Facial and Gesture Recognition using YOLOv3

A Capstone Project

Harman Deol – Salvatore Giambrone – Mir Ali Khan – Ernest Ramazani – Zackary Simpson
(Group 5)

Indiana University - Purdue University of Indianapolis
Fall 2023

Problem statement and challenges:

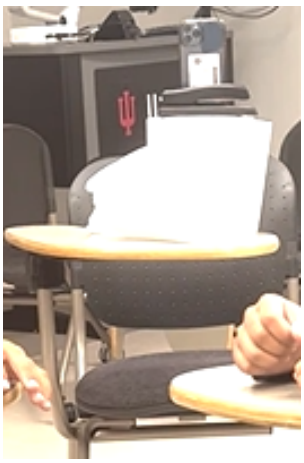
For this capstone project our goal was to develop a machine learning model capable of accurately tracking multiple human features, specifically faces, mouths, eyes, and each hand, in real-time within a dynamic environment. This system should be able to handle various scenarios including different lighting conditions, multiple individuals in the frame, and a range of human poses and movements. During this project we had to face many challenges that were difficult to overcome. Seeing our video recording we did we can immediately notice that lighting conditions would be one of our biggest concerts. Each video had some lighting issues where the exposure seemed higher or lower than we wanted and caused many areas of the frame to be too bright almost making it full white and causing issues. This biggest issue occurred when a hand would move too fast during certain frames causing the exposure to increase and cause the hand to be very bright. To counteract this model must be robust enough to function effectively under varying lighting conditions. This requires the algorithm to adapt to changes in shadows, brightness, and contrast without losing tracking accuracy. We figure with enough training and exposure to enough frames our model may be able to adapt to each frame and the lighting with time. Another large challenge is the multiple individuals in the frame. Each frame had multiple individuals meaning multiple faces, mouths, eyes and hands were present so that the model had to keep track of each one. The model must distinguish between different people and consistently track the specified features for each individual. This involves dealing with potential overlaps, occlusions, and the varying distances of individuals from the camera. The primary challenge is distinguishing between similar features (faces, hands, etc.) of different individuals. So to make sure our detector is able to detect each individual and their desired properties such as their hands with proper tracking for each frame we have to ensure the detector is properly trained with multiple samples and enough data to where it doesn't get lost with multiple individuals. Another issue can arise when some particular feature such as in our case the hand would sometimes leave the frame and reenter a few frames later. Another large issue that can be caused is due to overlap. As the camera is in a fixed position for each view, there are multiple frames where the hands will overlap another individual's hand or where a face including the eyes, and mouth are not clearly shown. This causes a challenge of the detector losing focus on the feature, not only must the model be able to detect that, that feature is no longer in the frame but must be able to detect it as soon as it reappears. This caused a challenge at the start as if our detector was to lose focus of a feature such as a hand going off the frame, it would have a difficult time to reacquire the tracking when it reenters. However after more training data and proper adjustments we were able to get our model trained to be able to constantly track these overlaps, or temporarily feature absences from the frame. The model must re-identify individuals upon re-entry and maintain consistent tracking throughout. I believe the biggest challenge was to feed the model with proper data that can be used throughout each video and be as accurate as possible. We took some time to mess with how many frames we wanted to train the model with, such as using all 1000 frames, or only

100 frames. Now the number varies and we realized the more frames we used means the better data we get but we noticed that using 250 frames from the video to train the model gave us consistent results while also saving time with each training. The more frames we used means the longer it would take to train the model, so after testing we realized 250 frames was a sweet spot that not only saves time but trains the model in an efficient way to detect each feature without compromising on results. We wanted to keep our model very consistent across all views as we had 3 different views we were testing each with their own 5 videos. We needed to consistently fine tune our model for each view to get the best tracking. For example our top view only includes hands, because of this we wanted to make sure that the model would not accidentally pick up unwanted features such as the faces. With all of this in my mind we decided to use YOLOv3 for our model. We chose this as our model must be able to process and analyze each video from each view in real time. The model had to be fast but we did not want to give up the consistency for speed and after testing multiple models we found to have the best luck with this model and decided to tackle our mentioned challenges above. We believe the biggest way to overcome these challenges was due to the step by step process we had taken. Being able to divide the work weekly and conquer each step at a time allowed us to get the results we wanted. Being able to take the time to ground truth each label allowed us to have the data we needed for any model we wanted to work with and enough data to train the models to be able to overcome our problems and get the results we wanted.

