Exploring the Potential of A-ResNet in Person-Independent Face Recognition and Classification

Ahmed Mahdi Obaid*, Aws Saad Shawkat and Nazar Salih Abdulhussein
Al Imam Al Adham University College, IRAQ
* Corresponding author's Email: ahmed.altaee1977@imamaladham.edu.iq

***Abstract*—This study offers a novel face recognition and classification method based on classifiers that use statistical local features. The use of ResNet has generated growing interest in a variety of areas of image processing and computer vision in recent years and demonstrated its usefulness in several applications, especially for facial image analysis, which includes tasks as varied as face detection, face recognition, facial expression analysis, demographic classification, etc. This paper is divided into two steps i.e. face recognition and classification. The first step in face recognition is automatic data cleansing which is done with the help of Multi-Task Cascaded Convolutional Neural Networks (MTCNNs) and face.evoLVe, followed by parameter changes in MTCNN to prevent dirty data. The authors next trained two models: Inception-ResNetV1, which had pre-trained weights, and Altered-ResNet (A-ResNet), which used Conv2d layers in ResNet for feature extraction and pooling and softmax layers for classifications. The authors use the best optimizer after comparing a number of them during the training phase, along with various combinations of batch and epoch. A-ResNet, the top model overall, detects 86/104 Labelled Faces in the Wild (LFW) dataset images in 0.50 seconds. The proposed approach was evaluated and received an accuracy of 91.7%. Along with this, the system achieved a training accuracy of 98.53% and a testing accuracy of 99.15% for masked face recognition. The proposed method exhibits competitive outcomes when measured against other cutting-edge algorithms and models. Finally, when it comes to why the suggested model is superior to ResNet, it may be because the A-ResNet is simpler thus it can perform at its best with little data, whereas deeper networks require higher data size.**

Keywords-Face Recognition; Face Imag; Local Binary Patterns; Labelled Faces In The Wild

I. INTRODUCTION

Although algorithms for face recognition and facial classification have been developed, effective face identification remains a significant problem for computer vision and pattern recognition researchers. The last decade has seen significant development because of advances in face modelling and analysis tools. Cons of traditional approaches include the need for identity verification in the digital environment becoming more critical, worries about public safety, the use of modelling techniques and face analysis in multimedia data management, and computer entertainment. Algorithms for accurate facial classification and facial recognition have grown quickly in the last ten years. Performance in a number of face recognition technology sectors is always improving, and it's important to note that current applications place new requirements on future development such as data security measures include using biometrics, encryption keys, passwords, and several other techniques. To communicate identities and facilitate social interactions, the human face is essential. Due to its potential applications in both law enforcement and non-law enforcement organisations, biometric facial recognition technology has attracted a lot of attention in recent years. Due to its non-contact process, face recognition has distinct advantages over other biometrics systems that use fingerprints, palm prints, and iris recognition. Without touching the subject, images can be captured from a distance and used to create a face. Identification doesn't need getting to know the person. Additionally, recognisable facial images can be gathered and stored to help with future identification. In this section, the problem is divided into several parts such as:

- Classification: Throughout the classification phase, comparisons between

the facial image and images from the database are made.

- Feature Extraction: The most valuable and distinguishing features of the facial image are extracted during the feature extraction stage.
- Face Representation: Face representation outlines the modelling process for faces and establishes the methodologies used for future face detection and recognition.

The following is the paper. The studies pertinent to the suggested strategy have been discussed in section 2. The study's mathematical foundation is given in section 3. The proposed study is contrasted with several algorithms based on various factors in section 4. The empirical study part, which describes and processes datasets, is developed in section 5. The outcomes of the suggested approach are detailed in section 6, and section 7 of the study includes some conclusions and suggestions for further work.

II. Literature Review

The studies that used the LFW dataset or the techniques for face recognition fall into one of two categories with the suggested strategy.

