UNIVERSITY OF CALIFORNIA
Los Angeles

Application of Siamese Neural Networks to Automotive Parts Image Recognition

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics & Data Science

by

Yewon Lee

2023

ABSTRACT OF THE THESIS

Application of Siamese Neural Networks
to Automotive Parts Image Recognition

by

Yewon Lee
Master of Applied Statistics & Data Science
University of California, Los Angeles, 2023
Professor Hongquan Xu, Chair

This paper focuses on enhancing the recognition of automotive parts by applying Convolutional neural network (CNN) and Siamese neural networks with contrastive loss and triplet loss. Simple CNN model encounters challenges due to class imbalance and the difficulty of learning distinct features for each category. To overcome these limitations, a Siamese neural network with Contrastive loss is implemented. This approach involves utilizing pairs of images and employing two identical models, leading to improved generalization to unseen data and achieving a test accuracy of 70.24%. However, when utilizing triplet loss in the Siamese neural network, the test accuracy remains low. The inadequate performance of the triplet loss model can be attributed to suboptimal sampling methods and a lack of control over the distances between anchors, positive anchors, and negative anchors during triplet selection. Overall, this study demonstrates the potential of Siamese neural networks with contrastive loss for enhancing automotive part recognition, highlighting the need for careful selection of loss functions and sampling strategies in improving model performance.

The thesis of Yewon Lee is approved.

Nicolas Christou

Ying Nian Wu

David Anthony Zes

Hongquan Xu, Committee Chair

University of California, Los Angeles

2023

To all the individuals

who have supported me

throughout my new journey

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Throughout history, image classification has been developed in many ways as there has been a huge improvement in technology. For consumers, image search technology allows them to find products based on visual similarity or specific visual features. This enables them to discover new products and brands that align with their preferences and style, expanding their choices and driving consumption.

These image recognition deep learning models can be applied to detect and identify car parts. The extensive variety of automotive parts and their different types can make it challenging for non-experts to find the right products when their cars break down or require specific parts. This lack of expertise and specificity among consumers can complicate online searches for suitable automotive products. The application of image recognition in the automotive industry will enhance the customer experience by simplifying the process of identifying and purchasing the correct car parts.

In prior studies, various deep learning models have been used in the analysis of automobile images and car part images. In the context of image segmentation, it has primarily been employed to develop models that segment the car's body into multiple sections and identify the damaged sections. [13] For image recognition, research has focused on creating models to detect defects by comparing images of normal parts with the images of defected parts. [12] Although there has been research where multiple deep learning models were applied to classify automobile parts, the scope was limited to a small number of categories (eight parts). [9] Due to the wide range of classes and the lack of distinct characteristics for each

part, there has been limited research on using deep learning models to classify automotive part images into specific part names.

In this paper, one of the well-known computer vision techniques, Convolutional neural network, will be employed to address the problem. In addition, Siamese neural networks, consisting of identical Convolutional neural networks connected at their outputs, with contrastive loss and triplet loss functions will be applied to yield improved outcomes.

The image data was sourced from CarParts.com., an online retailer specializing in automotive parts and accessories. It offers a wide range of products for various makes and models of vehicles, such as engine components, brakes, lighting, and suspension parts.

CHAPTER 2

Methodology

2.1 Convolution Neural Network

Convolutional Neural Networks (CNNs) [11] are a specialized type of neural network designed to handle data represented by multiple arrays. These arrays can be 1D for signals and sequences, including language; 2D for images or audio spectrograms; or 3D for video or volumetric images. The main advantage of CNNs lies in their ability to reduce the number of parameters compared to traditional Artificial Neural Networks (ANNs).

CNNs are particularly well-suited for image recognition tasks due to their remarkable proficiency in identifying intricate patterns and features within images. Moreover, they exhibit high efficiency in processing large volumes of data, making them indispensable for tasks such as object detection and classification.

The architecture of a CNN consists of four main layers: convolutional layer, pooling layer, and fully-connected layer. These layers work in tandem to enable the network to extract relevant features, downsample the data, and establish connections between different layers for comprehensive analysis.

2.1.1 Convolutional Layer

The first layer is the convolutional layer [6], which is a fundamental component of CNNs and plays a crucial role in processing and extracting features from input data, particularly images. The operation involves sliding the filter over the input image and computing the

element-wise product between the corresponding values of the filter and the image at each position. These products are then summed up to obtain the final output value at each position.

$$s(t) = \int x(a)w(t-a)da \tag{2.1}$$

where s represents the feature map as output, x represents input, w represents kernel or filter and t represents the variable or parameter over which the convolution operation is performed. It could be time, space, or any other dimension along which the functions x and w are defined. a represents the integration variable, a dummy variable used in the context of integration.

This operation is called convolution and it is typically denoted with an asterisk:

$$s(t) = (x * w)(t) \tag{2.2}$$

If the functions x and w are defined only on integer values, the discrete convolution equation can be represented as:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.3}$$

In machine learning, the typical scenario involves working with input data in the form of a multidimensional array. Similarly, the kernels used in the learning process are also multidimensional arrays, which are called tensors, comprised of adjustable parameters. For example, in the case of using a two-dimensional image I as the input, it is likely that a two-dimensional kernel K would be employed as:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \tag{2.4}$$

where (i, j) represents the coordinates of the output feature map and (m, n) refers to the column and row axes, respectively.

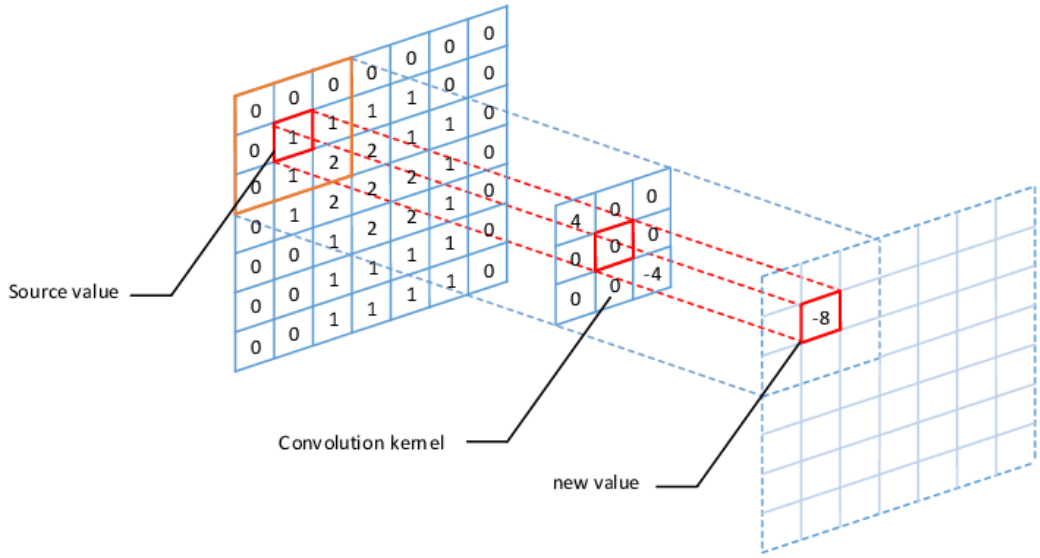


Figure 2.1: Convolution process illustration [7]

2.1.2 Pooling Layer

After obtaining the feature maps, we may perform pooling [6] to simplify the output by conducting downsampling. This downsampling process reduces the number of parameters that the model needs to learn. Two types of pooling are commonly used in CNNs: max pooling and average pooling. In max pooling, the maximum value within each pooling region is selected as the output. On the other hand, average pooling calculates the average value within each pooling region and uses it as the output.

2.1.3 Fully-connected (FC) Layer

In a fully connected layer [3], every neuron is connected to every neuron in the previous layer, forming a fully connected network structure. In the context of CNNs, fully connected layers are typically placed at the end of the network architecture, following the convolutional and pooling layers. The purpose of the fully connected layer is to combine the extracted features from previous layers and make predictions or classifications based on those features.

Each neuron in a fully connected layer receives inputs from all neurons in the previous layer. The output of each neuron in the fully connected layer is computed using a weighted sum of the inputs, followed by the application of an activation function. These weights are learned during the training process, allowing the network to adjust and optimize its performance.

2.1.4 Non-linearity Activation Function

An activation function [3] is applied to the output of each convolutional layer or fully connected layer. The activation function introduces non-linearity to the network, enabling it to model complex relationships and capture intricate features within the data. The activation function takes the weighted sum of inputs and applies a non-linear transformation to produce the output of a neuron. This output is then passed on to the next layer in the network. The activation function plays a crucial role in determining the output range and behavior of each neuron. Commonly used activation functions in CNNs include Rectified Linear Unit (ReLU), Sigmoid, Hyperbolic Tangent (tanh) and Leaky ReLU.

2.2 Siamese Neural Network

2.2.1 Siamese Neural Network

Siamese neural networks [4] represent a distinctive neural network architecture that deviates from the typical approach of training a model to classify inputs. The focus of siamese networks is on training the neural networks to distinguish and discern the dissimilarity or similarity between two inputs. The primary objective is to learn and quantify the level of similarity existing between the inputs.