Data acquisition:

Before we could begin with the project we needed data to work with. The data for this project needed to be 15 videos with a minimum of 1000 frames each all recording four of us at a table moving our hands. These videos are split into five sessions where three videos will be recorded at a time. The three videos had different angles which were a birds eye view, a left side view, and a right side view. So in total we gathered five videos of each view, one per session, for a total of 15 videos.

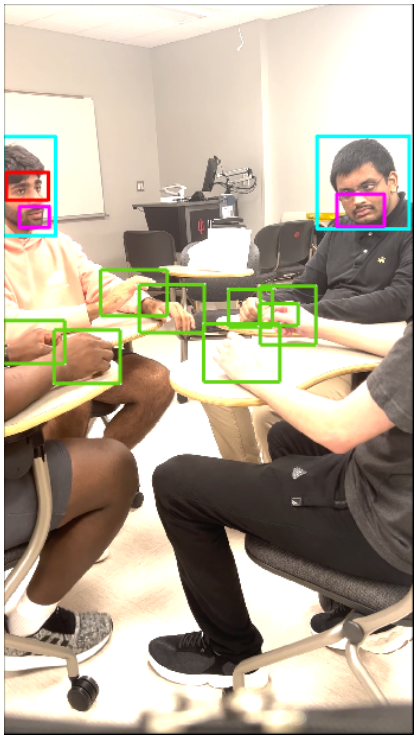
All of this data was collected using three Iphones for the cameras. The video settings for each camera was set at 30 frames per second at 1080p. Since we needed at least 1000 frames each session lasted around 35 seconds. Our birds eye view camera was placed onto a ceiling mounted projector facing down, the left side camera was on a backpack placed on two desks and stabilized with a highlighter, and the right side camera was on a paper towel roll on a desk and stabilized with a stapler. We made these elaborate camera stands so that we could reduce the number of occlusions in the videos and so the side would be as close to a 90 degrees difference from the bird's eye view as possible.