A. LFW Dataset

Several studies have been published such as the LFW benchmark's upper bound for naive-deep face recognition has been studied by Zhou et al. [1]. They started by looking into how data distribution and size have an impact on system performance. They use a variety of cutting-edge approaches that have been developed in earlier literature to describe their findings when they have a sizable training dataset. They summarised their findings by stating that classification, feature extraction, and face detection are the three primary issues that need to be resolved in order to improve face recognition.. The data is biased and the false positive rate is relatively low. Iqbal et al. [2] have investigated face detection using angularly discriminative features and Deep Learning (DL). To reduce model errors, they have been suggested in classification strategies. On the LFW dataset, they ultimately obtained 99.77% accuracy. Balaban [3] has suggested cutting-edge DL and facial recognition. To benchmark these systems, the authors have emphasised the need for larger and more challenging public datasets. The joint identification for DL face representation was suggested by Sun et al. [4]. By using DL and both face recognition and verification signals, they demonstrate in this study that it is possible to do so successfully. On the LFW dataset, supervised face verification accuracy of 99% was achieved. Table 1 lists the accuracy that the aforementioned investigations were able to obtain.

TABLE I. Accuracy achieved

Study	Accuracy (%)
Zhou et al. [1]	99.50%
Iqbal et al. [2]	96.40%
Balaban [3]	99.63%
Sun et al. [4]	67%

B. General: Face recognition and classification

Several studies have been published such as a computational framework for brain-inspired face recognition has been put out by Chowdhury et al. [5]. This work presents a novel idea for an ideal computational model of facial recognition software that incorporates both engineering counterparts of these cues from earlier studies and signals from the distributed face recognition mechanism of the brain. They discovered that accuracy decreased on average by 4%. In their study on face identification, Mao et al. [6]V employed a deep residual pooling network. They provide a complete learning architecture for recognising textures that integrates the CNN model's prior residual pooling layer for effective feature transfer. According to their claims, the dataset is randomly split into 60% for training and 40% for testing. Deep fair models for complex data labelling in graphs and explainable face recognition for Local Binary Pattern (LBP) have been developed by Franco et al. [7]. Their model's accuracy increased by 5%. In the future, they plan to extend their research to a wider variety of architectures and datasets, providing new information and guidelines on how to build more equitable models for challenging input data. An LBP face recognition survey system was proposed by Kortli et al. [8]. The strategies based on local, holistic (subspace) and hybrid characteristics are highlighted in this paper's summary of recent

research on 2D and 3D face recognition systems. Additionally, they asserted that they have compared the processing speed, complexity, discrimination, and resilience of numerous approaches. Utilizing a super-wide regression network, Liu et al [9] have investigated and researched unsupervised cross-database facial expression identification. In this study, they provide a Special Super Wide Regression Network (SWiRN) model that serves as the regression parameter to connect the original feature space and label space.

III. Empirical Study

The dataset description and dataset preprocessing will be covered in this part.

A. Dataset Description and Pre-Processing

A face image library called Labeled Faces in the Wild (LFW) was developed to study the issue of unrestricted face identification [10]. This database was created and is kept up to date by researchers at the University of Massachusetts, Amherst. 13,233 images of 5,749 people from the internet were recognised and centred using the Viola-Jones face detector. 1,680 of the people in the dataset had two or more different images. Four sets of LFW images and three different types of "aligned" images are included in the original database. The researchers found that for the majority of face verification techniques, deep-funnelled images performed better than alternative image formats. The dataset offered here is therefore the deep-funnelled form.

Since the 1970s, face recognition has been the subject of extensive research. To extract the faces from an input image that contains many faces, face detection is typically used by face recognition systems. A low-dimensional representation (or embedding) is produced and acquired after preprocessing each face. It is necessary to have a low-dimensional representation for effective classification. Face identification is challenging since faces are not solid objects and images might be taken from different perspectives. Face representations must be impervious to intrapersonal image fluctuations like those caused by age, expression, and style while yet being able to distinguish between interpersonal image variations between people. The preprocessed and enhanced input images are as follows:

- They are scaled to fall between [0, 1].
- The images are subjected to shearing alteration.
- To make the model more robust, various areas of the image are zoomed in.
- Each image is then horizontally flipped.

