A Novel Approach for Facial Attendance:AttendXNet

Kawal Arora[1], Ankur Singh Bist[2], Roshan Prakash[3],Saksham Chaurasia[4]
Signy Advanced Technologies, India
Address : Level 39, One Canada Square, Canary Wharf, London E14 5AB
e-mail: kawal@signy.io[1], ankur@signy.io[2], roshan@signy.io[3], saksham@signy.io[4]

Author Notification
03 June 2020
Final Revised
04 June 2020
Published
05 June 2020

(APA style, Justify, Arial 10pt) Example:
To cite this document:
Arora, K., Bist, AS., Prakash, R., & Chaurasia, S. (2020). A Novel Approach for Facial Attendance:AttendXNet. *Aptisi Transactions On Technopreneurship (ATT)*, *2*(2), 104-111.

DOI : https://doi.org/10.34306/att.v2i2.86

Abstract

Recent advancements in the area of facial recognition and verification introduced the possibility of facial attendance for various use cases. In this paper we present a system named as AttendXNet. Our method uses the ResNet and Multi-layer feed forward network to achieve the state of art results. Extensive analysis of various deep learning and machine learning techniques is described. Face anti-spoofing is a major challenge in facial attendance. Extended-MobileNet is used to resolve the same issue. We also introduced the end to end pipeline to implement an attendance system for various use cases.

Keywords: Feature extraction, Support Vector Machine, Muli-layer Neural Network, Face Anti-spoofing, Faiss

1. Introduction

Attendance is very crucial part in any organization for maintaining the proper workflow. In schools and colleges, lecture attendance normally takes 10 minutes. If we extend our analysis for manual attendance time for a month or year, we found long hours are going into vain. Automatic attendance system is the need of hour where without human intervention attendance can be marked. Facial attendance involves the process of face detection and verification. There are various popular techniques for face detection and verification. DeepFace[2], DeepID2[3], DeepID3[4], FaceNet[5], Baidu[6], VGGface[7], light-CNN[8], Center Loss[9], L-softmax[10], Range Loss[11], L2-softmax[12], NormFace[13], CoCo Loss[14], vMF loss[15], Marginal loss [16], SphereFace[17], CCL[18],AMS-loss[19],Cosface[20],Arcface[21],DPSSD[22], Face recognition with alignment learning[23] and Ring loss[24] are some important methods in this domain.

The question arises: face verification involves certain steps; will it be same for different use cases? Will unique selection of deep learning and machine learning models be sufficient for various use cases? Our proposed model comes out after diving into various models. After comparing different models, ML classifiers and distance functions, we found results that were suitable for facial attendance.

We utilize the LFW [1] dataset for quantifying and comparing the performance of our proposed system on face based attendance. LFW dataset is created and maintained by researchers at the University of Massachusetts, Amherst. Original database contains four different sets of LFW images and also three different types of "aligned" images. In order to build dataset for face anti-spoofing, in different lightning conditions videos are recorded for ~30 seconds then replay the same video facing phone towards desktop after this process we get two videos for real and fake.

An overview of the rest of the paper is as follows: in section 2 we present related work in this area; section 3 defines proposed model architecture used; section 4 defines results and discussions. Finally in section 5 we present conclusion and future work.

