Car Part classification and Car Verification using through deep learning approachs, CNN and SNN

Amirhossein Kargar Khabbazi Sardrod[†], Ali Erfanian Omidvar[†]

***Abstract*—This project explores deep learning models for car part classification and car verification using the large-scale CompCars dataset. For car part classification, two popular convolutional neural networks, ResNet50 and InceptionV3, were compared using different loss functions_sparse_categorical_crossentropy and focal_loss. The models were evaluated to determine their effectiveness in handling real-world automotive image data. In our experiments, ResNet50 combined with focal_loss achieved the highest performance, with no misclassified images and a validation accuracy of 97.81%. InceptionV3 also performed competitively, particularly with focal_loss, achieving a validation accuracy of 98.82%. For car verification, a Siamese Neural Network (SNN) was built using ResNet50 and MobileNetV2 to compare pairs of images. ResNet50 with focal_loss outperformed the other models, reaching an accuracy of 95.64%, while MobileNetV2 achieved 86% accuracy on the test set. These results highlight the strength of focal_loss in improving model accuracy and handling class imbalances across both classification and verification tasks.**

***Index Terms*—Unsupervised Learning, Optimization, Neural Networks, Convolutional Neural Networks, SNN, Deep Learning.**

I. Introduction

There have been observed many challenges in the past with image classification as one of the most difficult problems that require efficient methods of problem solving using deep learning has been identified as one of the most effective solution techniques for the automotive industries. Self-detected car parts and inspection of the cars from the images in such applications such as vehicle automatic examination, traffic surveillance and security are becoming more crucial. This project focuses on two key tasks in this domain: car part classification and car verification we use Comp-Cars dataset which is contain 11,059 images of different model of car and parts.

a) Car Part Classification: The first task is car part classification, where the goal is to identify specific components of a vehicle from an image. To tackle this challenge, we experimented with two popular convolutional neural network (CNN) architectures: **ResNet50** and **InceptionV3**. These models are widely recognized for their ability to learn complex visual features and achieve state of the art results on large-scale image datasets.

ResNet50 (Residual Networks) They brought about residual connections which eliminated the issue of vanishing gradients in very deep networks thus revolutionizing the field of deep learning. As for ResNet50, the identity mappings enabled the model to skip several layers and train deep networks better. This is beneficial since it comprises a 50 layers deep structure ideal for image classification tasks to settle on a variety of image features. In this work, ResNet50 was trained with two different loss functions for proving the competency which are the default loss function of keras, sparse_categorical_crossentropy and the new introduced focal_loss. From the results, ResNet50 converged with focal_loss was the most accurate with no chance at all of miscategorised images during testing with validation accuracy of 97. 81 % that enhances its capability in addressing class imbalance.

InceptionV3, n the other hand, it has its peculiar inception modules that facilitate the multi scale processing of the visual data within a single convolutional block. InceptionV3 architecture uses different filter sizes at different levels and works in parallel to capture images at different resolutions it is very efficient for image classification. As for the last experiment, our InceptionV3 also yielded reasonable results especially with focal_loss, having a validation accuracy of 98. 82 percent and only one image misclassified during the testing phase. The ability of this model to recognize both the fine and gross features in the images for this experiment has boosted this models performance.

b) Car Verification Using Siamese Neural Networks: The second task in this project is car verification, where the goal is to determine whether two images represent the same car. This is particularly relevant in security and surveillance systems, where identifying vehicles across different camera angles or timeframes is critical. For this task, we employed a **Siamese Neural Network (SNN)** architecture, which is designed to learn similarity between image pairs. In a Siamese network, two identical neural networks share weights and process two input images to generate embeddings, which are then compared to measure similarity.

For car verification, we used **ResNet50** and **MobileNetV2** as the base models in the Siamese network. ResNet50, as described earlier, excels in learning deep and abstract visual features, making it well suited for the car verification task. When trained with focal_loss, ResNet50 achieved an accuracy of 95.64% and a loss of 0.0162, making it the best performing model for this task. In contrast, **MobileNetV2**, a more lightweight architecture optimized for mobile and resource constrained environments, offered faster computation but lower accuracy. MobileNetV2 with binary_crossentropy loss achieved an accuracy of 86%, which, although decent, was lower than ResNet50's performance.

