Smart Attendance System Using Face Recognition

Anshu Kumar Pandey
Email: anshu.20scse1010553@galgotiasuniversity.edu.in
Anmol Kumar
Email: Anmol.20scse1010698@galgotiasuniversity.edu.in
Dr. Abdul Aleem
Email: er.aleem@gmail.com
Galgotias University Greater Noida, India

Abstract: By using modern technology and deep learning strategies, we can monitor attendance by creating a system that can recognize students' frontal faces in a classroom. The area of the head that extends from the forehead to the chin is known as a face (Oxford Dictionary). One of the biometric techniques that will be crucial for this system is face recognition. The front of the human face is the most informative part of social interactions because it conveys crucial information about a person or group. Because of their distinctive faces, people can be easily identified. The recommended course of action is to develop a working prototype of a system that will assist the classroom's attendance system by recognizing students' frontal faces in a nearby photograph. The problem of students and middlemen being listed as present even if they are not is easily resolved with this system. Systems for face identification and recognition have been developed recently as a consequence of research. Governmental organizations, financial institutions, and social media platforms all employ some of them. Face recognition is essential formobile devices to operate and execute specific duties. This concept serves as the foundation for a smart attendance system, which will be quicker than manual attendance. This strategy will be effective for keeping track of student attendance and records.

Keywords--- Deep Learning, Onyx Model, MXNet, TensorFlow, Convolutional Neural Network(CNN), Database, Training, and Recognition.

I. Introduction

To track students' attendance, every university requires a reliable system. And every institution has its unique methodology to follow. Many students still manually take attendance during lecturesby calling out their names in attendance registers. Fewer students have adopted biometric attendance systems like RFID Card readers [1], fingerprint scanners, and iris scanners. The traditional approach of manually calling out student names takes a lot of time. Each student has anRFID Card system allocated to them with their unique identity, but there is a chance that their Cardmay get lost or that someone not authorized will abuse it to make up for phony attendance. However, they fall short in other biometrics like voice, iris, and fingerprint detection.

Using a digital image or a stream of video frames, a facial recognition system can recognize or confirm a person. There are several ways that facial recognition systems can function; they distinguish between the provided information and faces in a database using chosen facial information. Artificial intelligence (AI) is also referred to as biometric artificial intelligence (BIAI) when it can uniquely identify a pupil by recognizing patterns based on the student's facial information and forms. The fastest and most efficient method of the attendance management system uses face recognition as its primary method [2]. Facial recognition reduces the likelihood of proxy attendance and is a more appropriate and quick technique than other approaches. Facial recognition helps with the static identification of a pupil who is to be identified without the requirement for the further activity to maintain its integrity. Face recognition involves the first of the two processes, which is the detection of faces, and the second, which is the recognition of the discovered face photos using the database of existing photographs [3]. As a result, numerous facedetection and recognition techniques have been developed.

Face recognition is a comprehensive process that includes both actualization-based methods, which encompass the information of the complete frontal face, and element-based methods, which emphasize the face's geometricalcharacteristics, including the cheeks, eyes, nose, and brows. With the aid of deep learning, our system accepts face recognition progress to mitigate the flaws of the current system. It requires a high-quality camera to take student photographs, and the face recognition process is carried out using a scatter diagram of an orientated angle. The backend side (server), which is made up of logic and Python, and the frontend side (client), which consists of a graphical user interface built on electron JS, are connected by an IPC (Inter-Personal Communication) bridge. The photos captured bythe camera are transferred to the system for additional processing, where they are compared to a database of reference images of each student to track their attendance.

II. Literature Survey

A little research has been done in the field of attendance system using facial recognition. /few of the most prominent and recent researches have been discussed next. Machine learning methods have been used as a new approach for attendance and feedback systems [4]. The student attendance and feedback system concepts from two technologies have been applied in this study using a machine learning method. The student's performance is automatically assessed by this system, which also records attendance, remarks, and grades in subjects like Science, English, and other courses. Therefore, recognizing a face makes the student’s attendance information available. Following identification, feedback on the student's attendance and grades is gathered.

