DeepSL: Deep Neural Network-based Similarity Learning

Mohamedou Cheikh Tourad*[1], Abdali Abdelmounaim[2], Mohamed Dhleima[3]
Cheikh Abdelkader Ahmed Telmoud[4], Mohamed Lachgar[5]
CSIDS, FST, University of Nouakchott, Mauritania [1, 3, 4]
CISIEV, FSTG, Cadi Ayyad University, Morocco[2]
LTI, ENSA, Chouaib Doukkali University, El Jadida, Morocco[5]

***Abstract*—The quest for a top-rated similarity metric is inherently mission-specific, with no universally "great" metric relevant across all domain names. Notably, the efficacy of a similarity metric is regularly contingent on the character of the challenge and the characteristics of the records at hand. This paper introduces an innovative mathematical model called MCESTA, a versatile and effective technique designed to enhance similarity learning via the combination of multiple similarity functions. Each characteristic within it is assigned a selected weight, tailor-made to the necessities of the given project and data type. This adaptive weighting mechanism enables it to outperform conventional methods by providing an extra nuanced approach to measuring similarity. The technique demonstrates significant enhancements in numerous machine learning tasks, highlighting the adaptability and effectiveness of our model in diverse applications.**

***Keywords*—Similarity learning; Siamese networks; MCESTA; triplet loss; similarity metrics**

I. Introduction

Similarity learning, a critical domain within machine learning, is dedicated to creating algorithms capable of determining the degree of similarity or relatedness between pairs of items [1].

This location of research reveals application throughout a huge spectrum of obligations, together with but not restrained to image type, item detection, and natural language processing, where know-how the nuances of similarity can drastically impact the effectiveness of the models deployed. At the heart of similarity getting to know lies the Siamese triplet network architecture [2], famed for its efficiency in learning excellent-grained similarity distinctions [3]. This architecture employs a specialized shape of schooling referred to as triplet loss, which optimizes the version to minimize the distance among comparable items at the same time as maximizing the space among diverse ones inside a learned embedding area [4].

Despite the plethora of distance metrics [5] to be had for deployment in the distance layer of Siamese models, including Euclidean distance and cosine similarity, the selection of the most appropriate metric stays critical to the achievement of the getting to know manner. In this context, the paintings introduces MCESTA [6], an innovative method that synergizes a couple of similarity metrics [5], every fine-tuned with task-precise weights, to achieve superior performance in similarity mastering responsibilities. Through this composite metric machine, MCESTA seeks to set up a new benchmark in the discipline, imparting a flexible and sturdy way to the challenges of similarity measurement.

Siamese triplet networks are a type of neural network that is often used for similarity learning. Siamese triplet networks are trained using a loss function called triplet loss. Triplet loss encourages the network to learn a similarity function that places similar images close to each other in the embedding space and dissimilar images far apart.

There are a variety of different distance metrics that can be used in the distance layer of a Siamese model. Some of the most common distance metrics include:

- Euclidean distance [5]: Euclidean distance, a key metric in the domain of vector spaces, measures the length of a straight segment directly connecting two vectors In mathematics, it is a standard or measure of the vector distance between two points in multidimensional space. This distance measure obeys the principles of Euclidean geometry, providing a sensitive measure of the separation between vectors.
- Cosine similarity [5]: A key concept in vector space analysis, cosine similarity, refers to the cosine of the angle formed between two vectors. This similarity measure is particularly sensitive to high-altitude areas, where it measures the directional alignment of the vectors rather than their magnitudes It ranges from -1 to 1, where 1 indicates perfect alignment, 0 indicates orthogonality, and -1. 1 indicates diametric opposition. Cosine equations excel in capturing systematic relationships and patterns in datasets, making them a common choice in scientific and machine learning applications.

In the study, MCESTA is employed as a metric of similarity, representing a combination of three standard similarity metrics [6].

This paper is organized into six main sections, each designed to systematically explore and present the research conducted. Section II delves into the existing literature and studies that have set the foundation for the current investigation, providing a context for the proposed methodology and highlighting the gaps and opportunities for innovation. Section III introduces the novel methodology developed for this study, detailing the theoretical underpinnings, the design of the Siamese Network, and the rationale behind the choice of encoder and feature vectors. Section IV describes the experimental framework, including the dataset used, the configuration of the network, and the specifics of the implementation that enable a thorough examination of the proposed approach. Section V presents the outcomes of the experiments, analyzes the findings in depth, compares them with existing methods, and discusses the implications and the potential impact of the study. Finally, Section VI summarizes the key findings, acknowledges the limitations of the study, and outlines directions for future research, encapsulating the contribution of the work to the field of similarity learning and face recognition technologies.

