



Siamese Neural Network Optimization Using Distance Metrics for Trademark Image Similarity Detection

Suyahman Suyahman¹, Sunardi Sunardi^{2*}, Murinto Murinto³ and Arfiani Nur Khusna⁴

¹Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

²Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

^{3,4}Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Received 26 June 2024, Revised 7 February 2025, Accepted 8 February 2025

Abstract: Trademark image similarity detection plays a crucial role in protecting intellectual property. Traditional methods, particularly those relying on Euclidean distance, often fail to capture subtle visual differences, leading to less accurate results. This study addresses this issue by optimizing a Siamese Neural Network (SNN) with improved distance metrics. Specifically, Chi-Squared and Manhattan distance methods are explored alongside the standard Euclidean approach to enhance trademark similarity detection. The objective is to develop a more precise and reliable system for trademark analysis, essential for effective intellectual property enforcement. The research utilizes a dataset of 255 trademark images across five classes, each with variations in color, texture, and design. To train and evaluate the model, 2000 triplet samples—comprising an anchor image, a similar (positive) image, and a dissimilar (negative) image—were generated, with 1600 pairs used for training and 400 for validation. The SNN model was built using the Xception CNN architecture and trained with a triplet loss function to distinguish between similar and dissimilar images. Performance was assessed using accuracy, precision, recall, and F1-score. Results demonstrated that the Chi-Squared distance metric outperformed the others, achieving an accuracy of 0.96, compared to 0.92 for Euclidean and 0.74 for Manhattan. The Chi-Squared metric proved particularly effective in capturing differences in color and texture, improving accuracy by 0.0435 over Euclidean. These findings highlight the significance of selecting appropriate distance metrics for image similarity tasks, as they directly impact performance. This study advances traditional trademark similarity detection by integrating optimized distance measures, making automated trademark protection more reliable. Future research may explore hybrid metrics or novel approaches to further improve accuracy across diverse trademark datasets, strengthening legal and business efforts in safeguarding intellectual property.

Keywords: Trademark Similarity, Siamese Neural Network, CNN, Deep Learning, Manhattan, Chi-Squared

1. INTRODUCTION

The protection of intellectual property, particularly trademarks, has become increasingly significant [1]. Trademarks serve as a vital component for companies, ensuring brand recognition and differentiation in a highly competitive market [2]. The advantages of owning a trademark are numerous. A registered trademark grants the owner exclusive rights to use the mark, which can inhibit others from using it without authorization [3]. This exclusivity helps build brand loyalty and trust among consumers, leading to increased business revenue [4], [5]. In addition, trademarks can be valuable assets that appreciate over time as the brand grows. They also offer legal protection against infringement, enabling companies to pursue legal recourse against unauthorized exploitation.

To be eligible for registration, a trademark must meet

certain criteria. It must be unique enough to differentiate between the goods or services of different enterprises ([6]). It should not be misleading, scandalous, or contrary to public order and morality [7]. Additionally, it must not be generic or merely descriptive without acquiring distinctiveness through use. Furthermore, the trademark should not conflict with existing registered trademarks. As the volume of digital content grows, so does the need for precise and effective methods to identify and compare trademark images, facilitating the enforcement of trademark rights and preventing infringement.

Traditional methods for comparing trademark images often rely on manual inspection, which is time-consuming and prone to human error. To address these issues, several studies have looked into using Siamese Neural Networks for finding images based on their content and for detecting

similarities between images.

For example, the study by Kumar and Madhavi used a stacked Siamese neural network with a Euclidean distance metric on the CIFAR-10 dataset, which showed superiority over conventional CBIR methods with significant improvements in accuracy. However, this study only explored one type of distance metric without comparing other alternatives, [8]. Zhang and others created a Convolutional Siamese Neural Network to tell the difference between CT scan images of lung cancer and tuberculosis. They used data from 719 patients and reached an accuracy of 94.7%, a mean average precision (mAP) of 95.3%, and an area under the curve score of 97.0% on their own dataset. Despite the superior results, this study focused on medical data and was limited in exploring distance metrics in non-medical domains [9].

Devi et al. proposed a batch-normalized Siamese network on the Fashion MNIST dataset with 30,000 images, achieving 0.9191 accuracy, 0.9293 precision, and 0.9072 recall. The advantage of this study is the use of batch normalization that improves training stability, but the disadvantage is the lack of comparison between distance metrics. Meanwhile, Jalilian and Mateu used SNN with a dissimilarity function to analyze spatial point patterns in 130 species, showing superiority over intensity-based and K-function methods. However, this approach was not applied in the context of image matching, so its usefulness in trademark detection remains untested. [10]. Meanwhile, Jalilian and Mateu used SNN with a dissimilarity function to assess the similarity of spatial point patterns in a dataset of 130 species. The results showed that this method was more accurate than intensity-based and K-function techniques, but this study was not applied to image matching, so its relevance to the task of trademark similarity detection is still limited [11].

In the specific task of trademark image similarity detection, previous studies by Suyahman et al. evaluated various CNN architectures in a Siamese Network with Test-Time Augmentation on a trademark dataset. The results showed that VGG19 performed the best with 0.9882 accuracy, while ResNet50 only achieved 0.5000 accuracy, indicating a significant difference in the effectiveness of the architectures. The advantage of this study is the use of various CNN architectures to evaluate the performance of SNN, but the disadvantage is the absence of a broader exploration of distance metrics to improve accuracy and efficiency. [12]. However, this study has not explored various distance metrics in improving the accuracy and efficiency of trademark image similarity detection.

Based on previous research, there is a gap that has not been explored, namely the optimal comparison of various distance metrics in the Siamese Neural Network architecture for the task of detecting trademark image similarity. Most previous studies have used only one or two types of

distance metrics without further analysis of their impact on model accuracy and efficiency. Therefore, this study contributes by exploring and optimizing various distance metrics, including Chi-Squared distance, which is sensitive to variations in the distribution of color and texture features, and Manhattan distance, which is more robust to outliers in high-dimensional space. With these optimizations, this study aims to improve the accuracy and efficiency of trademark image similarity detection, which ultimately supports the trademark protection process more effectively.

