Fuzzy ArcFace: Enhancing Face Recognition with Membership-Function-Integrated Angular Margin Loss

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Face recognition systems have significantly benefited from the introduction of the ArcFace loss function, which enhances the discrimination capability by focusing on angular relationships within facial feature vectors. However, ArcFace struggles with images positioned at or near class boundaries, such as those affected by occlusions, lighting, or complex facial expressions. This paper presents *FuzzyArcFace*, an enhanced loss function that integrates a fuzzy membership function into the traditional ArcFace framework. The fuzzy membership function dynamically adjusts the angular margin based on the certainty of class membership. This adjustment provides a flexible and nuanced mechanism for classifying boundary cases more effectively. Extensive experiments on the labeled faces in the wild (LFW), japanese female facial expression (JEFF), and celebrities in frontal-profile (CFP) datasets demonstrate that *FuzzyArcFace* consistently outperforms ArcFace, especially in challenging scenarios. Notably, *FuzzyArcFace* achieves the highest accuracy on JEFF and CFP datasets by allowing more fluctuation in margin to accommodate nuanced images. These results highlight *FuzzyArcFace*'s potential to enhance face recognition accuracy and robustness across diverse scenarios, making it a promising tool for applications requiring high precision in face verification.**

***Index Terms*—ArcFace, fuzzy logic, membership function, deep convolutional neural network, face recognition, histopathology image classification**

I. Introduction

Deep convolutional neural networks (DCNNs) have substantially influenced advancements in face recognition technology [1]–[3]. DCNNs leverage multiple layers of convolutional filters to automatically learn hierarchical representations of data, making them particularly effective for image classification and recognition tasks. Among the pivotal contributions to this field is the ArcFace loss function [4], which has markedly elevated accuracy levels by introducing an additive angular margin to enhance the discriminative power of learned features. However, the ArcFace algorithm has limitations, particularly when classifying images positioned at or near the decision boundaries of classes—scenarios that are quite prevalent in real-life datasets. This paper introduces an innovative enhancement to address these limitations: the so-called Fuzzy ArcFace loss function, which is an extension of ArcFace. The Fuzzy ArcFace loss function incorporates a novel fuzzy membership function $\mu(\theta_{y_i})$, based on the cosine distance and parameterized by a scalar τ. This function adjusts the angular margin dynamically, depending on the certainty of the class membership. The primary goal of Fuzzy ArcFace is to refine the network's ability to handle boundary cases by providing a flexible mechanism that adapts classification strength based on the proximity of data points to class centers in the feature space. This enhancement not only inherits the robustness of the original ArcFace loss but also provides a flexible and nuanced mechanism, resulting in improved accuracy and robustness in face recognition, particularly in scenarios with high ambiguity or overlapping class boundaries.

Our approach's ingenuity lies in its ability to modulate the margin multiplier in a context-sensitive manner, applying a variable degree of membership to input images. This mechanism is designed to handle the inherent ambiguities of facial images, permitting a more granular and adaptive classification. By implementing this approach, we aim to extend the network's discernment capabilities, especially in ambiguous cases that traditional methods might otherwise misconstrue.

Our paper lays out a comprehensive exploration of the Fuzzy ArcFace loss function. Following this introduction, section II delves into the theoretical basis of this function, providing a detailed mathematical exposition of incorporating the fuzzy membership function into the facial recognition framework. In section III, we outline the specific implementation details, and section IV describes the experimental setup and dataset used to test our hypotheses. Then, section V presents the results of our experiments, offering a discussion on their implications. Finally, in section VI, we conclude by summarizing our findings and suggesting potential pathways for future research. This structured approach is adopted to unfold our methodology meticulously and explicate the broader impact our findings could have on face recognition technology and medical imaging analysis, specifically in domains where class boundaries are not distinctly defined.

