State-of-the-Art Architectures and Methods for Face Recognition and Verification on the LFW Dataset

Andrew Turangan
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
turangan@usc.edu

Shuchan Zhou
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
zhoushuc@usc.edu

Yanbing Chen
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
yanbingc@usc.edu

Zhaoyi He
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
zhaoyihe@usc.edu

***Abstract*—With the widespread use of face recognition technology becoming more ubiquitous in the modern digital world, critical research in face recognition methods remains ongoing. Recent literature points to performance improvements resulting from certain architectural patterns, loss functions, and feature embeddings. This project investigates state-of-the-art deep learning models and techniques for face recognition using the Labeled Faces in the Wild (LFW) dataset. Initial experiments with Inceptionv3 and other advanced architectures, including Inception-ResNet, revealed limited performance improvements for this dataset. With a focus on optimizing loss functions, such as CosFace and ArcFace, the project achieved modest gains in accuracy. Efforts to address challenges associated with small datasets included exploring embedding-based verification and one-shot learning techniques, although a comprehensive performance comparison remains ongoing. Future work will expand experiments to larger datasets, incorporate liveness detection, and refine methods for small-sample scenarios.**

***Index Terms*—Face Recognition, Deep Learning, Labeled Faces in the Wild (LFW)**

I. Introduction

Face recognition is a key application of computer vision with widespread use in security, authentication, and social media. The Labeled Faces in the Wild (LFW) dataset, which contains more than 13,000 face images for unconstrained face verification, serves as a standard benchmark for evaluating face recognition systems under real-world conditions [1].

This project initially focused on replicating a study that proposed combining the InceptionV3 model with Q-learning to dynamically adjust data augmentation strategies, aiming to improve robustness under varying lighting conditions [2]. The paper demonstrated strong performance on the LFW dataset, presenting impressive robustness curves. However, challenges in replicating results led to a shift in focus.

During the replication process, we successfully implemented the InceptionV3 model, but the results of the Q-learning technique under different lighting conditions were not promising. Several challenges hindered accurate reproduction. First, the original paper lacked clear descriptions of how Q-learning was implemented and how its parameters were tuned, making it difficult to reproduce the results accurately. Additionally, it is possible that the authors used a large iteration set that exceeded the computational capacity of our devices, leading to results that could not be replicated effectively. Finally, our combined implementation of InceptionV3 and Q-learning performed significantly worse than the baseline model, especially under varying gamma correction conditions. As shown in Fig. 1, the validation accuracy peaked at approximately 80% with a gamma value of 1.8, whereas the baseline InceptionV3 model consistently achieved over 99% accuracy on the same dataset.

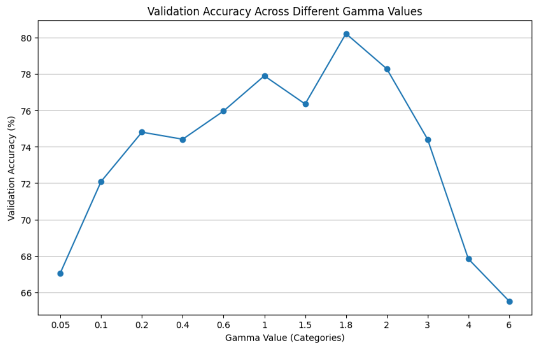

Fig. 1. Validation accuracy of InceptionV3 with Q-learning under different gamma values, peaking at approximately 80% at gamma = 1.8.

Given the challenges, we hypothesized that the paper may have omitted methodological details. Consequently, we shifted our goal to improve facial recognition performance on the LFW dataset by exploring alternative approaches, primarily focusing on model selection and optimization.

II. Key accomplishment/contributions

A. Implementing and Comparing Deep Learning Architectures

We implemented and evaluated InceptionV3, InceptionV4, and Inception-ResNet-V2, which are among the most advanced pre-trained convolutional neural network architectures. These models were analyzed for their effectiveness in face recognition tasks, with a particular focus on their performance under the constraints of the LFW dataset.