2. METHODS

The method for detecting trademark similarity utilizing a Siamese neural network is depicted in Fig. 1.

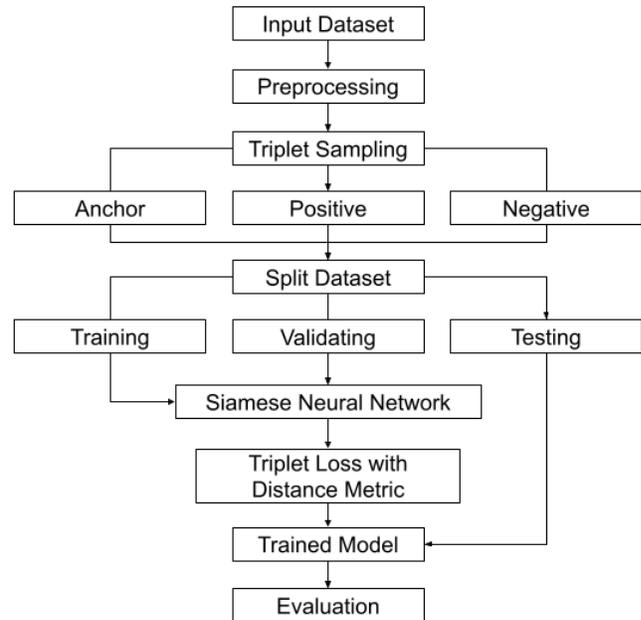


Figure 1. Research methods

The procedure initiates with the gathering and preparation of the initial dataset. The raw data is subject to preprocessing steps to maintain quality and uniformity, including steps like normalization and augmentation. Triplet samples, which consist of an anchor, a positive, and a negative image, are then selected from this preprocessed data. Subsequently, the dataset is divided into training, validation, and testing subsets. Evaluation is done using a confusion matrix to calculate accuracy, precision, recall, and F-1 Score. The research process is carried out using keras in Jupyter Notebook.

In the training phase, the Siamese neural network undergoes training using the training subset, where it is refined to optimize a similarity metric through the use of triplet loss. The validation subset serves to monitor the model's performance during training and to adjust hyperparameters as necessary. Following the training phase, the network is evaluated using the testing subset to gauge its overall

performance. In this study, the network employs triplet loss alongside a specific distance metric, such as the chi-square distance, to process the data.

Ultimately, the efficacy and accuracy of the model in recognizing visual similarities between trademarks are assessed. The aim of this approach is to forge an efficient model capable of detecting trademark similarities utilizing the methodology based on the Siamese neural network.

A. Dataset

The initial stage of this research involved collecting a dataset of 255 trademark images taken from Google Images and registered in the Indonesian Intellectual Property Database [13]. These images were carefully selected to represent a wide range of trademark variations, divided into five different classes, plus an additional set of 55 images for testing. The selection criteria for these trademarks were based on variations in design, color, and shape, as well as real-world relevance in trademark registration and infringement cases.

Specifically, the selected trademarks exhibit significant variation within their classes, with some brands displaying visual similarities despite being registered as different entities. These similarities often arise from shared design elements, color schemes, or usage across different media, making them challenging for traditional similarity detection methods. Details of the trademark classes and the amount of data used can be seen in Table I.

TABLE I. Dataset overview

Tardemark	Training & Validation	Testing
Miniso	40	11
Uniqlo	40	11
Grab	40	11
Gojek	40	11
Circle CI	40	11
Total	200	55

Within each class, the dataset includes one anchor image, 20 positive images that are visually similar to the anchor, and 20 negative images that are significantly different. Positive images were selected to reflect subtle variations in color, texture, or design, while negative images were selected to represent trademarks that are strikingly different.

This structure is designed to simulate real-world scenarios where trademarks may appear similar but are legally distinct, or where subtle visual differences are critical to distinguishing between brands. For convenient access and integration with Google Colab, all images are stored in Google Drive, with each image stored in a folder defined by its respective class. This organization facilitates efficient data handling and processing during the training and evaluation phases of this study. By combining trademarks with high variability and visual similarity, this dataset provides a

solid foundation for evaluating the effectiveness of various distance metrics in capturing subtle differences, which are critical for accurate trademark similarity detection. Fig. 2. illustrates samples of the trademark images used in the study.



Figure 2. Trademark image

B. Data Preprocessing

Pre-processing measures were implemented to purify and prepare the data for subsequent examination. This included converting all images to the PNG format to ensure lossless compression and consistent quality across the dataset. The images, which were in RGB color format, were then resized to a uniform dimension of 128x128 pixels to standardize the input size for the neural network. This resizing step guarantees uniformity in image dimensions, simplifying processing efforts and ensuring compatibility with the model's architecture. Additionally, the pixel values of the images were normalized to a scale of [0, 1] by dividing each pixel value by 255, which aligns with the neural network's input requirements. Additionally, the normalization step adjusts the pixel values to a scale that is most conducive to neural network functionality, thereby improving the model's efficiency and expediting its convergence rate [14].

C. Triplet Sampling

During the Triplet Sampling step, the data was arranged into groups of three images known as triplets [15]. The dataset initially consisted of 5 distinct trademark classes, with each class containing 40 variations of the same trademark. These variations included differences in color, texture, shape, and design, reflecting real-world scenarios where trademarks may appear similar but are legally distinct.

