Project Summary report

Our project was mainly focused on hand-gesture based verification. Where we used a pretrained caffe model which is used for hand key-point detection.

The process included recording the initial password and extracting the frames from the video, identifying the hand key-points extracting the required data from co-ordinates of each frame. Then saving the coordinates of these key-points in a file.

We record the video to be verified and follow the same procedure. Finally, we cross checked the coordinates of the initial video with the one to be verified.

We accomplished the verification with an accuracy of 85.88% in case of authorized person showing right gesture. We were also able to thaw forgery attempts with a success in thawing of 60%.

We experimented with different qualities of videos with different frame rates and resolutions and fount out that the processing speed highly depends on the quality of the video. The better the quality the lesser time it takes to process the video using the caffe model. However, higher fps for the same time period takes more time as it increase number of frames to be processed.

We also tried putting the skeletal frame against a black background to identify when and at what time the hand is going out of frame. We found out that this isn't feasible for a real-time solution so we had to ditch this part.

We further tried to have multiple layers which could detect and remove clutter and thus show only hand in the focus. However, we weren't able to do that as arms would also come in the focus and as the model was trained not just on the visual appearance but also for the curves at different places, the result were found misleading.

The removal of frames removing all the frames in the beginning of the video where the hand is out of frame.

We weren't able to successfully detect multiple hand in the gestures as a password as the keypoints aren't being differentiated by which hand it's connected to. We are hoping to do this later by breaking the images into pieces and bootstrap it.

Main Findings:

- The idea of deep convoluted neural networks
- Activation functions
- Working on videos and images using OpenCV
- Extracting required details from random data
- Comparing data using specific properties
- Exponential smoothening and de-cluttering of data
- Using pre-trained caffemodel
- Adjusting the retained information according to the depth of the actor

Analysis of results:

- The output with respect to authentication was pretty good. However, we found out how the different types of data input can impact the results. Like,
    - Having too many gestures having azimuth close to 90 or -90 degrees can decrease recognition of the data and might be removed supposing it to be a bad frame.
    - If hand goes out of the frame after all the key-points were once detected, then data can be lost at a major scale and hence we need a wideangle camera or otherwise a good depth from actor(which decreases the quality of detection)
- We also were assigning previous data frame key-point position in case it's not found in the current frame and hence if a key-point is not seen for long, suppose of x frames, as we are considering window length = 5 , therefore the key-point might look like it is static for x-10 frames whereas it might not be.
- The final results according to confusion matrix were like as follows:

Confusion Matrix (For the right person):

	Authenticated	Not Authenticated
Password Correct	85.883%	14.117%
Password Incorrect	10.15%	89.85%

Confusion Matrix (In case of Forgery):

	Authenticated	Not Authenticated
Original Person	85.883%	14.117%
Forgerer	~40%	~60%

Difficulties faced were:

- Extracting and standardizing images from video
- Resolving assumed keypoints in frames where they aren't found
- Smoothening data to have more logical motions
- Extracting required details from keypoints' co-ordinates of each frame
- Comparing two output files
- Increasing efficiency by assigning values for number of degree of match to each frame
- Further increasing efficiency by detecting major hand gestures and comparing them
- Removing frames where data becomes unresolvable
- Using both hands simultaneously as gesture

Of which unresolvable were:

- Further increasing efficiency by detecting major hand gestures and comparing them
- Using both hands simultaneously as gesture

Removing frames where data becomes unresolvable was partially resolvable as we were able to handle until the point when all key-points were simultaneously detectable. After it, we are checking if too many points go out of visibility then only we are considering as not found.