Another significant research proposes an automated attendance system, which utilizes facial recognition [5]. Once students enter the classroom, the system automatically recognizes them and records their attendance by using facial detection and recognition algorithms. Viola-Jone’s to find human faces, a method that combines a cascade classifier, the PCA feature selection method, and SVM classification has been utilized. When compared to the traditional technique of documenting attendance, this system saves time and helps with student monitoring.

Student attendance system have been also developed using iris detection [6]. The suggested method requires the pupils to stand in front of the device for the camera to identify and recognize the pupil and record the student's attendance. The iris can be located using techniques like Skin Pixel Detection, Six Segment Rectangular Filter, and Gray Scale Conversion. Waiting until the previous members are finished is one of the time-consuming tasks for a student or employee, but it helps to minimize proxy issues and it effectively protects the student's attendance.

Facial recognition-based attendance system for lectures have been designed making use of automatic attendance detection obtained from continuous monitoring [7]. The constant observation supports the estimation of attendance performance and its improvement. To keep track of attendance, pictures of the students' faces and postures are taken in the classroom. For attendance purposes, the system establishes each student's seating arrangement and location through constant monitoring and recording. The work's primary goal is to develop a technique for calculating each focused seat's precise weight based on its location. To speed hasten the process by which the imaging picture camera that recognizes-detects and recognizes the faces kids faces of the children can identify, the effectiveness of the image is also explored. The computer then compares the image of the student's face that was recognized with those that were previously saved in the database. The attendance is entered into the attendance database for further calculation after the student's face is found on the saved image. If the retrieved image does not match the student's face that is already in the database, a new image is added. The camera might not capture the entire group of kids or the image adequately using this method.

III. Proposed Approach

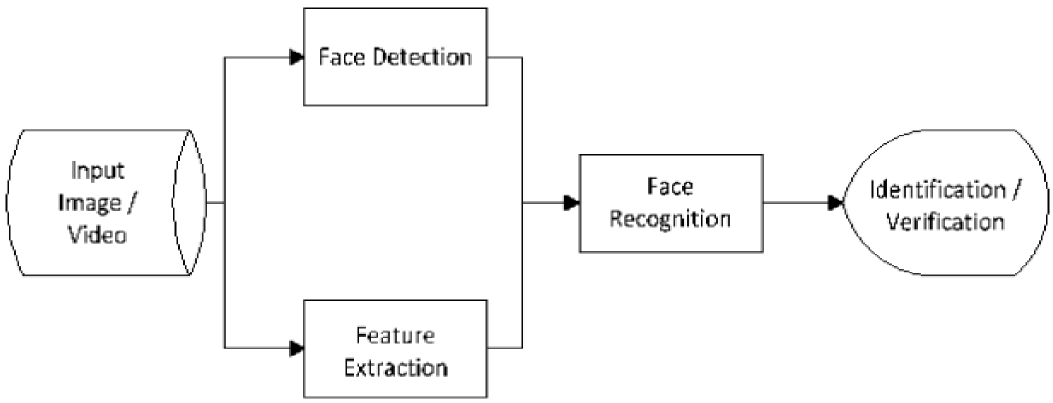

Fig. 1: Activity Flow of the Proposed Approach

3.1 Model

The adoption of an Open Neural Network Exchange can provide data scientists with the ability tooptimize. It advances the use of ONNX-compatible libraries for all tools that favor exported ONNXmodels and ensures the improvement of performance for specific realistic use cases. On a very largescale, ONNX is said to support framework interoperability. Numerous significant machine learning libraries, including as Py-Torch, TensorFlow[8], and MX.Net, are accessible in a variety of programming languages. The concept is that you may put up a model for assumption and predictionusing one tool stack and train it using another. Your model must be submitted in the model. To achieve this compatibility, the model is serialized and stored in a proto-buf file using the onyx format. Fig.1 shows the activity flow diagram of the proposed approach.

3.2 Face Detection

We would first need to identify a face from an image to detect a face. The Ultra-light face detector [9] introduced a relatively high level of accuracy while being unmatched in terms of speed. Fig. 2 shows the level of accuracy for a sample ultra-light face detection.

ultra light	80	0.102

Fig. 2: Level of Accuracy

Python is crucial for using the ultra-light model, and it needs a specific model to act. The pre-trained ultra-light 640 will be employed. Using the onyx model, The input image has to be resized to 640x480. Having pre-processed the provided image, we must set up an ONNX runtime and the onyx model [9].