Figure 2.2 shows that the architecture of Siamese networks consists of two identical feedforward neural networks connected at their outputs. These networks process input profiles and update weights through error back-propagation. The outputs of the networks are

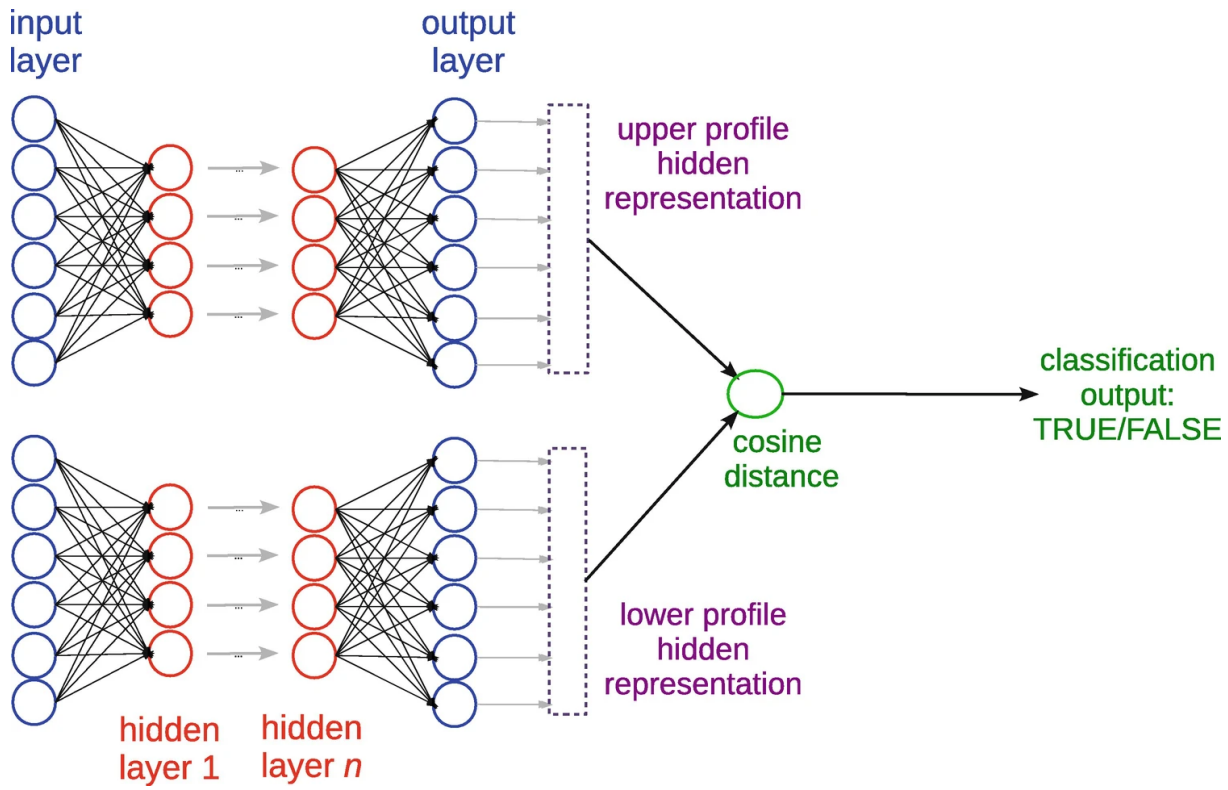


Figure 2.2: Siamese Neural Network: The Architecture [4]

compared using distance metrics such as Euclidean distance, Cosine similarity, Manhattan distance (L1 distance) and Hamming distance. Based on this similarity measure, the algorithm determines whether the profiles are similar or different and assigns positive or negative labels accordingly.

2.2.2 Contrastive Loss

In Siamese networks, the objective of training is to discover a parametric function, denoted as G_W, that transforms the images in such a way that the feature points of similar input pairs become close to each other in the feature space, while the feature points of dissimilar input pairs become distant. The contrastive loss is utilized to achieve this goal by minimizing its value. Let X_1, X_2 be a pair of input vectors and Y be a binary label assigned to this pair.

$$Y = \begin{cases} 0 & \text{if } X_1 \text{ and } X_2 \text{ are deemed similar} \\ 1 & \text{if } X_1 \text{ and } X_2 \text{ are deemed dissimilar} \end{cases} \tag{2.5}$$

The parameterized distance function D_W between X_1 and X_2 is defined as the euclidean distance between the outputs of G_W. This means that D_W measures the dissimilarity or similarity between X_1 and X_2 based on the feature representations obtained by passing them through G_W. By calculating the euclidean distance between these feature representations, the distance function captures the geometric separation or proximity between the samples in the learned feature space.

$$D_W(X_1, X_2) = \left\| G_W(X_1) - G_W(X_2) \right\|_2 \tag{2.6}$$

The general form is:

$$\mathcal{L}(W) = \sum_{i=1}^{P} L(W, (Y, X_1, X_2)^i) \tag{2.7}$$

$$L(W,(Y,X_1,X_2)^i) = (1-Y)L_S(D_W^i) + YL_D(D_W^i) \tag{2.8}$$

where $(Y,X_1,X_2)^i$ is the ith labeled sample pair, L_S is the partial loss function for a pair of similar points, L_D is the partial loss function for a pair of dissimilar points, P is the number of training pairs, and D_W^i is the shortened notation $D_W(X_1,X_2)$ of ith sample pair.

By jointly optimizing L_s and L_d (minimizing $\mathcal{L}$ with respect to W), the network can learn discriminative representations that effectively distinguish between similar and dissimilar pairs. With a margin of $m > 0$, the exact loss function is:

$$L(W,Y,X_1,X_2) = (1-Y)\frac{1}{2}(D_W)^2 + Y\frac{1}{2}max(0,(m-D_W)^2) \tag{2.9}$$

2.2.3 Triplet Loss and Triplet Anchor Sampling

2.2.3.1 Triplet Loss

Unlike Contrastive Loss, the triplet loss [5] considers triplets of samples instead of comparing pairs: an anchor sample, a positive sample (similar to the anchor), and a negative sample (dissimilar to the anchor). The goal is to ensure that the distance between the anchor and positive samples is smaller than the distance between the anchor and negative samples by a margin.

The key difference between the two loss functions lies in the number of samples they consider. While the contrastive loss compares pairs of samples, the triplet loss works with triplets. The triplet loss takes advantage of the relative relationships within a triplet to learn features that can effectively distinguish between different classes or concepts.

When $f(\theta)$ refers to the base model and x_i^a (anchor), x_i^p (positive), x_i^n (negative) are a set of triplet from the set of all possible triplets in the training set, we want to ensure that an anchor image is closer to all other similar images than it is to any other dissimilar images

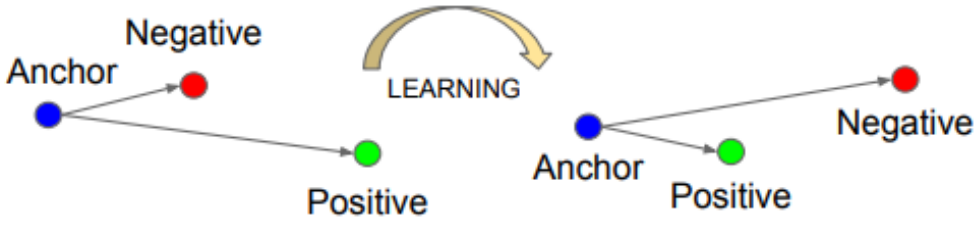


Figure 2.3: The Triplet Loss [5]

by margin $m > 0$.

$$\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 + m < \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 \tag{2.10}$$

The triplet loss encourages the anchor-positive distance to be smaller than the anchor-negative distance by at least the margin value. Thus, loss function that is being minimized can be defined as follows:

$$\sum_i^N \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + m \right] \tag{2.11}$$

2.2.3.2 Triplet Anchor Sampling

Since the triplet loss is designed to learn effective embeddings by leveraging relative distances between instances in a dataset, sampling methods play a crucial role in training Siamese neural networks with a triplet loss function. The process of training is sensitive to the selected triplets. [15]

Offline and Online Triplet Mining Triplets can be selected in two different ways: they are either pre-determined before training starts, or they are dynamically chosen during the training process.

