

# Age and Gender Prediction using Face Recognition

Sai Teja Challa, Sowjanya Jindam, Ruchitha Reddy Reddy, Kalathila Uthej



**Abstract:** Automatic age and gender prediction from face images has lately attracted much attention due to its wide range of applications in numerous facial analyses. We show in this study that utilizing the Caffe Model Architecture of Deep Learning Framework; we were able to greatly enhance age and gender recognition by learning representations using deep-convolutional neural networks (CNN). We propose a much simpler convolutional net architecture that can be employed even if no learning data is available. In a recent study presenting a potential benchmark for age and gender estimation, we show that our strategy greatly outperforms existing state-of-the-art methods.

**Keywords:** Caffe model, Deep-Convolutional neural networks, Deep Learning framework, Tensor Flow

Abbreviations and Acronyms:

CNN- Convolutional Neural Networks

DNN-Deep Neural Network

CV- Computer Vision

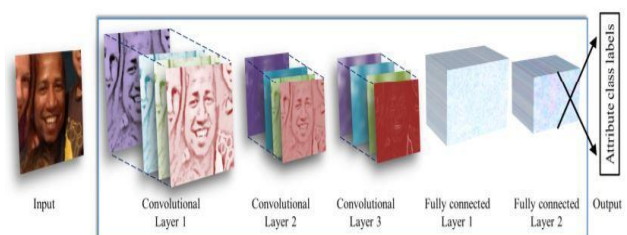
## I. INTRODUCTION

Social understanding, biometrics, identity verification, video surveillance, human-machine interaction, electronic customers, crowd behavior analysis, online advertising, product recommendations, and many more real-world applications rely on age and gender data. Due to the multiple causes of intra-class differences in people's facial pictures, calculating age and gender from face shots is challenging to solve. It restricts the models' applicability in real-world scenarios. Differences in facial feature dimensions have been used in the past to estimate or classify the attributes from face photos. Few of the prior methods were built to address the plethora of challenges that unconstrained photography brings. Furthermore, previous systems' machine learning algorithms did not correctly utilise the large numbers of picture samples and data available via the Internet to improve classification skills. We utilised Deep Learning in this study to consistently predict a person's gender and age from a single facial capture.

The expected gender will be 'Male' or 'Female.' For predicting age, we generally consider one of the following ranges: (0–2), (4–6), (8–12), (15–20), (25–32), (38–43), (48–53), (60–100) (In the final softmax layer, these eight ranges are used as nodes.). It is challenging to determine an accurate age from a single shot due to factors such as cosmetics, environmental lights, impediments, and facial expressions. As a result, instead of treating this as a regression problem, we treat it as a classification challenge. The Caffe framework of convolutional neural networks was utilised in the study to train pre-trained models. The Caffe framework beats TensorFlow by 1 to 5 times in internal testing. We use the .caffemodel trained model to make predictions about new data that hasn't been seen before after the training phase. A Python script will be written that uses OpenCV for the project code. The trained model will detect the person's face and correctly forecast their age and gender. Age and gender are two of the essential face characteristics that influence social relationships. In intelligent applications, estimating these properties from a single facial image is a critical challenge. Access control, human-computer interaction, marketing intelligence, visual surveillance, determining the total number of children and adults in hospitals, and automated ticket counters are among these uses (issuing the ticket according to age).

## II. METHODOLOGY

We implemented various techniques such as Face Detection, Face Alignment using OpenCV for processing the image, and CNN architecture for training Models. Later we performed Age group classification and Gender classification to predict age and gender while capturing photographs accurately.



The architecture of the work by Levi and Hassner, courtesy of [1].

Our CNN architecture is depicted in this diagram. There are three convolutional layers followed by a rectified linear operation and finally a pooling layer in the network. The first two layers also use a local response normalisation method for normalising.

Manuscript received on November 15, 2021.

Revised Manuscript received on November 17, 2021.

Manuscript published on December 30, 2021.

\*Correspondence Author

**Sai Teja Challa\***, Department of Bachelor of Engineering, Maturi Venkata Subba Rao Engineering College, Osmania University Hyderabad (Telangana) India.

**Sowjanya Jindam**, Department of Bachelor of Engineering, Maturi Venkata Subba Rao Engineering College, Osmania University Hyderabad (Telangana) India.

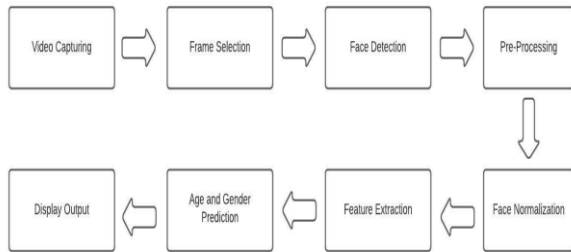
**Ruchitha Reddy Reddy**, Department of Bachelor of Engineering, Maturi Venkata Subba Rao Engineering College, Osmania University Hyderabad (Telangana) India.

