fundamental architecture but trained with different loss functions. This allows us to compare the effect of different loss functions on the classification performance.

- **Training Configuration** In the same manner as how we handled the car make classification task, we used the Cross-Entropy Loss for the first model as the baseline. Model 2 uses the focal loss with parameters $\alpha = 1$ and $\gamma = 2$ to check if there's some class imbalance between the 75 different classes that we work with in this task. Model 3 utilizes focal loss with parameters $\alpha = 0.25$ and $\gamma = 2$. This change allows for further tuning of the loss function to check its impact on the learning process, especially when we're working with imbalanced classes. All three models were used to classify the headlights as well as the taillights.
- **Optimization Process** The Adam optimizer was used to optimize the performance of all three models. The learning rate of 0.001 and a batch size of 32 was chosen for balancing the memory constraints and convergence stability during training. Each model was trained for ten epochs for the task of taillight classification, but for the headlight classification 25 was chosen as the number of epochs

Car Make Verification The model has 3 convolution layers to extract high-level features and followed by ReLu activation. After convolution layers output is flattened and passed through fully connected layers and while reducing the dimensionality. Dropout technique was used to implement regularization to the network while forward method take two input images processes each through the network and return their embedding.

- **Training Configuration** Model was trained by using contrastive loss serves as the baseline. The contrastive loss layer takes the output features of the last layer and the labels as the input to calculate the cost of the model. With the Stochastic Gradient Descent algorithm, the Siamese Neural Network is optimized with the contrastive loss.
- **Optimization Process** The RMSprop optimiser was selected to train the Siamese Network because it is an excellent fit for applications involving deep learning since it can handle noisy gradients and non-stationary targets.

VI. Results

In this section, we show the training and testing results for three tasks.

Car Make Classification Figure 11 shows the comparison of training and validation losses of all three models for ResNet18. And Figure 12 shows the comparison of training and validation accuracies of all three models for ResNet18. Table 1 summarizes the testing performances for 3 models of ResNet, we can see the Test Loss decreased by a significant amount for model 3, which means focal loss with $\alpha = 0.25$ would have better performance for top-3 error. Figure 13 plots the total number for every class in blue and hits number in red for model 3.

Model	Loss Type	Test Loss
Model 1	Cross Entropy Loss	1.532
Model 2	Focal Loss ($\alpha = 1$)	1.527
Model 3	Focal Loss ($\alpha = 0.25$)	0.166

TABLE 1: Test Loss for Different models of ResNet18

And the hits plot for the model 3 with best performance is as follow

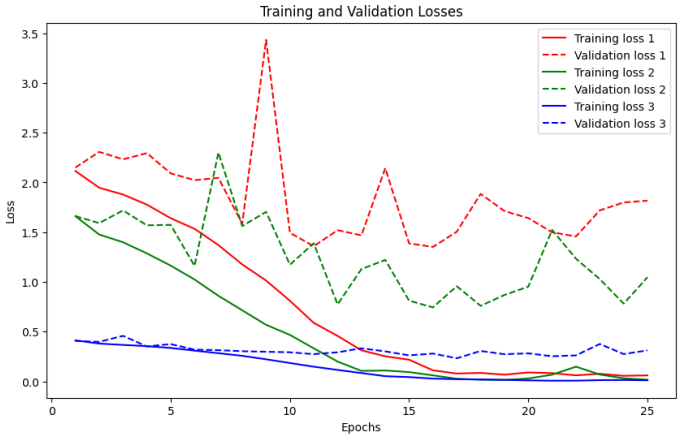

Fig. 11: Comparison of Training and Validation Losses-ResNet18.

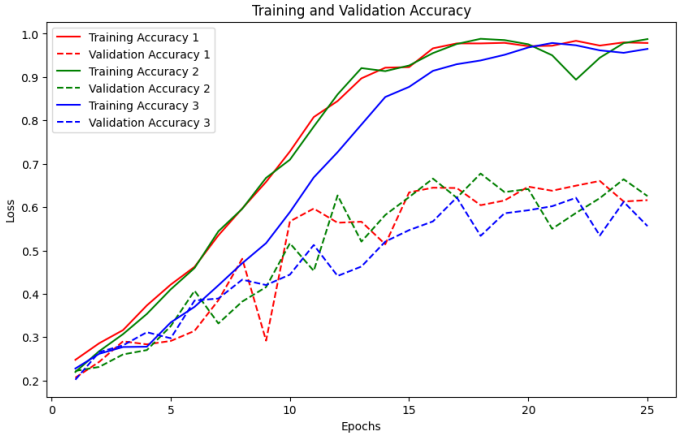

Fig. 12: Comparison of Training and Validation Accuracies-ResNet18.

