ACT
AOI :10.100.234513.0231
13th international conference on Microelectronics, Circuits and Systems, Micro2026

Few-Shot Facial Recognition: A Practical Implementation

(Using Quality-Aware Prototypes and Multi-View Embeddings)

Deeksha Varshney, Dr. Vidushi Sharma, Dr. Arpit Bhardwaj

School of ICT
Gautam Buddha University
Corresponding Author: vidushi@gbu.ac.in

ABSTRACT

The problem that I am going to address in this paper is a real-life issue of how to create a face recognition system with only a couple of handfuls of images of individuals? Majority of studies presuppose that we possess hundreds of photos, yet that is not the case in the real-life deployments. We have developed a system that operates with 3 or less to 10 enrolment photos to identity. This is not us developing a new neural architecture but approximately to have the existing facial embeddings significantly more effective with thorough engineering. We present four techniques that use test-time augmentation to minimize noise, quality-based weighting to deal with bad photos, k-means clustering to make use of pose diversity, and a hybrid similarity measure that, as a combination of cosine and Euclidean distances. Comparison with standard ImageNet models (VGG-16, ResNet-50, etc.) demonstrates that our method achieves 98.7% accuracy, within 0.3 points of VGG-16 with an incredibly low number of samples. The system is real-time and uses standard hardware and an automatic threshold calibration technique which eliminates manual tuning. We've also addressed the very often ignored problem of cross-device enrolment. When a person signs in using the phone, but is not identified by a web camera. The full implementation is modular so you can replace more desirable embedding models as they are developed.

Keywords: few-shot learning, face recognition, prototype networks, quality assessment, test-time augmentation, domain adaptation

I. INTRODUCTION

The problem that I am going to address in this paper is a real-life issue of how to create a face recognition system with only a couple of handfuls of images of individuals? Majority of studies presuppose that we possess hundreds of photos, yet that is not the case in the real-life deployments. We have developed a system that operates with 3 or less to 10 enrolment photos to identity. This is not us developing a new neural architecture but approximately to have the existing facial embeddings significantly more effective with thorough engineering. We present four techniques that use test-time augmentation to minimize noise, quality-based weighting to deal with bad photos, k-means clustering to make use of pose diversity, and a hybrid similarity measure that, as a combination of cosine and Euclidean distances. Comparison with standard ImageNet models (VGG-16, ResNet-50, etc.) demonstrates that our method achieves 98.7% accuracy, within 0.3 points of VGG-16 with an incredibly low number of samples. The system is real-time and uses standard hardware and an automatic threshold calibration technique which eliminates manual tuning. We've also addressed the very often ignored problem of cross-device enrolment. When a person signs in using the phone, but is not identified by a web camera. The full implementation is modular so you can replace more desirable embedding models as they are developed.

II. RELATED WORK

A. Deep Face Recognition

The progression in face recognition over the last decade has been pretty remarkable. DeepFace [7] started in 2014 by showing that a deep CNN could match human performance on face verification. They trained on 4 million Facebook images and got 97.35% on LFW, which was a huge deal at the time. But their approach required massive per-class softmax—literally millions of output neurons.

FaceNet [8] changed the game by moving to metric learning with triplet loss. Instead of classification, they learned embeddings where similar faces are close and dissimilar faces are far apart. The triplet loss directly optimized for this that pick an anchor, a positive (same person), and a negative (different person), then push the negative away while pulling the positive closer. They hit 99.63% on LFW, which is basically human-level. The problem? Triplet mining is computationally expensive and tricky to get right. You need careful sampling strategies or you spend most of your time on easy triplets that don't teach the network anything.

ArcFace [9] solved this elegantly by adding an angular margin to the softmax loss. The math is a little bit involved. It forces the network to work harder by requiring extra separation between classes. Training is easier than triplets because you can use standard softmax infrastructure, but you get the same kind of discriminative embeddings. They reached 99.83% on LFW and 98.02% on the much harder IJB-C benchmark. The InsightFace

ISBN: 978-81-985770-0-9

library [10] packages these models nicely, which is what we're using here.