**Kalathila Uthej**, Department of Bachelor of Engineering, Maturi Venkata Subba Rao Engineering College, Osmania University Hyderabad (Telangana) India.

©The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The first Convolutional Layer of CNN has 96 filters with a resolution of 77%, the second Convolutional Layer has 256 filters with a resolution of 55%, and the third and final Convolutional Layer has 384 filters with a resolution of 33%.

Finally, two ultimately linked layers with 512 neurons each are added. Figure 3 shows a detailed schematic depiction..



**Fig. 1. Architecture Diagram**

In this methodology, we capture the face using the VideoCapture() function. After receiving the input frame from the webcam, we identify the faces.

From the selected frames, we pre-process and normalize them using a user-defined function (detectFace()) with the use of a predefined function, namely blobFromImage(), which performs mean subtraction, scaling, and channel swapping (BGR to RGB). After normalization, we make use of the readNet() function which reads the deep learning network represented in one of the supported formats.

Send processed faces to the models and receive prediction outcomes through the help of pre-trained Caffe models. Render prediction outcomes with green color bounding boxes to screen.

The adience dataset, which is available at Kaggle, was utilised to perform this project. This dataset serves as the base for facial images and comprises noise, lighting, stance, and look, among other real-world imaging situations. It is about 1 Gigabyte in memory and comprises a total of 26,580 photographs of 2,284 subjects in eight age groupings (as noted above). The pre-trained models will be used on this dataset.

## III. IMPLEMENTATION

### Module -1: Data Preparation

In this stage, we cleaned the photos and saved them in a Caffe-compatible format. A Python script to handle picture pre-processing, as well as storage, is developed.

### Module -2: Model definition

In this phase, a CNN architecture is selected and its settings are described in a .prototxt configuration file.

### Module -3: Definition of a Solver

The solver is in-charge of optimizing the model. In a configuration file with the extension .prototxt, we define the solver parameters.

### Module -4: Model Training

One Caffe command from the terminal is used to train the model. The trained model in a file with the extension .caffemodel is then obtained after we have trained the model. We used a .pb file for face recognition, which is a protobuf file (protocol buffer) that carries the graph description and training weights of the model.

This is the file we will use to execute the trained model. The .pb file contains the protobuf in binary format(1's and 0's) and the .pbtxt contains in text format. The network configuration for age and gender is provided in .prototxt files, whereas the .caffemodel file describes the internal states of the layers' parameters. Further, TensorFlow files are also included along with the .caffemodel file while loading the networks.

To get the image argument from the command prompt, we utilised the argparse package to construct an argument parser. And to categorise gender and age, we parsed the input which holds the path to the image.

Next a protocol buffer is created and modeled for the face, age, and gender. The model's mean values are created and the list of age groups and genders are used to categorise. To load the networks, the readNet() function is used. The first parameter is used to store training weights, while the second is used to save network configuration.

Furthermore, we have collected the video stream through the feed from a webcam where padding is set to 20. We then read the stream and store the material in the hasFrame and frame names till any key is hit. If it is not a video, we will have to wait. For this, we will use cv2's waitKey() and then break.

Using the faceNet and frame arguments, we executed the detectFace() method, and the produced result was saved under the identifiers frameOpencvDNN and faceBoxes. And if we obtained 0 faceBoxes, that implies we did not find any faces. The net is faceNet, and the model is the DNN Face Detector, which takes up approximately 2.7MB of disc space.

For detecting the face using DNN, the following steps are taken:

1. A shallow copy of the frame is then created to get its height and width.
2. From the shallow copy, a blob is made.
3. After setting the input a forward pass to the network is made.
4. faceBoxes is currently an empty list. Hence we defined the confidence level for each number from 0 to 127. (between 0 and 1). We acquire the x1, y1, x2, y2 coordinates and append a list of them to faceBoxes whenever the confidence is greater than the confidence threshold, which is 0.7.
5. Then, for each such list of coordinates, rectangles were drawn on the picture and returned the following: the shallow copy and the list of faceBoxes. If there are faceBoxes, however, we must generate a four-dimensional blob from the image to characterise each one's face.

While doing so, we scale, resize, and send in the mean values.. We then feed the input into the network and give it a forward pass to determine the two classes' confidence.

The gender of the individual in the photograph is determined by the higher of the two numbers. After that, the process is repeated again to determine ages.