II. Background

A. Similarity Learning

The notion of similarity is very important in computer science and mathematics. Different methods of analogy can be used when comparing two vectors with different elements. The choice of method depends on the main objective of the comparison, which includes methods such as Euclidean distance, Pearson correlation coefficient, Spearman's rank correlation coefficient [7].

Similarity learning is a supervised machine learning technique in artificial intelligence. Regression is closely related to classification, but the goal is to find a similarity function that shows how similar or related two things are This has applications in ranking, recommendation systems, visual recognition tracking, face verification, and speaker verification [8].

Four patterns of similarity and metric distance learning are common [8]:

1) Learning Regression Analogy:

- There are two in this case (x_i^1, x_i^2) have given proof of their similarity $y_i \in \mathbb{R}$.
- The goal is to find a function that calculates $f(x_i^1, x_i^2) \sim y_i$ for each new sample written three times (x_i^1, x_i^2, y_i).
- This is usually achieved by reducing regular losses $\min_W \sum_i \text{loss}(W; x_i^1, x_i^2, y_i) + \text{reg}(W)$.

2) Study Taxonomic Similarity:

- Given two such elements (x_i, x_i^+) and unequal elements (x_i, x_i^-).
- As a binary label for each pair (x_i^1, x_i^2) $y_i \in \{0, 1\}$ determining their equations.
- The aim is to find a classifier that can decide whether two other objects are the same or not.

3) Study Group Equation:

- Given triple factors (x_i, x_i^+, x_i^-) with relative similarities following a predefined order.
- The objective is to find the function f which gives every other triple (x, x^+, x^-) that $f(x, x^+) > f(x, x^-)$ (inverse learning).
- This scheme assumes easier maintenance compared to regression.

4) Local Hot Hashing (LSH):

- LSH hashes input objects so that similar objects map to the same "buckets" in memory with high probability.
- Commonly used in nearest-neighbor searches in large, high-dimensional databases, such as image databases, document stacks, and genome databases.

A prevalent strategy for learning similarity involves modeling the similarity function as a bilinear form. For instance, in ranking similarity learning, the aim is to learn a matrix W that parameterizes the similarity function $f_W(x, z) = x^T W z$. When data is abundant, a common approach is to utilize a siamese network—a deep network model with shared parameters [3].

B. Similarity Models

Similarity models play a crucial role in various domains, ranging from information retrieval and data analysis to machine learning and pattern recognition. These models are designed to quantify the likeness or resemblance between different entities, such as documents, images, or data sets. They form the basis for numerous applications, aiding in tasks like recommendation systems, clustering, and classification. Here's an overview of key aspects related to similarity models:

1) Euclidean Distance: The Euclidean distance [9]is a fundamental measure of similarity, representing the straight-line distance between two points in Euclidean space.
2) Cosine Similarity: The cosine similarity metric represents a text as a vector of terms, and the similarity between two texts is determined by the cosine value between their respective term vectors. Nevertheless, cosine similarity struggles to accurately capture the semantic meaning of the text [10] [9].
3) Jaccard Index: The Jaccard Index [9] calculates the similarity between sets by measuring the intersection over the union. Predominantly used in areas like information retrieval, text analysis, and recommendation systems, where set-based comparisons are essential .
4) Fuzzy Similarity Models: Fuzzy similarity models [11], like those employing trapezoidal fuzzy numbers, are designed to handle uncertainty and vagueness in data. Particularly useful in situations where data is imprecise or lacks clear boundaries, such as in linguistic variables.
5) Machine Learning-Based Similarity Models: With the rise of machine learning, similarity models leveraging algorithms like k-nearest neighbors (KNN) or deep learning-based embeddings have gained prominence [12]. These models are applicable in diverse domains, including image recognition, recommender systems, and anomaly detection.
6) Hybrid Models: Hybrid similarity models combine multiple similarity measures to enhance performance and address specific challenges. Especially beneficial when dealing with diverse data types or when aiming for a more comprehensive understanding of similarity.
7) Graph-Based Similarity Models: Similarity models based on graph theory consider relationships and connections between entities in a network [13]. Applied in social network analysis, recommendation systems, and community detection.

