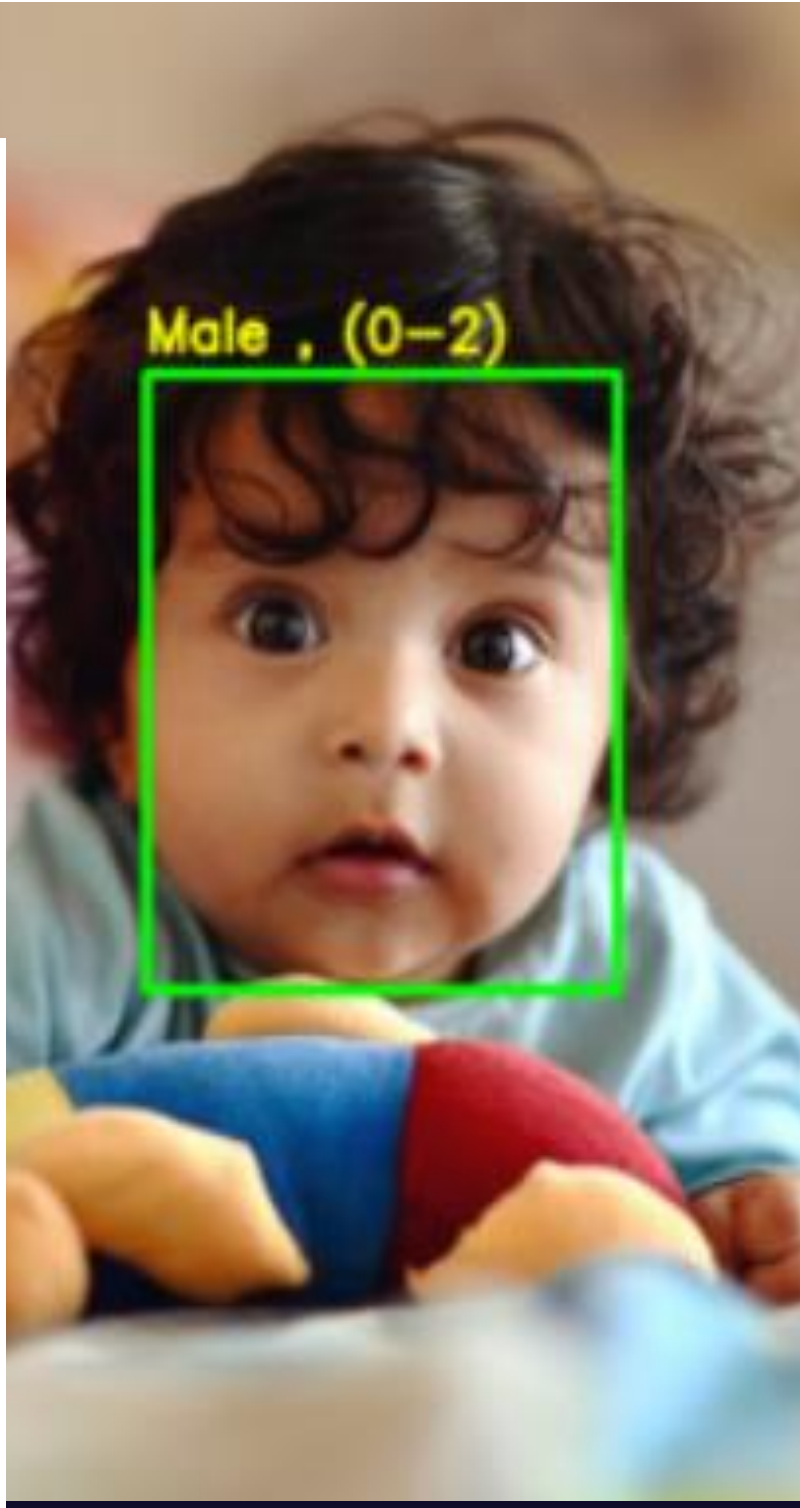
Age and Gender Detection using Deep Learning



JULY 6

Data Science Intern

Submitted by: Saifullah Rahimi

Solution Analysts
Analysing needs, delivering solutions

Index

Synopsis

The upgrading of image pictures taken from the camera sources, from satellites, airplanes, and the images caught in everyday lives is called picture processing. Processing of the image based on analysis undergoes many different techniques and calculations. Digital-formed pictures need to be carefully imagined and studied.

Image processing has two main steps followed by simple steps. The improvement of an image with the end goal of more good quality pictures; that can be adopted by other programs are called picture upgrades. The other procedure is the most pursued strategy utilized for the extraction of data from a picture. The division of images into certain parts is called segmentation.