B. Methodology

The technique is broken down into two sections in this section: general methodology and the suggested model design.

C. General Methodology

The broad methodology primarily consists of two things:

1) Face position. The authors begin by using face.evoLVe.PyTorch and MTCNN [11] for automatic face alignment. Figure 1 depicts the architecture of the deep cascaded multi-task framework that MTCNN proposes to improve ResNet's performance on face alignment by utilising their intrinsic correlation.

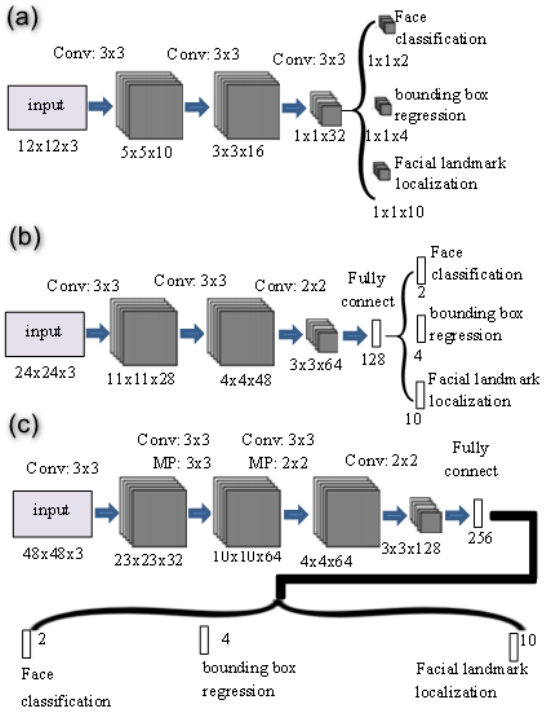

Figure 1. MTCNN's architecture: (a) P-Net (b) R-Net and (c) O-Net

However, the authors discover that even though MTCNN is quick, it occasionally makes mistakes and introduces erroneous data, such as in Figure 2, and these erroneous data will unquestionably have disastrous effects on model training. The authors then use the face-align tools from face.evoLVe to ultimately obtain accurate data. You may find this utility here [12]. This tool brings no dirty data but is around four times slower than MTCNN. However, the scientists are perplexed as to why MTCNN produces such incorrect results given that it is almost cutting edge. The face.evoLVe tool was created using the MTCNN. The authors evaluate a number of options, and the results demonstrate that when the default minim-window-size is undefinable, MTCNN starts at 10x10 and has a propensity to obtain incorrect faces. Therefore, all results are positive once the authors set the minimum size at 40x40.

Figure 2. Examples of corrupt data from MTCNN

2) Transforms. The authors performed additional preparations for the models' resilience after cleaning the data and aligning all the faces, and this work resulted in a 3-point improvement in test accuracy. The authors randomly *modify* the images after loading the data to enhance training. The authors experimented with a number of transforms, including Random-Color-Jitter, Random-Rotation, and Random-Horizontal-Flip. Finally, all of these transforms were chosen by the authors to increase the model's resilience. And due to the fact that LFW dataset take images under various lighting conditions, the Random-Color Jitter accuracy increases by around 2 points.

D. Model Architecture

The limited scale of the data the authors have makes it difficult to train a model without over-fitting. The authors believe that it is acceptable to fine-tune a model that has already been trained. The last levels must be created by the authors. The suggested model design primarily consists of two things:

1) Pre-trained ResNet. The pre-trained weight that the authors download is a version of Google's FaceNet. The high-level model structure of FaceNet is depicted in Figure 3 [13]. They apply triple loss and ultimately achieve 0.997 accuracy at LWF.