B. Few-Shot Learning

Few-shot learning involves the classification problem with a very small number of samples per class. The standard model is meta-learning: you can train the model, or many few shot tasks sampled some huge dataset, in such a way that it learns to learn with limited data [11]. The well-known ones are Matching Networks [12] and MAML [13]. They work well on Omniglot and mini-ImageNet benchmarks.

The Prototypical Networks [14] are more applicable and simple to what we are doing. The concept: calculate a prototype (means embedding) of each class given the support set, and classify queries based on nearest prototype. Simply embarrassingly easy, it happens to work. The point of it all is that in good embedding space, natural representatives of classes are their means.

But this is in itself interesting: face recognition actually does not require meta-learning. ArcFace embeddings are already trained to do generalization to new identities zero-shot- image of your face was not trained, and yet they still manage to differentiate between you and other people. It is not learning to adapt, then, but dealing with noisy samples of enrolment and pose change with little data. A different challenge, that is more about strong statistics than meta-learning.

C. Image Quality for Recognition

Not all face images are created equal. A blurry photo or one with bad lighting will produce worse embeddings, and if you're only working with 5-10 samples total, even one bad image can mess up your prototype. FIQA (face image quality assessment) has gotten a lot of research attention [15], [16].

There are two approaches: learned quality predictors trained to estimate recognition performance, and hand-crafted metrics using things like Laplacian variance for sharpness, mean intensity for brightness, and standard deviation for contrast. We use hand-crafted metrics because they're model-agnostic and don't require separate training. They're also surprisingly effective—in our testing, they correlate well with actual recognition accuracy.

III. METHODOLOGY

A. System Overview

There are six stages to the pipeline. The first thing we do is face recognition with MTCNN [11] and pose matching. Second, we determine quality scores and sieve out garbage. Third, we make the extraction of embeddings with the help of one of multiple backends (mainly ArcFace). Fourth, we construct prototypes based on the samples of enrolment- this is where the quality weighting and clustering take place. Fifth, the hybrid metric and adaptive thresholds are used to compute similarities and to make decisions. Sixth, in the case of video streams, we use temporal smoothing to minimize jitter.

All the components are modular. The embedding backend can be changed without any code modification, or it can be the clustering strategy, or quality metrics. This is important on maintenance and improvement in the future.

B. Face Detection and Alignment.

MTCNN is a three-step cascade network (P-Net, R-Net, O-Net) to refine face detections and produce five landmarks (both mouth corners and both eyes, and nose). We tune it a bit less strictly than default ([0.5, 0.6, 0.6] rather than [0.6, 0.7, 0.7]) since recall is more important here than precision: We want to miss a few false positives than to miss a true face at the enrolment time.

The alignment step is important. We approximate a similarity transform (uniform scale + translation + rotation) that can be applied to the identified landmarks in a 112 by 112 template to obtain fixed locations. This compensates the head tilt and camera angle and all faces are presented to the embedding network in the same orientation. Without alignment, the embeddings of the same person during two differing head poses would be different which pose a kill to few-shot settings.

C. Quality Assessment

To obtain a composite quality rating we use three components to come up with a score, q, of [0,1] of each face:

Sharpness: Laplacian operator variance of the grayscale image which is clamped at 500 and normalized. This is blur out of motion or defocusing.

Brightness: Offset of mean brightness with equal penalties of either too dark or too bright. Under-exposed pictures lose the shadowy areas; over-exposed pictures massacre highlights.

Contrast: SD of pixel values, capped at 60. Low contrast implies de-saturated photographs with minimal differentiating details.

The final score is $q = 0.55\times\text{sharpness} + 0.25\times\text{brightness} + 0.20\times\text{contrast}$. The weight is given on sharpness since it is the most prevalent quality issue in practice. We drop any frame whose $q < 0.28$, during enrolment. This is an empirically adjusted threshold, based on a validation set, though it is not very fussy- anything in the 0.25-0.30 regime is good.

ISBN: 978-81-985770-0-9

D. Embedding Extraction Test-Time Augmentation.

We support five embedding backends, but ArcFace (InsightFace buffalo_l) is the default. It is a ResNet-50 that has been trained on the angular margin loss on MS1MV2, and 512-dimensional L2-normalized embeddings come out. LFW accuracy is approximately 99.7 and zero-shot generalization is superb.