In conclusion, similarity models are versatile tools with applications spanning various domains. Their effectiveness depends on the nature of the data and the specific requirements of the task at hand. Advances in machine learning and data representation continue to contribute to the development of more sophisticated and context-aware similarity models.

C. An Intelligent Similarity Model MCESTA

The mathematical model proposed in this paper uses fuzzy estimation systems to determine the value of the effective load. These weights are associated with methods that are able to handle a significant amount of information. The importance weights are calculated using a Mamdani-type fuzzy inference system (FIS), using the cosine coefficient and the Jaccard index. Three properties of the model are also demonstrated, one of which is useful for use with large datasets [6]. MCESTA (**M**ohamedou **C**heikh **E**lghotob Cheikh **S**aad bouh Cheikh **T**ourad **A**bass) is the new estimation algorithm proposed in this paper, representing MC Tourad and A Abdali. It acts as a great similarity distance between generalized trapezoidal fuzzy numbers (GTFNs) and is a hybrid of the similarity measure. In order to distinguish between the proposed method and the existing methods, a comparative study is carried out based on 21 different generalized trapezoidal fuzzy numbers (GTFNs) This study shows that the proposed model is more reasonable than existing methods and can overcome current limitations system.

$$MCESTA(\widetilde{T}, \widetilde{H}) = \sum_{k=1}^{n} \alpha_k \cdot S_k(\widetilde{T}, \widetilde{H}), \tag{1}$$

where

$$S_k(\widetilde{T}, \widetilde{H}) = \sum_{q=1}^{m} \beta_q \cdot S_{qk}(\widetilde{T}, \widetilde{H}), \tag{2}$$

and

$$\sum_{k=1}^{n} \alpha_k \leq 1, \sum_{q=1}^{m} \beta_q \qquad \leq 1.$$

where S_k is a similarity method between $\widetilde{T}$ and $\widetilde{H}$, and S_{qk} is a similarity sub-measure between $\widetilde{T}$ and $\widetilde{H}$,

III. Related Work

The panorama of similarity learning is rich and sundry, with a wide array of strategies and fashions proposed to deal with the intricacies of measuring similarity. Among these, Siamese triplet networks have emerged as a cornerstone, specifically for his or her software in generating embedding that mirror the relative similarities amongst facts points. Central to the operation of those networks is the idea of triplet loss, a mechanism that has been extensively studied for its effectiveness in distinguishing among pairs of similar and assorted items [3].

In a study by Vorontsov et al [14], the authors addressed the challenge of comparing transcription factor binding site (TFBS) models, focusing on positional weight matrices (PWMs) in particular common PWMs to quantify TF binding; however, different ones arise when TF-binding DNA fragments obtained from different experimental methods give rise to similar but not identical PWMs. Existing tools often compare matrix elements directly to PWMs, which can be limiting, especially when dealing with log-odds PWMs where negative factors do not contribute to high-scoring TF binding sites To address this , Vorontsov et al. A practical method based on a Jaccard index was introduced, which takes into account PWM and the respective scores, this new method simplifies TFBS modeling if TFBS modeling is done by various methods, such as raw-state counts, log anomalies PWMs and comparison f The proposed algorithm, implemented in the software MACRO-APE (MAtrix CompaRisOn by Approximate P-value Estimation), efficiently computes similarities based on Jaccard index for two TFBS samples The software is more work, accommodating TFBS models of different lengths and construction methods. The authors also present a two-pass scanning algorithm for detecting query-like PWMs presented in the collection [14].

Concurrently, the exploration of distance metrics plays a essential role in the development of similarity learning models. Traditional metrics like Euclidean distance and cosine similarity were the situation of a great deal studies, every with its own set of advantages and barriers relying on the software domain. Recent improvements have sought to transcend those barriers via featuring hybrid or composite metrics that integrate the strengths of individual measures.

Against this backdrop, MCESTA represents a huge leap ahead, embodying the next generation of similarity metrics via harnessing the power of multiple metrics tailor-made through adaptive weighting. This approach now not best addresses the inherent boundaries of unmarried-metric procedures however also introduces a level of customization formerly unseen in the discipline. The evaluation of related works underscores the evolutionary trajectory of similarity learning, setting the stage for MCESTA's contribution to this ongoing narrative.