To create the triplets, the images were randomly sampled to ensure diversity and robustness in the training process. Each triplet consisted of an anchor image, a positive image, and a negative image. The anchor image was a randomly selected image from one of the 5 trademark classes. The positive image was another image from the same class as the anchor, representing a variation of the same trademark. The negative image was an image from a different class, representing a distinctly different trademark. The formation process involved constructing all possible combinations of these positive and negative images, resulting in a total of 400 unique triplet pairs for each trademark category. The results of triplet sampling can be seen in Table II.

With 5 trademark classes, this amounted to a total of 2000 triplet pairs for the entire dataset, which were used for training and validation. This random sampling approach ensured that the model was exposed to a wide



TABLE II. Triplet sampling overview

Taremark	Training & Validation	Testing
Miniso	400	110
Uniqlo	400	110
Grab	400	110
Gojek	400	110
Circle CI	400	110
Total	2000	550

range of visual similarities and differences, enhancing its ability to generalize across various trademark designs. By systematically organizing the data into triplets, this methodology enriched the training dataset, thereby bolstering the neural network's ability to accurately identify similarities and differences among trademarks [16].

D. Data Splitting

The dataset consists of 2000 triplet pairs, which were strategically split into training and validation subsets with a proportion of 80:20, respectively. This division resulted in 1600 triplet pairs allocated for training and 400 triplet pairs reserved for validation. This distribution is crucial for the model's development, as the training subset enables the model to learn and adapt to data patterns effectively. Meanwhile, the validation subset plays a critical role in controlling overfitting by providing a separate data pool to monitor and evaluate the model's performance continuously during training. This careful partitioning ensures comprehensive training while rigorously assessing the model's ability to generalize to new data. For testing, a separate set of 55 images was used, distinct from the 400 triplet pairs allocated for training and validation. These testing images were not included in the triplet formation process and were reserved exclusively for evaluating the model's performance after training. Adjustments based on validation results enhance the accuracy and general reliability of the model, leading to a robust and broadly applicable machine learning model [17].

E. Model Training

In the training phase, a Siamese Neural Network (SNN) employs a triplet loss function that leverages distance metrics such as Chi-Squared, Manhattan, and Euclidean. A Siamese Neural Network is a specialized neural network architecture designed to learn similarity metrics between pairs or groups of inputs [18]. Unlike traditional neural networks that process individual inputs for classification or regression tasks, SNNs are structured to process two or more inputs simultaneously and compare their feature representations [19]. These subnetworks process the input pairs independently but share the same set of parameters and weights. This weight-sharing mechanism ensures that the feature representations of the inputs are extracted using the same transformation, enabling a consistent and fair comparison. The outputs of these subnetworks are then mapped into a shared embedding space, where the similarity or dissimilarity between the inputs is computed using distance

metrics [20]. The function is designed to train the model to reduce the distance between the anchor and its corresponding positive image and increase the distance between the anchor and the negative image. This mechanism ensures effective learning by differentiating between similar and dissimilar items [21].

Euclidean Distance is one of the most commonly used distance metrics. The Euclidean distance between two feature vectors X and Y is defined as the straight-line distance (hypotenuse) connecting two points in n -dimensional space. This metric measures the direct distance between two points in Euclidean space [22]. The smaller the Euclidean distance, the more similar the two vectors are [23]. Euclidean distance calculation uses the following equation:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (1)$$

where X_i and Y_i are the i -th components of vectors X and Y , and n is the dimension of the vector.

Manhattan Distance, also known as city block distance or taxicab distance, measures the distance between two points by summing the absolute differences of their coordinates [24]. This metric measures the total absolute distance between the components of two vectors [25]. It is often used in contexts where movement is restricted to straight lines and vertical or horizontal directions. The calculation of Manhattan distance is performed using the following equation:

$$D(X, Y) = \sum_{i=1}^n |X_i - Y_i| \quad (2)$$

where X_i and Y_i are the i -th components of the vectors X and Y , and n is the dimension of the vector.

Chi-Squared Distance is primarily used in statistical contexts and image processing. It measures the difference between two distributions by comparing each element of the vectors, normalizing the difference by the average value of the two elements [26]. This metric accounts for the magnitude of the difference relative to the combined size of the two elements. It is particularly useful in situations where we want to emphasize relative differences between elements with low values. The following equation is used to compute the Chi-Squared distance:

$$D_{\chi^2}(X, Y) = \sum_{i=1}^n \frac{(X_i - Y_i)^2}{X_i + Y_i} \quad (3)$$

where X_i and Y_i are the i -th components of vectors X and Y , and n is the dimension of the vector. This formula

is very useful in situations where we want to emphasize the relative difference between elements with low values, because it takes into account the magnitude of the difference relative to the combined size of the two elements.

The objective of the triplet loss function is to minimize the distance between the anchor and the positive example (which belong to the same class) while simultaneously maximizing the distance between the anchor and the negative example (which belong to different classes). This is achieved by maintaining a fixed margin, denoted by α , which ensures that the positive example is closer to the anchor than the negative example by at least this margin [27]. Mathematically, the triplet loss function is designed to push the model to learn embeddings where similar images (anchor and positive) are clustered together in the feature space, while dissimilar images (anchor and negative) are pushed apart. The triplet loss function is particularly effective in scenarios where the dataset size is small, as it encourages the model to focus on relative distances between samples rather than absolute feature values [28]. This relative comparison allows the model to learn more discriminative features even with limited data, as it directly optimizes the relationships between samples rather than relying on large amounts of labeled data. By enforcing a margin α , the triplet loss ensures that the model does not simply collapse all embeddings into a single point but instead learns a well-structured feature space where similarities and differences are clearly defined. This calculation of triplet loss is governed by a specific equation:

$$L(A, P, N) = \max(0, D(A, P) - D(A, N) + \alpha) \quad (4)$$

In this formulation, D symbolizes the distance metric, which can be Chi-Squared, Manhattan, or Euclidean. A stands for the Anchor, P for the Positive, and N for the Negative, with a fixed margin, α , of 1.0. The function operates by increasing the distance between the Anchor and Negative while decreasing the distance between the Anchor and Positive. This approach effectively prompts the model to develop representations that distinctly segregate different data classes, enhancing its discriminative capability.