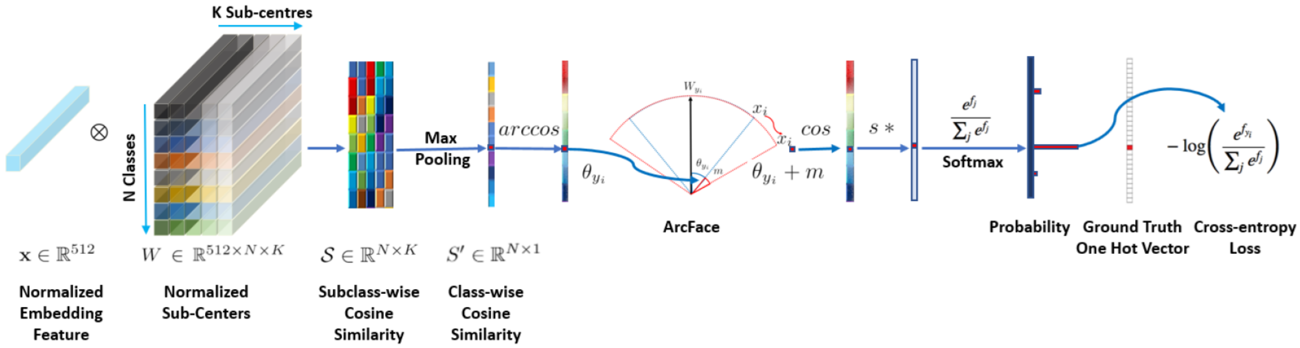

Fig. 1. End to end Arcface architecture [4]

II. Background and Related Work

The evolution of loss functions in face recognition technology has been integral to enhancing model accuracy. Over the years, several significant methodologies have been developed, each contributing uniquely to the field. This section provides a chronological overview of these advancements and their impact on face recognition systems.

The introduction of the Softmax loss function [5], [6] marked a pivotal moment in the development of face recognition models. Softmax loss is widely used in classification tasks, including face recognition, due to its simplicity and effectiveness in optimizing the separation of classes in high-dimensional space. However, it lacks sufficient margin between classes, leading to potential misclassifications in more complex scenarios.

To address the limitations of Softmax, SphereFace [7], [8] was introduced. SphereFace incorporates an angular margin, transforming the feature space into a hypersphere. This method enhances the discriminative power by enforcing a margin between different classes, thus improving the robustness of the face recognition model. SphereFace significantly improved the angular separation between classes, leading to better performance in distinguishing subtle differences in facial features.

Following SphereFace, CosFace [9] was developed to further enhance the margin-based approach. CosFace introduces a cosine margin penalty, which directly optimizes the cosine similarity between feature vectors and class centers. This method simplifies the optimization process and has demonstrated superior performance in various face recognition benchmarks by achieving higher inter-class separability and intra-class compactness.

Building upon these advancements, ArcFace [4] introduced an additive angular margin loss. ArcFace enhances the discriminative power of face recognition models by adding an angular margin to the arc-cosine function. This approach ensures that the learned features are more discriminative and have better generalization capabilities, particularly in recognizing faces under different poses and lighting conditions.

The integration of fuzzy logic into deep learning has further refined face recognition methodologies [10], [11]. The Fuzzy ArcFace loss function is an enhancement of ArcFace, incorporating a fuzzy membership function $\mu(\theta_{y_i})$, which dynamically adjusts the angular margin based on the certainty of class membership. This method provides a flexible and nuanced mechanism that adapts classification strength dynamically, improving the network's ability to handle boundary cases and enhancing overall robustness and accuracy.

A comparative analysis of these loss functions is presented in Table I, highlighting their key contributions and performance improvements.

This section provides a comprehensive background on the role of DCNNs in facial recognition, focusing on the ArcFace loss function and the integration of fuzzy logic into deep learning. The comparative table highlights the key contributions and performance improvements of each loss function, illustrating the advancements and their impact on face recognition technology.

A. DCNNs in Facial Recognition and the Role of ArcFace

DCNNs, such as ResNet [3], VGG [12], and Inception [13], are crucial in extracting complex features from facial images, significantly improving facial recognition accuracy. Leveraging these DCNN architectures, ArcFace specifically enhances the discrimination capability by focusing on the angular relationships within facial feature vectors, marking a significant advancement in facial recognition technology.

B. ArcFace Loss Function

The ArcFace loss function, applied to architectures like ResNet, projects facial embeddings onto a hyperspherical space, enhancing class separability through an angular margin m. The ArcFace loss is given by:

$$L = -\log\left(\frac{e^{s\cdot\cos(\theta_{y_i}+m)}}{e^{s\cdot\cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq i}^{C} e^{s\cdot\cos(\theta_j)}}\right) \tag{1}$$

where θ_{y_i} is the angle between the input feature vector and the true class weight vector, m is the margin added to enhance the decision boundary, and s scales the distribution of the outputs.