3.3 Face Recognition

The deep learning system Mobile-Face-Net can recognize faces in the supplied dataset with an accuracy of 90% and 85% of the faces in the collection can be labeled[10]. The network runs anddisplays results on the GPU in roughly 24ms and employs almost a million parameters. When contrasting this performance Shuffle-Net's accuracy results, which had more parameters and required a little more CPU processing time, had accuracies of 98.70 percent and 89.27. The developers' simple replacement of a CNN's global average pooling layer with a depth-wise convolution layer will enhance facial recognition performance [11]. The amazing feat is that Mobile-Face-Nets manages to achieve equivalent accuracy on a very tight budget.

IV. Experimental Details

A system with a single CPU running at 1.8 GHz, Processor - Intel I3, 4.00 GB RAM, Storage – 1TB, and Web camera have been employed for performing the experiments. We utilized the same maximum iterations, one CPU, and starting learning rate for the three networks, but we varied the number of epochs for each network. To use the ultra-light model, the following Python (version 3.6) packages are TensorFlow1.13.0, OpenCV Python 4.1.1.26, Onyx 1.6.0, Onyx-TF 1.3.1, Onyx Runtime 0.5.0, Operating system - Windows 10 and Front-End - Python Tkinter.

All the required libraries first have been imported initially. The libraries utilized are as follows:,

1. Open-CV: To process images and videos, use.
2. Numpy: To manipulate pixels as arrays.
3. Onyx: To work with onyx models.
4. Dlib: For face mapping
5. OS: To modify or create directories as well as open files in existing ones.
6. Imutils: A tool for manipulating images (preprocessing).
7. Tensor-flow: Create and load models to produce layers.
8. Pickle technique: The serialization or deserialization of Python objects (such as lists, tuples, and dictionaries)is done.
9. Xlsx: To create generate, read, or write into an Excel spreadsheet.

The system we suggested operates in three actions:

1. Establishing a Database.
2. Making the example ready.
3. Identifying the data in real time.

4.1 Database

The first step is to construct a database into which students' videos can be submitted. For the model to extract features from it, it should be in video format. Your video could produce more features if it is more distinct and varied. Also, each pupil is given a folder in which to keep the corresponding video. The model must then be trained after that. They are trained using the video's captured visuals. Preprocessing is done [12] to generate variants or alignments from the raw video and extract featuresfrom the video. Shape predictor 68 face landmarks. Imutils Dat and Face Multis capability is utilized for this [12]. The names and images of these extracted features are then correctly mapped usinga pickle and dumped into a pkl file. Fig.3 shows a single image in database which is augmented to multiple images for training, as shown in Fig. 4.