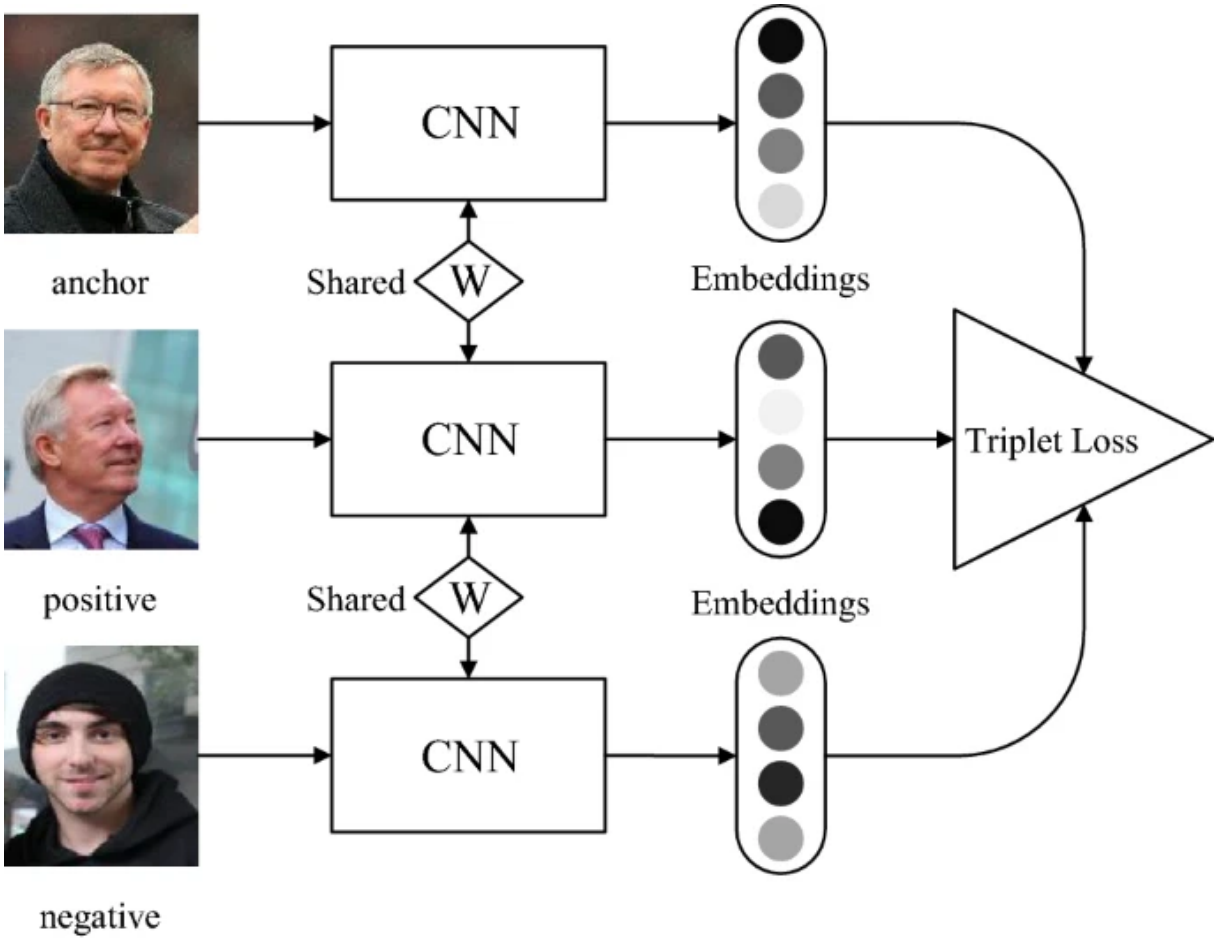


Figure 2.4: Siamese Neural Network with Triplet Loss: The Architecture [17]

Offline Triplet Mining One straightforward and intuitive approach is offline triplet mining, where all triplets are pre-generated by selecting a random image from the dataset before the training session, while all training samples are taken into account. It involves randomly selecting an anchor first, sampling a positive anchor from the same category which the anchor is from and a negative image from all other categories. After sampling, triplets that are either excessively easy or excessively challenging for the model are excluded and the remaining triplets are employed for training purposes. Though this approach has notable advantages that it is characterized by its simplicity and straightforward implementation, its effectiveness heavily relies on the construction of triplets. If the triplets are excessively simple, the learning process becomes trivial, whereas if they are too challenging, the embedding may collapse into a singular point. Thus, generating the right triplets requires additional processing, such as selection based on image features.

Online Triplet Mining The online triplet mining strategy involves conducting data processing during the training phase within mini-batches of data. In this approach, the model learns to distinguish between similar and dissimilar examples by inputting new and unfamiliar data. A drawback of online triplet mining can be that this process is computationally expensive and time-consuming.

Triplet Categories Triplets can be selected in two different ways: either they are predetermined before training starts, or they are dynamically chosen during the training process. [2]

Hard Triplets (Hard negatives, Hard Mining) Hard triplets consist of an anchor, a positive anchor, and hard negatives. The hard negatives refer to the negative samples that are closer to the anchor than the positive samples. These particular samples are challenging for the model to distinguish and provide the model the most valuable information for training purposes. However, relying solely on hard triplets for training can lead to a highly

focused learning process, potentially neglecting the broader distribution of negative samples and resulting in a model that is overly specialized to specific instances. To address these challenges, various techniques such as semi-hard mining have been proposed to balance the training process and optimize the model's performance.

Semi-hard negatives (Semi-hard negatives, Semi-hard Mining) Semi-hard triplets have proven to be an effective method for training Siamese neural networks with triplet loss, with the primary goal of identifying sets of triplets that contribute to the continued progression of network training. Semi-hard negative samples are farther from the anchor than the positive sample but still have a positive loss. [5] These samples are easier for the model to distinguish than hard negatives but are still useful for training. However, identifying a sufficient number of valid semi-hard triplets can be challenging, and therefore semi-hard mining requires a large batch size to search for informative pairs. Typically, this mining process is conducted over the stochastic subset of samples utilized within each mini-batch.

$$d(f(x_i^a), f(x_i^p)) < d(f(x_i^a), f(x_i^n)) < d(f(x_i^a), f(x_i^p)) + m \tag{2.12}$$

where d is a distance function.

Easy negatives (Easy negatives, Easy Mining) Negative samples that are farthest from the anchor are referred to as easy negatives. However, these samples are excessively simplistic for the model to differentiate and consequently lack substantial value in terms of training information.

In Figure 2.5, Diagram (a) shows a triplet violating the condition of the triplet loss. Diagram (b) presents three possible types of triplets, determined by the distance between an anchor and the positive anchor.

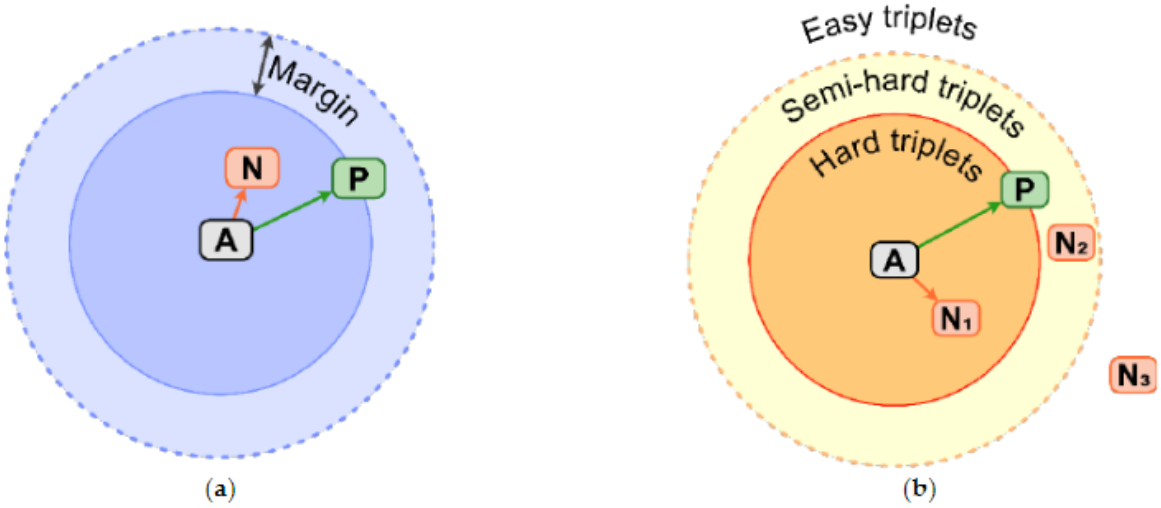


Figure 2.5: Triplet Loss Sampling [10]

2.3 Principal Components Analysis

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction and feature extraction. This method transforms a dataset consisting of possibly correlated variables into a set of uncorrelated variables called principal components.