C. Visualization of ArcFace Mechanism

Figure 2 provides a detailed illustration of the ArcFace mechanism. It shows an input feature vector $\mathbf{x}$ and its relationship with the class centroids, represented by the weights.

TABLE I
COMPARATIVE ANALYSIS OF FACE RECOGNITION LOSS FUNCTIONS

Loss Function	Year	Key Contribution	Margin Type	Optimization Focus	Performance Improvement
Softmax [5], [6]	2015	Baseline classification loss	-	Class separation in high-dimensional space	Basic
SphereFace [7], [8]	2017	Angular margin on hypersphere	Angular	Enhanced angular separation	Moderate
CosFace [9]	2018	Cosine margin penalty	Cosine	Inter-class separability and intra-class compactness	High
ArcFace [4]	2019	Additive angular margin	Angular	Discriminative power and generalization	Very High
Fuzzy ArcFace (current)	2024	Dynamic angular margin with fuzzy logic	Angular	Boundary case handling and robustness	Superior

ArcFace computes the cosine distance between $\mathbf{x}$ and these weights, which is depicted by the angles θ_1 and θ_2. The margin m is introduced to enhance class separation, effectively increasing the angular distance between different classes. This visualization underscores how ArcFace modifies the feature space, improving discriminability and accuracy in facial recognition.

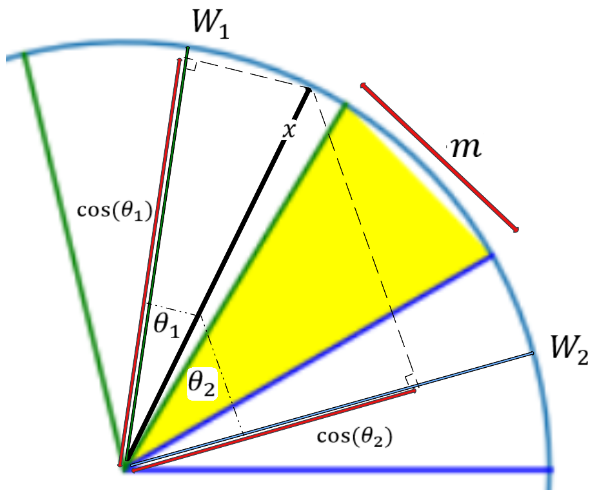

Fig. 2. Representation of the ArcFace mechanism showing the input vector $\mathbf{x}$, the angles θ_1 and θ_2, and the margin m, illustrating class separation.

D. Fuzzy Logic in Deep Learning

Recent approaches in deep learning have introduced fuzzy logic to handle the uncertainties in data more effectively [10], [11]. Fuzzy logic provides a way to encode ambiguous, noisy, or incomplete information and make decisions based on partial information. This is particularly useful in face recognition, where images may be at the class boundaries or exhibit variability due to factors like occlusion [14], lighting [15], or facial expressions [16].

E. Limitations and Research Gap

While the ArcFace margin m has significantly advanced facial recognition, its effectiveness diminishes in complex facial expressions, heavy occlusions, or significant age progression. These situations present high data variability and ambiguity, challenging the crisp margin of ArcFace. Existing models may struggle to differentiate between features altered by external factors versus inherent facial characteristics [17], [18]. This gap underscores the need for a more adaptable approach to handle real-world facial data's nuanced and dynamic nature [19]. An example of such challenges is illustrated in Figure 3 using images from MIT-Synthetic [20], MIT-Test [21], and JEFF [22] datasets showcasing the complexity of recognizing faces under varying conditions.

Fig. 3. Example of facial recognition challenges in the presence of heavy occlusions. JEFF (left), MIT-Synthetic (middle), and MIT-Test (right) datasets.

F. Contribution of This Work

Our contribution lies in integrating fuzzy logic with the ArcFace loss function to form the Fuzzy ArcFace loss function. This integration aims to address ArcFace's limitations in scenarios involving data variability and ambiguity, such as complex facial expressions, occlusions, or age progression [23]. The Fuzzy ArcFace loss function enhances the neural network's robustness and adaptability, promising improved performance in facial recognition tasks, especially in ambiguous scenarios.

III. THEORETICAL FOUNDATION OF FUZZY ARCFACE LOSS FUNCTION

The Fuzzy ArcFace loss function is a novel contribution to the repertoire of neural network loss functions specifically tailored for facial recognition tasks. It extends the ArcFace loss function by incorporating fuzzy logic, addressing inherent uncertainties and ambiguities in real-world data. See Figure 4. The fuzzy membership function is at the core of this extension, providing a dynamic, context-dependent degree of membership to each class. This section provides a detailed dissection of each component of the Fuzzy ArcFace loss function, highlighting our approach's theoretical motivations and mathematical implications.