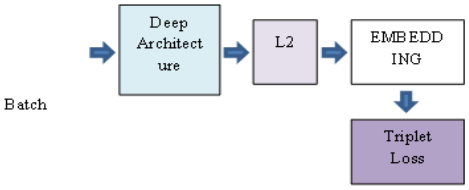

Figure 3. FaceNet's high level model structure

The first model, which was created to be improved upon by FaceNet, is tuned by the authors using Inception-ResNet [14]. Figure 4 depicts the Inception-ResNet architecture.

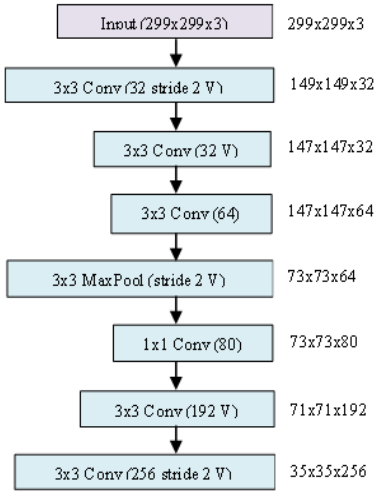

Figure 4. Inception-ResNet

2) Altered-ResNet (A-ResNet). The authors altered the final layers of the ResNet before testing which model performed best. According to the code snippet, the model has six final levels as shown in Figure 5.

As a result, the authors wish to remove the layers after Conv2d, utilise some of their algorithms, and just update the final layers to include an additional 104 faces. This is because earlier levels contained the fundamental data necessary to recognise face traits and fundamental characteristics. In the modified model, the last linear, pooling, batchnorm, and sigmoid layers have been removed, leaving only a torch model. In order to leverage the features retrieved by Cov2d

layers, the authors then construct a final layer's class with sample Flatten and Normalize layers. Figure 6 depicts the architecture in this manner. It can be called A-ResNet. The authors will train these two models and provide some details to determine the best in the following part.

```
[Block8
 (branch0): BasicConv2d(
  (conv): Conv2d(1792, 192, kernel_size=(1, 1),
   stride=(1, 1), bias-False)
  (bn): BatchNorm2d(192, eps-0.001, momentum=0.1,
   affine=True, track_running_stats=True)
  (relu): ReLU()
 )
 (branch1): Sequential(
  (0): BasicConv2d(
(branch1): Sequential(
  (0): BasicConv2d(
   (conv): Conv2d(1792, 192, kernel size=(1, 1),
    stride=(1, 1), bias=False)
   (bn): BatchNorm2d (192, eps=0.001, momentum=0.1,
    affine=True, track_running_stats=True)
   (relu): ReLU()
  )
  (1): BasicConv2d(
   (conv): Conv2d(192, 192, kernel_size=(1, 3),
    stride=(1, 1), padding=(0, 1), bias=False)
   (bn): BatchNorm2d(192, eps=0.001, momentum=0.1,
    affine=True, track_running stats=True)
   (relu): ReLU()
  )
  (2): BasicConv2d(
   (conv): Conv2d(192, 192, kernel_size=(3, 1),
    stride=(1, 1), padding=(1, 0), bias=False)
   (bn): BatchNorm2d(192, eps=0.001, momentum=0.1,
    affine=True, track_running_stats=True)
   (relu): ReLU()
  )
 )
 (conv2d): Conv2d(384, 1792, kernel size=(1, 1),
  stride=(1, 1))
), AdaptiveAvgPool2d(output_size=1), Sequential(
 (0): Flatten()
 (1): Linear(in features=1792, out features=512,
  bias=False)
 (2): normalize()
), Linear(in_features=512, out features=104, bias=True),
 Softmax(dim=1)]
```

Figure 5. Six-final layers

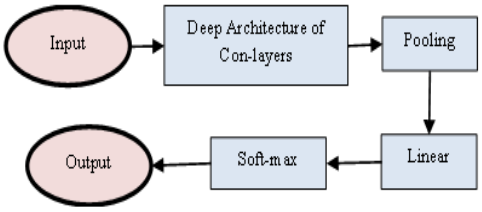

Figure 6. A-ResNet Architecture

IV. Model Implementation

The authors start the training phase after designing the model. Various epochs, batch sizes, learning rates, and models were tested in this section.