B. Optimizing Face Recognition with Advanced Loss Functions

We studied and compared Cross-Entropy, CosFace, and ArcFace loss functions to optimize feature embeddings for face recognition. Our experiments revealed that ArcFace excels in achieving higher accuracy on larger datasets, while CosFace strikes a balance between accuracy and training efficiency, particularly on medium-sized datasets.

C. Conducting Embedding-Based Pair Verification

We conducted an image verification experiment using the LFW pair-matching test dataset to evaluate the effectiveness of different feature embeddings. Six recent deep learning models were tested alongside the classical InceptionV3 baseline, providing insights into the suitability of various embeddings for similarity-based classification tasks.

D. Exploring Few-Shot Learning for Small Datasets

To address challenges posed by small datasets, we explored Few-shot learning techniques. Using embedding-based methods, we achieved promising results in scenarios with limited samples, showcasing the potential of these approaches to overcome data scarcity.

III. Background and dataset with related work

A. Labeled Faces in the Wild (LFW) Dataset

The Labeled Faces in the Wild (LFW) dataset, consisting of 13,233 images representing 5,749 individuals, serves as a standard benchmark for unconstrained face verification [1]. Among these, only 1,680 individuals have two or more images, highlighting the dataset's significant class imbalance and limited sample size per class. Its variability in pose, lighting, and occlusion makes it valuable for evaluating face recognition systems under real-world conditions, but these challenges also complicate model training and generalization.

B. Inception Models and Their Evolution

Inception models are advanced deep learning architectures designed for image classification tasks. They are based on the concept of ”Inception modules”, which split the input into parallel paths with filters of different sizes to capture spatial features at multiple scales [3]. This modular design makes Inception architecture highly tunable, meaning that there are a lot of possible changes to the number of filters in the various layers that do not affect the quality of the fully trained network.

InceptionV3 and InceptionV4 are highly similar, with V4 being more cost efficient and more architecturally simplified. Inception-ResNet-V2 is a combination of InceptionV4 with residual networks, which is more distinguished in memory saving and accuracy boosting at the cost of more computational time. Earlier Inception models including InceptionV3 used to be trained in a partitioned manner, where each replica was partitioned into multiple sub-networks to be able to fit the whole model in memory, whereas the most recent models, namely Inception-ResNet-V2 can be trained without partitioning the replicas [3].

C. Advanced Loss Functions for Face Recognition

Loss functions play a fundamental role in deep learning, directly influencing the quality of learned features for tasks like face recognition. Traditional Cross-Entropy loss is widely used due to its simplicity and efficiency, but it lacks the capability to enforce strong intra-class compactness and inter-class separability, which are critical for robust face recognition systems. Margin-based loss functions, such as CosFace and ArcFace, extend beyond Cross-Entropy by introducing explicit constraints to improve feature discriminability.

CosFace [4] operates in the cosine similarity space, where an additive margin m is subtracted to enlarge the decision boundary between classes. The formula is:

$$L_{lmc} = \frac{1}{N}\sum_{i} -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{j,i})}}, \tag{1}$$

ArcFace [5], in contrast, introduces an additive angular margin m in the angular space, optimizing the geodesic distance between classes on a normalized hypersphere. The formula is:

$$L_3 = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{N} e^{s\cos\theta_j}}. \tag{2}$$

These loss functions not only enhance class separability but also provide a more geometrically interpretable framework for feature learning. CosFace is computationally efficient due to the simplicity of cosine margin calculation, while ArcFace's angular margin optimization is particularly suited for scenarios with high inter-class variability.

D. Embedding-Based Verification and Few-Shot Learning

Embedding-based verification, or pair matching, is a common method in face recognition that measures the similarity between feature embeddings of image pairs to determine if they belong to the same individual. Models like FaceNet, GhostFaceNet, and SFace [6], trained with techniques such as triplet loss, produce high-quality embeddings that enhance intra-class compactness and inter-class separability.

In scenarios with limited data, such as the LFW dataset, few-shot learning provides an effective solution. This approach focuses on classifying query images based on their similarity to a small labeled support set, typically using a few labeled examples per class. By conducting similarity comparisons, few-shot learning reduces the reliance on extensive labeled data, making it particularly useful for addressing the challenges of small sample sizes and class imbalance in face recognition tasks [7].