This research employs the Xception architecture within a Siamese Neural Network, featuring three identical sub-networks that share weights. Each sub-network includes a sequence of convolutional layers, pooling layers, and fully connected layers [29]. The Xception architecture was chosen due to its efficiency and effectiveness in feature extraction, particularly through its use of depthwise separable convolutions. Depthwise separable convolutions significantly reduce the computational cost and number of parameters compared to traditional convolutional layers, while maintaining high performance in capturing complex spatial features [30]. This makes Xception particularly suitable for image similarity tasks, where the ability to extract fine-

grained features, such as texture, color, and shape, is critical [31].

Using identical hyperparameters for a fair comparison, the models were assessed concurrently with the Chi-Squared, Manhattan, and Euclidean distance metrics [32]. This method highlights the influence of each metric on improving the performance of the models. The details of the specific hyperparameters used in this evaluation are meticulously outlined in Table III, ensuring transparency and replicability of the assessment process.

TABLE III. Hyperparameters of model

Hyperparameters	Value
Batch Size	128
Epoch	15
Optimizer	Adam
Learning Rate	0.001

Upon completion of its training, the model operates by mapping new inputs into a predefined feature space. Within this space, the distances, as learned from the training process, serve to determine the similarity or dissimilarity of the inputs. This determination is based on the criteria established by the triplet configuration, effectively using the learned distances to categorize inputs relative to each other.

F. Evaluation Metrics

Ultimately, the model's performance is evaluated through several metrics, including the confusion matrix [33]. This matrix is an essential instrument for gauging the accuracy of a classification model. It visually presents the count of both correct and incorrect predictions in a structured table format, providing a clear depiction of the model's predictive capabilities. This matrix is particularly valuable in binary classification tasks. It consists of four elements: True Positives (TP), where the model correctly predicts the positive class; True Negatives (TN), where it correctly predicts the negative class; False Positives (FP), cases where the model incorrectly predicts the negative instance as positive; and False Negatives (FN), where it fails to recognize a positive instance, marking it as negative. These elements help quantify the number of correct and incorrect predictions made by the model, facilitating the calculation of performance metrics such as accuracy, precision, recall, and the F1-score.

Once the confusion matrix is obtained, the following metrics will be calculated: accuracy, precision, recall, and F1-Score, according to the specified formulas.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

Accuracy measures the overall correctness of the model and is calculated as the ratio of correct predictions (both true positives and true negatives) to the total number of cases examined.

Precision assesses the accuracy of positive predictions made by the model and is defined as the ratio of true positive predictions to the total number of positive predictions (true positives plus false positives).

Recall, also known as sensitivity, measures the ability of the model to identify all relevant instances within a dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives, indicating how many actual positives were correctly identified.

The F1-Score is a harmonic mean of precision and recall, providing a single score that balances both the precision and the recall. It is particularly useful when dealing with imbalanced datasets, where one class is significantly underrepresented. The F1-Score is calculated as 2 times the product of precision and recall divided by the sum of precision and recall, offering a measure of the model's accuracy in terms of both precision and recall.

This comprehensive approach, from data preparation to detailed evaluation, ensures a robust assessment of the model's ability to differentiate between various classes of trademark images.

3. RESULT AND DISCUSSION

This section presents the results of the Siamese Neural Network models trained using Euclidean, Manhattan, and Chi-Squared distance metrics. It includes a detailed analysis and comparison of the performance of these models, focusing on key metrics such as accuracy, precision, recall, and F1-score.

A. Result

After image preparation and pre-processing steps, the images are randomly organized into triplet sampling, resulting in 400 triplet image pairs derived from 20 positive and 20 negative images. An example of the triplet sampling results can be seen in Fig. 3.

The triplet samples were trained using Siamese Neural Network models with Euclidean, Manhattan, and Chi-Squared distance metrics. Fig. 4. presents the training loss over 15 epochs for three distance metrics—Chi-Squared, Manhattan, and Euclidean—employed in a Siamese neural network.



Figure 3. Result of triplet sample

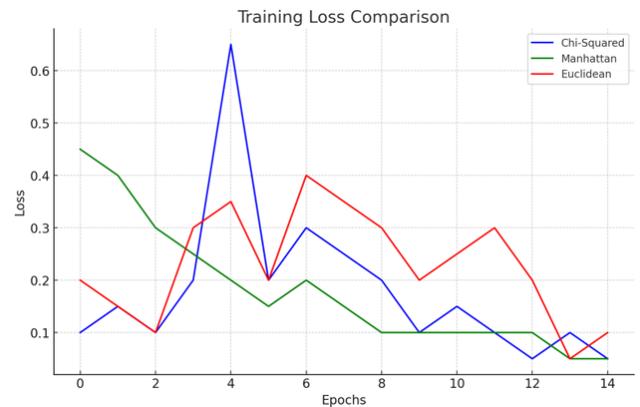


Figure 4. Training loss comparison

The Euclidean loss results show that at the beginning of training, the loss value decreased significantly, indicating that the model is in the process of learning and increasing its accuracy. However, during the process, there are quite striking fluctuations, where the loss value sometimes increases before finally decreasing again. This phenomenon

can be interpreted as an indication that the model may have difficulty finding the right pattern or even potentially overfitting on certain data. However, the general trend shows an overall decrease in loss, meaning that the model is gradually showing improvements in its capabilities. The Euclidean loss graph can be seen in Fig. 5.



Figure 5. Training loss of Euclidean

Furthermore, the analysis of Manhattan Loss, as seen in Fig. 6., shows more consistent results. The graph shows a steady decrease in the loss value from the beginning to the end of training, indicating that the model is learning well from the given data. After reaching a certain point, the loss value appears to stabilize around a very low number, approaching zero, indicating that the model has achieved a good level of accuracy and is not experiencing significant overfitting. In addition, minimal fluctuations in the loss value indicate that the model is not experiencing difficulties in the learning process, thus providing confidence in the effectiveness of the method used.