2. Research Method

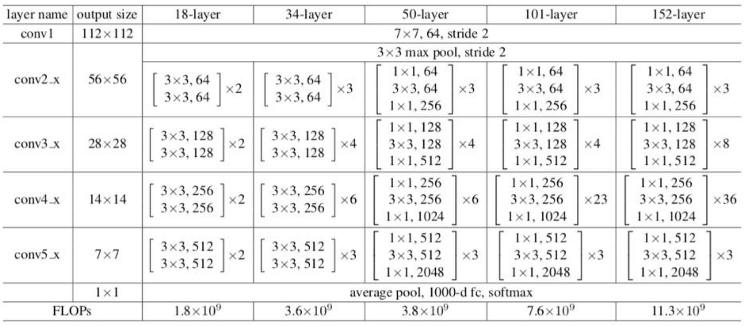
2.1 Related Work

Face recognition and verification is the domain where deep learning dominated as per recent literature. DeepFace method used Alexnet architecture with softmax as loss function with training dataset Facebook(4.4M,4K)and obtained accuracy 97.35%. DeepID2 method used Alexnet architecture with contrastive loss with training dataset CelebFaces+(0.2M,10K) and obtained accuracy 99.15%. DeepID3 used VGGNet-10 architecture used contrastive loss with training dataset CelebFaces+(0.2M,10K) and obtained accuracy 99.53%. Facenet method used GoogleNet-24 architecture, triplet loss on Google(500M,10M) dataset and obtained accuracy 99.63%. Baidu used CNN-9 architecture, triplet loss on dataset baidu(1.2M,18k) and obtained 99.77% accuracy. VGGface used VGGNet-16 architecture, triplet loss on dataset VGGface(2.6m,2.6K) and obtained 98.95% accuracy. Light-CNN used light CNN architecture, softmax loss on dataset MS-Celeb-1M(8.4M,100K) and obtained 98.8% accuracy. Center Loss used Lenet+-7 architecture, center loss on dataset CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K) and obtained 99.28% accuracy. L-softmax used VGGNet-18, L-softmax on CASIA-WebFace (0.49M,10K) dataset and obtained 98.71% accuracy. Range Loss used VGGNet-18 architecture, range loss on dataset MS-Celeb-1M, CASIA-WebFace(5M,100K) and obtained 99.52% accuracy.

L2-softmax used ResNet-101 architecture, L2 softmax on dataset MS-Celeb-1M (3.7M,58K) and obtained 99.78% accuracy. Normface used Resnet-28 architecture, contrastive loss on dataset(CASIA-WebFace (0.49M,10K)) and obtained 99.19% accuracy. CoColoss used loss function CoCo on dataset MS-Celeb-1M (3M,80K) and obtained 99.86% accuracy. vMF method used ResNet-27, vMF loss on dataset MS-Celeb-1M (4.6M,60K) and obtained 99.58% accuracy. Marginal Loss used ResNet-27, marginal loss on dataset (MS-Celeb-1M (4M,80K)) and obtained 99.48% accuracy. SphereFace used ResNet-64, A-softmax on dataset CASIA-WebFace (0.49M,10K) and obtained 99.42% accuracy. CCL used ResNet27, center invariant loss on training dataset (CASIA-WebFace (0.49M,10K)) and obtained 99.12% accuracy. AMS Loss ResNet-20, AMS Loss on training dataset (CASIA-WebFace (0.49M,10K)) and obtained 99.12% accuracy. Cosface used ResNet-64, cosface on training dataset CASIA-WebFace (0.49M,10K) and obtained 99.33% accuracy. ArcFace used ResNet-100, arcface loss on MS-Celeb-1M (3.8M,85K) dataset and obtained 99.83% accuracy. Ring Loss used ResNet-64, Ring Loss on MS-Celeb-1M (3.5M,31K) dataset and obtained 99.50% accuracy.

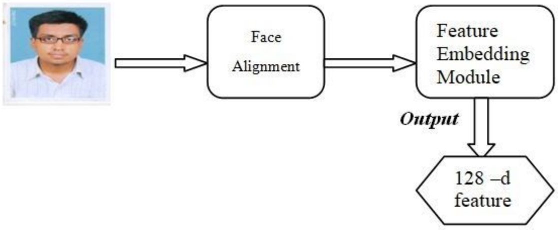
3. Findings

AttendXNet method involves the best fitted combination of Face Alignment, Face embedding extraction, Classification based on embedding. In the real time scenario, input image of the face may not be aligned so we used face alignment technique to obtain accurate results. Face alignment is the process of localization of predefined landmarks. We designed the python script to retrieve output facial coordinate such that eyes comes under horizontal axis. Second crucial step is extraction of feature vector from aligned face. We trained ResNet-34 on LFW dataset and obtained128-d feature vector.