Gender and age texts will then be added to the final image and use imshow to display it.

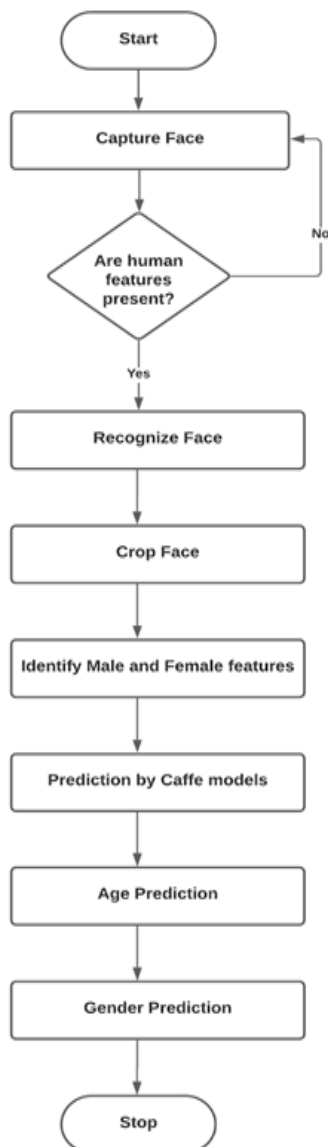
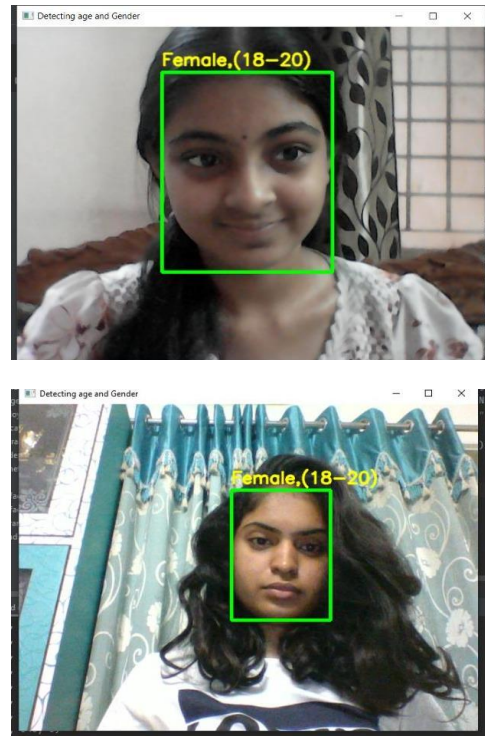
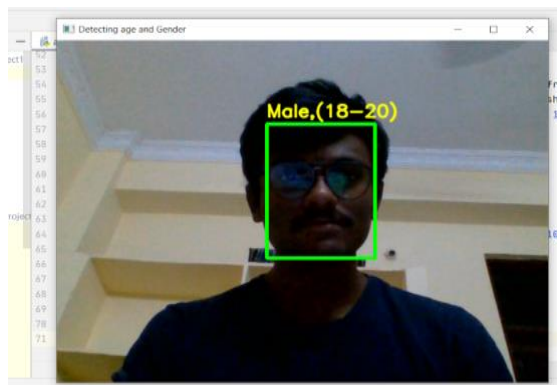


Fig. 2. Flow chart

#### IV. RESULTS

As mentioned before, upon a live real-time video capturing, a rectangle green colored facial bounding box is displayed around the face of the user. The two parameters that are displayed are gender and age. For gender, male and female categorizations are done. And in the second parameter, a range of values for age. This application does not predict the birth year as it only classifies the age range, that is, (0–7 years), (8–12), (61– ...), etc.



Screenshots of age and gender real-time prediction.

As mentioned, the Caffe framework has a performance of 1 to 5 times more than TensorFlow in the internal benchmarking and better performance compared to other deep learning frameworks.

Table 1: Comparison of popular deep learning frameworks, courtesy of [5].

Framework	License	Core language	Binding(s)	CPU	GPU	Open source	Training	Pretrained models	Development
Caffe	BSD	C++	Python, MATLAB	✓	✓	✓	✓	✓	distributed
cuda-convnet [7]	unspecified	C++	Python	✓	✓	✓	✓	✓	discontinued
Dexif [2]	BSD	Python		✓	✓	✓	✓	✓	discontinued
OverFeat [9]	unspecified	Lua	C++, Python	✓	✓	✓	✓	✓	centralized
Theano/Pylearn2 [4]	BSD	Python		✓	✓	✓	✓	✓	distributed
Torch7 [1]	BSD	Lua		✓	✓	✓	✓	✓	distributed

#### V. CONCLUSION

The findings of this project on gender categorization and age estimation contributions may be applied to real-time application problems. Although previous techniques addressed the issues of age and gender classification, much of this research was limited to constrained photographs acquired in lab conditions until recently. Such settings don't adequately reflect the appearance differences prevalent in real-world photographs on social media platforms and web archives. On the other hand, images on the internet are not only more difficult to find but also more numerous. We study how well Deep-CNN performs on these tasks utilising Internet data, and a facial recognition example from a related field. Due to the lack of labelled data, we describe our findings using a lean Deep-Learning architecture that prevents overfitting. In comparison to particular current network topologies, our network is "shallow", minimising the number of parameters and the risk of overfitting.