Figure 6. Training loss of Manhattan

Meanwhile, in the Chi-Squared Loss measurement, there is a significant peak at the beginning of training, where the

loss value reaches its highest point around 0.7. This indicates that the model may have difficulty learning from the data in the early stages. However, after reaching the peak, the loss value begins to decrease, indicating that the model is starting to learn and improve its predictions. Although there are some fluctuations between the lower loss values, indicating challenges in finding consistent patterns in the data, at the end of the graph, the loss value appears to be stable around a lower number. This indicates that the model has achieved a better level of accuracy, indicating significant progress in the training process. The Chi-Squared loss graph can be seen in Fig. 7.

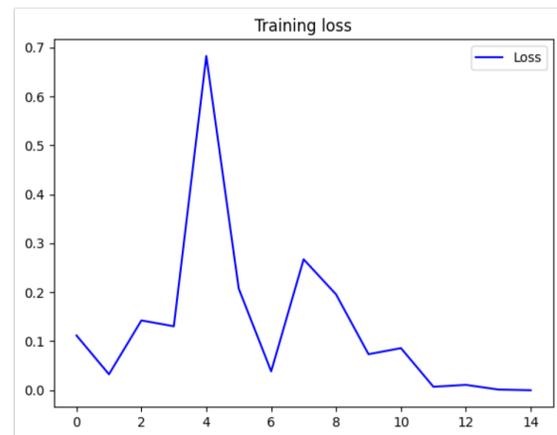


Figure 7. Training loss of Chi-Squared

The accuracy results during training also show different results for each distance metric used. A comparison of model accuracy can be seen in Fig. 8.

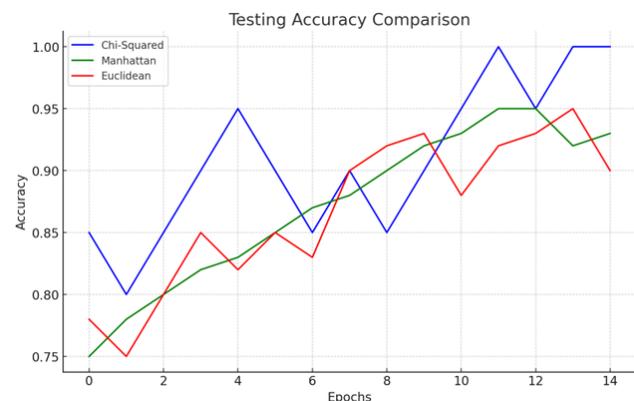


Figure 8. Training accuracy comparison

In the Euclidean Accuracy measurement, as seen in Fig. 9., the graph shows quite significant fluctuations at the beginning of training, with the lowest value around 0.75. This indicates that the model may not have fully understood the patterns in the data provided. However, after several iterations, there is a gradual increase in accuracy, eventually reaching a value above 0.90. This increase indicates that the

model is starting to learn and improve its ability to predict data. At the end of the graph, the accuracy appears stable in the range of 0.90 to 0.95, indicating that the model has achieved a good and consistent level of accuracy, which is in line with the decrease in loss discussed earlier.

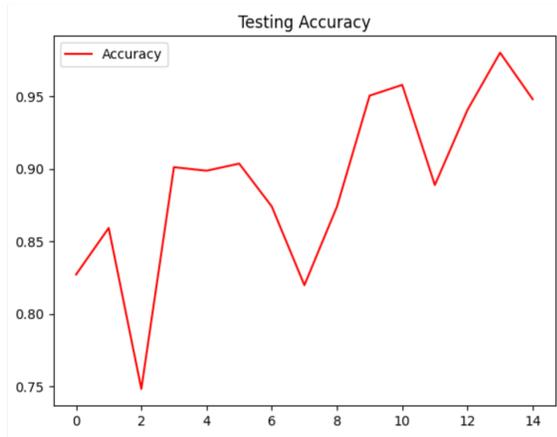


Figure 9. Training accuracy of Euclidean

Furthermore, the analysis of Manhattan Accuracy shows a consistent increasing trend from the beginning to the end of training. Starting from an accuracy of around 0.75, the model shows a significant increase to nearly 0.925. After several iterations, the accuracy appears stable with little fluctuation, indicating that the model has learned well from the data. At the end of the graph, the accuracy reaches its highest value, indicating that the model can be relied on to predict data with high accuracy. This reflects the effectiveness of the training method applied and the consistency in model performance. The accuracy graph per epoch of Manhattan can be seen in Fig. 10.

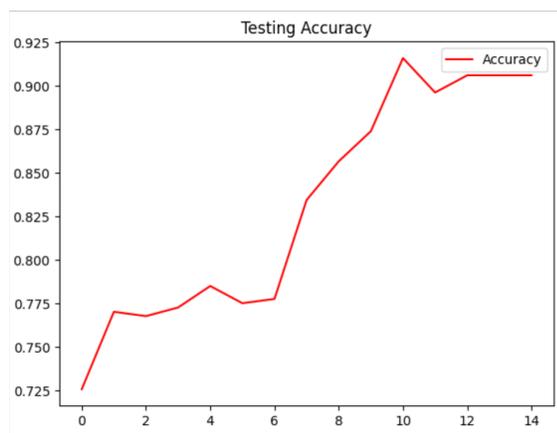


Figure 10. Training accuracy of Manhattan

Finally, in the Chi-Squared Accuracy measurement, there is a pattern similar to the previous measurement, where the initial fluctuations appear quite significant with

the lowest value around 0.75. This indicates that the model has difficulty learning from the data in the early stages. However, after several iterations, the accuracy begins to increase, reaching values above 0.90, indicating that the model is starting to improve its ability to predict data. At the end of the graph, the accuracy reaches its highest value near 1.00, indicating that the model can be relied on to predict data with very high accuracy. This increase in accuracy is in line with the previously observed decrease in loss, indicating that the model is not only learning well, but is also able to apply that knowledge effectively in predictions. The training accuracy results of Chi-Squared are shown in Fig. 11.

