

Jonathan Torres
Brian Dang
Mohamed
Jugaillo
Team TitanGo
CPSC 481 W 7:00

Facial Recognition

Age and gender are two of the most key facial features, they play a major role in most social interactions and play an even bigger role in artificial intelligence applications. There are many industries that would benefit from the use of age and gender recognition, some of them include security, marketing, and advertising. For our face recognition application we are able to use detect faces in images and live video. We are using OpenCV for our face detector, it is based on the single shot detector (SSD) framework with a ResNet based network. Most OpenCV SSDs usually use MobileNet for its base network. OpenCV is a library for python designed to solve computer vision problems, Computer vision is a field of artificial intelligence that deals with how computers can interpret and understand data. OpenCV was originally developed by Intel in 1999. It supports a number of languages including Python, C++, and java, and is compatible with several platforms including MacOS, Windows, and Linux. Along with OpenCV we use numpy, numpy is a highly optimized library for numerical operations. It adds support for the large multi-dimensional arrays and matrices, it also comes with a large collection of functions to operate on arrays and matrices. Having access to all these additional tools gave us the resources to successfully implement facial recognition.

Our application begins by importing the necessary packages and libraries, this includes numpy, argparse, and cv2. Then it follows by constructing the argument parser and parsing the arguments. Followed by loading a serialized model. Next we initialize the video stream allow the camera sensor to warm up, we initialize an object that specifies a camera with the index of zero which is usually the built-in camera or the first camera it detects. The next step is to loop over several frames and begin the face recognition with OpenCV. We grab a frame from the video stream and resize it, then grab the frame and convert it into a blob. Once we have a blob we pass it through a deep neural net to begin to obtain face detections. Now we begin to loop over detections and compare them to the confidence threshold. Once the confidence values are calculated we display face boxes with the confidence values on the screen. The loop continues until input breaks out of it. Once the loop terminates, cleanup begins and the application closes.

In the end, our entire group began with little to no experience with this type of application, and we finished with the confidence and tools to work on other artificial intelligence projects. We initially wanted to create a chess bot, similar to our Go bot, but as a group we decided it would be more beneficial and a better learning experience to work on a project completely unrelated from our previous projects to be exposed to a different area of artificial intelligence. Although the material was new to all of us as a group, we did our best to understand and learn the material. Along with learning the new concepts, we gained more experience with OpenCv.