This is where test-time augmentation enters in. Embeddings of individual images are noisy--minimal positional or lighting variation results in image vectors of different feature values. We minimize this variance by deriving embeddings with seven augmented versions of each face: original, horizontal flip, flip + brightness reduction, + 15 rotation, -15 rotation, zoom-in + 110% and zoom-out + 90%. We then average the seven embeddings and normalize. The geometric and photometric differences cannot be complementary and the mean is more steady than any of the views.

This increases computational cost -7 times more forward passes/image, but it pays off. Empirically we achieve 1-3% accuracy gains and it can be achieved offline during enrolment when latency is not such a critical factor. To recognize in real-time we trade off speed versus accuracy with a 5-view schedule that is reduced.

E. Prototype Construction

Here the things become interesting. Naive prototype building simply averages all the embeddings of a class. All well when your samples of enrollment are clean and consistent, but in the real world, data is messy.

There are two things that we do; quality weighting and outlier rejection. We initial estimate a weighted mean, first by the quality scores, which serve as weights. Next we see the correlation of each sample to this mean (cosine similarity). Any sample more than 1.5 standard deviations below the mean similarity gets thrown out as an outlier. And lastly, we recalculate the weighted mean, but only with the surviving samples.

What is the rationale behind 1.5s, as opposed to 2s? Due to the small number of samples we do- 5-10 per person. Even a bad picture with a tiny sample can overshadow others so we should be more vigilant in filtering. We used 1.5 to and decided that 1.5sigma was better than the 2sigma and other thresholds we experimented with.

We also used k-means clustering with k=2 to identify identities that had at least 6 enrollment images. This procures us twice the prototypes per individual than otherwise, to assist in recording the various modes of appearance--perhaps of the face in front and at the sides, or under varying conditions of light. In recognition we compare ourselves with both prototypes and select the highest similarity. It is the differences of having two reference points per person. Classes less than 6 samples have a single prototype in order to prevent overfitting.

F. Similarity Computation and Decision Logic

For L2-normalized embeddings, cosine similarity is the standard choice. But we found that combining it with Euclidean distance works slightly better:

$$s\_ensemble = 0.75 \times \cos(q, p) + 0.25 \times (1 - ||q - p||_2 / 2)$$

The Euclidean term is normalized by dividing by 2 (max distance between unit vectors) and flipped to convert distance to similarity. The 75:25 weighting favors cosine but incorporates geometric information. The improvement is modest—about 1% in most configurations—but consistent.

We also implement an adaptive margin: before accepting a match, we require the best score to exceed the second-best by at least

$$MARGIN = \max(0.04, 0.10 \times (1 - second\_best))$$

This prevents misidentification when multiple people have similar embeddings. If the runner-up is close, we demand a bigger gap. If it's far away, a smaller gap suffices. Without this, lookalike individuals get confused.

G. Automatic Threshold Calibration

It is a pain to have to set thresholds manually. Which threshold do you use? It relying on enrolled people, how similar they are, the embedding model you are using, etc. So we automate it.

Once enrolled we do a leave-one-out cross-validation, which is as follows: derive a temporary prototype (without that sample) on that sample, run the similarity of that temporary prototype to that LOO prototype (genuine score) and to all other class prototypes (impostor scores). We take all the impostor sample, and reduce the threshold to the 99th percentile, with a 1% false acceptance. This threshold is saved and utilized in recognition.

It's not the best--LOO with 5 samples each person, does not provide you with great statistics--but it is much better than guessing. And it fits itself to your particular deployment.

H. Cross-Domain Domain Correction.

The following is an issue that is not referred to much in papers and is practical: people take up their phone camera to enrol but then a web-cam recognizes them later. Phone cameras capture at a higher resolution, another color balance, shallow depth of field- essentially another distribution of images. This will be in the form of instilling drift and all of a sudden, a person who has scored well has not kept up.

We process this in preprocessing enrolment images that have been captured by a mobile phone. The first step is to resize to the standard resolution (224 enough). Second, use CLAHE (Contrast-Limited Adaptive Histogram Equalization) clip limit 2.5 on L channel of LAB color

ISBN: 978-81-985770-0-9

space. This averages local contrast without distorting colors. Third, use unsharp masking (Gaussian 6=1.5, amount=0.8) to recover detail that JPEG compression tend to rub in. Then re-detect the face and crop out the normalized crop.