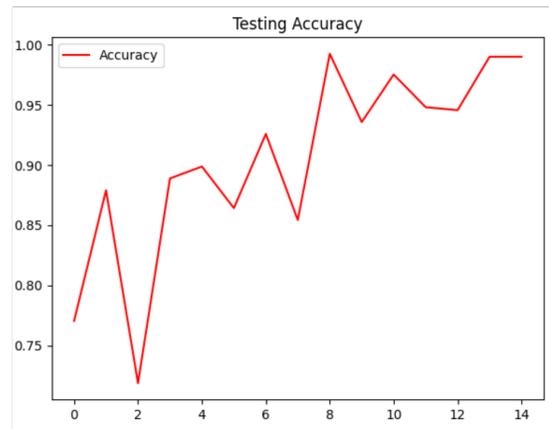


Figure 11. Training accuracy of Chi Squared

At the testing stage, model comparison is evaluated using a confusion matrix. Table IV features the comparison of confusion matrix for this model.

As seen in Fig. 12., The Euclidean results show that the model correctly classifies 42.19% of similar images as True Similar and 50.00% of different images as True Different, demonstrating balanced performance in distinguishing both classes. However, there was a 7.81% error rate where different images were incorrectly classified as similar (False Similar), while no instances were misclassified as different when they were similar (False Different).

The Manhattan distance metric shows slightly higher accuracy in identifying similar images with 45.90% (True Similar), but lower accuracy of 29.49% for dissimilar images (True Different). Notably, this metric suffers from a higher misclassification rate for dissimilar images that are misclassified as similar at 4.10% (False Similar) and a significant misclassification rate of 20.51% when similar images are misclassified as different (False Different). The confusion matrix results from Manhattan can be seen in Fig. 13.

Finally, the Chi-Squared distance metric achieves the highest accuracy in correctly classifying similar images at 50.00% (True Similar) and maintains good performance

TABLE IV. Confusion matrix metrics comparison in testing

Metric Distance	True Similar (%)	False Similar (%)	False Different (%)	True Different (%)
Euclidean	42.19	7.81	0.00	50.00
Manhattan	45.90	4.10	20.51	29.49
Chi-Squared	50.00	0.00	3.52	46.48

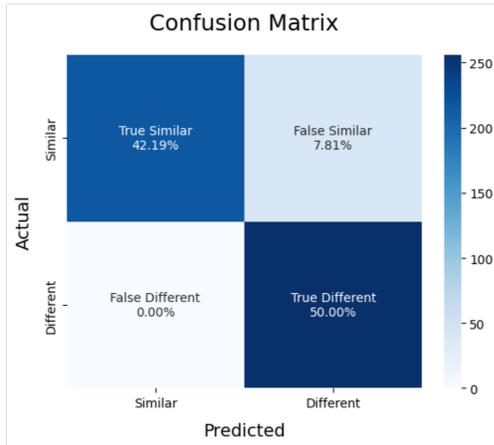


Figure 12. Training loss of Chi-Squared Distance

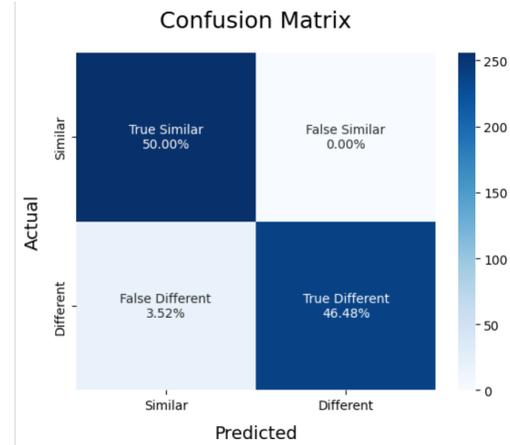


Figure 14. Training loss of Chi-Squared Distance

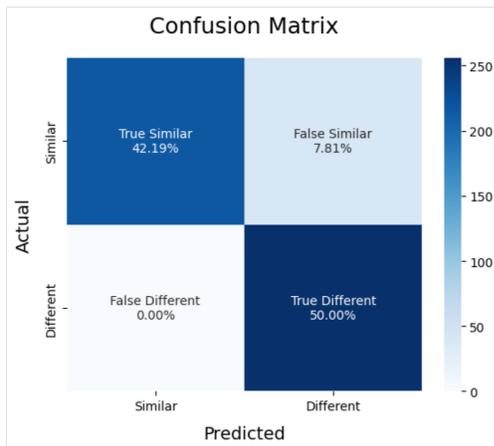


Figure 13. Training loss of Chi-Squared Distance

by the model are accurate. The model achieved a recall rate of 100%, effectively identifying every true instance of similarity, which resulted in a flawless F1-score of 1.00. This score indicates exemplary performance, with the model excellently balancing precision and recall.

with an accuracy of 46.48% for dissimilar images (True Different), as seen in Fig. 14. This metric exhibits the lowest error rate among the three metrics with no instances being misclassified as similar (Incorrect Similar) and a minimum of 3.52% similar images being misidentified as different (Incorrect Different).