A. *Adam Optimizer*

In deep learning, the optimizer is crucial, and different optimizers can perform completely differently. As is well known, "Adam" is a highly effective optimizer, but should authors also utilise it in their work? Figure 7 displays the outcomes of the authors' testing of RMS-prop, another theoretically sound optimizer, in Tensorborad-X.

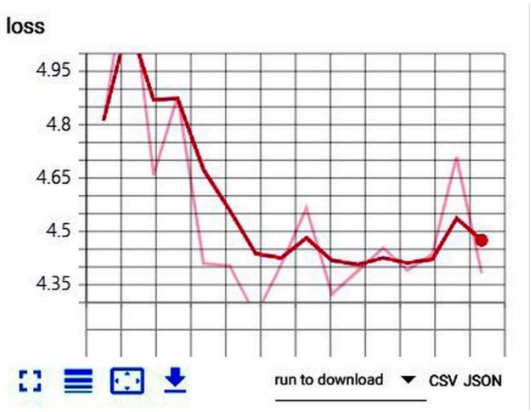

Figure 7. RMS tracking loss in TensorboradX

It demonstrates that the loss of the RMS optimizer actually decreases very quickly in the initial stages, and finally converges at a value of roughly 4.5. However, Figure 8 illustrates how much better the Adam optimizer performs with the identical epochs and batch sizes of 32 and 128.

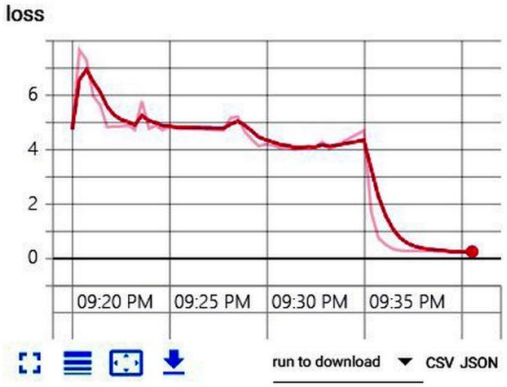

Figure 8. Adam tracking loss in TensorboradX

B. *Epoch and Batch Size*

The findings on Inception-ResNet that the authors obtain after selecting various epoch and batch size combinations are shown in Table 2. More batch size typically results in improved performance, as shown in Table 2, although

sometimes more epochs are required to minimise the loss. For example, 256 batch size performs worse than 128 batch size in 24 epochs before improving in 32 epochs. Finally, the ResNet achieves 82 true positives at 24 epochs, 128 batch size, and highest performance. The authors can quickly select a few combinations for A-ResNet using Table 2, and the outcomes are given in Table 3.

TABLE II. RECORDS OF COMBINATION FOR RESNET

Epochs	Batch size	True Positive	Train FPS
10	16	20	426.4
24	16	25	421.7
24	32	40	278.6
24	64	74	151.2
32	64	70	160.7
24	128	79	148.9
32	128	76	232.4
24	256	69	182.5
32	256	76	192.8
64	256	75	154.3

TABLE III. RECORDS OF COMBINATION FOR A-RESNET

Epochs	Batch size	True Positive	Train FPS
24	64	70	151.9
24	128	81	170.4
32	128	85	254.4
32	256	76	209.8
64	256	76	194.3

Fortunately, the A-ResNet outperforms ResNet at its peak performance of 24 epochs, 128 batch size, and 81 true positives. The authors are therefore pleased to declare that A-ResNet has won this combination with 10 more ture-positives. The least loss for ResNet during training is approximately 0.27, whereas the minimum loss for A-ResNet is approximately 3.8. This likely indicates that ResNet is constructed more intelligently in order to track and minimise the loss.