The location of the information accessible in the pictures is much-needed information. The information that the image contains is to be changed and adjusted for discovery purposes.

There are different sorts of procedures required, just as the expulsion of the issue. In a Facial identification strategy: The articulations that the faces contain hold a great deal of data. At whatever point the individual associates with the other individual, there is an association of a ton of ideas.

The evolution of ideas helps in figuring out certain boundaries. Age assessment is a multi-class issue in which the years; are categorized into classes. Individuals of various ages have various facials, so it is hard to assemble the pictures.

To identify the age and gender of several faces' procedures, are followed by several methods. From the neural network, features are taken by the convolution network. In light of the prepared models, the image is processed into one of the age classes. The highlights are handled further and shipped off the preparation frameworks.

Introduction

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture using *Deep Learning* on the Adience dataset. A gender Detection and age prediction web application developed using Flask, Open CV, HTML & CSS.

Objective

Mankind has advanced technology to the extent that the 21st century is the crack of the dawn for unimaginable achievements. The aforementioned technology can be used for our benefit in identifying one's age and even their gender just based on their glimpse from a camera, image, and even a video. This research paper will methodically chalk out the whole procedure, multiple methodologies and algorithms that can be used, which one is the most accurate, and how it all comes together. It will also stress its importance and how it can be implemented to benefit our day-to-day life. The paramount objective of the paper is to build a gender and age detector that can approximately guess the gender and age of the face of an individual in a picture using Deep Learning on the audience dataset. Moreover, to get the most effective predictions and result by overcoming the problem of accuracy and time. Moreover, the map for the ways this technology can be used to our benefit looks at the huge spectrum where it can be implemented: ranging from security services, CCTV surveillance, and policing to dating applications, and matrimonial sites.

The main aim of this article is to detect age and gender through the given data set. We will use simple python and Keras methods for detecting age and gender.

Methodology

In this Python Project, we will use Deep Learning to accurately identify the gender and age of a person from a single image of a face. We will use the models trained by Tal Hassner and Gil Levi. The predicted gender may be one of 'Male' and 'Female', and the predicted age may be one of the following ranges- (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100) (8 nodes in the final softmax layer). It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions. And so, we make this a classification problem instead of making it one of regression.

The terminologies used in this advanced python project on gender and age detection.

What is Computer Vision?

Computer Vision in the field of study enables computers to see and identify digital images and videos as a human would. The challenges it faces largely follow from the limited understanding of the biological vision. Computer Vision involves acquiring, processing, analyzing, and understanding digital images to extract high-dimensional data from the real world in order to generate symbolic or numerical information which can then be used to make decisions. The process often includes practices like object recognition, video tracking, motion estimation, and image restoration.

What is OpenCV?

OpenCV is short for Open Source Computer Vision. Intuitively by the name, it is an open-source Computer Vision and Machine Learning library. This library is capable of processing real-time images and videos while also boasting analytical capabilities. It supports the Deep Learning frameworks *TensorFlow*, Caffe, and PyTorch.

What is a CNN?

A *Convolutional Neural Network* is a deep neural network (DNN) widely used for the purposes of image recognition and processing and *NLP*. Also known as a ConvNet, a CNN has input and output layers, and multiple hidden layers, many of which are convolutional. In a way, CNNs are regularized multilayer perceptrons.

CNN-based gender and age detection method

This study is a proposal to apply deep learning-based learning to estimate the gender and age of a robust face image in various external environment changes. Figure 1 shows the block diagram of the gender and age estimation method based on CNN.

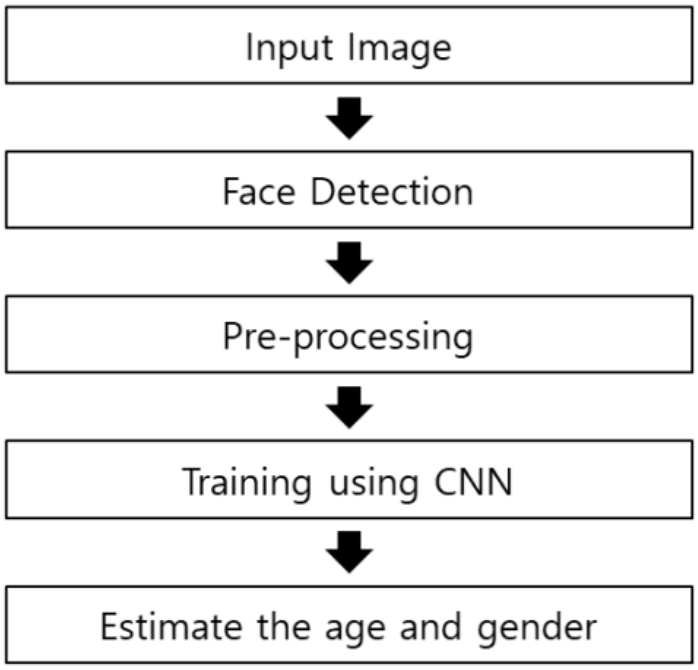


Figure 1: CNN-based gender and age detection method

The Dataset

For this python project, we'll use the Adience dataset; the dataset is available in the public domain and you can find it from Kaggle. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size. The models we will use have been trained on this dataset.