[†]Department of Information Engineering, University of Padova, email: {name.surname}@unipd.it
Special thanks / acknowledgement go here.

MobileNetV2 is specifically designed for efficiency and speed, utilizing depthwise separable convolutions to reduce the number of parameters and computational cost. While it trades off some accuracy for faster inference, MobileNetV2 remains a viable option for real-time applications where computational resources are limited. However, in this project, the deep feature extraction capabilities of ResNet50 proved more advantageous for the car verification task, particularly in handling more challenging images.

c) Final Results: Comparing the results for the classification of car parts, it is highest with the ResNet50 with focal_loss having no misclaased images with a validation accuracy of 97. 81%. Said, InceptionV3 I get validation accuracy of 98. 82% when using focal_loss. In car verification, the model constructed with resnet50 and siamese network with focal loss rate was 95. 64%, though MobileNetv2, formed for less efficacy provided 86% in testing models, and ResNet50 provided 88. 36%. on the hard pair images with binary cross entropy loss and accuracies of 91. 33.

This experiment also show that ResNet50 is able to perform well in both classification and verification tasks, especially if combined with focal_loss. There is also the possibility of using InceptionV3 for classification, with MobileNetV2 still being good for application which do not require a large emphasis on the networkâ€™s accuracy.

In the next section, we describe the dataset used, data preprocessing steps, various models employed in the experiment and the experimental protocol used in this work, as well as details of analysis on results achieved and the comparison of performance of various configurations.

II. Related Work

New findings in computer vision and deep learning have enormous effects, evolving around the automobile industry in particular, on image classification and verification. Indeed, there is a significant volume of work done in the area of classification and verification of car parts, which becomes very important for auto-checking, traffic monitoring, and recognition of vehicles. This section discusses major advances in these fields and situates current work with respect to them.

a) Car Part Classification: Traditional methods for car part classification rely on handcrafted features as an input to machine learning, for instance Histograms of Oriented Gradients and Scale-Invariant Feature Transform [1]. Such methods were always plagued by the existence of high intra-class variation and complex backgrounds. High variability in images for the same class and complex backgrounds were often limitations. Deep learning replaced this domain entirely after the introduction of Convolutional Neural Networks (CNN) that allow automatic learning of hierarchical features from raw images. This has employed notable architectures such as ResNet [2] and Inception [3] in many tasks of classification, with state-of-the-art performance on large-scale image datasets..

In the automotive field, research by Yang et al. [4] used CNNs to classify fine-grained car models from images, achieving significant improvements over traditional methods. Similarly, the CompCars dataset [4] has been widely used for car model and part classification due to its extensive variety of car models and part annotations. Deep architectures like ResNet and Inception, with their ability to learn complex visual patterns, have become the preferred choice for these tasks. The current project builds on these findings by utilizing both ResNet50 and InceptionV3 for car part classification, comparing their effectiveness with different loss functions, including focal loss, which has shown to improve performance in imbalanced datasets.

b) Car Verification: Vehicle verification, also known as vehicle re-identification, is a critical task in automotive security and surveillance systems. The goal is to determine whether two images represent the same car, even when captured from different viewpoints or under different conditions. Traditional approaches for vehicle verification used handcrafted features and similarity metrics, such as SIFT and SURF . However, these methods struggled with robustness across different lighting conditions and angles.

Recent work has focused on leveraging deep learning models, particularly Siamese Neural Networks (SNNs), for vehicle verification tasks. SNNs are designed to compare two images by learning a similarity function that minimizes the distance between representations of the same object while maximizing the distance between different objects. Studies like Zhou et al. [5] applied Siamese networks using deep feature extraction models like ResNet and demonstrated significant improvements in vehicle re-identification accuracy. The use of deeper CNN models, such as ResNet50 and MobileNetV2, has enabled more robust and efficient feature extraction, even in resource-constrained environments.