A. Siamese Neural Network

Siamese neural networks consist of two identical artificial neurons, each capable of learning a hidden representation of the input vector Both neurons are feedforward perceptrons and use error surface propagation during training. They operate simultaneously, process the input vector independently, and subsequently compare their output, usually using a cosine distance measure. The execution result of the Siamese neural network can be interpreted as the logical similarity between the predicted values of the two input vectors [7]. See Fig. 1 for illustration.

1) Architecture: Siamese Network is a type of network architecture that contains two or more identical sub-network used for generate feature vectors for each input and compare them. A Siamese Neural Network is a class of neural network architectures that contain two or more identical sub-networks. Identical, here means, they have the same configuration with the same parameters and weights. Parameter updating is mirrored across both sub-networks. It is used to find the similarity of the inputs by comparing its feature vectors, so these networks are used in many applications [15] [16].

The architecture is as follows:

- Feature Extraction layers: Each sub-network contains a encoder that converts input into a dense vector. This encoder typically consists of multiple layers of neural

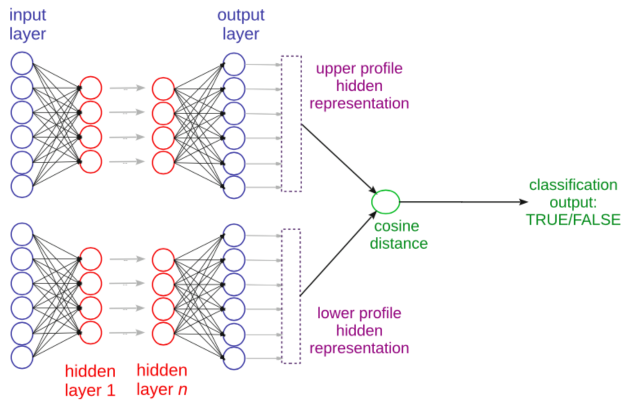

Fig. 1. Understanding the Siamese neural network: Architecture and cosine distance metric [7].

units, such as CNN, LSTM, GRU, or fully connected layers. The shared weights ensure that both sub-networks learn similar representations for similar inputs [17]. The encoded vectors are then passed through additional layers for feature extraction. These layers learn to extract high-level features that are important for measuring text similarity.

- Distance layer: The final output of the sub-networks is a pair of feature vectors. The similarity between the inputs is computed using a distance metric, such as Euclidean distance or cosine similarity, between these vectors. Smaller distances indicate higher similarity [18].

The Siamese Deep Neural Network's architecture and training process make it a powerful tool for measuring similarity, as it can capture subtle semantic relationships between inputs and provide accurate similarity scores.

B. Triplet Loss

Triplet loss are similar to Contrastive Loss, but it take three inputs instead of two: an anchor A, a positive P, and a negative N. The goal of the network is to learn a representation for each image such that the distance between the anchor and positive image is smaller than the distance between the anchor and negative image [19] [20].

$$\text{d(A,P)} = \|f(A) - f(P)\|, \tag{3}$$
$$\text{d(A,N)} = \|f(A) - f(N)\| \tag{4}$$

And we want:

$$\|f(A) - f(P)\| \leq \|f(A) - f(N)\|, \tag{5}$$
$$\|f(A) - f(P)\| - \|f(A) - f(N)\| \leq 0. \tag{6}$$

When the input are the same, and so

$d(A, P) = d(A, N) = 0$, the loss i equal to zero. This is call **trivial solution**.

To prevent trivial output, a new term called *margin* is introduced, which pushes the anchor-positive pair and the anchor-negative pair further away from each other

$$\|f(A) - f(P)\| + margin - \|f(A) - f(N)\| \leq 0 \tag{7}$$

$$L(A, P, N) = \max(|f(A) - f(P) \\ | + margin - \|f(A) - f(N)\|, 0) \tag{8}$$

The *Cost function*:

$$J = \sum_{i=0}^{n} L(A^{(i)}, P^{(i)}, N^{(i)}). \tag{9}$$

IV. Approach

The innovative proposed approach consists of integrating MCESTA into the Siamese Network architecture by replacing the cosine distance in the existing Siamese architecture (see Fig. 2) with the MCESTA model. This modification has yielded extraordinary results compared to other methods mentioned in the related works.