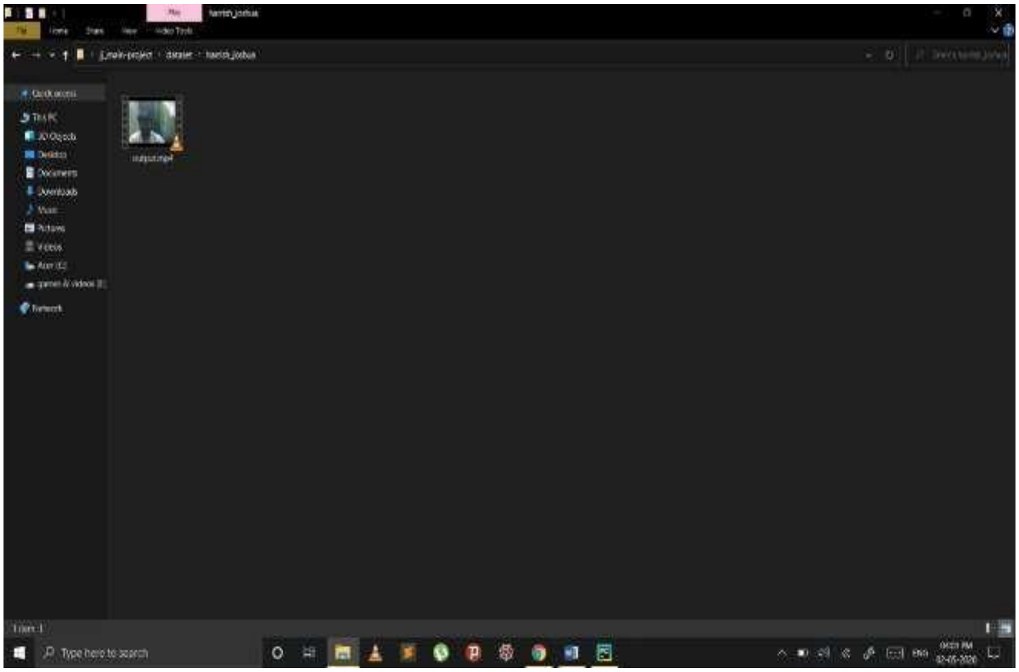

Fig. 3: Database

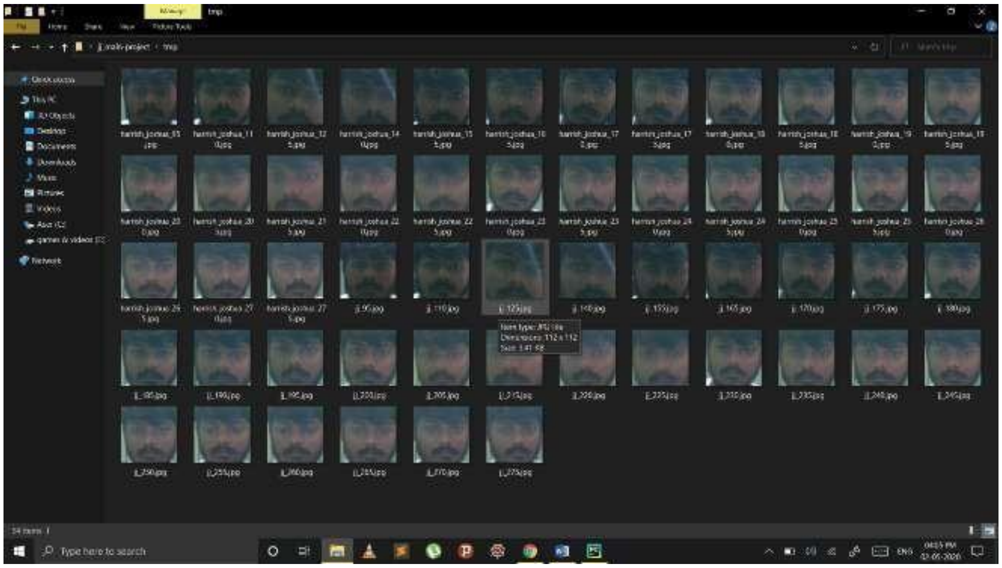

Figure 4: Training Database

V. Result and Discussion

5.1 Training

The experiment's face photos come in a variety of forms, including variations in emotion, changes in position, and modifications in facial features. Using the ORL face database, this goal is accomplished. 350 photos are associated with 35 people in the database. For experimentation, every image in the ORL database is taken into consideration. The effectiveness of facial recognition has been used as a performance indicator. The features are broken down using SGWT, after which performance is assessed.

Decay Level	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	sample 6	Sample 7	Sample 8	Sample 9	Sample 10
1	85.78	83.23	81.23	85.44	88.55	90.12	92.04	91.23	88.54	88.34
2	88.91	82.67	82.35	86.52	88.43	90.34	92.45	93.21	88.32	88.35
3	89.23	84.34	82.56	87.65	88.65	92.23	92.34	91.23	88.91	84.33
4	90.21	85.43	83.45	89.01	88.43	92.13	92.36	93.5	86.54	83.28
5	90.45	84.32	83.43	88.43	87.66	94.32	92.36	90.23	87.34	83.51
6	86.32	82.31	86.71	88.32	88.43	91.26	92.56	90.26	87.31	85.49
7	84.57	85.43	87.54	88.49	89.43	94.31	92.31	91.26	85.43	86.41

Fig. 5: Recognition Accuracy

The SGWT was trained using a total of 60 epochs, with 10 iterations for each epoch, allowing the network to extremely effectively train and validate the data. The validation accuracy was 100% after 720 iterations, which is flawless accuracy and shows that the network is well-trained. In comparison to the previous two networks, the network's training process took 76 minutes to finish. Additionally, a 10-iteration validation method was used to make sure the system was properly trained without overfitting the data.