The revised Fuzzy ArcFace loss function is formalized as follows:

$$L = -\log\left(\frac{e^{s\cdot\cos(\theta_{y_i}+m\cdot\mu(\theta_{y_i}))}}{\sum_{j=1}^{C} e^{s\cdot\cos(\theta_j)}}\right) \tag{2}$$

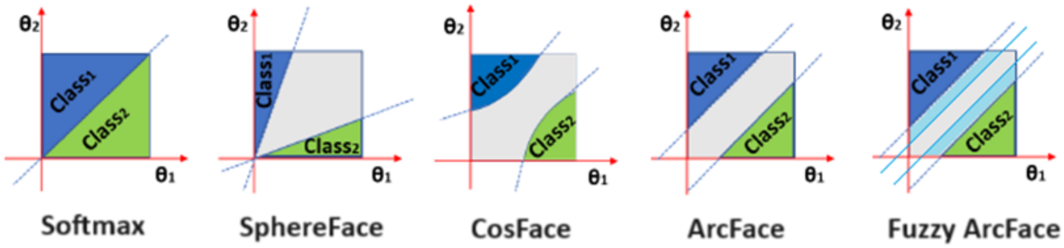

Fig. 4. Fuzzy Arcface defines variables boundaries to support a flexible inter-class separation.

In this formula:

- $\cos(\theta_{y_i} + m \cdot \mu(\theta_{y_i}))$: The cosine of the sum of the angle θ_{y_i} and the product of m and $\mu(\theta_{y_i})$ represents the angular margin modified by the fuzzy membership function for the true class y_i.
- s: A scaling factor that enhances the discriminative power by amplifying the range of the cosine function.
- m: A fixed angular margin in the original ArcFace algorithm is added to the angle between the feature vector and its corresponding class weight in the hyperspherical space.
- C: The number of classes in the original dataset.
- $\mu(\theta_{y_i})$: The fuzzy membership function that provides a value between 0 and 1, based on the absolute value of the cosine of the angle between the feature vector and the class center, and is parameterized by a scalar τ.

The fuzzy membership function $\mu(\theta_{y_i})$ is defined as follows, where $\mu(\theta_{y_i})$ dynamically adjusts m based on the degree of certainty in class membership:

$$\mu(\theta_{y_i}) = \begin{cases} |\cos(\theta_{y_i})|, & \text{if } |\cos(\theta_{y_i})| \geq \tau \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

This function ensures that when the feature vector is at a higher angular distance from the class center (lower cosine value), the margin m remains unchanged, preserving the enforcement of m for larger distances.

1) Role of m and Fuzzy Membership: In the Fuzzy ArcFace loss function, m represents the original margin factor from ArcFace, designating a fixed angular distance between classes. The novel aspect of our approach is the modification of this margin by the fuzzy membership function $\mu(\theta_{y_i})$, which dynamically adjusts m according to the degree of certainty in class membership [10], [11], [24]. The fuzzy membership function is computed based on the absolute value of the cosine similarity, providing a value between 0 and 1 [25]. This value reflects how closely the feature vector x aligns with the true class y_i, considering the absolute angle between them. By adapting m using the absolute value of cosine similarity, the margin becomes flexible and responsive to each data point's particular characteristics and uncertainties.

2) Visualization of the Fuzzy ArcFace Mechanism: Figure 5 provides a comprehensive visualization of the Fuzzy ArcFace mechanism, illustrating the nuanced differences from the standard ArcFace approach. Like ArcFace, it depicts an input feature vector $\mathbf{x}$ and its relationship with class centroids, represented by the weights. However, the key distinction lies in the dynamic adjustment of the angular margin m through the fuzzy membership function $\mu(y_i, x)$.