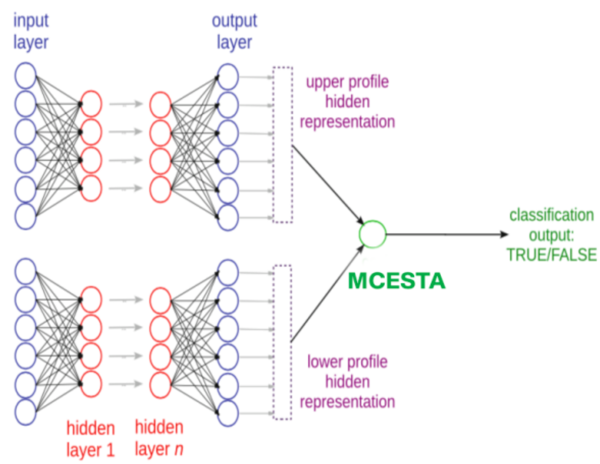

Fig. 2. Understanding the Siamese neural network: Architecture and MCESTA similarity model.

Training the Siamese Deep Neural Network involves optimizing the shared weights to minimize the distance between feature vectors of similar pairs and maximize the distance between feature vectors of dissimilar pairs.

The training process includes the following steps (see Fig. 3) :

- Creating Triplets: The train and test lists are utilized to create triplets of **(anchor, positive, negative)** face data, where the positive instance is the same person as the anchor, and the negative is a different person than the anchor.
- Creating the Model : Unlike a conventional CNN, the Siamese Network does not classify the images into certain categories or labels, rather it only finds out the distance between any two given images. If the images have the

same label, then the network should learn the parameters, i.e. the weights and the biases in such a way that it should produce a smaller distance between the two images, and if they belong to different labels, then the distance should be larger.
The Encoder is responsible for converting the passed images into their feature vectors. We're using a pretrained model, **Xception model** which is based on **Inception-V3 model**. By using transfer learning, it is possible to significantly reduce both the training time and the size of the dataset required.
The Model is connected to **Fully Connected (Dense) layers** and the last layer normalises the data using **L2 Normalisation**. (L2 Normalisation is a technique that modifies the dataset values in a way that in each row the sum of the squares will always be up to 1).
A Siamese Network is created to process **3 input images** (anchor, positive, negative), utilizing the **encoder** to encode the images into their respective feature vectors.
Those features are passed to a **distance layer** which computes the distance between (anchor, positive) and (anchor, negative) pairs. A custom layer is defined for computing the distance, wherein **MCESTA** is employed as the metric of similarity instead of other metrics.

- Training: The network is trained using a triplet loss function. This loss penalizes the model when the similarity of positive pairs is below a certain threshold and when the dissimilarity of negative pairs is above another threshold. This encourages the network to learn meaningful and discriminative representations.

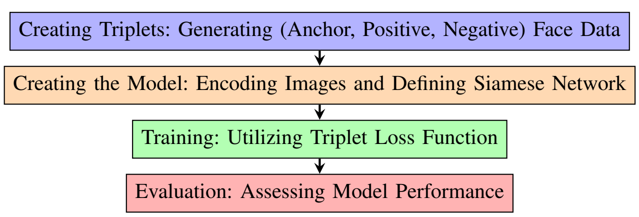

Fig. 3. Siamese Neural Network Training Process

V. Experimental

A. Implementation

Implementing a model necessitates integrating a custom training loop, a custom layer for distance computation utilizing MCESTA, and a loss function. This configuration facilitates the calculation of triplet loss using the three embeddings generated by the Siamese network. A Mean metric instance is established to monitor the training process's loss. The training of the *Siamese-model* will proceed on batches of triplets, with the training loss and additional metrics from testing reported every epoch. Model weights will be saved whenever an improvement over the previous *max-accuracy* is achieved.

B. Dataset

The Face Recognition Dataset, derived from the Labeled Faces in the Wild Dataset (LFW) which is a database of face photographs designed for studying the problem of unconstrained face recognition. This database was created and maintained by researchers at the University of Massachusetts, Amherst (specific references are in Acknowledgments section). 13,233 images of 5,749 people were detected and centered by the Viola Jones face detector and collected from the web. 1,680 of the people pictured have two or more distinct photos in the dataset. The original database contains four different sets of LFW images and also three different types of "aligned" images. According to the researchers, deep-funneled images produced superior results for most face verification algorithms compared to the other image types. Hence, the dataset uploaded here is the deep-funneled version. The dataset is utilized for developing face detection and recognition models. This dataset comprises JPEG images of famous individuals collected from the internet (see Fig. 4). More details can be found on the official website: `http://vis-www.cs.umass.edu/lfw/`.