Although SGWT has the highest validation accuracy, training takes the longest because of the large number of parameters. The second-best option is SqueezeNet, which provides an accuracy of 98.33% after a minimum of 26 min 53 sec of training. The least accurate of the three networks is GoogleNet. Fig 6 provides an overview of training and validation results.

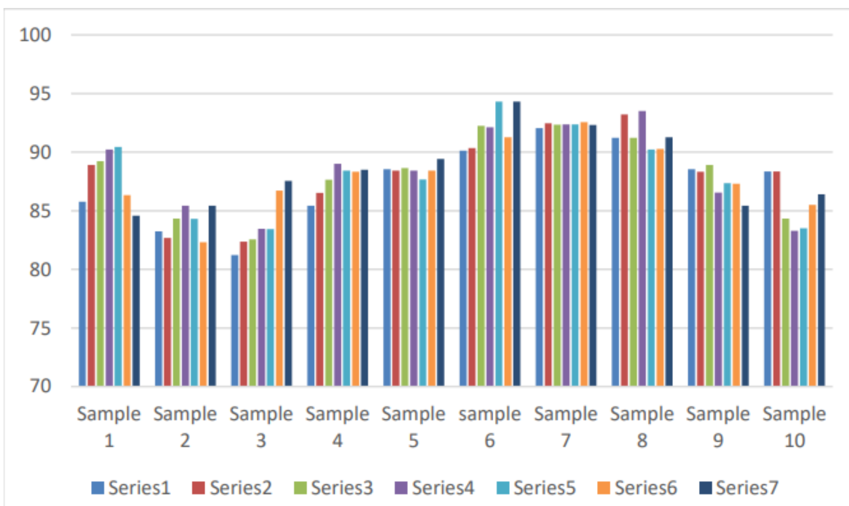

Fig. 6: Result in Graph

5.2 Testing

A face must be detected to be recognized. Several face detectors, including Yoloface, MTCNN, HOG, and Ultralight face detectors, have been tested. We choose an ultralight face detector because it allows for accurate detection with less processing power. Even in a single frame, it can possibly distinguish 70–80 faces. Moreover, compared to other detectors, the computing time is shorter[13].

Index	Method	No. Faces	Time (s)
a	face_recognition cnn	70	108.499
b	face_recognition hog	74	0.76
c	mtcnn	123	2.209
d	yoloface	129	1.053
e	ultra light	80	0.102

Fig. 7: Recognition

We need to scale the input image to 640x480 because the pre-trained Ultra-light 640 will be used onyx model. Please adjust the size if you are using 320 models. Before building up the ONYX model and an inference session, the image must first be pre-processed. After detection, recognition is a vital phase. Several methods, including Open-face, Resnet, Face-Net, VGG-Face-Net, and Mobile-NetV2 can be used to recognize faces. Because of MobileNetV2's incredibly accurate real-time face verification on mobile and embedded devices, we selected it from among these [14]. To identify a face, just load the appropriate names into the embedding dataset (.pkl file). Euclidean distance and threshold can then be used to determine how different the target face is from a person's in the database. The person from the dataset will be projected to be the one with thesmallest difference. To be regarded as a known individual, the minimal difference must be greaterthan the threshold. An individual will be treated as unknown when the disparity is greater. Fig 7 provides an overview number of faces.

As a result, we have developed a system that utilizes a CPU to do real-time facial recognition. Even though it only has a frame rate of about 13, it is still far faster than utilizing sophisticated CNNs[15]. But, there are still a lot of things we could do to boost this system's efficiency and accuracy. The existing models may be compressed by knowledge distillation, and their size may be further minimized by low-bit quantization. Another option is embedding.