The goal of PCA is to identify the directions in the data that contain the most information, known as the principal components. These components are obtained by calculating the eigenvectors and eigenvalues of the covariance matrix of the dataset. The eigenvectors represent the directions of maximum variance, while the eigenvalues indicate the amount of variance captured by each component. [8]

Let the random vector $\mathbf{X}^\intercal = [X_1, X_2, \cdots, X_p]$ have the covariance matrix Σ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

Consider the linear combinations:

$$Y_1 = \mathbf{a_1}^\intercal \mathbf{X}$$

$$Y_2 = \mathbf{a_2}^\intercal \mathbf{X}$$

$$\vdots$$

$$Y_p = \mathbf{a_p}^\intercal \mathbf{X}$$

Then we have,

$$Var(Y_i) = \mathbf{a_i}^\intercal \mathbf{\Sigma a_i}$$

$$Cov(Y_i, Y_k) = \mathbf{a_i}^\intercal \mathbf{\Sigma a_k}$$

Principal components are those uncorrelated linearcombinations $Y_1, Y_2, \cdots, Y_p$ whose variances are as large as possible. For example, the first principal component Y_1 is the linear combination with maximum variance. That is, $Y_1 = \mathbf{a_1}^\intercal \mathbf{X}$ that maximizes $Var(Y_1) = \mathbf{a_1}^\intercal \mathbf{\Sigma a_1}$ subject to $\mathbf{a_1}^\intercal \mathbf{a_1} = 1$

$$\max_{a_1 \neq 0} \frac{\mathbf{a_1}^\intercal \mathbf{\Sigma a_1}}{\mathbf{a_1}^\intercal \mathbf{a_1}} = \lambda_1$$

is attained when $a_1 = e_1$, where $e_1, e_2, \cdots, e_p$ are the associated normalized egenvectors of the eigenvalues of Σ, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

Thus, the ith principal component is given by

$$Y_i = \mathbf{e_i}^\intercal \mathbf{X}$$

and

$$Var(Y_i) = \lambda_i$$

$$Cov(Y_i, Y_k) = 0$$

In the context of image analysis using PCA, the random vector $\mathbf{X}^\intercal = [X_1, X_2, \cdots, X_p]$ represents the collection of pixel values at a specific pixel location within the image data.

CHAPTER 3

Data

3.1 Data Source

A total of 24,682 automotive parts images of 990 different categories (part names) were sourced from CarParts.com's cloud-based images, with each image labeled accordingly with its corresponding part name.

In the sample images below, the significant visual differences among the four images of the same category (Distributor Rotor) highlights the variability that can be observed within a single class. Despite sharing the same part name, the images demonstrate variations in appearance and shape. Additionally, these sample images inform that the effect of colors on image classification should be eliminated.

3.2 Data Cleaning

Categories (part names) with a single image were removed from the dataset to ensure that multiple samples are available for each category. The siamese network relies on learning the similarity or dissimilarity between pairs of samples (Contrastive Loss) or the difference of distance between anchor and positive samples and anchor and negative samples (Triplet Loss). If a category has only one image, there is no other image to form a pair or triplet with, making it impossible to train the network effectively on that category. By removing categories with only one image, the network can focus on learning meaningful representations

Distributor Rotor

and relationships between different categories that have sufficient sample pairs or triplets.

The dataset also excluded certain categories containing strings listed below. Reasons for exclusion include similar shapes from different categories and high dispersion within the same category.

- Seal: Axle Seal, Door Seal, Fuel Pump Seal, etc.
- Bearing: Center Bearing, Clutch Pilot Bearing, Clutch Release Bearing, etc.
- Shaft: Axle Shaft, Driveshaft, Steering Shaft, etc.
- Strap: Cargo Strap, Door Pull Strap, Fuel Tank Strap, etc.
- Hose: Air Intake Hose, Clutch Hose, Heater Hose, etc.
- Belt: Accessory Belt Tensioner, Drive Belt, Serpentine Belt, etc.
- Link: Center Link, Drag Link, Lateral Link, etc.
- Arm: Control Arm, Idler Arm, Pitman Arm, etc.
- Bushing: Control Arm Bushing, Leaf Spring Bushing, Sway Bar Bushing, etc.
- Kit: Automatic Transmission Shift Kit, Brake Caliper Repair Kit, Brake Drum and Shoe Kit, etc.
- Set: Brake Pad Set, Extractor Set, Piston Ring Set, etc.

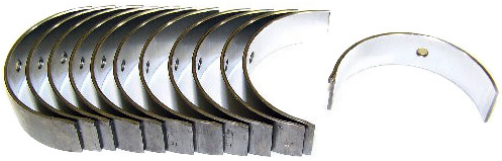
After data cleaning, 15,165 images from 352 categories were left in the dataset to be used to train, validate and test the models.

The following examples were deleted from the dataset:

Rear Window Seal

Door seal

Main Bearing



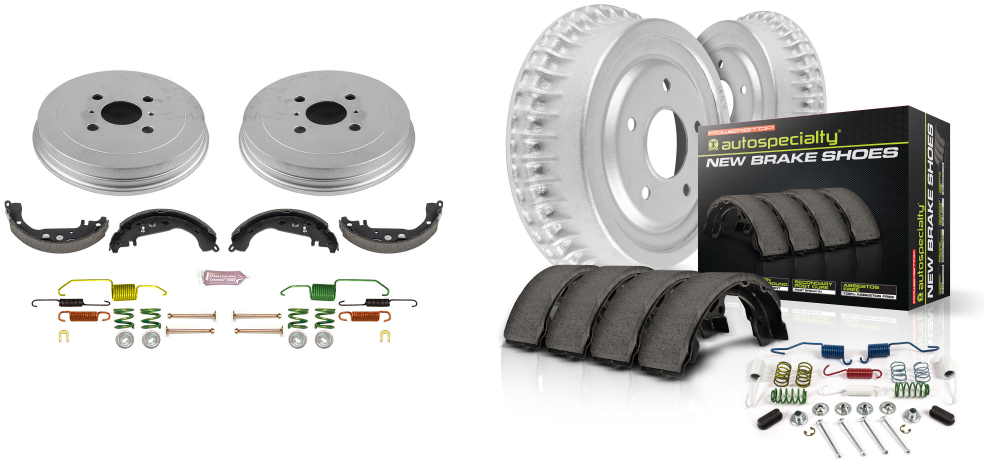
Rod Bearing Set

Driveshaft

Steering Shaft

Heater Hose

Power Steering Hose



Brake Drum and Shoe Kit

3.3 Data Preparation

Grayscale is an image representation where each pixel is represented by a single intensity value ranging from 0 to 255, instead of dealing with three color channels (red, green, and blue). The color images were converted to grayscale images within the automotive image dataset to eliminate the potential impact of color on image classification and recognition for each category.

White padding is a technique used in image data preparation where additional white pixels are added around the borders of an image. This is done to increase the overall size of the image while maintaining its original aspect ratio. The images in the dataset were white padded to ensure that they do not lose their original aspect ratios when they are resized as the same size.

As CNNs expect images of consistent dimensions as inputs, the images in the dataset were resized as (100, 100).

To train Siamese neural network models using contrastive loss, it is necessary to create input pairs that represent both similarity and dissimilarity. Here is the process for creating these pairs:

- Similar image pairs: Randomly select two images from the same category for each category. Label these pairs as positive pairs with a value of 0, indicating their similarity.
- Dissimilar image pairs: From each positive pair, randomly choose one of the images. Then, randomly select a single image from a different category and pair it with the chosen image. Label these pairs as negative pairs with a value of 1, indicating their dissimilarity.
- To maintain balance in the pairs, approximately 50% of the pairs should be positive pairs, while the remaining pairs are negative pairs.

To train Siamese neural network models using triplet loss, three types of input sequences are required: anchor, positive anchor, and negative anchor. The anchor represents a randomly chosen image. The positive anchor is chosen based on its relevance to the anchor, meaning it belongs to the same category as the anchor images. Conversely, the negative anchor is selected to be dissimilar or unrelated to the anchor, coming from different categories than the anchor images.

3.4 Exploratory Data Analysis

Table 3.1 shows the image count for each category. Among all the categories, "Bumper Cover," "Mirror," "Headlight," "Brake Disc," and "Fender Liner" are the top five categories with the highest number of images. These five categories collectively comprise 3,931 images out of a total of 15,165 images. Additionally, each category has a minimum of two images.

Part Name	**Image Count**
Bumper Cover	952
Mirror	920
Headlight	761
Brake Disc	741
Fender Liner	557
...	...
Flashlight	2
Tire Pressure Gauge	2
Trailer Brake Control	2
Transfer Case Chain	2
Headlight Guard	2