Fig. 4. LFW-facial-recognition-benchmark-database.

Each picture is centered on a single face, and every image is encoded in RGB. The original images are of the size 250 x 250. The dataset contains 1680 directories, each representing a celebrity. Each directory has 2-50 images for the celebrity. Extracted Faces Faces extracted from the original image using Haar-Cascade Classifier (cv2) encoded in RGB and size of image is 128, 128

VI. Results and Discussion

A. Training Loss

Fig. 5 shows the training loss for a machine learning model. The x-axis represents the number of training epochs, and the y-axis represents the loss. The loss is a measure of how well the model is performing on the training data. A lower loss indicates that the model is performing better.

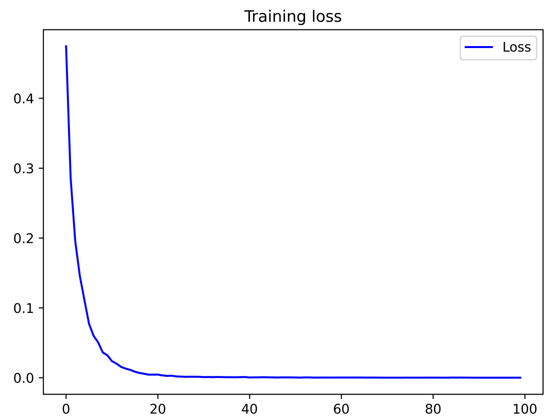

Fig. 5. Testing loss.

Fig. 5 shows that the loss decreases over time, which indicates that the model is learning. The loss is still decreasing at the end of the training, which suggests that the model could continue to improve with more training.

Fig. 5 shows a plot of training loss over time. The training loss is measured on a scale of 0 to 0.47131. The training loss decreases over time, starting at 0.47131 and decreasing to 0.00015 at the end of training.

Fig. 5 shows that the model is training well and is likely to perform well on new data.

B. Testing Accuracy

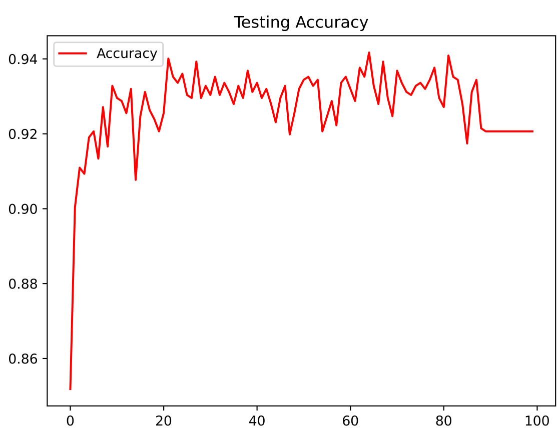

Fig. 6. Testing accuracy.

Fig. 6 shows a graph of testing accuracy over time. The x-axis represents the number of training epochs, and the y-axis represents the testing accuracy. The testing accuracy is a measure of how well the model performs on data that it has not seen before.

Fig. 6 shows that the testing accuracy increases over time, which indicates that the model is learning to generalize to new data. The testing accuracy is still increasing at the end of the training, which suggests that the model could continue to improve with more training.

Fig. 6 shows a plot of testing accuracy over time. The testing accuracy is measured on a scale of 0.9 to 0.94. The testing accuracy increases over time, starting at 0.9 and increasing to 0.94 at the end of training.

Fig. 6 shows that the model is training well and is likely to perform well on new data. However, it is important to monitor the testing accuracy to ensure that the model is not overfitting to the training data.

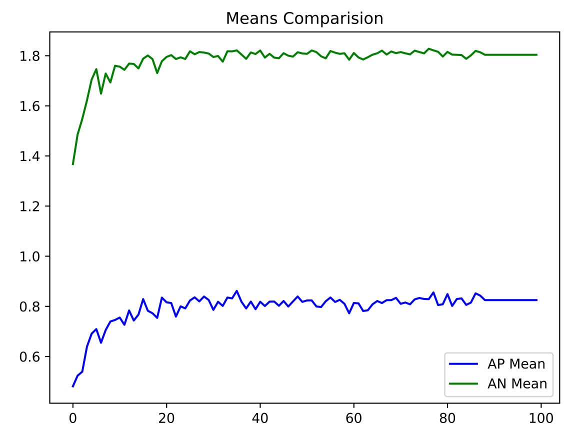

Fig. 7. Performance comparison of similarity metrics.