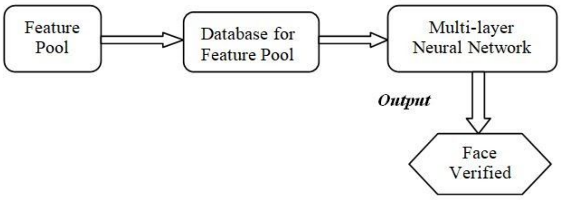
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Picture 1. Convolutional Kernels and size of outputs for ResNet 34 [25]



Picture 2. AttendeXNet face embedding

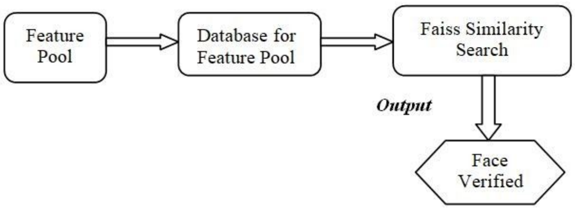
Picture 2 depicts the process of face embedding generation from aligned input of face. For attendance task we have to store face embeddings in database. There are two approaches used by us to register user. Registration through single image or by multiple images at different angles. Picture 3 explains the process, after extraction of feature vectors from input samples, database is maintained to store the features. AttendeXNetV1 used Multi-layer Neural Network to learn from database of feature pool.



Picture 3. AttendeXNetV1, version1 of module

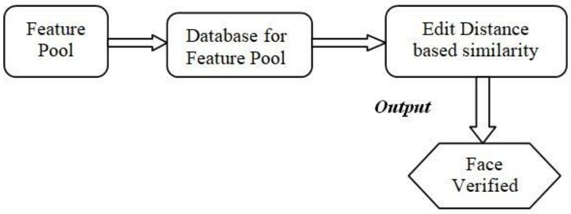
We used other machine learning models like Support vector machine, k-nearest neighbors (**KNN**) , Decision Tree and Naïve Bayes classifier. After testing on different IT workspaces and colleges we found Multi-layer Neural Network was most appropriate in face verification pipeline.

AttendeXNetV2 used Faiss similarity search [26] to learn from database of feature pool. Efficient similarity search is very crucial in context of our problem. Faiss is used in this module and then tested on different IT workspaces and colleges. We found Faiss similarity is fast as compared to multi-layer Neural Network. Accuracy of AttendeXNetV2 is comparable with AttendeXNetV1.



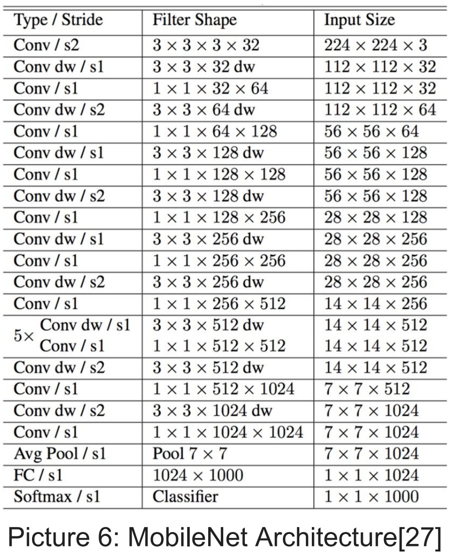
Picture 4. AttendeXNetV2, version2 of module

AttendeXNetV3 used edit distance based similarity [27] to learn from database of feature pool as shown in Picture 5. Edit distance is used in this module and then tested on different IT workspaces and colleges. We found edit distance is also working well and accurate as compared to manhattan distance. Accuracy of AttendeXNetV3 is less as compared with AttendeXNetV1and AttendeXNetV2.



PIcture 5: AttendeXNetV3, version3 of module

There are two major attacks in case of facial attendance, print attack and video attack. Intruder can spoof the attendance system using photo or video of someone else i.e. print attack and video attack respectively. There are various gesture based techniques to sort out this problem but these techniques don’t work well for video attacks. Secondly user has to perform certain actions like eye blink etc. Advantage of deep learning based technique is better results as well as good user experience. To implement the face Anti-spoofing, first of all we preprocessed the dataset using random resize python script. Figure6 shows the basic architecture of MobileNet. After different set of experiments we found current architecture of MobileNet is not suitable for getting standard results. We then added three layers in existing network and tested it over different cases. Accuracy of our Extended-MobileNet model is 98%. We used RTX 2080 for performing the experiments, it’s recommended to use current or better version of GPU for better results.



Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1 Conv / s1	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

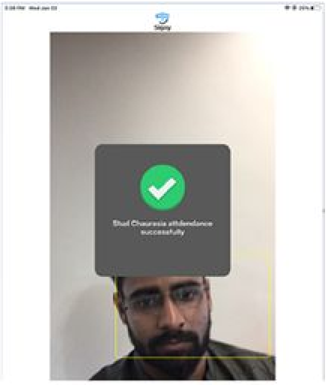
Picture 6: MobileNet Architecture[27]

4. Result and Discussion

Picture 7: Signy App for on boarding

To test proposed methods we developed Signy mobile app for on boarding the users. Figure7 shows first screen of app where admin has to start app once after that face detection module will detect face from live feed. Figure8 shows check-in process using Signy.

After detection of face, same input will move into two APIs deployed on server. Initially Extended-MobileNet API will extract features from input image and classify it as real and spoof. If someone is trying to spoof the system then we will store image for security purpose. In second step, AttendeXNet API will take face and extract 128-d vector then as per architecture of AttendeXNetV1, AttendeXNetV2 and AttendeXNetV3, face will be verified. For experimental analysis, we used all versions of AttendeXNet for facial attendance. After using three variants of AttendeXNet, we found multilayer Neural Network and Faiss as most appropriate method. We selected Faiss in our final module because it was accurate and relatively fast as compared to other techniques.



Shail Chaurasia attendance successfully

Picture 8: User Check-in using Signy

To test proposed methods for group attendance or classroom attendance, we developed IP camera based solution for taking inputs as shown in Figure9. In context of classroom scenario, input will be sent to AttendexNet API after fix interval of time.

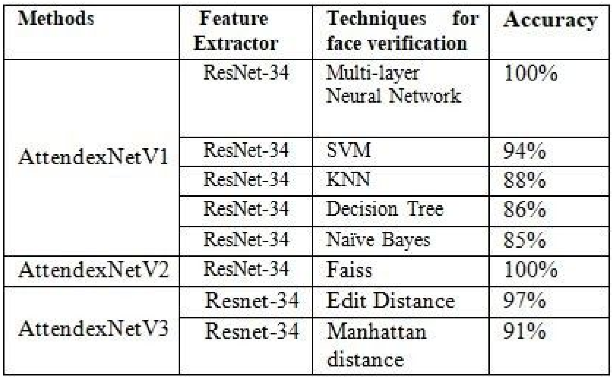
Picture 9: Group Attendance

AttendxNet API will take input as shown in Figure9 and returns confidence, face id, face count and status.

AttendXNet Output

```
{"IDs":

{"confi":[97.6,96.88,96.88,96.29,98.83,98.45,97.3,96.47,96.38],

"id":["5e283b1901dda02112a3f5a3",
"5e283b8501dda02112a3f5a5",
"5e283acf01dda02112a3f5a1",
"5e282a3601dda02112a3f584",
"5e283a7401dda02112a3f59f",
"5e282b1901dda02112a3f58a",
"5e282a9401dda02112a3f586",
"5e282ed201dda02112a3f599",
"5e27edd184fa810fff221d3f"],

"locs":[[1032,237,1114,319],
[468,140,525,197],
[548,215,617,284],
[946,105,1003,162],
[277,257,395,375],
[862,221,980,339],
[146,145,202,202],
[1707,318,1806,417],
[1347,220,1429,302]]},

"faceCount":9,

"status":"success"}
```

Output from different frames will be consolidated to return final output as per requirements of client. In the following table we will present comparative analysis of different approaches used during experiments.