Does it work? Empirically, it decreases cross-device recognition failures, which are approximately 60-70 percent, though. It is not flawless, but good enough so that individuals are able to enroll using their phone and be recognized.

IV. EXPERIMENTAL RESULTS

A. Evaluation Setup

Our test is one-vs-rest binary classification, which resembles real deployment. Given a threshold and a specific enrolled individual, each image of the enrolled individual is a positive query with all other images a negative query. This will result in true positives, false negatives, true negatives, and false positives which are used to calculate sensitivity, specificity, and balanced accuracy.

We compare to five ImageNet-pretrained CNNs (VGG-16, ResNet-50, GoogLeNet, MobileNet, AlexNet) as fixed feature extractors on a comparison basis 5. We strip the heads of classification, pool the result of the final conv layer globally, and L2-normalize it. Next we classify using the one-nearest-neighbor method versus class prototypes. It is a just comparison between embedding without regard to complexities in classifiers.

The test is configured to support/ query split 50:50- one half of every individual images are used to construct the prototype, and the other half are test queries. To model realistic deployment constraints, the number of identities, which we present to our enrolment dataset, is 9, and those identities have 10 images taken at different distances and angles.

B. Main Results

Table I shows the results. VGG-16: 99.0% accuracy is understandable, such a large network (138M parameters) has 4096-dimensional features. ResNet-50 reaches 94.0% with the aid of residual connections.

TABLE I
COMPARISON WITH BASELINE ARCHITECTURES

Model	Specificity (%)	Sensitivity (%)	Accuracy (%)
VGG-16	98.65	99.45	99.00
ResNet-50	92.50	95.00	94.00
GoogLeNet	88.24	90.00	89.00
MobileNet	86.00	83.40	88.30
AlexNet	84.00	88.46	87.70
Proposed (ArcFace)	98.20	99.10	98.70

The superficial nets (GoogLeNet 89.0, MobileNet 88.3, AlexNet 87.7) find it harder to detect fine-grained discrimination of faces.

We have 98.7% accuracy on our system using ArcFace embeddings. That is merely 0.3 percentage points lower than VGG-16 which is extraordinary bearing in mind that we have 3-10 samples per individual as opposed to the huge support sets that such baselines normally consider. Specificity (98.2) and sensitivity (99.1) are both good-we are not sacrificing one on the other.

The main point is that face-specific training (ArcFace) outperforms generic ImageNet features by far in this task. All AlexNet as well as ResNet-50 are trained on object classification; they never witnessed a face during training. It is natural that ArcFace is more trained to discriminate on millions of faces with discriminative loss.

V. DISCUSSION

And what have we made here? It is an experimental few-shot face recognizer that can be used tomorrow. This rate is accurate enough to be used in most fields, 98.7% accuracy will allow you to get approximately 1-2 errors per 100 recognition attempts, which is really only acceptable in an access control or attendance tracking situation.

The point is that you do not have to use more complex few-shot algorithms or meta-learning. ArcFace embeddings already extrapolate to unseen faces in a zero-shot fashion--they have learned a good face representation space. The difficulty lies in processing noisy data of enrolments and poses variation with small samples. That is a largely an engineering issue and not a research issue.

Naturally, there are limitations. We have tested on a custom dataset instead of LFW or IJB-C hence cannot be easily compared with the published work. The enrolment population is minimal (9 identities), but that would be realistic in many deployments. We have not tried extreme situations such as heavy occlusion, such large angles of poses (>45 degree), or day light. Interesting follow up those would be.

The cross-domain correction across devices works yet not flawlessly- circa 60-70 cuts in failures. This will likely go even more in-depth with more advanced preprocessing or domain adaptation methods but eventually you find yourself contending with the realities of physics (different optics, sensors, ISPs).

One of the things that we avoided doing in detail: models themselves were not trained. All of them use pre-trained ArcFace embeddings. This makes the system accessible to people without ML expertise or GPU clusters. This can be executed on a laptop CPU, but needless to say, more quickly with a GPU.