Pretrained Models

The contents of project directory include:

- opencv_face_detector.pbtxt
- opencv_face_detector_uint8.pb
- age_deploy.prototxt
- age_net.caffemodel
- gender_deploy.prototxt
- gender_net.caffemodel
- a few pictures to try the project on

For face detection, we have a .pb file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model. We can use this to run the trained model. And while a .pb file holds the protobuf in binary format, one with the .pbtxt extension holds it in text format. These are TensorFlow files. For age and gender, the .prototxt files describe the network configuration and the .caffemodel file defines the internal states of the parameters of the layers.

2. We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We make it parse the argument holding the path to the image to classify gender and age for.

3. For face, age, and gender, initialize protocol buffer and model.

4. Initialize the mean values for the model and the lists of age ranges and genders to classify from.
5. Now, use the readNet() method to load the networks. The first parameter holds trained weights and the second carries network configuration.

6. Let's capture video stream in case you'd like to classify on a webcam's stream. Set padding to 20.

7. Now until any key is pressed, we read the stream and store the content into the names hasFrame and frame. If it isn't a video, it must wait, and so we call up waitKey() from cv2, then break.

8. Let's make a call to the highlightFace() function with the faceNet and frame parameters, and what this returns, we will store in the names resultImg and faceBoxes. And if we got 0 faceBoxes, it means there was no face to detect.

Here, the net is faceNet- this model is the DNN Face Detector and holds only about 2.7MB on disk.

- Create a shallow copy of the frame and get its height and width.
- Create a blob from the shallow copy.
- Set the input and make a forward pass to the network.
- faceBoxes is an empty list now. for each value from 0 to 127, define the confidence (between 0 and 1). Wherever we find the confidence greater than the confidence threshold, which is 0.7, we get the x1, y1, x2, and y2 coordinates and append a list of those to faceBoxes.

- Then, we put up rectangles on the image for each such list of coordinates and return two things: the shallow copy and the list of faceBoxes.

9. But if there are indeed faceBoxes, for each of those, we define the face, and create a 4-dimensional blob from the image. In doing this, we scale it, resize it, and pass in the mean values.

10. We feed the input and give the network a forward pass to get the confidence of the two classes. Whichever is higher, that is the gender of the person in the picture.