Methods	Feature Extractor	Techniques for face verification	Accuracy
AttendexNetV1	ResNet-34	Multi-layer Neural Network	100%
	ResNet-34	SVM	94%
	ResNet-34	KNN	88%
	ResNet-34	Decision Tree	86%
	ResNet-34	Naïve Bayes	85%
AttendexNetV2	ResNet-34	Faiss	100%
AttendexNetV3	Resnet-34	Edit Distance	97%
	Resnet-34	Manhattan distance	91%

Table1: Comparative analysis of accuracy for different users(1000 requests)

5. Conclusion and Future Work

In this paper we proposed AttendXNetV1, AttendXNetV2 and AttendxNetV3 which can effectively perform the task of face verification for attendance. Use of ResNet-32 for estimating face embedding with a combination of identified classifier and similarity measuring metric produced a pipeline for real world application. Extended-MobileNet ensures security from print and video attack. In future, we will improve datasets by collecting more samples in different lighting conditions for face anti-spoofing. Deep learning architectures are evolving with very fast pace, and that will be helpful for designing robust systems. Current work will be very useful for industrial or academic purposes.

5.1 Acknowledgement

This project is fully funded by Signy Advanced Technologies, Level 39 One Canada Aquare, Canary Wharf, London E14 5AB. We want to extend our thanks to Parmesh, Sourabh and Matangi for developing Signy.

References

[1] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik 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.

[2] Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[3] W.-S. T. WST. Deeply learned face representations are sparse, selective, and robust. perception, 31:411–438, 2008.

[4] Sun, Yi, et al. "Deepid3: Face recognition with very deep neural networks." *arXiv preprint arXiv:1502.00873* (2015).

[5] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[6] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. arXiv preprint arXiv:1506.07310, 2015.

[7] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In BMVC, volume 1, page 6, 2015.

[8] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. arXiv preprint arXiv:1511.02683,2015.

[9] Y. Wu, H. Liu, J. Li, and Y. Fu. Deep face recognition with center invariant loss. In Proceedings of the on Thematic Workshops of ACM Multimedia 2017, pages 408–414. ACM, 2017.

[10] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In ICML, pages 507–516, 2016.

[11] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deepface recognition with long-tail. arXiv preprint arXiv:1611.08976, 2016.W.-S. T. WST. Deeply learned face representations are sparse, selective,and robust. perception, 31:411–438, 2008.

[12] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. arXiv preprint arXiv:1703.09507, 2017.

[13] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l 2 hypersphere embedding for face verification. arXiv preprint arXiv:1704.06369, 2017..

[14] Y. Liu, H. Li, and X. Wang. Rethinking feature discrimination and polymerization for large-scale recognition. arXiv preprint arXiv:1710.00870, 2017.

[15] M. Hasnat, J. Bohn´e, J. Milgram, S. Gentric, L. Chen, et al. von mises-fisher mixture model-based deep learning: Application to faceverification. arXiv preprint arXiv:1706.04264, 2017.

[16] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In CVPR Workshops, volume 4, 2017.

[17] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In CVPR, volume 1, 2017.

[18] X. Qi and L. Zhang. Face recognition via centralized coordinate learning. arXiv preprint arXiv:1801.05678, 2018.

[19] F. Wang, W. Liu, H. Liu, and J. Cheng. Additive margin softmax for face verification. arXiv preprint arXiv:1801.05599, 2018.W.-S. T. WST. Deeply learned face representations are sparse, selective, and robust. perception, 31:411–438, 2008.

[20] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. arXiv preprint arXiv:1801.09414, 2018.

[21] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. arXiv preprint arXiv:1801.07698, 2018.

[22] Ranjan, Rajeev, et al. "A fast and accurate system for face detection, identification, and verification." *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.2 (2019): 82-96.

[23] Tang, Fenggao, et al. "An End-to-End Face Recognition Method with Alignment Learning." *Optik* (2020): 164238.

[24] Y. Zheng, D. K. Pal, and M. Savvides. Ring loss: Convex feature normalization for face recognition. In CVPR, June 2018.

[25] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[26] Douze, Matthijs, Jeff Johnson, and Hervé Jegou. "Faiss: A library for efficient similarity search." (2017).

[27] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).