This project extends these techniques by employing Siamese networks for car verification, comparing the performance of ResNet50 and MobileNetV2 as backbone models. The use of focal loss further enhances the modelâ€™s ability to handle class imbalance, improving the accuracy of verification.

c) Handling Class Imbalance with Focal Loss: Class imbalance is a common issue in real-world datasets, particularly in tasks like car part classification, where certain parts or models may be overrepresented compared to others. Traditional loss functions like categorical cross-entropy often perform poorly in such scenarios, leading to biased model predictions. Focal loss [6], introduced as a modification of cross-entropy loss, mitigates this issue by focusing on hard-to-classify examples. It down-weights the loss contribution from easy examples, allowing the model to focus more on underrepresented or challenging cases. Several studies have reported improved classification performance on imbalanced datasets using focal loss, particularly in tasks involving object detection and fine-grained image classification.

This project adopts focal loss for both car part classification and car verification tasks, demonstrating its effectiveness in improving model generalization and handling dataset imbalances.

d) Summary: To that end, previous experiments in the area of car part classification as well as verification of automobiles have indicated that deep learning architectures such as ResNet and Inception are ideal when it comes to modeling difficult visual patterns. Siamese networks coupled with strong CNN backbones have been used to achieve good results in verification tasks because they learn good distance metrics to use when comparing images. The approach of using focal loss has been found to be useful in solving problems of class imbalance, which makes it spot on for the current project. This project extends these developments further by utilizing the most recent deep learning strategies for the classification of car parts and the verification of car identity utilizing the CompCars dataset to analyse the performance of different models and loss functions

III. PROCESSING PIPELINE

The project employs two distinct deep learning pipelines to address the tasks of car part classification and car verification. These tasks are critical in automotive applications such as automated inspection, vehicle tracking, and security systems. Both tasks utilize the CompCars dataset, which contains thousands of images, with car part classification focusing on categorizing specific car components, and car verification determining whether two images depict the same car.

For car part classification, the primary goal is to accurately identify specific components of a vehicle from images. The deep learning architectures used for this task are ResNet50 and InceptionV3, two widely known Convolutional Neural Networks (CNNs). These models were trained from scratch, and two loss functions_sparse_categorical_crossentropy and focal_loss were employed to evaluate their performance on imbalanced datasets. The models were trained using a pipeline that involves data loading, preprocessing, and constructing TensorFlow datasets to ensure efficient training.

In the car verification task, a Siamese Neural Network (SNN) is used to learn whether two images represent the same car. This task is often framed as a binary classification problem, where the network outputs a similarity score between two input images. The backbone architectures used in this task are ResNet50 and MobileNetV2, with L1 distance calculations used to compare feature vectors. Similar to car part classification, the models were trained using a flexible pipeline that allows interchangeable use of different models and loss functions, including binary_crossentropy and focal˙loss.

This section details the complete processing pipeline for both tasks, including data preparation, model construction, training, and evaluation. Despite the differences in the tasks, the pipelines share common elements such as efficient data loading, modular architectures, and robust model training, allowing for easy experimentation with different neural networks and loss functions.

A. Car Part Classification

The processing pipeline for car part classification is designed to handle the complexities of the CompCars dataset, including the preprocessing of images, training deep neural networks, and evaluating the model's performance. The key stages of this pipeline include data preparation, model construction, and the training and evaluation processes. Notably, both the ResNet50 and InceptionV3 architectures are utilized interchangeably, with only minor adjustments in the model initialization, and all models are trained from scratch without using pre-trained weights.

1. Data Preparation The initial phase of the pipeline is the preparation of the dataset, which includes loading image paths, validating the corresponding labels, and applying image preprocessing steps to standardize the input for the neural networks.

Loading Image Paths and Labels: Image paths and their associated labels are loaded from text files containing the dataset split for training and testing. Each class (car part) is represented by its corresponding image paths, which are mapped to a unique integer label. The read_image_paths() function loads these images into memory, followed by a label validation step where the correctness of the labels is ensured by extracting labels from the file path structure and comparing them to the provided labels.