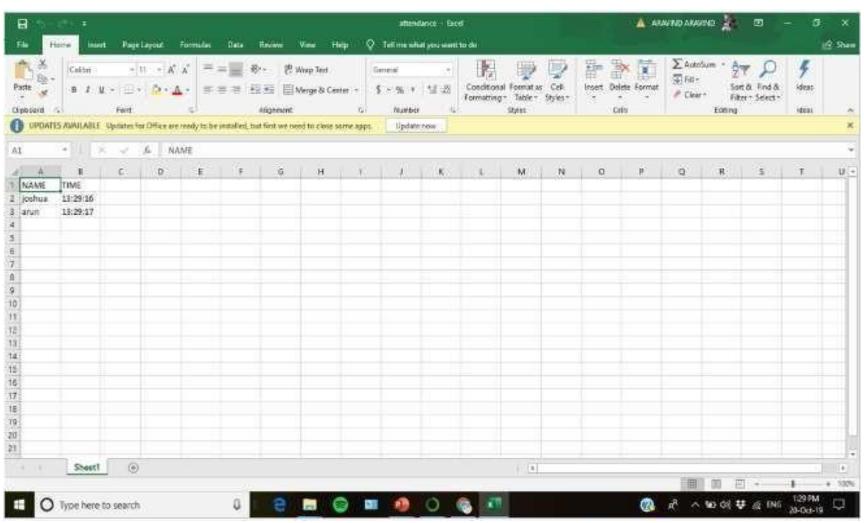

Fig. 8: Attendance Dataset Created by the Proposed System

VI. Conclusion and Future Work

The research was carried out more effectively to take attendance. The deep learning system Mobile-Face-Net can identify faces in the given dataset with an accuracy of up to 85% when they are labeled and 90% when they are recognized. This would benefit management by cutting down on the time needed to manually take attendance and replacing the RFID card system that assigns eachstudent a unique identification. Hence, the possibility of card loss won't have an impact on students'attendance, and fraudulent attendance will be reduced. So when the termination is put into practice, By controlling attendance and preventing unlawful admission, will benefit the institution.

References

[1] T.S. Lim, S.C. Sim, M.M. Mansor, ― "RFID Based Attendance System‖," 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009)", October 4-6, 2009, Kuala Lumpur, Malaysia

[2] M. K. P. Basheer, C. V. Raghu, ― "Fingerprint attendance system for classroom needs," in Proc. India Conference (INDICON)," 2012 Annual IEEE, pp. 433-438, 7-9 Dec. 2012.

[3] S. Chintalapati; M.V. Raghunadh, "Automated attendance management system based on face recognition algorithms," 2013 IEEE International Conference on.: " Computational Intelligence and Computing Research (ICCIC), vol., no., pp.1,5, 26-28 Dec. 2013, doi: 10.1109/ICCIC.2013.6724266.

[4] N. Kar and A. Saha; "Study of implementing automated attendance system using face recognition technique unit," Computer Architecture (ISCA) 2017 ACM/IEEE 44th Annual International Symposium on, pp. 1-12, 2017

[5] N. Dalal, and B. Triggs, .: "Histograms of oriented gradients for human detection," In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 886–893. (2005).

[6] S. Kadry; K. Smaili, ― "A Design and Implementation of a Wireless Iris Recognition Attendance Management System," ISSN 1392 – 124X Information Technology and Control, 2007, Vol.36, No.3

[7] N. Rotem, J. Fix, S. Abdulrasool, S. Deng, R. Dzhabarov, J. Hegeman, et al., "Glow Graph lowering compiler techniques for neural networks," CoRR, vol. abs/1805.00907, 2018.

[8] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, et al., "In-datacenter performance analysis of a tensor processing unit," Computer Architecture (ISCA) 2017 ACM/IEEE 44th Annual International Symposium on, pp. 1-12, 2017.

[9] T.Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´S. Belongie. "Feature pyramid networks for object detection," In CVPR, 2017. 1, 2, 4.

[10] G. A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al.: Mobilenets: "Efficient convolutional neural networks for mobile vision applications," CoRR, abs/1704.04861 (2017).

[11] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: "A unified embedding for face recognition and clustering," In CVPR (2015).

[12] S. Han, H. Mao, J.W. Dally,: Deep compression: "Compressing deep neural network with pruning, trained quantization, and Huffman coding," CoRR, abs/1510.00149 (2015).

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. "Focal ´loss for dense object detection," In ICCV, 2017. 1, 2, 4.

[14] 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. 6

[15] Y. Sun, X. Wang, X. Tang, : "Deeply learned face representations are sparse, selective, and robust," In: Computer Vision and Pattern Recognition, pp. 2892–2900 (2015).