Fig. 7 shows a confusion matrix for a binary classification problem. The confusion matrix is a square table that shows how many instances were predicted to be in each class, and how many were actually in each class.

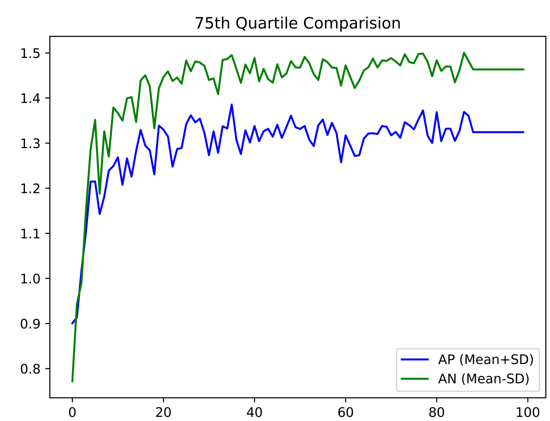

Fig. 8. Evolution of model accuracy using proposed methodology.

The rows of the confusion matrix represent the actual classes, and the columns represent the predicted classes. The diagonal cells of the matrix show the number of instances that were correctly predicted, and the off-diagonal cells show the number of instances that were incorrectly predicted (see Fig. 8).

In the confusion matrix you sent, the actual classes are "true similar" and "true different", and the predicted classes are "predicted similar" and "predicted different".

The diagonal cells of the matrix show that 41.80% of the instances were correctly predicted to be similar, and 44.34% of the instances were correctly predicted to be different.

The off-diagonal cells of the matrix show that 8.20% of the instances were incorrectly predicted to be similar, and 5.66% of the instances were incorrectly predicted to be different.

The confusion matrix shows that the model is performing well on this problem. The model is correctly predicting more instances than it is incorrectly predicting, and the off-diagonal cells of the matrix are relatively small.

The choice of comparing MCESTA with Euclidean and Manhattan distance guided by the characteristics of dataset. This table presents a comparison of different methods based on two key metrics: loss and accuracy on a test dataset. In this comparison:

The MCESTA method has the lowest loss (0.00015), indicating that it performs the best in terms of minimizing errors during training. This suggests that it's effective in optimizing the model's parameters. The MCESTA method also has the highest test accuracy (0.91438). This means that it performs best in making correct predictions on unseen data, which is a crucial measure of a model's overall performance.

The Euclidean method also demonstrates strong performance with a low loss (0.00040) and good test accuracy (0.87695).

The Manhattan method has a higher loss (0.00122) compared to the other two methods, indicating that it incurs more errors during training. Its test accuracy (0.86132) is lower than that of the MCESTA and Euclidean methods.

TABLE I. METRICS COMPARISON

Methods	Metrics Loss	Accuracy on test
Euclidean	0.00040	0.87695
Manhattan	0.00122	0.86132
MCESTA	0.00015	0.91438

In summary, this Table I allows you to compare the performance of different methods in a specific task. The choice of the most suitable method may depend on the specific requirements of your project, but based on these metrics, the "MCESTA" method appears to be the best-performing one.

VII. CONCLUSION

The conclusion drawn from this study effectively captures the key insights and breakthroughs in the realm of similarity learning. It emphasizes the critical importance of selecting an appropriate similarity metric, meticulously customized to align with the specific demands of the task and the peculiarities of the data involved. This strategic customization is vital for the optimal performance of machine learning models, especially in scenarios that necessitate precise measurements of data point similarities.The approach of using Siamese Network and MCESTA method boasts the lowest loss (0.00015), signifying its superior performance in minimizing errors during training, and a corresponding high test accuracy (0.91438), indicating its proficiency in making accurate predictions on unseen data. This underscores its effectiveness in optimizing the model's parameters.

Highlighting the cutting-edge performance of Siamese triplet networks within similarity learning, the study showcases these networks as exemplars of significant advancements in both architecture and methodological approaches within this sphere.

At the heart of the study's contributions is the unveiling of MCESTA, an innovative method poised to substantially elevate the domain of similarity learning. MCESTA's unique approach, which amalgamates multiple similarity functions each accorded with a task-specific weighting, presents a more adaptable and efficacious strategy for addressing a broad spectrum of challenges. This comprehensive approach not only facilitates a deeper and more nuanced application of similarity metrics but also opens up prospects for ongoing innovation and enhancement within machine learning tasks.