Image Preprocessing: Each image is preprocessed before being fed into the neural network models. The preprocessing involves resizing the images to the required input size of 224x224 pixels and normalizing pixel values to the [0,1] range. This step ensures that the images are compatible with both ResNet50 and InceptionV3 models, which expect fixed-size inputs. Additionally, the labels are one-hot encoded to match the output shape of the softmax activation in the network.

Dataset Construction: The TensorFlow dataset API is utilized to construct efficient pipelines for training and validation datasets. The entire dataset is shuffled and split into training (70%) and validation (30%) sets. These datasets are batched with a batch size of 16 and prefetched for optimal data loading performance during training. To ensure the robustness of the training process, the train˙dataset and val˙dataset are both repeated to provide continuous data for multiple epochs.

2. Model Construction The pipeline allows for flexibility in the choice of neural network architecture, with both ResNet50 and InceptionV3 being used for car part classification. These models are trained from scratch, without any pre-trained weights, ensuring that the network learns directly from the provided car part images.

Base Model: For both architectures, the base model (either ResNet50 or InceptionV3) is initialized without pre-trained weights (weights=None) to start training from scratch. This enables the network to learn specific features relevant to the task of car part classification from the CompCars dataset.

Custom Layers: On top of the base architecture, custom layers are added to adapt the models for classification. These custom layers include:

GlobalAveragePooling2D: Applied to reduce the spatial dimensions of the feature maps output by the convolutional layers. **Dropout (0.5):** Added to prevent overfitting by randomly disabling some neurons during training. **Dense Layer (512 units):** A fully connected layer with ReLU activation and L2 regularization is introduced to provide nonlinearity. **BatchNormalization:** Applied to normalize activations and accelerate training convergence. **Final Dense Layer (softmax):** The output layer contains num_classes units (equal to the number of car parts), with softmax activation to produce class probabilities. This modular construction ensures that the architecture remains flexible, allowing easy replacement of models (e.g., switching between ResNet50 and InceptionV3) while maintaining the same pipeline structure.

3. Loss Function and Optimization The training process evaluates the performance of two loss functions: sparse_categorical_crossentropy and focal_loss. These loss functions are used interchangeably by simply swapping them during model compilation.

Sparse Categorical Crossentropy: This loss function is used for standard multiclass classification tasks, where the goal is to minimize the difference between the predicted class probabilities and the true class labels.

Focal Loss: Focal loss is introduced to address the issue of class imbalance, which is common in realworld datasets. By focusing on hard to classify examples (through alpha=0.25 and gamma=2.0), focal loss helps the model prioritize these samples during training, resulting in improved performance for underrepresented classes.

Optimizer: The Adam optimizer is used across all training configurations with a learning rate of 1e-4. Additionally, gradient clipping is applied (clipnorm=1.0) to ensure stability during training, preventing the issue of exploding gradients that can occur when training deep networks from scratch.

4. Training Process The models are trained for 20 epochs, with the training and validation sets being fed continuously to the neural network. Both ResNet50 and InceptionV3 follow the same training process, where the only changes involve the underlying architecture and the choice of loss function.

Batch Size: A batch size of 16 is used for both the training and validation datasets. **Steps per Epoch:** The number of steps per epoch is calculated based on 70% of the dataset size, ensuring sufficient iterations per epoch for effective learning. **Validation Steps:** Validation is performed at each epoch, with the number of steps calculated based on 30% of the dataset. The training process monitors both accuracy and loss across epochs for both the training and validation datasets, providing insights into the model's convergence and generalization capabilities.

5. Evaluation and Testing After training, the models are evaluated on a set of test images that were not part of the training or validation datasets. The test phase includes the following steps:

Image Loading and Preprocessing: Similar to the training data, test images are preprocessed by resizing them to 224x224 pixels and normalizing the pixel values to [0,1].

Prediction: Each test image is passed through the trained model using the predict_image() function. The predicted class is determined by selecting the class with the highest softmax probability. These predictions are compared to the true labels to calculate the number of correct and incorrect classifications.