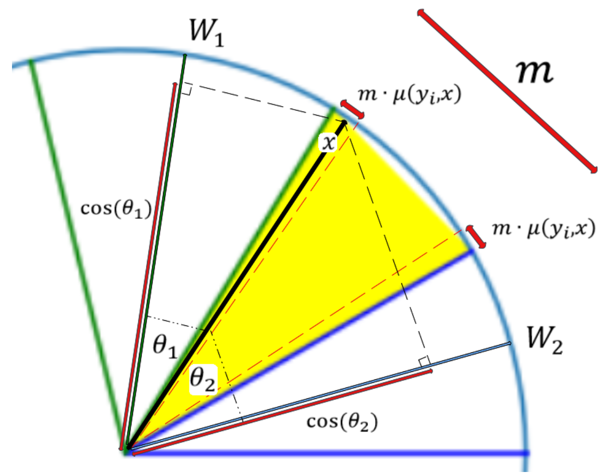

Fig. 5. 2D representation of Fuzzy Arcface loss function, showing the adaptive angular margin m influenced by the fuzzy membership function $\mu(y_i, x)$.

In standard ArcFace (as shown in Figure 2), the margin m is a fixed value, enhancing class separation by increasing the angular distance between different classes. This is achieved by computing the cosine distance between $\mathbf{x}$ and the class weights, represented by angles θ_1 and θ_2.

Conversely, in Fuzzy ArcFace, the margin m is adaptively modified based on the fuzzy membership value $\mu(y_i, x)$. This value, ranging between 0 and 1, determines the degree to which the feature vector x belongs to the true class y_i. A higher membership value indicates a closer alignment with the class, leading to a larger margin m, thereby increasing the angular separation. Conversely, a lower membership value suggests less certainty, reducing the margin and allowing the feature vector x to be closer to other classes. This adaptive nature of the margin in Fuzzy ArcFace, as visualized in Figure 5, allows for a more flexible and context-aware classification, accommodating the varying degrees of certainty and ambiguity in real-world data.

Overall, this novel approach in the Fuzzy ArcFace loss function offers a more sophisticated and adaptable model better suited for the complexities and uncertainties prevalent in facial recognition tasks.

IV. Implementation Details

This section elaborates on the implementation process of the Fuzzy ArcFace loss function within a deep learning framework. The process encompasses setting up the Python environment, defining the fuzzy membership function, detailing the *FuzzyArcFace* Loss class, and integrating it with a deep learning model. Below, we provide a detailed description and a pseudo code example to illustrate the algorithm used.

A. Environment Setup and Dataset Preparation

- Libraries such as PyTorch [26], Torchvision, and DataLoader [26] are imported for foundational deep learning functionalities.
- The labeled faces in the wild (LFW) [27] dataset, apanese female facial expression (JEFF) dataset [28], and celebrities in frontal-profile (CFP) [29] dataset are preprocessed using PyTorch's transforms and loaded using DataLoader.
- The datasets undergo resizing and normalization to 112 x 112 pixels, aligning with the input requirements of pre-trained models.

B. FuzzyArcFace Class

- The `FuzzyArcFace` class, extending `nn.Module` [26], initializes parameters like input features, output features, scaling factor s = 60, margin m = 0.5, and class weights.
- The forward method involves normalizing input features, calculating cosine similarity, applying the fuzzy membership function, and dynamically adjusting the margin m.

1) Pseudo Code for FuzzyArcFace Algorithm: The following algorithm outlines the forward pass of the *FuzzyArcFace* during training, detailing the process of normalizing input features, calculating cosine similarity, and dynamically adjusting the margin using a fuzzy membership function. This algorithm is designed to enhance the robustness and accuracy of face recognition models by incorporating a flexible margin adjustment mechanism based on the confidence of class membership.

Algorithm 1 *FuzzyArcFace* Forward Pass

Require: input, label, weight, s, m, τ
Ensure: output
Normalize input features and weights
$cosine \leftarrow$ cosine similarity between input and weight
$fuzzy\_membership \leftarrow cosine$
$mask \leftarrow (fuzzy\_membership > 0) \wedge (fuzzy\_membership \geq \tau) \wedge (fuzzy\_membership \leq 1)$
$fuzzy\_membership \leftarrow$ apply mask to fuzzy_membership
$m\_adjusted \leftarrow m \times fuzzy\_membership$
$sine \leftarrow \sqrt{1.0 - cosine^2}$
$cos\_m\_adjusted \leftarrow \cos(m\_adjusted)$
$sin\_m\_adjusted \leftarrow \sin(m\_adjusted)$
$\phi \leftarrow cosine \times cos\_m\_adjusted - sine \times sin\_m\_adjusted$
if easy_margin **then**
  $\phi \leftarrow$ where$(cosine > 0, \phi, cosine)$
else
  $\phi \leftarrow$ where$(cosine > \cos(\pi - m_adjusted), \phi, cosine - \sin(\pi - m_adjusted) \times m_adjusted)$
end if
$one\_hot \leftarrow$ one hot encoding of labels
$output \leftarrow (one\_hot \times \phi) + ((1.0 - one\_hot) \times cosine)$
$output \leftarrow output \times s$ **return** $output$