Ultimately, this study sets a solid foundation for subsequent research and practical applications of similarity learning, spotlighting MCESTA as a pioneering innovation. It advocates for a detailed and task-specific consideration of similarity metrics, alongside introducing an architecture that markedly propels the field forward. This exploration heralds new paths for augmenting machine learning models and their utility across a vast array of domains, promising significant implications for future advancements

ACKNOWLEDGMENT

We would like to express our gratitude to all those who contributed to this study, especially the originator of the MCESTA model, Pr. Mohamedou El Ghtotb Cheikh Tourad. Additionally, we extend our thanks to the creators of the LFW face dataset for the outstanding work they have done.

REFERENCES

[1] Sarker, I. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*. **2**, 160 (2021)

[2] Chopra, S., Hadsell, R. & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. *2005 IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR'05)*. **1** pp. 539-546 (2005)

[3] Koch, G., Zemel, R., Salakhutdinov, R. & Others Siamese neural networks for one-shot image recognition. *ICML Deep Learning Workshop*. **2** (2015)

[4] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. & Pietikäinen, M. Deep learning for generic object detection: A survey. *International Journal Of Computer Vision*. **128** pp. 261-318 (2020)

[5] Bishop, C. Pattern Recognition and Machine Learning by Christopher M. Bishop. (Springer Science+ Business Media, LLC,2006)

[6] Tourad, M. & Abdali, A. An intelligent similarity model between generalized trapezoidal fuzzy numbers in large scale. *International Journal Of Fuzzy Logic And Intelligent Systems*. **18**, 303-315 (2018)

[7] Chicco, D. Siamese Neural Networks: An Overview. *Artificial Neural Networks*. pp. 73-94 (2021), doi

[8] Brian Kulis, Foundations and Trends® in Machine Learning *Metric Learning: A Survey*. Vol. 5: No. 4, pp. 287-364 (2013), http://dx.doi.org/10.1561/2200000019

[9] Manning, C. Prabhakar raghavan, and hinrich schutze. *Introduction To Information Retrieval.* (2008)

[10] Rahutomo, Faisal, Teruaki Kitasuka, and Masayoshi Aritsugi. ”Semantic cosine similarity.” The 7th international student conference on advanced science and technology ICAST. Vol. 4. No. 1. 2012.

[11] Ross, T. Fuzzy logic with engineering applications. (John Wiley and Sons,2009)

[12] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. Learning deep features for discriminative localization. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 2921-2929 (2016)

[13] Deng, C., Zhao, Z., Wang, Y., Zhang, Z. & Feng, Z. Graphzoom: A multi-level spectral approach for accurate and scalable graph embedding. *ArXiv Preprint ArXiv:1910.02370*. (2019)

[14] Vorontsov, Ilya E., Ivan V. Kulakovskiy, and Vsevolod J. Makeev. ”Jaccard index based similarity measure to compare transcription factor binding site models.” Algorithms for Molecular Biology 8.1 (2013): 1-11.

[15] Zhan, T., Song, B., Xu, Y., Wan, M., Wang, X., Yang, G. & Wu, Z. SSCNN-S: A spectral-spatial convolution neural network with Siamese architecture for change detection. *Remote Sensing*. **13**, 895 (2021)

[16] Melekhov, I., Kannala, J. & Rahtu, E. Siamese network features for image matching. *2016 23rd International Conference On Pattern Recognition (ICPR)*. pp. 378-383 (2016)

[17] Li, M., Chang, K., Bearce, B., Chang, C., Huang, A., Campbell, J., Brown, J., Singh, P., Hoebel, K., Erdoğmuş, D. & Others Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging. *NPJ Digital Medicine*. **3**, 48 (2020)

[18] Cheng, X., Zhang, L. & Zheng, Y. Deep similarity learning for multimodal medical images. *Computer Methods In Biomechanics And Biomedical Engineering: Imaging & Visualization*. **6**, 248-252 (2018)

[19] Dong, X. & Shen, J. Triplet loss in siamese network for object tracking. *Proceedings Of The European Conference On Computer Vision (ECCV)*. pp. 459-474 (2018)

[20] Trigueros, D., Meng, L. & Hartnett, M. Enhancing convolutional neural networks for face recognition with occlusion maps and batch triplet loss. *Image And Vision Computing*. **79** pp. 99-108 (2018)