11. Then, we do the same thing for age.

12. We'll add the gender and age texts to the resulting image and display it with imshow().

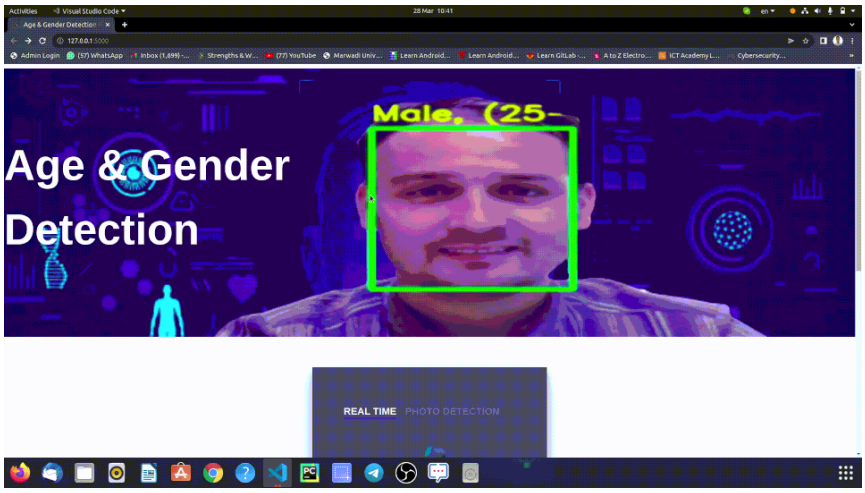


Figure 2: Age and Gender live detection

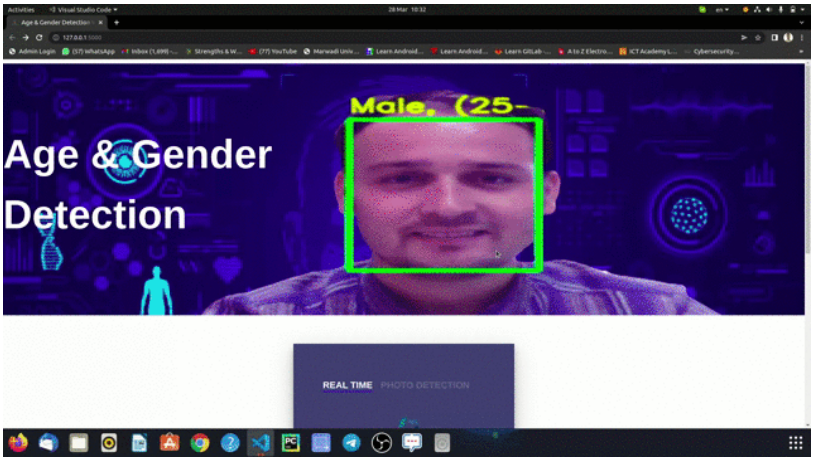


Figure 3: Age and Gender detection from photo

Conclusion

The task of recognizing age and gender, nonetheless, is an innately troublesome issue, more so than numerous other PC vision undertakings. The fundamental justification for this trouble hole lies in the information needed to prepare these kinds of frameworks. While general article discovery errands can regularly approach many thousands or even large numbers of pictures for preparing, datasets with age and gender names are extensively more modest, as a rule in the large numbers or, best case scenario, several thousand. Python obtained images and the Model did not do well in the accuracy rate, further, improvement is required in the model algorithm.

References

1. Convolutional Neural Networks tutorial – Learn how machines interpret images https://data-flair.training/blogs/convolutional-neural-networks-tutorial/
2. **Getting Started with TensorFlow https://data-flair.training/blogs/tensorflow-tutorials-home/**
3. NLP (Natural Language Processing) – A Data Science Survival Guide https://data-flair.training/blogs/nlp-natural-language-processing/
4. Deep Learning Tutorial – What is Neural Networks in Machine Learning https://data-flair.training/blogs/deep-learning-tutorial/
5. Data Science Blogathon https://datahack.analyticsvidhya.com/contest/data-science-blogathon-10/True/
6. D. Nguyen, S. Cho, and K. Park, "Human Age Estimation Based on Multi-level Local Binary Pattern and Regression Method," Future Information Technology, Vol. 309, pp. 433-438, 2014. DOI: https://doi.org/10.1007/978-3-642-55038-6_67
7. R. Rothe, R. Timofte and L. V. Gool, "DEX: Deep EXpectation of apparent age from a single image," in Proc. IEEE International Conference on Computer Vision Workshops, Dec, 2015. DOI: https://doi.org/10.1109/ICCVW.2015.41
8. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," in Proc. IEEE Conference Computer Vision and Pattern Recognition, 2015. DOI: https://doi.org/10.1109/CVPR.2015.7298594
9. T. R. Kalansuriya, and A. T. Dharmaratne, "Neural Network based Age and Gender Classification for Facial Images," International Journal on Advances in ICT for Emerging Regions, vol. 7, no. 2, 2014. DOI: http://doi.org/10.4038/icter.v7i2.7154
10. Y. Taigman et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 1701-1708, 2014. DOI: https://doi.org/10.1109/CVPR.2014.220
11. O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep face recognition," in Proc. British Machine Vision Vol. 1, No 3, pp. 6-17, 2015. DOI: https://dx.doi.org/10.5244/C.29.41

12. F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 815-823, 2015. DOI: https://doi.org/10.1109/CVPR.2015.7298682
13. N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 886-893, 2005. DOI: https://doi.org/10.1109/CVPR.2005.177
14. A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Neural Information Processing Systems Conference, 2012. DOI: https://doi.org/10.1145/3065386
15. G. Huang et al., "Labeled Faces in the wild: A Database for Studying Face Recognition in Unconstrained Environments," Univ. of Massachusetts, Amherst, Technical Report 07-49, 2007.
16. L. Wolf, T Hassner, and I. Maoz, "Face Recognition in Unconstrained Videos with Matched Background Similarity," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 529-534, 2011. DOI: https://doi.org/10.1109/CVPR.2011.5995566
17. B. F. Klare et al., "Pushing the Frontiers of Unconstrained Face Detection and Recognition: IARPA Janus Benchmark A," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 1931-1939, 2015. DOI: https://doi.org/10.1109/CVPR.2015.7298803
18. I. Kemelmacher-Shlizerman et al., "The MegaFace Benchmark: 1 Million Faces for Recognition at Scale," in Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 4873-4882, 2016. DOI: https://doi.org/10.1109/CVPR.2016.527
19. IMDB-WIKI-500k+ face images with age and gender labels. https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/\