Kenneth McAvoy

Kmcavoy1

Deliverable 3

For my test data, there are two parts, since my project is broken into two pieces. One is the stillimage.py piece which operates with still images. The other part of my project , video.py, works with a video feed from the camera of my laptop. The size of the datasets on both parts is essentially unlimited to the user, however, on the still images part of my project I have 264 images for training and 31 for validation. I then took another 10 unseen photos from the BioID database and my camera roll for the stillimage testing. All of the generic people in my datasets are from the BioID database. The stillimage portion my project performed very well, It had 90%+ accuracy on validation and testing. The video portion of my project was a bit harder to determine accuracy because it will constantly run through a loop detecting new faces, recognizing them, and then relaying that information to the window. It is accurate about 70% of the time, but little distortions in the way the face orients itself to the camera or odd lightings will throw the model off. I believe the best way to enhance the recognition at this point is through the amount of data provided during training. Including more “lively” photos that present a more natural stature, lighting, and facial orientation. My model achieved 100% accuracy on my testing set for stillimage. It is a small testing set, but that is on purpose because the end goal is the video interface and accepting any number of faces to be recognized in the frame at any point. In order to combat the low training data, I added a function to the video aspect, so that a user can save correctly names faces to their respective training folders.