Table 3.1: Image count for each category

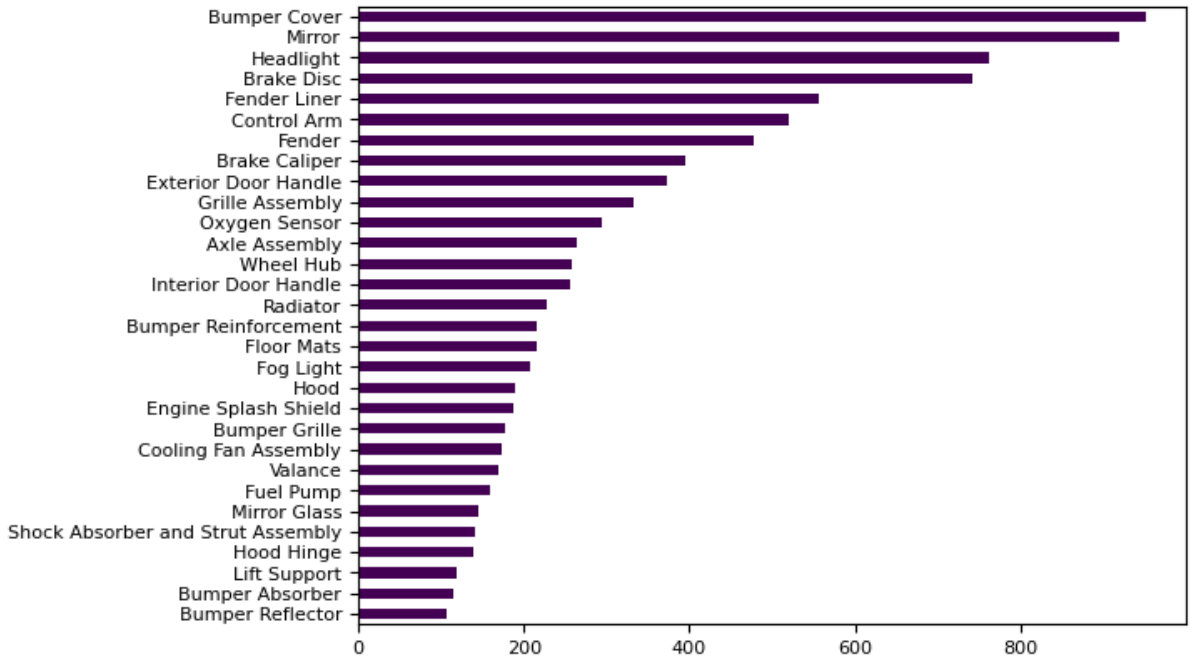


Figure 3.5: The number of images in the top 30 categories

Principal Components Analysis (PCA) was conducted on the 6,029 images from the top 30 categories to identify the most distinguishing features. This analysis involved analyzing the importance of the principal components. Figure 3.6 showcases the first 30 principal components with the largest eigenvalues out of a total of 6,029 principal components. The total number of principal components is determined by taking the minimum value between the number of features (100x100) and the number of images.

The first five factors provide the most comprehensive explanation for the variability in the image data. These factors encompass the presence of shapes with darker backgrounds and lower image values (e.g., Mirror), trapezoid shapes that occupy a significant portion of the image width (e.g., Hood), narrower concave shapes with darker backgrounds and lower image values, a shape with bright areas on the left side and dark areas on the right side, and a bright ring-shaped pattern. In contrast to facial image analysis, which reveals distinct features like eyes, nose, mouth, and facial contours, distinguishing features in autopart images tend to be

more ambiguous and less well-defined.

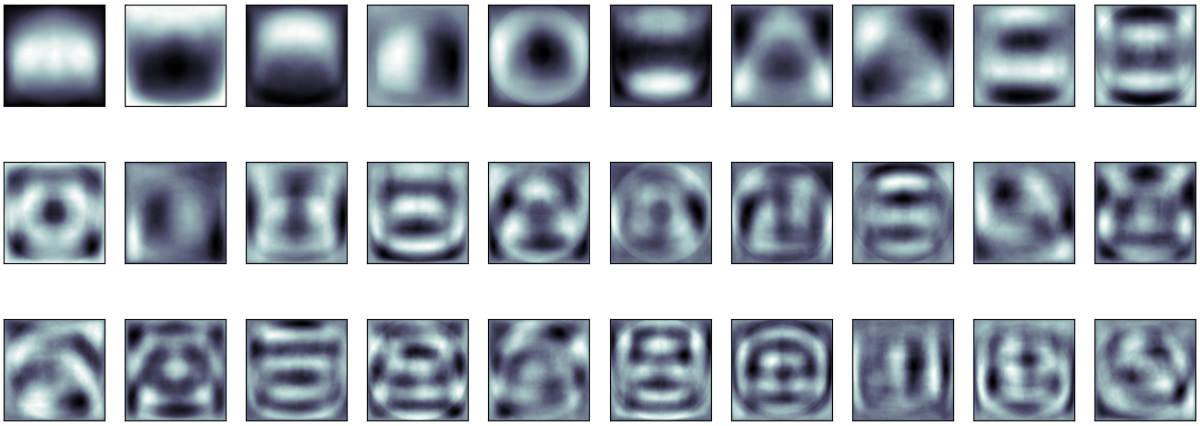


Figure 3.6: Principal Components Analysis for the automotive parts images

CHAPTER 4

Models

The image dataset was split into training, validation, and testing datasets in an 8:1:1 ratio. The basic configurations for each model were the same as follow:

- Train batch size = 64
- Train number epochs = 100
- Optimizer = Adaptive Moment Estimation (Adam)
- Learning rate = 0.0005

4.1 Convolution Neural Network

A simple CNN model was designed with two sets of convolutional layers (Conv layers) that operate on 2D input data. Each convolutional layer utilizes a 3x3 kernel and applies the Rectified Linear Unit (ReLU) activation function. Following the Conv layers, max-pooling layers were stacked for downsampling. These layers are then connected to a fully-connected layer responsible for predicting the category. The model outputs category probabilities by applying a log softmax function. Cross Entropy was used as a loss function to compare the predicted output probabilities to the true target.

Figure 4.1 shows that as the model is trained, validation loss keeps increasing while the training loss decreases. This indicates that CNN has a limitation in image classification with a large number of categories, because of class imbalance and difficulty in learning distinct