Car Part Classification Figure 14 shows the comparison of training and validation accuracy of all three models for ResNet18 for classification of headlights. And Figure 15 shows the comparison of training and validation losses of all three models for ResNet18 for the headlights classification. Figure 16 plots the total number for every class in blue and hits number in red for the third model of classifying headlights. In the same manner, the comparison of training and validation loss and accuracy of taillight classification can be seen in the figures 17 and 18 respectively using ResNet18 architecture. Figure 19 plots the total number for every class in blue and hits number in red for model 3 in the case of classification of taillights.

Car Make Verification Figure 20 shows that the comparison between training and validation loss of the constructed Siamese network. Figure 21 shows that the dissimilarity score of the given image pair with the corresponding label.

VII. Concluding Remarks

In this study, we successfully utilized the CompCars dataset to perform car make classification, car part classification,

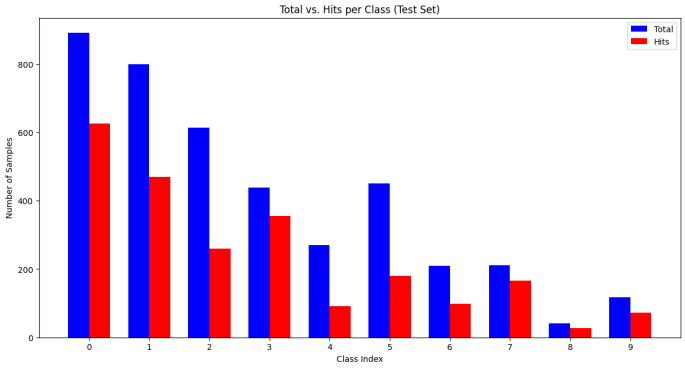

Fig. 13: Total vs. Hits per Class for Best Model 3

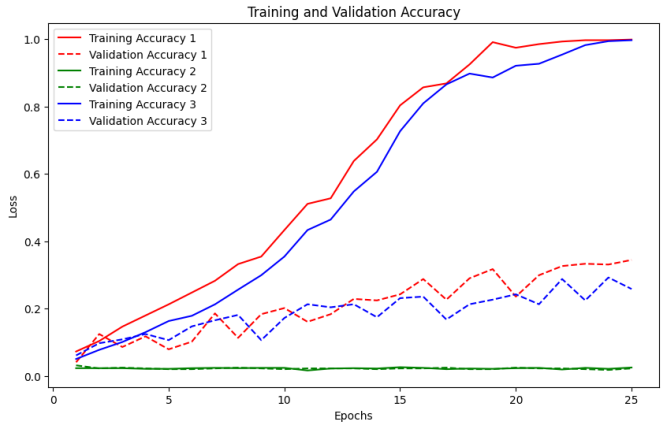

Fig. 14: Comparison of Training and Validation Accuracy for Headlight Classification using ResNet18

and car make verification using advanced CNN architectures like ResNet18, GoogLeNet, and Siamese networks. Our models demonstrated high accuracy, significantly improving over baseline methods. The accuracy could be improved with more resources as we struggled to improve our models due to lack of GPU, and sometimes we had to compromise to have small number of epochs due to this reason. These results highlight

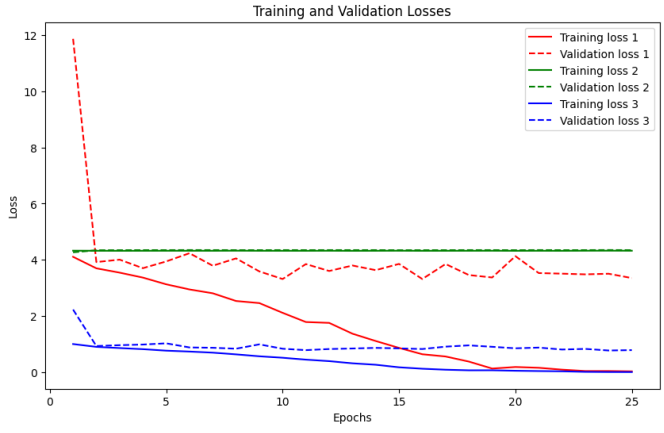

Fig. 15: Comparison of Training and Validation Loss for Headlight Classification using ResNet18