We purposefully include cropped copies of the images in our training set to increase the amount of the training data. Two key conclusions emerge from our research. First, CNN may be utilised to enhance age and gender classification results even with today's substantially reduced amount of unconstrained picture sets labelled for age and gender. Second, while the approach presented here is straightforward, more complicated systems with more training data may be able to achieve more significant outcomes.

## FUTURE ENHANCEMENTS

We will look into a more complex CNN architecture and a more reliable image processing approach for estimating exact ages for future work. We can use this project for electronic customers, crowd behavior analysis.

## REFERENCES

1. G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 34–42.
2. Koichi Ito, Hiroya Kawai, Takehisa Okano and Takafumi Aoki, "Age and Gender Prediction from Face Images Using Convolutional Neural Network" in Proceedings of APSIPA Annual Summit and Conference 2018.
3. Kyoungson Jhang, Junsoo Cho, "CNN Training for Face Photo based Gender and Age Group Prediction with Camera" in proceedings of 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), pages 548-551
4. Kundan Nigam, Sahil Sharma, Prashant Singh Rana, "Gender, Makeup, Age and Illumination Prediction from Faces using Ensemble Modeling," 2018 3rd International Conference for Convergence in Technology (I2CT). The Gateway Hotel, XION Complex, Wakad Road, Pune, India. Apr 06-08, 2018. Pages 1-6.
5. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding" in proceedings of the 22nd ACM international conference on Multimedia November 2014, Pages 675–678.
6. Amirali Abdolrashidi, Mehdi Minaei, Elham Azimi, Shervin Minaei "Age and Gender Prediction From Face Images Using Attentional Convolutional Network".
7. T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. Trans. Pattern Anal. Mach. Intell., 28(12):2037–2041, 2006. 2
8. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.
9. Héctor A. Sánchez-Hevia, Roberto Gil-Pita, Manuel Utrilla-Manso, Manuel Rosa-Zurera, "Convolutional-recurrent Neural Network for Age and Gender Prediction from Speech" in proceedings of 2019 Signal Processing Symposium (SPSymo). pages 242-245.
10. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In ICML, 2014.
11. E. Eiding, R. Enbar, and T. Hassner, "Age and Gender Estimation of Unfiltered Faces," 2170 IEEE Trans. Inf. FORENSICS Secur., vol. 9, no. 12, pp. 2170–2179, 2014.
12. E.M. Hand and R. Chellappa, "Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification," Proc. the Thirty-First AAAI Conf. Artificial Intelligence, pp. 4068–4074, Feb. 2017.
13. The Adience dataset"[Online]. Available: <https://www.kaggle.com/ttungal/adience-benchmark-gender-and-age-classification>

## AUTHORS PROFILE



and Pedestrian Detection Model Using Haar Cascade.

**Sai Teja Challa**, is a Bachelor of Engineering student, pursuing his final year in Information Technology at MVSR Engineering College, Osmania University. His research interests include in the field of Data Science, Machine Learning, and Artificial Intelligence. Recently, he worked on the projects, Credit Card Fraud Detection



**Sowjanya Jindam**, is an Assistant Professor from MVSR Engineering College with qualifications in MTech. Her research areas include Big Data Analytics and the Internet of Things. Her recent works include an Integrated Dynamic Approach for Prediction of Rainfall.



**Ruchitha Reddy Reddy**, is an undergraduate from MVSR Engineering College, Osmania University. Her research interest comprehends in the field of Computer vision, Machine Learning and Data analytics. Recently she worked on the project, Diabetic Retinopathy Detection, using eye fundus images and CNN.



**Kalathila Uthej**, is a Bachelor of Engineering student, pursuing his final year in Electronics and Communications at MVSR Engineering College, Osmania University. His research interests incorporate the fundamentals of research and design thinking, data analytics, Natural language processing, Machine learning and Internet of Things. He persistently worked on projects whose objectives include the concepts of NLP model for visually challenged, Smart Security System using IOT, Image Compression and Quick Response scanning system.