IV. Approach

To improve the performance of face recognition models on the LFW dataset, we explored four distinct approaches, each addressing specific challenges posed by the dataset.

First, we explored and tested newer iterations of the Inception family, specifically InceptionV4 and Inception-ResNet-V2, to compare with the baseline InceptionV3 model. These models were chosen for their architectural improvements, with the expectation that they could enhance memory efficiency and accuracy when applied to the LFW dataset. By comparing these models, we aimed to determine whether these architectural advancements could translate into improved performance on LFW.

Second, we focused on exploring alternative loss functions. Building upon the baseline Cross-Entropy loss, we implemented and tested two margin-based loss functions, CosFace and ArcFace. These loss functions were chosen for their theoretical ability to enforce stronger class separability through margin constraints in cosine and angular spaces, respectively.

Third, we conducted an embedding-based image verification experiment to evaluate feature embeddings for similarity-based classification. This method, often referred to as pair matching, determines whether a pair of images belongs to the same individual by comparing their embeddings. By testing embeddings from various pre-trained models, we sought to identify the embeddings best suited for face verification tasks.

Finally, we addressed the challenges of small sample sizes through a few-shot learning framework. This approach leveraged pre-trained embeddings to classify query images based on their similarity to a small labeled support set. By categorizing individuals into groups with 2, 3, or 4 images and allocating support and query sets, we evaluated the feasibility of few-shot learning for handling class imbalance and data scarcity.

V. Experiments

A. Comparison of Model Architectures

N, the minimum number of pictures an image subject must have to be considered a class of their own, was set to $N = 6$, $N = 30$, $N = 70$, for testing models' performance under different data sample sizes. For instance, a smaller choice of N for an experiment yields more classes with less training instances and vice versa for a larger choice of N.

TensorFlow was the primary deep learning framework for experimentation. The default loss function was Categorical Cross Entropy. The default optimizer was Adam. All models were trained with 10 epochs. The choice of 10 epochs was made heuristically given limits in computational resources. We observed that at around 10 epochs, validation accuracy made marginal and variable improvements apart from training accuracy, suggesting that the choice of 10 epochs was optimal.

Given the test outcome, mixed results were observed (see Table I, Table II). While both InceptionV4 and Inception-ResNet-V2 boosted memory efficiency and allowed training without partitioning, accuracy gains were minimal. A possible explanation is that the models and comparison task was conducted over a large dataset, ImageNet, in the original paper [3], while the advantages of the both InceptionV4 and Inception-ResNet-V2 over InceptionV3 is not evident for a comparingly smaller dataset, LFW.

In the case of accuracy improvement, especially in smaller data samples, the performance of both InceptionV4 and Inception-ResNet-V2 proved even more inferior than InceptionV3. It is assumed that the InceptionV4 and Inception-ResNet models use more simplified CNN architectures [3] than InceptionV3, causing greater effect in small datasets during training.

TABLE I
Training Time per Epoch for Inception Models on LFW Dataset

Model	$\mathbf{N \geq 6}$	$\mathbf{N \geq 30}$	$\mathbf{N \geq 70}$
InceptionV3	00:09	00:03	00:02
InceptionV4	00:16	00:06	00:03
Inception-ResNet-V2	00:20	00:08	00:04

TABLE II
Validation Accuracy for Inception Models on LFW Dataset

Model	$\mathbf{N \geq 6}$	$\mathbf{N \geq 30}$	$\mathbf{N \geq 70}$
InceptionV3	81.57%	91.77%	99.61%
InceptionV4	73.18%	90.93%	98.45%
Inception-ResNet-V2	73.18%	91.98%	98.45%

Despite its strengths in memory efficiency and stability, Inception-ResNet's slower performance(Fig. 2) and limited accuracy gains(Fig. 3) on small datasets made it less ideal for the goal of improving accuracy on LFW dataset, especially for smaller N. Instead, we found InceptionV3 to be more effective for improving accuracy. Nevertheless, Inception-ResNet's flexibility and ability to handle large datasets without memory crashes is worth noticing and powerful in dealing with large datasets in the future.