The test results show high performance, particularly when using focal˙loss, with only 0 or 1 misclassified images during testing, depending on the model and loss function configuration. For instance, when using ResNet50 with focal_loss, no misclassifications occurred in a test set of 53 images, highlighting the model's robustness in handling diverse car part images.

B. *Car Verification*

The car verification task involves training a Siamese Neural Network (SNN) to determine whether two images represent the same car. The pipeline for this task is designed to handle the specific challenges of working with image pairs and a binary classification problem. The steps include data preparation, model construction, and training, with flexibility to interchange models and loss functions. For this project, both ResNet50 and MobileNetV2 were used, with minimal changes in the pipeline, as well as interchangeable loss functions.

1. Data Preparation The data preparation step is crucial in constructing the verification dataset, where each sample consists of a pair of images and a label indicating whether the two images depict the same car.

Loading and Splitting the Dataset: The dataset is loaded from a file containing image pairs and their associated labels. Each line in the file consists of two image paths and a binary label (1 for a match and 0 for no match). The function read˙verification˙file() reads the file and splits the data into three sets: training (60

After splitting, the dataset contains 12,000 pairs for training, 4,000 pairs for validation, and 4,000 pairs for testing. This ensures a sufficient number of examples for the model to learn the verification task.

Image Preprocessing: Each image in the pair is preprocessed to meet the input requirements of the neural network models. The preprocessing steps include resizing each image to 224x224 pixels and normalizing the pixel values to the [0,1] range. The images are then loaded into a TensorFlow dataset pipeline, which is used to efficiently feed batches of image pairs and their corresponding labels into the model during training.

TensorFlow Dataset Construction: The TensorFlow dataset API is employed to create pipelines for training, validation, and testing datasets. The dataset is shuffled and split into batches of 128 pairs. For optimal data loading performance, the dataset is pre-fetched using TensorFlow AUTOTUNE, ensuring that data is continuously fed to the model without interruptions during training.

2. Model Construction The core of the car verification task lies in the construction of the Siamese Neural Network (SNN). The SNN consists of two identical subnetworks, each processing one of the two input images to extract features. The features from both images are then compared to determine their similarity.

Base Model: The base architecture for feature extraction is flexible, with ResNet50 and MobileNetV2 being used interchangeably in different experiments. In the provided example, ResNet50 is used as the base model. It is initialized with pre-trained ImageNet weights to take advantage of general visual features and is then modified to output 128-dimensional feature vectors. The base model is frozen (non-trainable) to focus training on the final layers.

Custom Layers: The base model is followed by a few custom layers for feature extraction:

GlobalAveragePooling2D: Applied to reduce the spatial dimensions of the feature maps to a fixed-size vector.

Dense Layer (128 units): A fully connected layer with ReLU activation is used to produce a compact representation of the features.

L2 Normalization: Applied to the feature vectors to ensure that the distance metric used later in the network is consistent. The same feature extraction model is used for both input images in the Siamese network, ensuring that the feature vectors for both images are extracted in the same way.

3. Siamese Network and Distance Calculation The Siamese Neural Network is designed to compute the similarity between two images by comparing their feature vectors.

Siamese Network Construction: The SNN takes two images as input. Both images are passed through the feature extraction model, which produces two 128-dimensional feature vectors. These feature vectors are then compared using the L1 distance (the absolute difference between the two vectors).

Output Layer: The output layer consists of a single neuron with a sigmoid activation function. This layer produces a probability score representing the likelihood that the two images are of the same car. A score closer to 1 indicates a match, while a score closer to 0 indicates that the images do not depict the same car.

4. Loss Function and Optimization The model is compiled with the binary_crossentropy loss function for the binary classification task. However, in different runs, focal_loss was also used to handle any class imbalance that might occur between matching and non-matching pairs.

IV. Results

A. Car Part Classification

The car part classification task was evaluated using two architectures ResNet50 and InceptionV3 trained with two loss functions: sparse categorical crossentropy and focal˙loss. Both models were trained for 20 epochs, and their performance was evaluated on the training, validation, and test sets. The results highlight the impact of the loss function on model accuracy, loss, and generalization to unseen data.