2) Explanation of Key Concepts:

a) Easy Margin: The `easy_margin` parameter, from the standard ArcFace implementation, allows for a more lenient decision boundary. It ensures that the margin m is applied only when the cosine similarity is positive, thus preventing overly strict boundaries and improving stability and convergence during training. Easy Margin involves a binary adjustment: applying the margin or not based on whether the cosine similarity is positive.

b) One-Hot Encoding: `one_hot` refers to the one-hot encoding of class labels, a standard practice in classification tasks. It ensures that the margin adjustment (ϕ) is applied only to the correct class while keeping the original cosine similarities for the other classes. This selective application helps in precise and effective class separation.

c) Difference Between Easy Margin and Fuzzy Membership Function: While both `easy_margin` and the fuzzy membership function aim to improve model performance, they address different aspects: - `Easy Margin`: Ensures stability by relaxing the decision boundary for near-boundary examples. It involves a binary adjustment (applying the margin or not based on a condition) to prevent the model from being overly strict. - `Fuzzy Membership Function`: Provides a continuous and adaptive adjustment to the margin based on cosine similarity, enhancing the model's ability to capture nuanced differences and handle ambiguity. The `fuzzy_membership` makes the margin m variable, controlled by τ, and dynamically adjusts based on the certainty of class membership, making it more flexible compared to the binary adjustment of `easy_margin`.

C. Model Setup and Training

- A pre-trained iResNet100 model [3] is adapted, modifying its final layer to incorporate the *FuzzyArcFace* loss function.
- The training process utilizes an stochastic gradient descent (SGD) optimizer [30] and a focal loss function [31].

The training environment setup involves using four NVIDIA DGX A100 GPUs for training on the LFW dataset, consisting of 5749 classes, taking approximately 8 hours.

D. iResNet100 Architecture

- iResNet100, chosen for its deep feature extraction capabilities, does not include bottleneck structures. Instead,

it relies on basic blocks (IBasicBlock) [3] for layer construction.
- The architecture of iResNet100 is designed for effective facial feature extraction, crucial for the *FuzzyArcFace* mechanism.

E. Training Loop

- The training loop involves 100 epochs of embedding generation, loss computation using the *FuzzyArcFace* loss, and backpropagation for optimization [32].

Project Repository: To facilitate access to code used in this work, a GitHub repository is accessible (the link to the repository will be added after the review process) to enable the reproducibility and validation of findings. This repository is a central hub for the code and data associated with this study.

V. Experimental Results

Our experiments evaluated the performance of the *FuzzyArcFace* loss function under various configurations of the membership function threshold τ. These tests were conducted using the standard LFW, the JEFF, and CFP datasets. It is important to note that the LFW dataset was included in the training data, whereas the JEFF and CFP datasets were not.

Specifically, we analyzed 10 different classes of faces from the JEFF database, taking one face randomly from each class, obfuscating those images between 5% to 30%, and then calculating the verification accuracy. Additionally, we analyzed 500 different classes of faces from the CFP dataset, applying the same obfuscation procedure to assess the performance on a broader set of identities. Embeddings for each image were extracted using the *FuzzyArcFace* models with different configurations ($\tau = 0.1, 0.5, 0.9$) and the standard ArcFace model. Cosine similarities between embeddings were computed to measure verification accuracy. Similarities were calculated for pairs of images within the same class to evaluate intra-class variance handling.

A. Face Verification with FuzzyArcFace

FuzzyArcFace demonstrated remarkable performance improvements over the standard ArcFace model, particularly at varying thresholds of τ. The adaptive nature of the fuzzy membership function has proven effective in enhancing face verification accuracy, especially in handling images at class boundaries and dealing with previously unseen images. The high precision achieved at low values of τ (e.g., 0.1) can be intuitively explained as allowing more room for m (crisp boundaries in ArcFace) to fluctuate and accommodate more nuanced images.

B. Benchmarking and Evaluation

We benchmarked the performance of *FuzzyArcFace* against ArcFace across different scenarios, measuring the accuracy of face verification as τ was adjusted. The results indicate that *FuzzyArcFace* excels in higher uncertainty and ambiguity scenarios, providing robust performance even when faced with challenging facial recognition tasks.