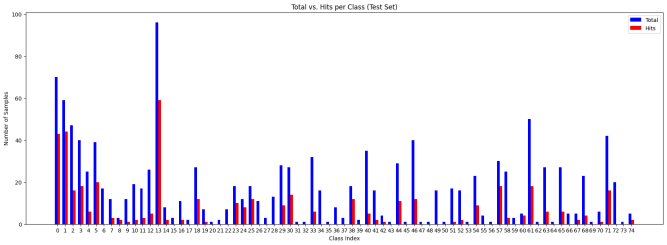

Fig. 16: Total vs. Hits per Class for Headlight Classification

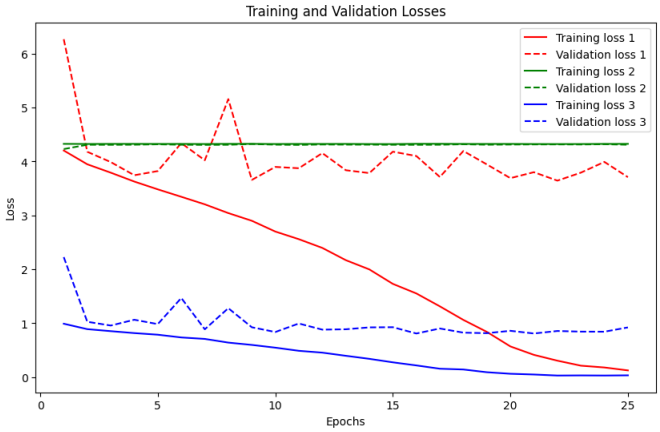

Fig. 17: Comparison of Training and Validation Loss for taillight classification using ResNet18

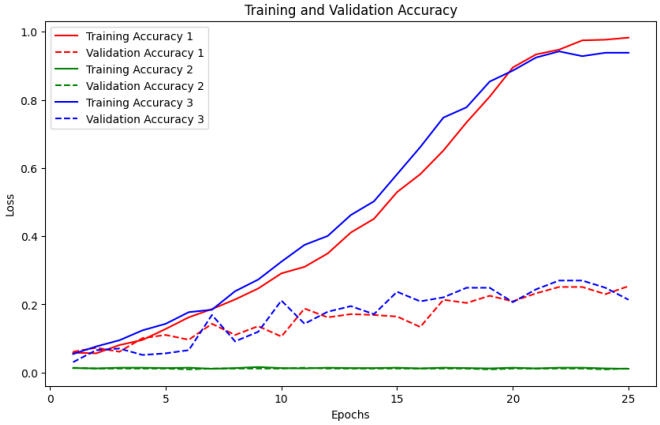

Fig. 18: Comparison of Training and Validation Accuracy for taillight classification using ResNet18

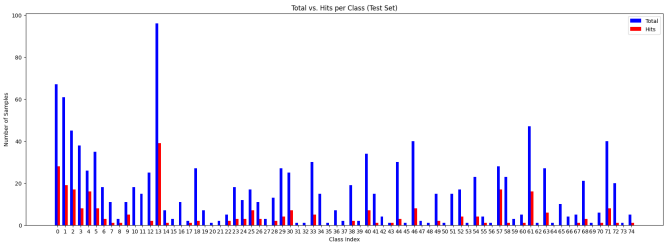

Fig. 19: Total vs Hits per Class for Best Model 3 of the taillight Classification