Training and Validation Results The following table summarizes the training and validation results for both models with different loss functions:

TABLE 1: Training and Validation Results for Car Part Classification

Model	Loss Function	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
ResNet50	Sparse Categorical Crossentropy	0.9867	0.2303	0.9662	0.2992
ResNet50	Focal Loss	0.9609	0.0910	0.9781	0.0666
InceptionV3	Sparse Categorical Crossentropy	0.9835	0.1209	0.9894	0.0921
InceptionV3	Focal Loss	0.9674	0.0444	0.9882	0.0276

The results show that focal˙loss consistently results in lower validation loss for both models, indicating better generalization to unseen data. InceptionV3 with focal˙loss achieved the lowest validation loss (0.0276), while ResNet50 with focal˙loss also demonstrated strong performance with a validation loss of 0.0666.

Test Results To assess the generalization capabilities of the models, a separate test set of 53 images was used. The test results are summarized in the table below:

TABLE 2: Test Results for Car Part Classification

Model	Loss Function	Total Images	Wrong Predictions	Accuracy
ResNet50	Sparse Categorical Crossentropy	53	2	96.23%
ResNet50	Focal Loss	53	0	100%
InceptionV3	Sparse Categorical Crossentropy	53	2	96.23%
InceptionV3	Focal Loss	53	1	98.11%

ResNet50 with focal_loss achieved a perfect test accuracy of 100%, correctly classifying all 53 images. InceptionV3 with focal_loss also performed well, misclassifying only one image with an accuracy of 98.11%. Models trained with sparse_categorica_crossentropy demonstrated slightly lower accuracy, with both ResNet50 and InceptionV3 misclassifying 2 images.

Overall Analysis

The results demonstrate that both models, ResNet50 and InceptionV3 are highly effective for car part classification. However, focal_loss consistently outperforms sparse_categorical_crossentropy in terms of both validation loss and test accuracy. This suggests that focal˙loss helps mitigate class imbalances, leading to improved generalization on unseen data.

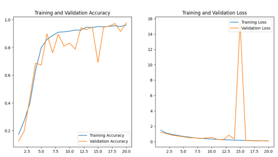
ResNet50 with focal_loss achieved the best overall performance, including a perfect classification accuracy on the test set. InceptionV3 also performed strongly, with focal_loss leading to better validation and test results compared to sparse_categorical_crossentropy. The results indicate that the choice of loss function plays a crucial role in optimizing model performance, especially when dealing with imbalanced datasets like car part classification.

B. Car Verification

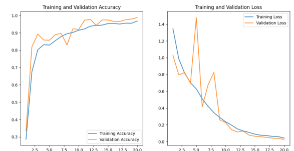
The car verification task was evaluated using ResNet50 and MobileNetV2 models, trained on different datasets and loss functions. The models were trained for 20 epochs, and their performance was evaluated on both the training and validation datasets for easy and medium difficulty image pairs. The results indicate the varying effectiveness of each model and loss function, with focal˙loss yielding the best performance in terms of accuracy and validation loss.

Training and Validation Results The following table summarizes the training and validation results for the models:

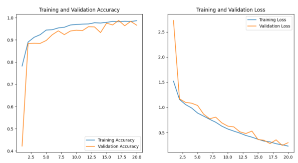
The results indicate that ResNet50 with focal˙loss outperforms other configurations, particularly on the easy pair



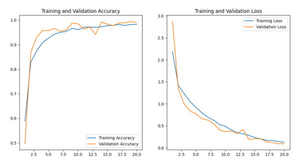
Results for ResNet50 using Focal Loss



Results for InceptionV3 using Focal Loss



Results for ResNet50 using sparse categorical crossentropy



Results for InceptionV3 using sparse categorical crossentropy

Fig. 1: Comparing the traning results for different models on different loss functions.