To illustrate the robustness of *FuzzyArcFace* in handling obfuscated images, Figure 6 shows an example from the JEFF database with varying levels of obfuscation, ranging from 5% to 30%.

Fig. 6. Example of obfuscated faces from the JEFF database, with obfuscation levels ranging from 5% to 30%.

Similarly, Figure 7 provides an example from the CFP database, showcasing the original image and the obfuscated images with obfuscation levels ranging from 5% to 30%. This dataset consists of 500 classes and demonstrates the application of *FuzzyArcFace* on a broader set of identities.

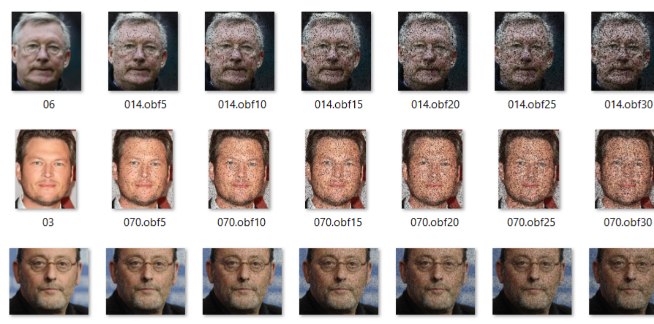

Fig. 7. Example of obfuscated faces from the CFP database, with obfuscation levels ranging from 5% to 30%.

TABLE II
Average face verification precision of ArcFace and *FuzzyArcFace* under different configurations on LFW and JEFF

Configuration	LFW	JEFF (5%-30%) obfuscation
ArcFace	98.8%	82.40%
FuzzyArcFace ($\tau = 0.1$)	52.21%	**85.75%**
FuzzyArcFace ($\tau = 0.5$)	74.6%	83.20%
FuzzyArcFace ($\tau = 0.9$)	**99.1%**	82.82%

The experiments with the CFP dataset, as summarized in Table III, further validate these findings. The trends observed in the JEFF dataset are consistent with those in the CFP dataset, reinforcing the robustness of *FuzzyArcFace* under varying conditions.

TABLE III
AVERAGE FACE VERIFICATION PRECISION OF ARCFACE AND *FuzzyArcFace* UNDER DIFFERENT CONFIGURATIONS ON CFP

Configuration	CFP (5%-30%) obfuscation
ArcFace	83.25%
FuzzyArcFace ($\tau = 0.1$)	**86.50%**
FuzzyArcFace ($\tau = 0.5$)	84.10%
FuzzyArcFace ($\tau = 0.9$)	83.80%

Overall, these experiments validate that *FuzzyArcFace* is particularly effective at high thresholds, where it substantially enhances verification accuracy, demonstrating its advantage over traditional methods in managing more complex, real-world scenarios. It is important to note that the trend of verification accuracy is opposite in the LFW dataset compared to the JEFF and CFP datasets. This discrepancy is because the *FuzzyArcFace* model was trained with the LFW dataset, meaning the inferencing was done with known data, whereas the JEFF and CFP datasets represent unseen data. In real-world applications, the performance with unseen data, as demonstrated with the JEFF and CFP datasets, is crucial.

VI. Conclusion and Future Work

The experimental results presented in this paper demonstrate the effectiveness of the *FuzzyArcFace* loss function in enhancing face recognition performance. This section discusses the advantages and disadvantages of the model based on these evaluations.

A. Advantages

a) Improved Accuracy: FuzzyArcFace consistently outperforms the standard ArcFace model in various challenging scenarios. The adaptive nature of the fuzzy membership function allows the model to handle boundary cases more effectively. This is particularly evident in the JEFF and CFP datasets, where *FuzzyArcFace* achieved superior accuracy, especially at lower values of τ. By allowing the margin m to fluctuate, the model can better accommodate nuanced images, resulting in higher precision in face verification tasks.

b) Versatility Across Datasets: FuzzyArcFace demonstrates strong performance across both seen (LFW) and unseen (JEFF and CFP) datasets. This indicates its potential applicability in various real-world settings where the model may encounter a wide range of facial images. The consistent improvement over ArcFace in different datasets highlights the generalizability of the *FuzzyArcFace* approach.

c) Handling of Obfuscated Images: The experiments with obfuscated images from the JEFF and CFP datasets show that *FuzzyArcFace* can maintain high verification accuracy even when faces are partially obscured. This ability to handle incomplete or noisy data is crucial for practical applications, such as surveillance and identity verification, where perfect image quality cannot always be guaranteed.