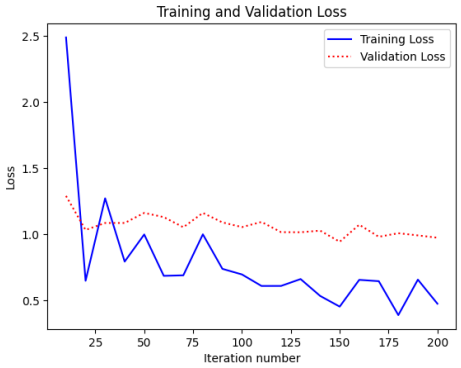

Fig. 20: Comparison of Training and Validation Accuracy for Siamese network

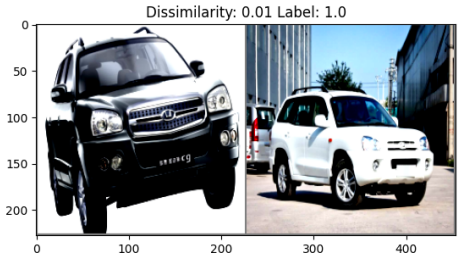

Fig. 21: Dissimilarity score

the potential of deep learning techniques in enhancing vehicle recognition systems, which can be integrated into commercial applications such as automated vehicle inspection and intelligent traffic monitoring. Future work could explore additional car parts, refine model architectures, and address class imbalances to further improve performance. This research contributes valuable insights and benchmarks for the ongoing development of intelligent transportation systems. With a test accuracy of 70.16, the Siamese Network demonstrated its strong performance in distinguishing between image pairs. The model might be overfitting, though, given that the validation error is somewhat larger than the training error. This indicates that even if the model has improved its performance on the training set, it has not yet fully generalised to new, untested data.

REFERENCES

[1] L. Yang, P. Luo, C. C. Loy, and X.Tang, "A large-scale car dataset for fine-grained categorization and verification.," in *CVPR*, 2015.

[2] Z. Zhang, T. Tan, K. Huang, and Y. Wang, "Three-Dimensional Deformable-Model-Based Localization and Recognition of Road Vehicles," *IEEE Transactions on Image Processing*, vol. 21, pp. 1–13, Jan. 2012.

[3] E. Hsiao, S. N. Sinha, K. Ramnath, S. Baker, L. Zitnick, and R. Szeliski, "Car make and model recognition using 3d curve alignment," in *In IEEE Winter Conference on Applications of Computer Vision*, (Steamboat Springs, CO, USA), June 2014.

[4] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly Optimizing 3D Model Fitting and Fine-Grained Classification," in *In ECCV*, Sept. 2014.

[5] J. Krause, M. Stark, J. Deng, and L. FeiFei, "3D object representations for fine-grained categorization.," in *In ICCV Workshops*, (Sydney, NSW, Australia), Dec. 2013.

[6] P. W. P. P. C. Wah, S. Branson and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," in *In ICCV Workshops*, (Technical Repport CNS-TR-2011-001), July 2011.

[7] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *In ECCV* , Oct. 2012.

[8] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *In ICVGIP*, (Bhubaneswar, India), Dec. 2008.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Technical Report 07-49*, (University of Massachusetts, Amherst), Oct. 2007.

[10] P. N. B. N. Kumar, A. C. Berg and S. K. Nayar, "Attribute and simile classifiers for face verification.," in *In ICCV*, 2009.

[11] Y. Sun, X. Wang, and X. Tang., "Deep learning face representation from predicting 10,000 classes," in *In CVPR*, 2014.

[12] Z. Zhu, P. Luo, X. Wang, and X. Tang., "Multi-view perceptron: a deep model for learning face identity and view representations.," in *In NIPS*, 2014.

[13] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun., "Bayesian Face Revisited: A Joint Formulation," in *In ECCV*, 2012.

[14] L. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," in *In ICCV*, (Barcelona, Spain), Nov. 2011.

[15] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Pedestrian Attribute Recognition At Far Distance," in *In ACM MM*, 2014.

[16] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev., "Panda: Pose aligned networks for deep attribute modeling.," in *CVPR*, 2014.

[17] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review.," in *T-PAMI*, 2006.

[18] H. S. S. B. C. Matei and S. Samarasekera, "Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features," in *In CVPR*, 2011.

[19] Y. Xiang, C. Song, R. Mottaghi, and S. Savarese., "TMonocular multiview object tracking with 3d aspect parts.," in *In ECCV* , 2014.

[20] K. He, L. Sigal, and S. Sclaroff, "Parameterizing object detectors in the continuous pose space," in *In ECCV*, 2014.

[21] J. L. L. Yang and X. Tang., "Object detection and viewpoint estimation with auto-masking neural network.," in *In ECCV* , 2014.

[22] M. Z. Zia, M. Stark, K. Schindler, and R. Vision, "Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects.," in *CVPR*, 2014.

[23] Y. J. Lee, A. A. Efros, and M. Hebert, "Style-aware mid-level representation for discovering visual connections in space and time," in *In ICCV*, 2013.