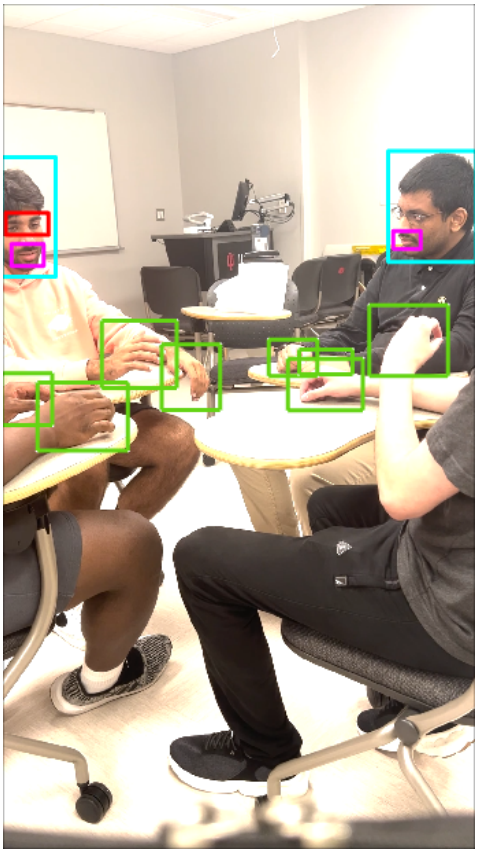


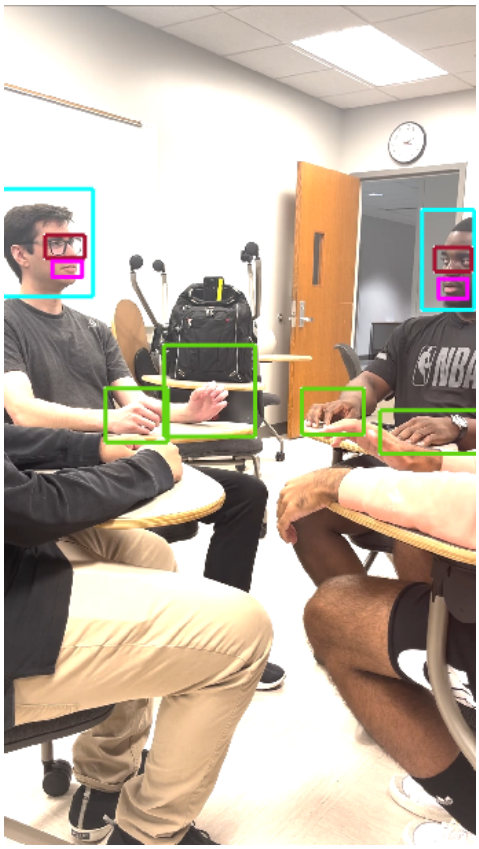
The next part of the data collection process was to sync all three videos from each session to start and end at the same time. We did this by having someone snap their fingers at the point where the videos would begin. Then from that starting point the videos were made 1000 frames long or around 33.33 seconds.

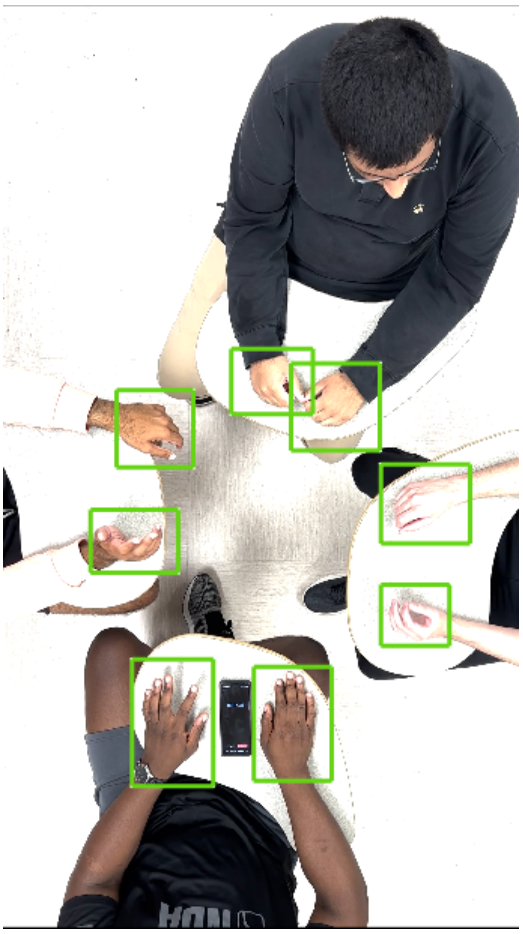
Ground-truth:

Ground-truthing is a very important part of creating a detection software, regardless of the object that is being detected. Ground-truthing provides a foundation for validation, improvements, and real world deployment. It adds credibility to the results and supports the development of more accurate and reliable detection algorithms. As previously mentioned, we completed five sessions of three videos in three views then after we got our videos synced from start to finish, we split up the work for the ground-truthing. Each member had around 3000 frames to manually create the bounding boxes using Matlab. This process wasn't necessarily hard but I don't think any of us realized how long this process would take. For this process, we each needed to upload the video into a video labeller in Matlab. After opening this video labeller we created four different labels: face, mouth, eyes, and hands. For each frame, we needed to create a bounding box around each region of interest with the correct label. For each session for two videos the face, eyes, and mouth have eight ground truth regions, while the hands have thirty-two. For the top view video, the face, eyes, and mouth have zero regions, while the hands still have thirty-two regions.This is because the face, eyes, and mouth aren't visible in this view. Each of these videos contain 1001 frames.









Detection:

Following the comprehensive groundwork laid in the previous section of this report, this segment focuses on the detection processes and outcomes using the YOLOv3 model.

Model Configuration and Training:

- Annotations Loading: The process began with the loading of ground truth face annotations from a ".mat" file. This critical step provided the baseline data necessary for training the YOLOv3 model.
- Data Division: We divided the annotated images into training and testing sets, ensuring a balanced representation of the various conditions captured in the videos.
- YOLOv3 Setup: The YOLOv3 detector was meticulously configured with specific parameters such as the number of epochs, batch size, and learning rate. These were

fine-tuned to optimize the model's learning process and improve performance, considering our unique dataset characteristics.