V. RESULT ANALYSIS

Because the authors used face.evoLVe to analyse face images during the training phase, employing this tool during the testing phase would be cumbersome. As a result, the authors turned to MTCNN, and by adjusting its parameters, it rarely detected incorrect images. The authors loaded the top A-ResNet model, and Table 4 lists the results of the face-recognition test. Face recognition takes roughly 0.46 seconds per image, and the top A-ResNet model achieves an accuracy of 82.7%. Not bad. But as seen in Table 5, this outcome is slower than ResNet.

TABLE IV. RESULTS FOR A-RESNET

Metrics	Value
Accuracy	0.9169230769230769
Time	50s

TABLE V. RESULTS FOR RESNET

Metrics	Value
Accuracy	0.7884615384615384
Time	38s

As a result of the authors' testing of ResNet and hand-modified A-ResNet, all of which were based on pretrained weights, A-ResNet ultimately emerged as the winner in terms of accuracy. The Adam optimizer is used by the authors since it minimises loss the best. For the best model, face recognition accuracy is 91.7% and it takes 0.50 seconds per image.

A. Face Recognition under Different Resolutions

The outcomes of face recognition for a number of low-resolution input images are covered in this section. Table 6 displays the identification rates utilising our created database LFW and a rotating head around the camera. As image resolution improves, the identification rate rises. Additionally, a key element in determining recognition accuracy is the quantity of images in the database. Table 7 shows that the findings demonstrate that as the input images' pixel count increases, so does the recognition accuracy. Identification accuracy is strong even when the camera is surrounded by a

moving head. This is because when the head is angled toward the camera, the cropped face image is aligned before being recognised, increasing recognition precision.

TABLE VI. RECOGNITION RATE BASED ON LFW DATABASE

Recognition	Correct Times	Wrong Times	Correct Image Accuracy	Incorrect Image Accuracy
At 15 pixels	84	20	80.76%	19.24%
At 20 pixels	86	18	82.69%	17.31%
At 30 pixels	88	16	84.61%	15.39%
At 35 pixels	90	14	86.53%	13.47%
At 45 pixels	92	12	88.46%	11.54%

TABLE VII. RECOGNITION RATE BASED ON LFW DATABASE

Recognition at 45 px	Correct Times	Wrong Times	Correct Image Accuracy	Incorrect Image Accuracy
Front facing	87	17	83.65%	16.35%
Facing 30’ Right	89	15	85.57%	14.43%
Facing 30’ Left	91	13	87.50%	12.5%

B. Masked Face Recognition

Even with the high number of epochs and steps in each epoch, the system performed with a testing set accuracy of 99.15% and a training set accuracy of 98.35%, proving that the model was not overfit (75 epochs of 276 steps). The A-ResNet's accuracy suggests that determining whether a face is wearing a mask is an easy problem to solve. The mask recognition model is not the most challenging aspect of the system, as has been discovered in earlier research on the subject. The real challenge is finding the locations of hidden faces in images. The authors classified the faces by using the A-ResNet and the Haar Cascade facial recognition system. The model worked well, according to the authors' manual examination of group images. Since the Haar Cascade approach required unique parameters for each image, this system is not automated, but it serves as a proof-of-concept for the model's ability to function with real-time input. Figures 9 (a and b) findings demonstrate that only actual faces are detected, and each face is correctly identified. Each classification is also accurate. Given that the model can classify each image in as little as 200 milliseconds and that the Haar Cascade technique can operate in real-time on a video stream, it is clear that once a facial detector is made autonomous, the model itself might be used to process real-time, on-the-fly data.