Table V and Fig. 15 summarizes the performance comparison, showing that the Euclidean distance metric exhibited strong effectiveness with an accuracy rate of 92%. This high level of accuracy suggests that the metric performs well in correctly classifying both similar and different images. The precision achieved was 84%, indicating a strong likelihood that predictions of similarity

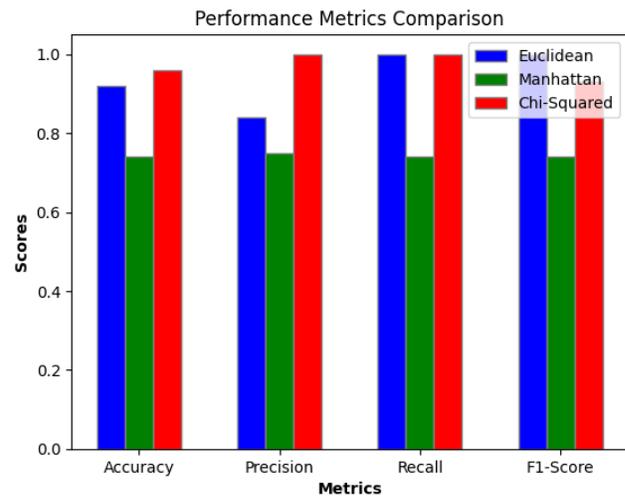


Figure 15. Performance metrics comparison

The Manhattan distance metric demonstrated a moderate level of effectiveness, achieving an overall accuracy of 74%. Both precision and recall were nearly equivalent, at 75% and 74% respectively, suggesting a balanced yet modest proficiency in accurately identifying and capturing true positives. The F1-score, a direct reflection of this balance, stood



TABLE V. Performance result of Euclidean, Manhattan, and Chi-Squared

Metric Distance	Accuracy	Precision	Recall	F1-Score
Euclidean	0.92	0.84	1.00	1.00
Manhattan	0.74	0.75	0.74	0.74
Chi-Squared	0.96	1.00	1.00	0.93

at 0.74, suggesting consistent but less optimal performance across these metrics compared to the Euclidean distance.

Conversely, the Chi-Squared showed superior performance with the highest accuracy of 96% among the metrics evaluated. It achieved a precision of 100%, indicating that every prediction of similarity was accurate. Similarly, the recall was also perfect at 100%, showing that the metric identified all similar images without fail. The F1-score, at 0.93, although slightly lower than the perfect scores, still indicates an exceptionally high level of performance in both precision and recall.

B. Discussion

The analysis of the trained Siamese Neural Network model, which uses different distance measures to check trademark image similarities, has revealed several important results. The Chi-Squared metric shows the best performance with an accuracy of 96% and a perfect precision of 1.00. These findings confirm that this model is very effective in identifying similarities between trademark images, which is the main objective of this study. Thus, the application of this model can make a significant contribution to protecting intellectual property rights and preventing trademark infringement.

However, there are several limitations that need to be considered. The model is accurate, but at first, the loss and accuracy values using Euclidean and Manhattan metrics fluctuate a lot. This suggests the model might struggle to learn from the data early on. This could mean that the model requires more training data or the application of better regularization techniques to avoid overfitting. In addition, although the Chi-Squared metric shows very good results, its use may not always be optimal for all types of data or application contexts, so further evaluation is needed.

Future research directions can be focused on several aspects. First, exploration of different combinations of distance metrics can be done to improve model performance. Combining multiple metrics in a single model can provide more robust results. Additionally, research can focus on creating better data enhancement methods to make training data more varied. This will help the model learn more effectively from the different types of data available. The application of transfer learning techniques, where a model that has been trained on a large dataset is tailored for the specific task of trademark image similarity detection, can also be an interesting approach to improve model accuracy and consistency.

This study shows that Siamese Neural Networks are

effective for detecting similarities in trademark images. It also suggests ways to continue researching this technology, which could lead to its broader use in industry. Further research is expected to overcome existing limitations and improve model performance in a wider context.

4. CONCLUSION

This study has successfully developed and analyzed a Siamese Neural Network model optimized using various distance metrics for trademark image similarity detection. The results obtained show that the Chi-Squared metric provides the best performance, with an accuracy of 96% and perfect precision. These findings demonstrate the enormous potential of the model in practical applications, especially in protecting intellectual property rights and preventing trademark infringement.

Although the model shows promising results, there are some limitations that need to be considered, such as initial fluctuations in loss and accuracy values in the Euclidean and Manhattan metrics. This indicates the need for further development, including exploring different combinations of distance metrics and implementing more sophisticated data augmentation techniques.

Future research directions can focus on developing transfer learning techniques and testing the model on more diverse datasets to improve generalization and accuracy. This study helps us understand how well Siamese Neural Networks work for finding similar images. It also creates chances for more research that can improve how this technology is used in different industries.

5. CONFLICTS OF INTEREST

The authors declare no conflict of interest.

6. AUTHOR CONTRIBUTIONS

Conceptualization, Suyahman; methodology, Suyahman; validation, Sunardi and Murinto; writing – original draft, Suyahman; supervision, Sunardi; writing – review & editing, Sunardi, Murinto, and Arfiani Nur Khusna; funding acquisition, Sunardi.

REFERENCES

- [1] World Intellectual Property Organization (WIPO), *World Intellectual Property Indicators 2023*. Geneva: WIPO, 2023, accessed: May 10, 2024. [Online]. Available: <https://www.wipo.int/publications/en/details.jsp?id=4678>.
- [2] B. Sharp and J. Dawes, "What is differentiation and how does it work?" *Journal of Marketing Management*, vol. 17, no. 7-8, pp. 739-759, 2001.