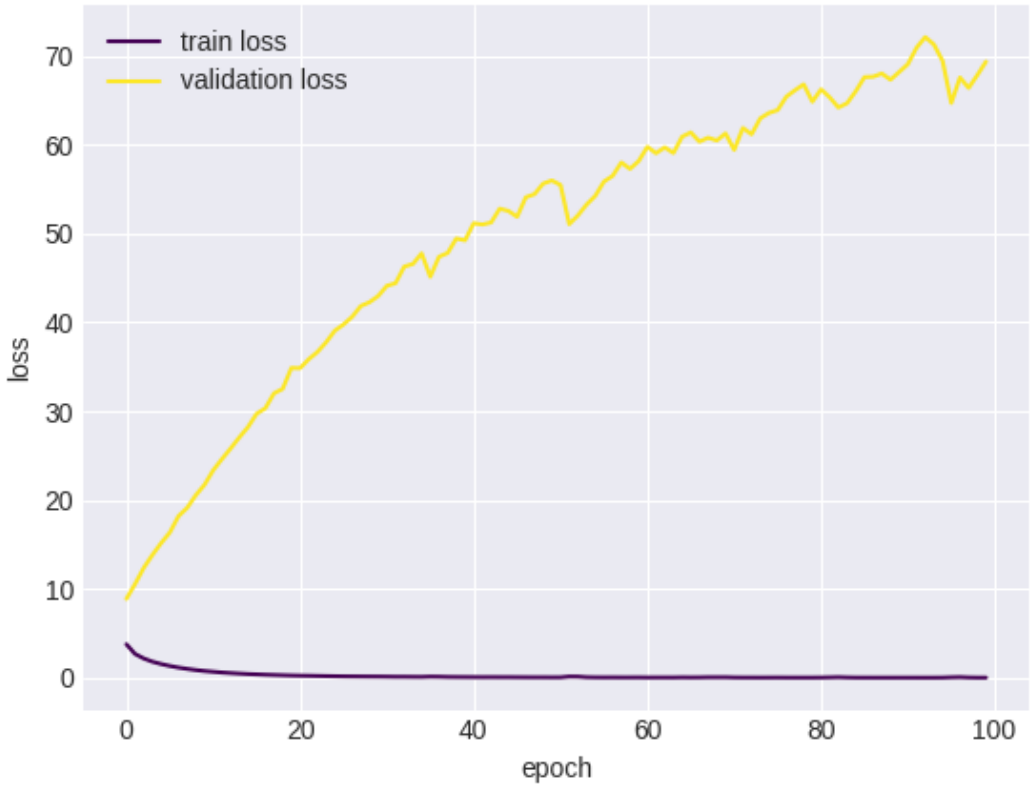


Figure 4.1: Training loss and validation loss of simple CNN model

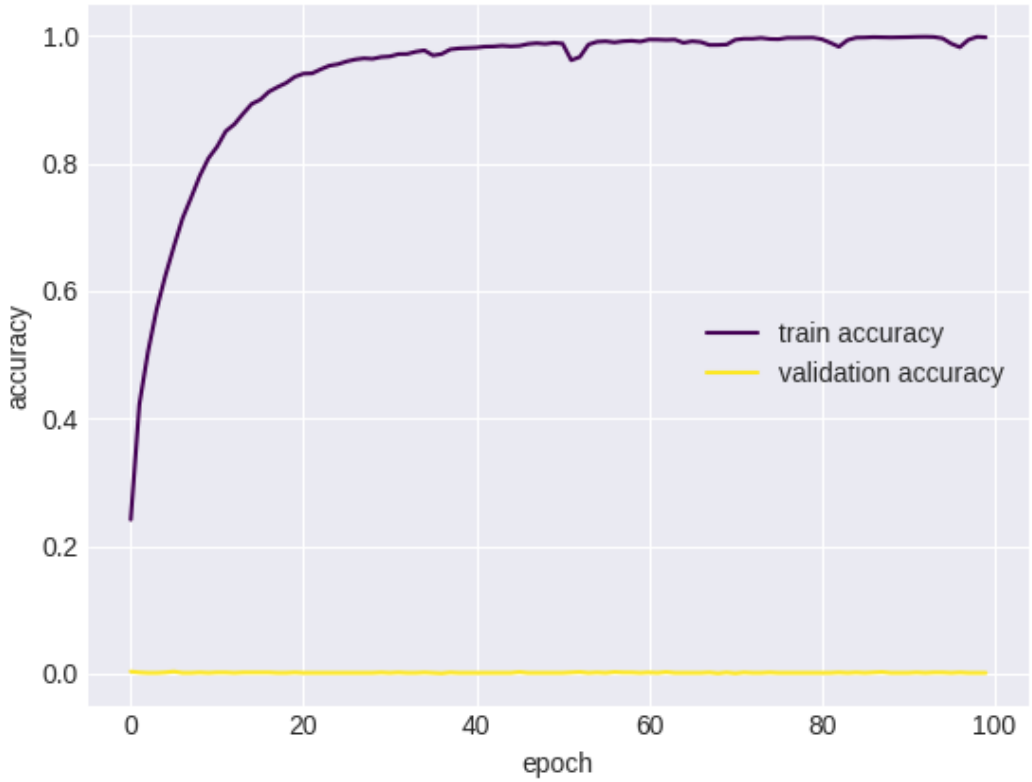


Figure 4.2: Training loss and validation accuracy of simple CNN model

features for each category. CNN struggles to effectively represent and differentiate between all classes, especially with limited training examples, leading to biased performance favoring majority classes. Furthermore, the limited size of the training dataset has led to overfitting of the model, resulting in its excessive specialization to the training data and inadequate generalization to unseen instances.

Test Accuracy of simple CNN : 50.80%

4.2 Siamese Neural Network with Contrastive Loss

To overcome the limitation of CNNs in image classification with a high number of categories, a solution was adopted using a Siamese neural network with Contrastive loss. This approach involved utilizing pairs of images and constructing the Siamese neural network by employing two identical models with the same CNN layers as the previous model. Input image pairs were randomly chosen from the dataset during each mini-batch (online mining) and the positive and negative pairs were approximately balanced, with a ratio of approximately 50% each.

The accuracy in a Siamese neural network was measured by evaluating how well the network can discriminate between similar and dissimilar pairs of inputs. The accuracy metric in this context indicates the percentage of correctly classified pairs.

To evaluate the model's performance, it was necessary to establish a threshold that determines whether images should be classified as similar or dissimilar, based on the distances observed in the outputs. Figure 4.3 represents the histogram showing the distribution of dissimilarities between the outputs of 9,254 pairs, which were randomly chosen from the training dataset. According to the histogram, the majority of values are concentrated within the range of 0 to 3.

Figure 4.4 represents the model metrics (accuracy, precision, recall, f1) based on the classification threshold. According to this graph, the training accuracy reaches its maximum

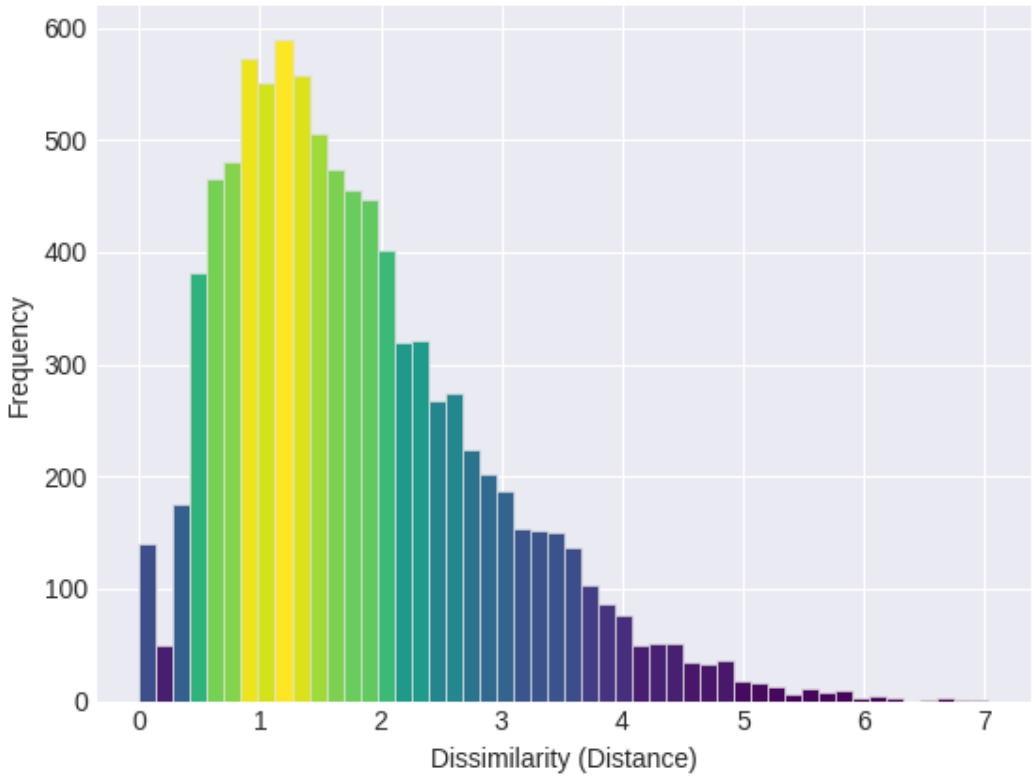


Figure 4.3: Histogram of dissimilarity (distance)

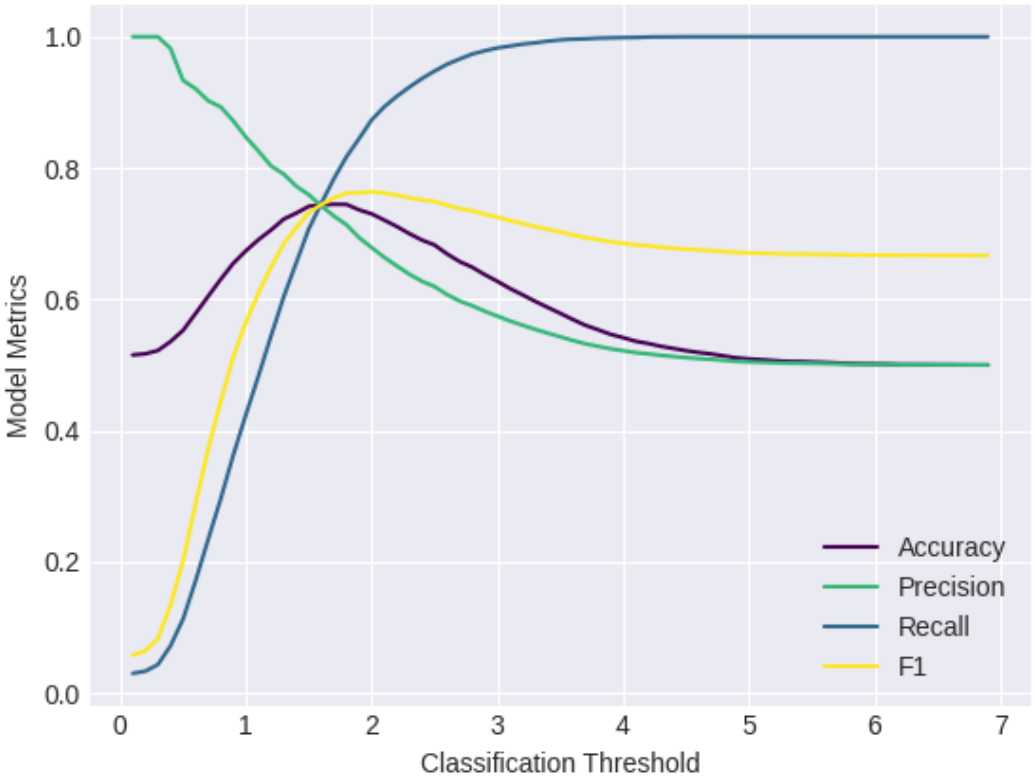


Figure 4.4: Model metrics by thresholds of dissimilarity (distance)

value of 74.47% at a threshold of 1.7. As the threshold increases beyond 1.7, the recall (sensitivity) increases but the precision (specificity) decreases. This indicates that the model correctly identifies positive instances from the actual positive instances in the data, while minimizing false negatives. Conversely, reducing the threshold below 1.7 increases precision but decreases recall, making the model correctly classify positive instances among all instances predicted as positive. [14] A higher precision represents that the model is better at minimizing false positives. In this dataset, as the numbers of positive and negative pairs are balanced, the classification threshold is set as 1.7 which generates the highest value of model accuracy.

Figure 4.5: Training loss and validation loss of Siamese model with contrastive loss

Figure 4.5 indicates that the model's performance on new data is not as robust as expected as the validation loss goes up throughout the training process, although Figure 4.6 illustrates

Figure 4.6: Training loss and validation accuracy of Siamese model with contrastive loss

a simultaneous increase in training accuracy and validation accuracy. The model shows 70.24% of accuracy.