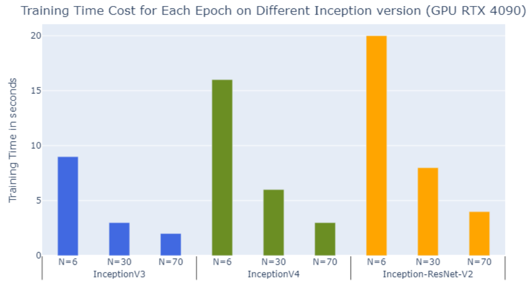

Fig. 2. Training Time for Inception Models. Embodied with Residual Networks, the Inception-ResNet-V2 model consumes the longest time in training, and InceptionV3 is the shortest in all N=6, N=30, N=70.

Given our conclusions backed by experiment results, the baseline InceptionV3 model was chosen as the most optimal pre-trained model among the Inception Family when tested on the LFW dataset.

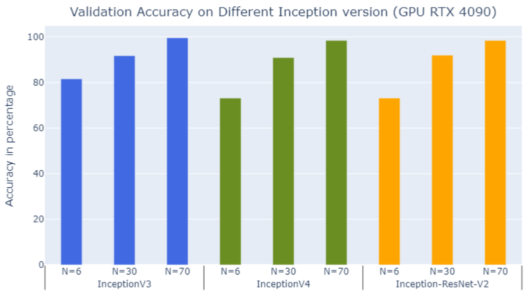

Fig. 3. Validation accuracy for Inception Models.The accuracy gain of InceptionV4 and Inception-ResNet-V2 is less ideal, especially for smaller N than the InceptionV3 model.

B. Comparison of Loss Functions

Unlike the previous setup, this experiment adopted variable epochs for different loss functions to accommodate their impact on convergence speed. Training was terminated once the loss stabilized, ensuring optimal convergence for each case. All other settings were kept consistent with the prior experiments to allow for direct comparisons across loss functions.

The results, presented in Table III and Table IV, highlight the trade-offs between training efficiency and accuracy among the tested loss functions. The Baseline (Cross-Entropy) loss function demonstrated the fastest training time, making it particularly suitable for scenarios requiring rapid deployment. Its accuracy at $N \geq 70$ proved sufficient, indicating that for classes with a larger number of samples, the baseline approach performs reliably without facing significant challenges.

ArcFace, on the other hand, showed clear advantages on datasets with fewer samples per class, achieving higher accuracy at $N \geq 6$ and $N \geq 30$. However, this improvement came at the cost of increased training time and the need for more epochs to achieve convergence, reflecting its inherent complexity and emphasis on angular margin optimization.

CosFace exhibited a different pattern, with limited improvement at $N \geq 6$ but achieving better accuracy at $N \geq 30$. Notably, this gain did not require additional training time, demonstrating CosFace's effectiveness in medium-sized datasets and its ability to balance performance and efficiency.

TABLE III
TRAINING TIME PER EPOCH FOR DIFFERENT LOSS FUNCTIONS

Loss Function	**N $\geq$ 6**	**N $\geq$ 30**	**N $\geq$ 70**
Baseline	00:09	00:03	00:02
ArcFace	00:16	00:08	00:01
CosFace	00:11	00:03	00:02

TABLE IV
VALIDATION ACCURACY FOR DIFFERENT LOSS FUNCTIONS

Loss Function	**N $\geq$ 6**	**N $\geq$ 30**	**N $\geq$ 70**
Baseline	81.57%	91.77%	99.61%
ArcFace	83.23%	95.15%	98.84%
CosFace	77.97%	93.88%	98.84%

Overall, these findings underscore the complementary strengths of the tested loss functions and their varying applicability across different dataset sizes and training objectives.

C. Embedding-Based Verification

We evaluated six recent deep learning model embeddings for face recognition, using the classical InceptionV3 embeddings as our baseline. The LFW dataset offers two configurations for training and testing: pairs and people. In our experiments, we adopted the pairs configuration, which included 1,100 matched pairs and 1,100 mismatched pairs in the training set, resulting in a total of 2,200 pairs. For testing, the dataset consisted of 500 matched pairs and 500 mismatched pairs, making a total of 1,000 pairs.

