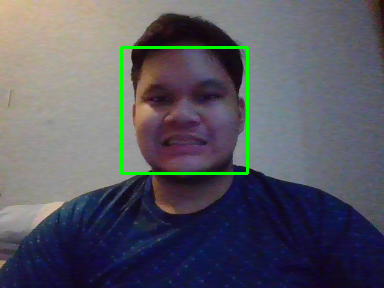
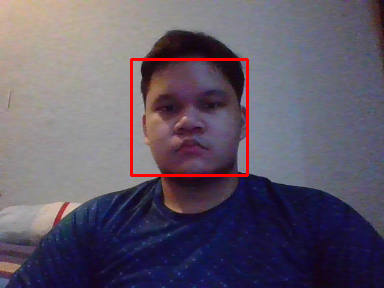
**Liveness Detection Weekly Report**

For the first week of the internship, I was tasked to research about liveness detection. The supervisor directs me to an article as a good way to start the research. The article in question is a project about a liveness detection for authorization of a person through them blinking to prevent spoof images. Using the *face\_recognition* library for encoding the faces to their own “database”, the project was able to display a person’s name when the person has blinked AND has a record in their database. To determine if the person had blinked or not, they used *OpenCV’s Haar Cascade* face and eye classifier and run it through a function to match the status of a blinking eye within 3 frames.

In order to familiar myself with the implementation of a liveness detection, I tried to replicate the project, only this time it requires the person to smile instead of blinking. I choose smiling so that it could prevent the use of paper masks which usually had holes in their eyes, which would be able to let someone to bypass the blinking authorization. The same as the previous project, I used *OpenCV’s Haar Cascade* classifier, only this time it’s the face and smile classifier instead of an eye classifier.



The project counts as an “Active” liveness detection, as it requires the person to do certain action for the system to authorize them. While this kind of method works effectively against a spoof photo, this method of liveness detection is still vulnerable to video spoofs. The best method for a liveness detection is through a “Passive” version where a model is trained to differentiate between a real and a spoof face. Although the article given from the supervisor had a code for a model in their site, I still don’t understand where and when would the model be implemented.