Test Accuracy of Siamese model with contrastive loss : 70.24%

4.3 Siamese Neural Network with Triplet Loss

To apply the concept of the relative distances between anchor, positive, and negative samples, Siamese neural network was trained with Triplet loss function. Triplets were randomly selected from the dataset during each mini-batch (online mining). However, due to the limit of time and computational cost, no control was conducted on the distances between the anchors and positive anchors, or negative anchors.

The accuracy in a Siamese neural network with a triplet loss function measured the model's ability to correctly rank the similarity of input triplets by ensuring that the distance between the anchor and positive sample is smaller than the distance between the anchor and negative sample by the margin.

Figure 4.7 shows that the validation goes down during the training process. However, the test accuracy remains at a low level.

Test Accuracy of Siamese model with triplet loss : 40.36%

The poor predicting performance of Siamese neural network with Triplet loss can be attributed to the inadequate sampling method. The lack of control over the distances between anchors, positive anchors, or negative anchors when selecting triplets hindered the model's ability to learn effectively.

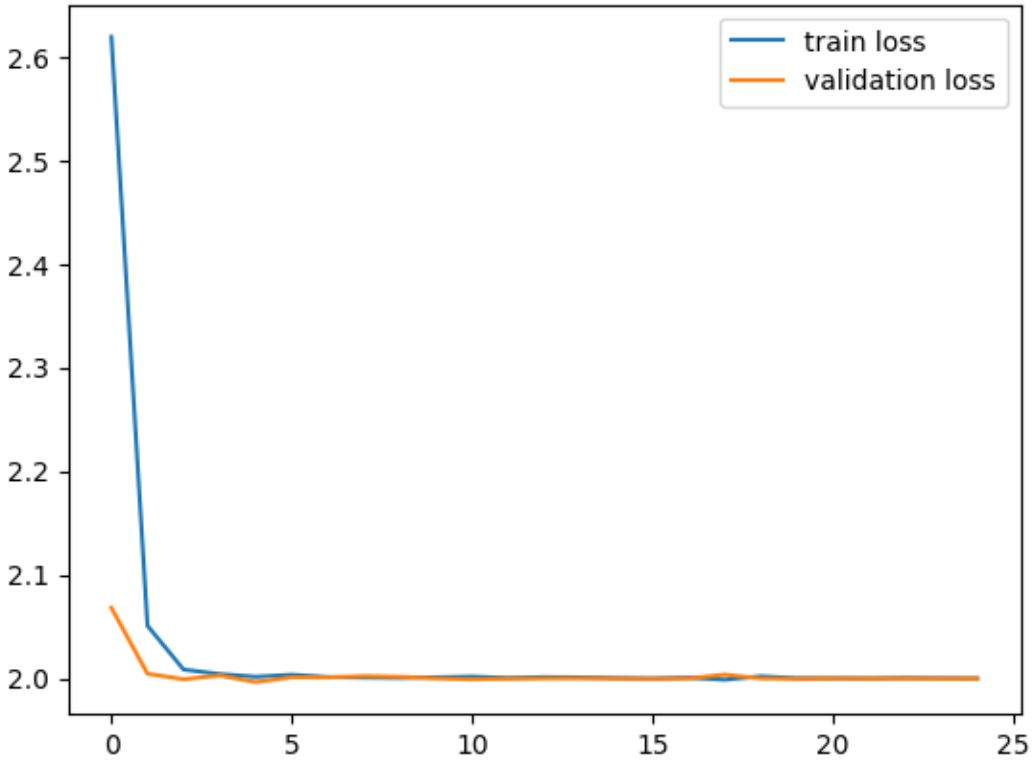


Figure 4.7: Training loss and validation loss of Siamese model with triplet loss

CHAPTER 5

Conclusion

Model metrics	Simple CNN	Siamese model with contrastive loss	Siamese model with triplet loss
Training accuracy	99.79%	97.94%	-
Validation accuracy	0.0007%	91.98%	-
Test accuracy	50.80%	70.24%	40.36%

Table 5.1: Model Comparison

The CNN model trained for image classification with a large number of categories faces limitations due to class imbalance and the difficulty of learning distinct features for each category. The validation loss increases while the training loss decreases, indicating biased performance favoring majority classes. Limited training examples and overfitting further hinder the model's generalization, resulting in a test accuracy of 50.80%. To address these limitations, a Siamese neural network with Contrastive loss is adopted. By utilizing pairs of images and employing two identical models, this approach achieves better generalization to unseen data with a test accuracy of 70.24%. However, when using triplet loss in the Siamese neural network, the test accuracy remains low at 40.36%. The poor performance of the triplet loss model can be attributed to inadequate sampling methods and the lack of control over anchor distances, positive anchors, and negative anchors during triplet selection.

As the dataset contains 15,165 images after data cleaning, increasing the sample size in dataset can lead to better model performance by enhancing model complexity, improving

generalization, and reducing overfitting.

In addition to leveraging a larger sample size, there are several other future discussions and strategies that can strengthening model training.

CHAPTER 6

Further Discussion

6.1 Semi-hard triplets

The absence of control over the distances between anchors, positive anchors, and negative anchors during triplet selection has resulted in subpar predictive performance for siamese neural networks with triplet loss. Employing the semi-hard triplet constraint can significantly enhance performance compared to the simple selection method.

6.2 Different loss functions

In addition to the contrastive loss function and triplet loss function mentioned in the paper, various other loss functions have been proposed to improve the performance of deep learning models in image analysis tasks. One of these alternative loss functions is quadruplet loss, [16] consisting of four samples: an anchor sample, a positive sample (belonging to the same class as the anchor), a negative sample (belonging to a different class than the anchor), and a negative sample that is further away from the anchor than the previous negative sample. This method is designed to minimize the distance between the anchor and the positive sample, and maximize the distance between the anchor and both negative samples. Applying these alternative loss functions to Siamese model can enhance its learning performance.

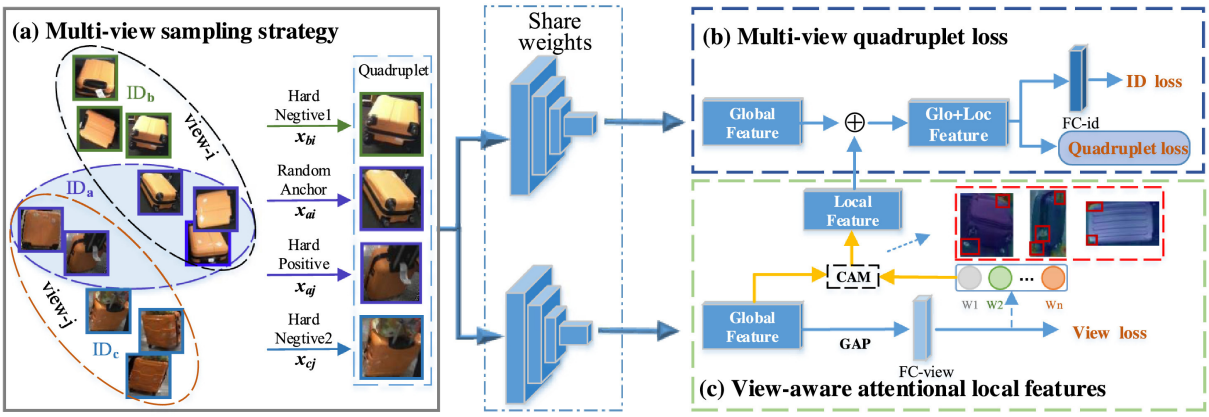


Figure 6.1: QuadNet: Deep learning model with quadruplet loss function [16]

6.3 Other distance measures

In this paper, euclidean distance was used to measure the distance between outputs. However, Figure 5.1 shows numerous alternative options exist, such as cosine similarity, hamming distance, and many more. Incorporating these alternative measures and conducting comparisons can contribute to performance improvements.

6.4 Image augmentation

As certain classes have significantly fewer samples than others, class imbalances exist within our dataset. Augmentation serves as a valuable solution by generating synthetic samples specifically for the underrepresented classes. This approach helps to balance the distribution of training data, mitigating bias towards dominant classes. Furthermore, augmentation exposes the model to diverse angles of samples, promoting robustness and invariance by introducing variations in the training data.