In the pair-matching test configuration, the task is to determine whether two images represent the same individual (match) or different individuals (mismatch). For each pair, we calculated similarity scores using the embeddings generated by the respective models and evaluated performance using Precision, Recall, F1 Score, and Accuracy. The results are summarized in Table V.

TABLE V
IMAGE VERIFICATION METRICS FOR DIFFERENT EMBEDDINGS (IN PERCENTAGES)

Model	**Precision**	**Recall**	**F1 Score**	**Accuracy**
InceptionV3	63.97%	63.20%	63.58%	63.80%
OpenFace	92.86%	2.69%	5.22%	50.70%
DeepID	83.33%	8.26%	15.04%	52.80%
Facenet	98.47%	66.32%	79.26%	82.40%
Facenet512	100.00%	51.03%	67.58%	75.20%
GhostFaceNet	99.13%	70.25%	82.22%	84.60%
SFace	99.43%	72.52%	83.87%	85.90%

The embeddings from Facenet, Facenet512, GhostFaceNet, and SFace showed significant improvements over the baseline InceptionV3 model. InceptionV3 achieved moderate performance with an F1 Score of 63.58% and an accuracy of 63.8%, reflecting its limitations for face verification tasks. OpenFace and DeepID performed poorly, with recall values of 2.69% and 8.26%, resulting in F1 Scores of 5.22% and 15.04%, respectively. Facenet achieved an accuracy of 82.4% and an F1 Score of 79.26%, while Facenet512 demonstrated perfect precision but lower recall (51.03%), yielding an F1 Score of 67.58%. GhostFaceNet and SFace delivered the best results, with SFace achieving the highest accuracy (85.9%) and F1 Score (83.87%), closely followed by GhostFaceNet with an accuracy of 84.6% and an F1 Score of 82.22%.

The superior performance of GhostFaceNet and SFace stems from their architectures and training methodologies. GhostFaceNet employs the Sub-Center ArcFace loss, while SFace utilizes a Self-supervised Angular Margin loss, both of which effectively enhance intra-class compactness and inter-class separability. Pre-trained on large-scale datasets, these models produce highly discriminative embeddings, reflected in their high precision and recall scores.

These results validate the effectiveness of embedding-based verification and emphasize the importance of optimized feature embeddings for face recognition. Based on their performance, Facenet, Facenet512, GhostFaceNet, and SFace were selected for further exploration in our few-shot learning experiments.

D. Few-Shot Learning for Small Sample Sizes

To address the challenges of limited sample datasets, we implemented a few-shot learning approach. Individuals with 2, 3, or 4 images were divided into three experimental settings, with one image per class allocated to the support set and the rest to the query set. This one-shot learning setup classified query images based on their similarity to the single support image for each class.

We utilized four pre-trained models: FaceNet512, FaceNet, GhostFaceNet, and SFace, selected for their strong performance in the embedding-based verification task. Using these models, we extracted embeddings for both support and query images and computed cosine similarity to classify each query image by assigning it to the class in the support set with the highest similarity score.

Table VI presents the details of the dataset splits for the three experimental groups. As the number of classes increased from 779 (n=2) to 1,257 (n=2 + n=3 + n=4), the query set size grew from 779 to 1,922 images. The results of classification accuracy for each model across the groups are summarized in Table VII.

TABLE VI
DATASET DETAILS FOR FEW-SHOT LEARNING EXPERIMENTS.

Group	**Number of Classes**	**Total Images**	**Support Set Size**	**Query Set Size**
$n = 2$	779	1558	779	779
$n = 2+n = 3$	1070	2431	1070	1361
$n = 2+n = 3+n = 4$	1257	3179	1257	1922

TABLE VII
CLASSIFICATION ACCURACY ACROSS MODELS AND SAMPLE GROUPS.