- [3] S. L. Dogan and M. A. Lemley, "What the right of publicity can learn from trademark law," *Stan. L. Rev.*, vol. 58, p. 1161, 2005.
- [4] R. Gul, "The relationship between reputation, customer satisfaction, trust, and loyalty," *Journal of Public Administration and Governance*, vol. 4, no. 3, pp. 368–387, 2014.
- [5] G. T. Lau and S. H. Lee, "Consumers' trust in a brand and the link to brand loyalty," *Journal of market-focused management*, vol. 4, pp. 341–370, 1999.
- [6] W. M. Landes and R. A. Posner, "Trademark law: an economic perspective," *The Journal of Law and Economics*, vol. 30, no. 2, pp. 265–309, 1987.
- [7] M. Bartholomew, "Trademark morality," *Wm. & Mary L. Rev.*, vol. 55, p. 85, 2013.
- [8] G. V. M. Kumar and D. Madhavi, "Stacked siamese neural network (ssinn) on neural codes for content-based image retrieval," *IEEE Access*, 2023.
- [9] K. Zhang, S. Qi, J. Cai, D. Zhao, T. Yu, Y. Yue, Y. Yao, and W. Qian, "Content-based image retrieval with a convolutional siamese neural network: Distinguishing lung cancer and tuberculosis in ct images," *Computers in biology and medicine*, vol. 140, p. 105096, 2022.
- [10] M. S. Devi, J. A. Pandian, A. Joshi, and Y. Praveen, "Batch normalized siamese network deep learning based image similarity estimation," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2023, pp. 1–5.
- [11] A. Jalilian and J. Mateu, "Assessing similarities between spatial point patterns with a siamese neural network discriminant model," *Advances in Data Analysis and Classification*, vol. 17, no. 1, pp. 21–42, 2023.
- [12] Suyahman, Sunardi, and Murinto, "Comparative analysis of cnn architectures in siamese networks with test-time augmentation for trademark image similarity detection," *Scientific Journal of Informatics*, vol. 11, no. 4, Nov. 2024.
- [13] Suyahman, "Indonesian trademark logo dataset," <https://www.kaggle.com/dsv/8401698>, 2024, dOI: 10.34740/KAGGLE/DSV/8401698.
- [14] Muis, S. Sunardi, and A. Yudhana, "Comparison analysis of brain image classification based on thresholding segmentation with convolutional neural network," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 2, p. 664–673, Jun. 2023.
- [15] Dehghan, P. Razzaghi, K. Abbasi, and S. Gharaghani, "Triplet-multitditi: Multimodal representation learning in drug-target interaction prediction with triplet loss function," *Expert Systems With Applications*, vol. 232, p. 120754, Dec. 2023.
- [16] I. N. et al., "Cholectriplet2022: Show me a tool and tell me the triplet — an endoscopic vision challenge for surgical action triplet detection," *Medical Image Analysis*, vol. 89, p. 102888, Oct. 2023.
- [17] K. Vanitha, S. S. Sree, and S. Guluwadi, "Deep learning ensemble approach with explainable ai for lung and colon cancer classification using advanced hyperparameter tuning," *BMC Medical Informatics and Decision Making*, vol. 24, no. 1, p. 222, 2024.
- [18] Q. Li and S. He, "Similarity matching of medical question based on siamese network," *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, p. 55, 2023.
- [19] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1420–1429.
- [20] D. Chicco, "Siamese neural networks: An overview," *Artificial neural networks*, pp. 73–94, 2021.
- [21] H. Oki, M. Abe, J. Miyao, and T. Kurita, "Triplet loss for knowledge distillation," in *2020 International Joint Conference on Neural Networks (IJCNN)*, July 2020, pp. 1–7.
- [22] Yudhana, S. Sunardi, and A. J. S. Hartanta, "Algoritma k-nn dengan euclidean distance untuk prediksi hasil penggergajian kayu sengon," *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 22, no. 4, pp. 123–129, Nov. 2020.
- [23] N. Trisanti, S. Sunardi, and A. Fadlil, "Penerapan metode euclidean pada pengenalan wajah siswa taman kanak-kanak," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 10, no. 1, pp. 903–914, 2023.
- [24] Fadlil and N. Trisanti, "The application of the manhattan method to human face recognition," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 6, pp. 939–944, 2022.
- [25] A. Fadlil and N. Trisanti, "Comparative analysis of euclidean, manhattan, canberra, and squared chord methods in face recognition," *Revue d'Intelligence Artificielle*, vol. 37, no. 3, p. 593–599, Jun. 2023.
- [26] U. I. Larasati, M. A. Muslim, R. Arifudin, and A. Alamsyah, "Improve the accuracy of support vector machine using chi square statistic and term frequency inverse document frequency on movie review sentiment analysis," *Scientific Journal of Informatics*, vol. 6, no. 1, pp. 138–149, 2019.
- [27] Yan, G. Pang, X. Bai, C. Liu, X. Ning, L. Gu, and J. Zhou, "Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss," *IEEE Transactions on Multimedia*, vol. 24, pp. 1665–1677, 2021.
- [28] J. Zhang, C. Lu, J. Wang, X.-G. Yue, S.-J. Lim, Z. Al-Makhadmeh, and A. Tolba, "Training convolutional neural networks with multi-size images and triplet loss for remote sensing scene classification," *Sensors*, vol. 20, no. 4, p. 1188, 2020.
- [29] S. H. Kassani, P. H. Kassani, R. Khazaeinezhad, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Diabetic retinopathy classification using a modified xception architecture," in *2019 IEEE international symposium on signal processing and information technology (IS-SPIIT)*. IEEE, 2019, pp. 1–6.
- [30] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [31] S. D. Pujari, M. M. Pawar, and M. Wadekar, "Multi-classification of breast histopathological image using xception: Deep learning with depthwise separable convolutions model," in *Techno-Societal 2020: Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications—Volume 1*. Cham: Springer International Publishing, May 2021, pp. 539–546.
- [32] M. Murinto, S. Winiarti, and I. Faisal, "Particle swarm optimization



algorithm for hyperparameter convolutional neural network and transfer learning vgg16 model,” *Journal of Computer Science, Information Technology and Telecommunication Engineering*, vol. 5, no. 1, pp. 474–480, 2024.

[33] J. T. Townsend, “Theoretical analysis of an alphabetic confusion matrix,” *Perception & Psychophysics*, vol. 9, pp. 40–50, 1971.