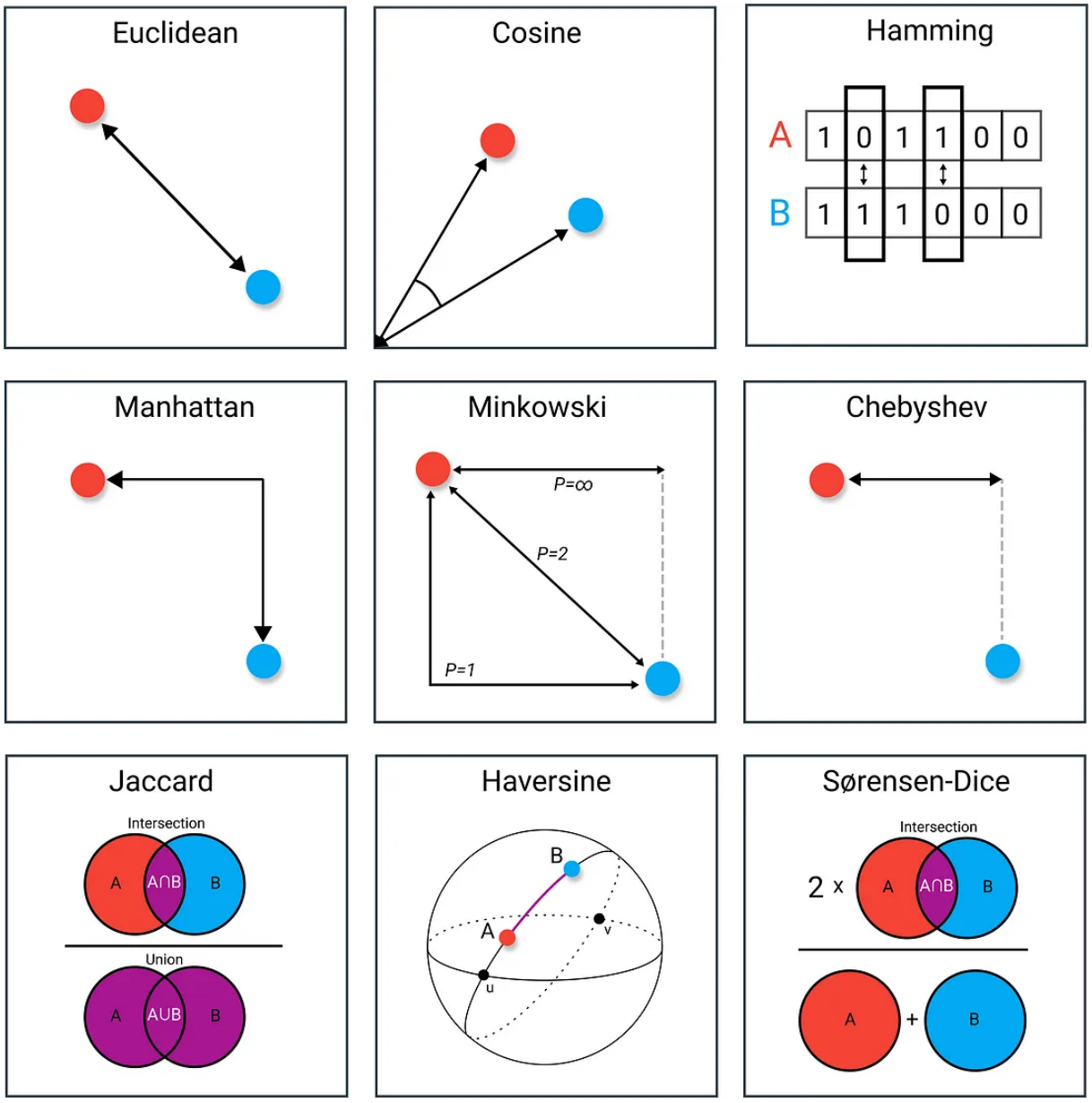


Figure 6.2: Different types of distance measures [1]

Figure 6.3: Examples of images in different angles from the same part name (Similarity result from Siamese neural network with contrastive loss)

6.5 Dimensionality reduction of input dataset

Principal Component Analysis (PCA) can be used in image recognition with deep learning models to reduce dimensionality. As the first 500 largest Principal Components explain 96.41% of the total 10,000 (100x100) dimension, replacing the input data with dimensionality-reduced representations can alleviate computational requirements and potentially enhance the efficiency and performance of the model.

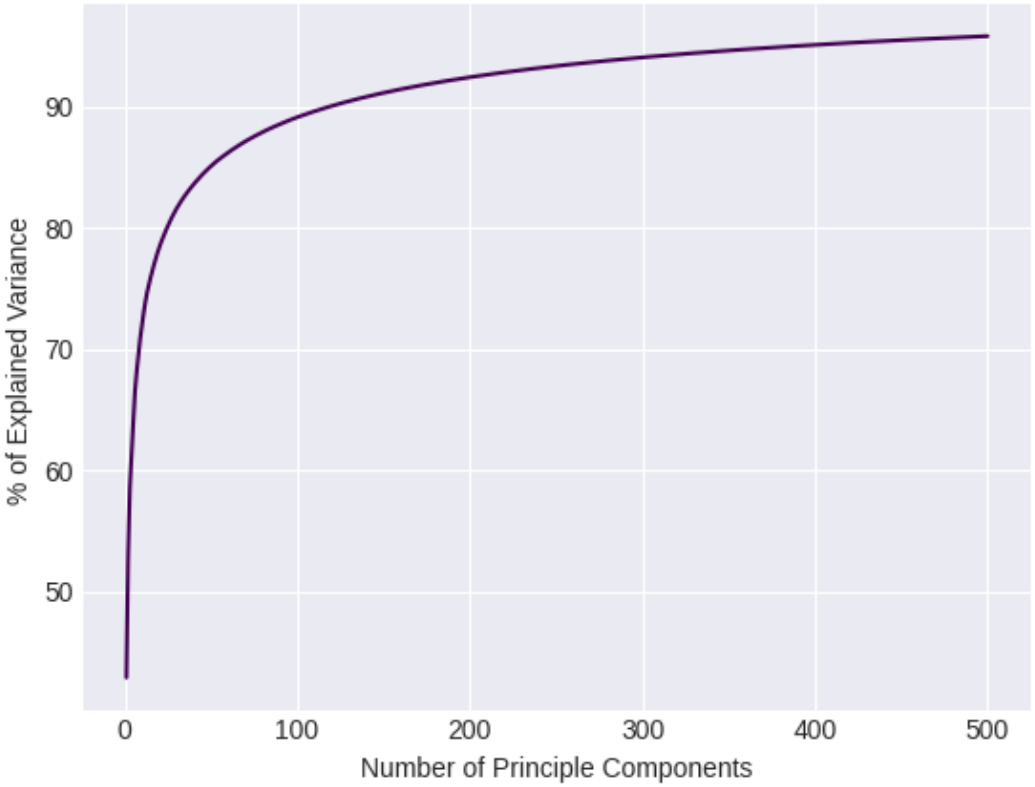


Figure 6.4: Proportion of variance explained by principal components

REFERENCES

[1] 9 distance measures in data science. https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa.

[2] Taghreed Abdullah, Yakoub Bazi, Mohamad Al Rahhal, Mohamed Mekhalfi, Lalitha Rangarajan, and Mansour Zuair. Textrs: Deep bidirectional triplet network for matching text to remote sensing images. *Remote Sensing*, 12:405, 01 2020.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[4] Davide Chicco. *Siamese Neural Networks: An Overview*, pages 73–94. Springer US, New York, NY, 2021.

[5] James Philbin Florian Schroff, Dmitry Kalenichenko. Facenet: A unified embedding for face recognition and clustering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 06 2015.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[7] Zhao Guyu, Guoyan Huang, Hongdou He, Haitao He, and Jiadong Ren. Regional spatiotemporal collaborative prediction model for air quality. *IEEE Access*, PP:1–1, 09 2019.

[8] Richard Arnold Johnson, Dean W Wichern, et al. Applied multivariate statistical analysis. 2002.

[9] Salik Ram Khanal, Eurico Vasco Amorim, and Vitor Filipe. Classification of car parts using deep neural network. In José Alexandre Gonçalves, Manuel Braz-César, and João Paulo Coelho, editors, *CONTROLO 2020*, pages 582–591, Cham, 2021. Springer International Publishing.

[10] Mariusz Kurowski, Andrzej Sroczynski, Georgis Bogdanis, and Andrzej Czyżewski. An automated method for biometric handwritten signature authentication employing neural networks. *Electronics*, 10:456, 02 2021.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[12] Wang Liqun, Wu Jiansheng, and Wu Dingjin. Research on vehicle parts defect detection based on deep learning. *Journal of Physics: Conference Series*, 1437(1):012004, jan 2020.

[13] Kitsuchart Pasupa, Phongsathorn Kittiworapanya, Napasin Hongngern, and Kuntpong Woraratpanya. Evaluation of deep learning algorithms for semantic segmentation of car parts. *Complex & Intelligent Systems*, 8(5):3613–3625, Oct 2022.

[14] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, page 27, 10 2020.

[15] Milad Sikaroudi, Benyamin Ghojogh, Amir Safarpoor, Fakhri Karray, Mark Crowley, and Hamid R. Tizhoosh. Offline versus online triplet mining based on extreme distances of histopathology patches. *CoRR*, abs/2007.02200, 2020.

[16] Hao Yang, Xiuxiu Chu, Li Zhang, Yunda Sun, Dong Li, and Stephen J. Maybank. Quadnet: Quadruplet loss for multi-view learning in baggage re-identification. *Pattern Recognition*, 126:108546, 2022.

[17] Jian Yu, Chang-Hui Hu, Xiao-Yuan Jing, and Yu-Jian Feng. Deep metric learning with dynamic margin hard sampling loss for face verification. *Signal, Image and Video Processing*, 14(4):791–798, Jun 2020.