Figure 9. System fully utilised to identify (a) faces and (b) face masks

VI. CONCLUSIONS

In this paper, the viability and utility of using high-order local patterns for face recognition and identification are examined. The experimental results show that, in comparison to other existing feature representation strategies, the suggested approach provides an efficient and cost-effective means of encoding facial features with strong discriminative ability. In this study, despite the model's outstanding accuracy results, the authors still have certain questions they want to answer. For instance, the face-verification function is too sluggish to verify all images and names; the authors speculate that this is because their algorithm is O(n2), and they write too much code to transfer data between the GPU and CPU, which takes time. And according to the authors, using a B+ tree or another data structure would be able to speed up the search process while also preventing the need to move data from one device to another. Additionally, even though the model performs admirably on the LFW-dataset, for actual industrial need, faces are occasionally very small, slanted, and only have side faces, similar to surveillance films. Perhaps the authors will need to

create a 3D model of faces and employ other skills to avoid overfitting, such as knowledge distillation, in order to identify faces in these settings. In conclusion, there is still a lot of room to adapt this work to a particular situation. Because it is less complicated, more computationally valuable, and simpler than other algorithms, the method is thought to be effective.

REFERENCES

[1] E. Zhou, Z. Cao, and Q. Yin, "Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not?," Jan. 2015, Accessed: Nov. 11, 2022. [Online]. Available: http://arxiv.org/abs/1501.04690.

[2] M. Iqbal, M. S. I. Sameem, N. Naqvi, S. Kanwal, and Z. Ye, "A deep learning approach for face recognition based on angularly discriminative features," Pattern Recognition Letters, vol. 128, pp. 414–419, 2019, doi: 10.1016/j.patrec.2019.10.002.

[3] S. Balaban, "Deep learning and face recognition: the state of the art," in Biometric and Surveillance Technology for Human and Activity Identification XII, 2015, vol. 9457, p. 94570B, doi: 10.1117/12.2181526.

[4] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Advances in Neural Information Processing Systems, 2014, vol. 3, no. January, pp. 1988–1996, Accessed: Nov. 11, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/e5e63da79fcd2bebbd7cb8bf1c1d0274-Abstract.html.

[5] P. R. Chowdhury, A. S. Wadhwa, and N. Tyagi, "Brain Inspired Face Recognition : A Computational Framework," pp. 1–26, May 2021, Accessed: Nov. 11, 2022. [Online]. Available: http://arxiv.org/abs/2105.07237.

[6] S. Mao, D. Rajan, and L. T. Chia, "Deep residual pooling network for texture recognition," Pattern Recognition, vol. 112, 2021, doi: 10.1016/j.patcog.2021.107817.

[7] D. Franco, N. Navarin, M. Donini, D. Anguita, and L. Oneto, "Deep fair models for complex data: Graphs labeling and explainable face recognition," Neurocomputing, vol. 470, pp. 318–334, 2022, doi: 10.1016/j.neucom.2021.05.109.

[8] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," Sensors (Switzerland), vol. 20, no. 2. 2020, doi: 10.3390/s20020342.

[9] N. Liu et al., "Super Wide Regression Network for Unsupervised Cross-Database Facial Expression Recognition," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2018, vol. 2018-April, pp. 1897–1901, doi: 10.1109/ICASSP.2018.8461322.

[10] 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," hal.inria.fr. 2007, Accessed: Nov. 11, 2022. [Online]. Available: https://hal.inria.fr/inria-00321923/.

[11] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.

[12] Q. Wang, P. Zhang, H. Xiong, and J. Zhao, "Face.evoLVe: A cross-platform library for high-performance face analytics," Neurocomputing, vol. 494, pp. 443–445, Jul. 2022, doi: 10.1016/j.neucom.2022.04.118.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in 31st AAAI Conference on Artificial Intelligence, AAAI 2017, 2017, pp. 4278–4284, doi: 10.1609/aaai.v31i1.11231.