B. Disadvantages

a) Increased Computational Complexity: Integrating the fuzzy membership function introduces additional computational overhead during training compared to the standard ArcFace model. Calculating the membership values and dynamically adjusting the margin for each input increases the complexity of the training and inference processes. This could potentially slow down the model, especially when deployed in real-time applications.

b) Parameter Sensitivity: The performance of *FuzzyArcFace* is sensitive to the choice of the threshold parameter τ. Selecting the optimal value for τ is critical for achieving the best results, and this may require extensive hyperparameter tuning. Inappropriate values of τ can lead to suboptimal performance, making the model less robust to varying conditions.

c) Training Instability: While the fuzzy membership function enhances flexibility, it can also introduce instability during training if not properly managed. The dynamic adjustment of the margin can lead to fluctuations in the loss landscape, potentially making the optimization process more challenging. Ensuring stable and consistent training requires careful initialization and monitoring of the training dynamics.

C. Future Directions

Future work could explore several avenues to address the disadvantages and further enhance the *FuzzyArcFace* model: - **Optimizing Computational Efficiency**: Developing more efficient algorithms for calculating the fuzzy membership function and adjusting the margin could reduce the computational overhead. - **Automated Parameter Tuning**: Implementing automated methods for hyperparameter tuning, such as Bayesian optimization, could help in finding the optimal τ values more effectively. - **Stability Enhancements**: Investigating techniques to stabilize training, such as advanced initialization methods or adaptive learning rate schedules, could mitigate the instability introduced by the dynamic margin adjustment.

Overall, the *FuzzyArcFace* loss function represents a significant advancement in face recognition technology, offering improved accuracy, robustness, and versatility. Addressing the identified disadvantages will further enhance its applicability and performance in real-world scenarios.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[4] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” *arXiv preprint arXiv:1801.07698*, 2019.
[5] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *British Machine Vision Conference (BMVC)*, 2015, pp. 41.1–41.12.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[7] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220.

[8] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *International Conference on Machine Learning (ICML)*, 2016, pp. 507–516.

[9] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[10] L. A. Zadeh, “Fuzzy sets,” *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[11] B. Kosko, *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Prentice Hall, 1992.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Occlusion-aware face detection using cnn-based single shot multi-box detector,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3420–3429.

[15] X. Chen, A. Dantcheva, and J.-L. Dugelay, “Lighting robust face recognition using a novel face relighting approach,” *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 969–981, 2017.

[16] H. Li, Z. Lin, X. Shen, and Y. Zhang, “Facial expression recognition with a multi-task cascaded convolutional network,” *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5825–5835, 2019.

[17] L. Best-Rowden, “Facial recognition: Challenges and opportunities,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 2, pp. 70–85, 2019.

[18] L. Georgopoulos and D. K. Iakovidis, “Convolutional neural networks for facial expression recognition: A review,” *Pattern Recognition Letters*, vol. 143, pp. 66–72, 2021.

[19] Y. Wang, X. Li, Z. Li, and B. Smith, “Adaptable deep learning models for facial recognition under varying conditions,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 217–221.

[20] M. D. to AI Lab, “Synthetic data vault (sdv),” 2020, a comprehensive ecosystem for generating and using synthetic data for machine learning applications. [Online]. Available: https://github.com/sdv-dev/SDV

[21] MIT Computer Science and Artificial Intelligence Laboratory, “Mit-cbcl-synthetic and mit-cbcl-test,” http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html, accessed: 2023-06-15.

[22] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 1998, pp. 200–205.

[23] D. Ramanan, S. Baker, and S. Kakade, “Face verification across age progression using generative models,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[24] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2004.

[25] G. J. Klir and B. Yuan, *Fuzzy set theory: foundations and applications*. Prentice Hall, 1995.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[27] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Technical Report 07-49, October 2007.

[28] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Jaffe: Japanese female facial expression database,” https://zenodo.org/record/3451524, 1998, accessed: 2023-06-15.

[29] B. Chen, C.-S. Chen, and W. H. Hsu, “Cross-age reference coding for age-invariant face recognition and retrieval,” in *European Conference on Computer Vision (ECCV)*, 2014, pp. 768–783.

[30] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.