Group/Model	**FaceNet512**	**FaceNet**	**GhostFaceNet**	**SFace**
$n = 2$	72.91%	66.50%	65.34%	63.03%
$n = 2 + n = 3$	69.51%	62.75%	64.00%	60.25%
$n = 2 + n = 3 + n = 4$	67.85%	62.54%	63.42%	61.08%

In the smallest dataset ($n = 2$), FaceNet512 achieved the highest accuracy of 72.91%, followed by FaceNet (66. 50%), GhostFaceNet (65. 34%) and SFace (63. 03%). As the complexity of the data set increased ($n = 2+n = 3$), the precision decreased across all models, with FaceNet512 maintaining the best performance at 69.51%, followed by GhostFaceNet (64.00%), FaceNet (62.75%), and SFace (60.25%). In the largest data set ($n = 2 + n = 3 + n = 4$), FaceNet512 again outperformed others, achieving 67.85%, while GhostFaceNet, FaceNet, and SFace scored 63.42%, 62.54% and 61.08%, respectively.

These results highlight the robustness of FaceNet512 in handling small-sample datasets, consistently outperforming other models across varying dataset complexities. Its strong performance is attributed to its high precision in the embedding-based verification task, which ensures accurate similarity-based classification in one-shot learning tasks. The decline in accuracy across all models with increasing dataset complexity reflects the inherent challenges of distinguishing between embeddings as the number of classes and diversity of samples grow.

Overall, the few-shot learning approach proved effective for small-sample classification, demonstrating that pre-trained models like FaceNet512 can offer scalable and efficient solutions. Using embeddings and similarity metrics, this method avoids the need for fine-tuning and minimizes the risk of overfitting. Future enhancements could include refining embeddings with additional related data or developing specialized architectures, such as Siamese networks, to further improve performance in diverse scenarios.

VI. FUTURE PLANS AND CONCLUSION

This study explored a variety of approaches to improve face recognition performance in the LFW dataset, including testing advanced Inception models, optimizing feature embeddings with margin-based loss functions, conducting embedding-based verification, and implementing few-shot learning. Although InceptionV3 emerged as the most efficient model for smaller datasets, ArcFace demonstrated significant advantages in accuracy for larger datasets, and CosFace showed a balance of performance and efficiency. Embedding-based methods and few-shot learning frameworks proved effective in addressing challenges posed by class imbalance and limited samples. Together, these approaches offer valuable information and pave the way for further advancements in face recognition tasks.

Looking forward, this study can be extended in several directions. Exploring larger and more diverse datasets, potentially including images captured from real-world scenarios, could offer new challenges and opportunities to improve model generalization. Incorporating liveness detection—distinguishing between real individuals and photographs—would further improve the applicability of face recognition systems in security contexts. Analyzing specific image attributes, such as lighting or pose, that influence model performance, may provide deeper insights into embedding quality and robustness. For few-shot learning, future work could focus on moving beyond pre-trained FaceNet embeddings by integrating more related data and training Siamese or meta-learning-based networks. These extensions would allow for the development of more versatile and resilient face recognition systems capable of addressing a broader range of real-world challenges.

REFERENCES

[1] 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”, Technical Report, 2007, pp. 7-49.

[2] P. Wang, W. -H. Lin, K. -M. Chao and C. -C. Lo, ”A Face-Recognition Approach Using Deep Reinforcement Learning Approach for User Authentication,” 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE), Shanghai, China, 2017, pp. 183-188, doi: 10.1109/ICEBE.2017.36.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, AAAI, vol. 31, no. 1, Feb. 2017.

[4] H. Wang et al., ”CosFace: Large Margin Cosine Loss for Deep Face Recognition,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 5265-5274, doi: 10.1109/CVPR.2018.00552.

[5] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia and S. Zafeiriou, ”ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690-4699.

[6] F. Boutros, M. Huber, P. Siebke, T. Rieber and N. Damer, ”SFace: Privacy-friendly and Accurate Face Recognition using Synthetic Data,” 2022 IEEE International Joint Conference on Biometrics (IJCB), Abu Dhabi, United Arab Emirates, 2022, pp. 1-11.

[7] A. Chowdhury, M. Jiang, S. Chaudhuri, and C. Jermaine, ”Few-Shot Image Classification: Just Use a Library of Pre-Trained Feature Extractors and a Simple Classifier,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9445-9454, Oct. 2021.