ISBN: 978-81-985770-0-9

Looking ahead, the modular architecture implies you can easily add or replace the model of better embeddings as they emerge. ArcFace is superb yet there are on-going efforts on enhancing the loss functions, training processes, larger datasets, etc. When such improvements become a reality, you simply replace the backend - the other parts of the system remain the same.

VI. CONCLUSION

We have introduced the full few-shot face recognition system that is applicable in the real world situations, where only a limited enrolment information is available. Our use of face-specific ArcFace embeddings, quality-sensitive prototyping, test-time augmentation, multi-prototype clustering, and automatic threshold calibration, give a 98.7% accuracy using only 3-10 samples of their face at enrolment.

The key idea: this problem does not require advanced few-shot-based learning algorithms. You must be prudent about engineering going around a good pre-trained embedding model. Minimize variation, model capture diversity with multiple samples, minimize noise with augmentation and automatically calibrate thresholds. Do all that right and you will have a working system.

It is a real-time system that operates using a normal computer hardware system, has some functionalities such as correction of cross-device domain and temporal smoothing of video, displaying a GUI that is usable by non-experts. The code and models can be used by whoever wishes to deploy this or extend it.

Future studies might consider more extended-scale applications (hundreds or thousands of registered identities), more advanced domain adaptation procedures, anti-spoofing, and fairness testing by demographics. However, with small-to-medium deployments where there is a limited pool of enrolments, the above would work quite well.

REFERENCES

[1] M. Bansal and D. Sharma, “Facial Recognition System for Security Resolutions in Smart City,” *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 10, pp. 146–151, 2020.

[2] G. Rajeshkumar, V. Sankar, P. Shanmugam, and M. Ramasamy, “Smart Office Automation via Faster R-CNN Based Face Recognition and IoT,” *Measurement: Sensors*, vol. 27, pp. 100606–100613, 2023.

[3] A. Balamurugan, R. Arulmurugan, and S. Priya, “Surveillance Using Face Recognition in Smart Cities,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 6, pp. 2042–2045, 2020.

[4] A. M. Rashid, A. A. Yassin, A. A. Alkadhmawee, and A. J. Yassin, “Smart city security: Face-based image retrieval model using gray level co-occurrence matrix,” *Journal of Information and Communication Technology*, vol. 19, no. 3, pp. 437–458, 2020.

[5] M. Negi, “Face Recognition Web App using One Shot Learning with Siamese Networks,” *Medium*, 2022. [Online]. Available: https://medium.com/@manishnegi101/face-recognition-web-app-using-one-shot-learning-with-siamese-networks-6e918bd5823f. [Accessed: Mar. 13, 2026].

[6] S. Li, C. Yue, and H. Zhou, “Few-shot face recognition: Leveraging GAN for effective data augmentation,” *Electronics (MDPI journal)*, vol. 14, no. 10, p. 2003, May 2025. doi: 10.3390/electronics14102003.

[7] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701-1708.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815-823.

[9] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690-4699.

[10] J. Guo, J. Deng, A. Lattas, and S. Zafeiriou, "Sample and computation redistribution for efficient face detection," *arXiv preprint arXiv:2105.04714*, 2021.

[11] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and K. Kavukcuoglu, "Matching networks for one shot learning," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630-3638.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126-1135.

[13] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077-4087.

[14] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6902-6911.

ISBN: 978-81-985770-0-9

[15] P. Terhorst, J. N. Kolf, N. Damer, F. Kirchbuchner, and A. Kuijper, "SER-FIQ: Unsupervised estimation of face image quality based on stochastic embedding robustness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5651-5660.
[16] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499-1503, Oct. 2016.

Cite this article as:

Deeksha Varshney, Vidushi Sharma and et. al. "Few-Shot Facial Recognition: A Practical Implementation (Using Quality-Aware Prototypes and Multi-View Embeddings)", Proceedings of 13th international conference on Microelectronics, Circuits and Systems, Micro2026.
Displayed as online on 01st July 2026.

Link: http://actsoft.org/science/micro2026-pro/378-micro2026.pdf

ISBN: 978-81-985770-0-9