TABLE 3: Training and Validation Results for Car Verification

Model	Dataset	Loss Function	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
ResNet50	Easy	Focal Loss	0.9564	0.0162	0.7675	0.0431
ResNet50	Easy	Binary Crossentropy	0.8836	0.4695	0.7430	0.5651
ResNet50	Medium	Binary Crossentropy	0.9133	0.4841	0.7295	0.5828
MobileNetV2	Easy	Binary Crossentropy	0.7588	0.5705	0.6270	0.6421

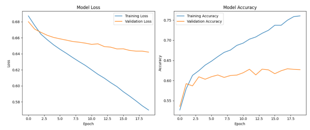
dataset, achieving a high training accuracy of 95.64% and a low validation loss of 0.0431. In contrast, ResNet50 trained with cross˙entropy on the same dataset achieved lower accuracy and higher validation loss (0.5651). Additionally, training on the medium pair dataset with cross˙entropy resulted in a slightly higher training accuracy (91.33%) but a similar validation accuracy (72.95%) and higher validation loss (0.5828).

MobileNetV2, while more computationally efficient, achieved the lowest performance, with a training accuracy of 75.88% and a validation accuracy of 62.70% on the easy pair dataset. The higher validation loss (0.6421) indicates that MobileNetV2 struggled to generalize as effectively as ResNet50.

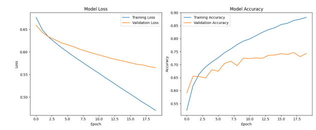
Overall Analysis The results demonstrate that ResNet50 with focal˙loss is the best-performing model for car verification tasks, particularly on the easy dataset, where it achieved the highest accuracy and lowest validation loss. In contrast, MobileNetV2 and ResNet50 with cross˙entropy showed weaker performance, especially in terms of validation loss, suggesting that these models struggled to generalize as well as ResNet50 with focal˙loss.

V. Concluding Remarks

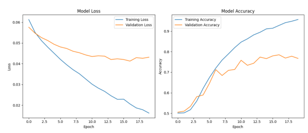
In conclusion, this project successfully demonstrated the application of deep learning techniques for car part classification and car verification using the CompCars dataset. By employing two prominent CNN architectures, ResNet50 and InceptionV3, along with Siamese Neural Networks for verification, we explored the potential of these models in real-world automotive scenarios.



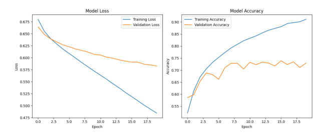
Results for MobileNet using binary˙crossentropy on easy pair dataset



Results for ResNet50 using binary˙crossentropy on easy pair dataset



Results for ResNet50 using Focal Loss on easy pair dataset



Results for ResNet50 using binary˙crossentropy on medium pair dataset

Fig. 2: Comparing the traning results for different models on different loss functions for car verification using SNN.

The results highlight the effectiveness of ResNet50, especially when combined with focal loss, which proved crucial in handling class imbalances and delivering near-perfect accuracy in both classification and verification tasks. InceptionV3 also showed strong performance, particularly in classification, while MobileNetV2, despite being optimized for efficiency, achieved reasonable results in verification but with a trade-off in accuracy.

The flexibility of the pipelines used in this project allows for easy experimentation with different architectures and loss functions, making it a robust framework for further improvements. This study emphasizes the importance of model selection and loss function optimization, particularly in complex tasks with imbalanced datasets, and provides a solid foundation for future advancements in automotive image recognition systems.

References

[1] Ù. G. Lowe, "Distinctive image features from scale," *The Bell System Technical Journal*, vol. 60, pp. 379–423, Nov. 2004.

[2] S. R. ÙK. He, X. Zhang and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 90, pp. 379–423, June 2016.

[3] S. I. J. S. ÙC. Szegedy, V. Vanhoucke and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 19, June 2016.

[4] P. . L. C. C. . T. X. ÙYang, Linjie Luo, "A large-scale car dataset for fine-grained categorization and verification," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 19, June 2015.

[5] Y. ÙZhou, "A large-scale car dataset for fine-grained categorization and verification," *IEEE Trans Image Process.*, vol. 19, July 2018.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. DollÃ¡r, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.