Custom Training and Evaluation:

- Training Approach: A custom training loop was implemented, allowing for direct control over the training process. This included the use of dlfeval for gradient calculations and sgdmupdate for optimizing parameters.
- Model Performance Metrics: The effectiveness of the model was quantified using Average Precision (AP). This metric is crucial in object detection scenarios, especially given the dynamic nature of our project environment.

Practical Application and Results:

- Video Processing Methodology: In our approach, each video frame underwent processing through the YOLOv3 detection system. This included the detection of bounding boxes for each region of interest (ROI) - faces, mouths, eyes, and hands - which were then reinserted into the frames.
- Results Documentation: Detection results were meticulously documented in a timetable “labelData”, which included both bounding boxes and labels. These results were saved in various formats, including a MAT-file for further analysis and an Excel file for accessibility and review.

Challenges and Model Tuning:

- Encountered Difficulties: Key issues included distinguishing each hand separately and maintaining detection accuracy when faces or hands were partially visible. Variations in lighting and rapid movements added layers of complexity to the detection process.
- IoU Evaluation and Tuning: We used the Intersection over Union (IoU) metric to evaluate the precision of the model in identifying various features. While the detection of faces and eyes yielded promising results, the accuracy for mouths and hands, particularly in certain positions, was less precise. This was reflected in the varying IoU scores, and the number of false positives and negatives recorded.

IoU	**Face**	**Eye**	**Mouth**	**Hands (Side1 and Top)**	**Hands (Side1 and Top)**
0-.25	905	3339	10489	13862	14614
.25-.50	2351	5763	3729	127	69

IoU	Face	Eye	Mouth	Hands (Side1 and Top)	Hands (Side1 and Top)
.50-.75	10253	5841	1440	455	181
.75-1	4509	2073	2360	1571	1152

Detection Accuracy	Face	Eye	Mouth	Hands (Side1 and Top)	Hands (Side1 and Top)
False Positive	522	2263	478	187	170
False Negative	336	1494	2197	850	748

Projections:

Model Description: We have designed a fully connected network (shallow Neural Network) as a regressor for projection of detected data from one view to another using the ground truth data.

The architecture of the shallow neural network contains a single hidden layer with 56 neural nodes. The input layer contains input nodes of size . The input nodes represent detection bounding box data x-coordinates of center, y-coordinates of center, x-coordinates of top left corner, y-coordinates of top left corner, width, and height of the detected bounding boxes. Bounding box data for one object detected are termed as one sample with 6 features. The output layer represents again bounding box data in another plane in which projection is required to be done (say, second plane). The training of the network is done with the help of ground truth of the bounding boxes in the second plane, the shape of the ground truth data in the second plane becomes output shape.

Training method: The neural model is trained using Adam optimizer with learning rate drop factor equal to 0.2 for maximum epochs of 50.

Performance evaluation: The performance evaluation is done based on IOU. IOU counts for each projection are given as follows:

Projection	IOU counts	False Positive/Negatives
1to3	IoU 0-.25 = 15468 IoU .25-.50 = 26 IoU .50-.75 = 19 IoU .75-1 = 3	False Positives = 15902 False Negatives = 0
2to3	IoU 0-.25 = 15459 IoU .25-.50 = 74 IoU .50-.75 = 19 IoU .75-1 = 2	False Positives = 23569 False Negatives = 0
3to1	IoU 0-.25 = 5187 IoU .25-.50 = 66 IoU .50-.75 = 24 IoU .75-1 = 0	False Positives = 23219 False Negatives = 0
3to2	IoU 0-.25 = 12800 IoU .25-.50 = 37 IoU .50-.75 = 7 IoU .75-1 = 0	False Positives = 28998 False Negatives = 0

Issues and Discussions: The IoU evaluation of the data revealed two issues with how our bounding boxes are saved. One issue is that we have no way to differentiate between different hands in the image so we cannot track which ground truth bounding box goes with which detection bounding box. This results in so many IoU's in the 0-.25 range. The second issue is the FCN that we used to get the projection data. The design of FCN in terms of number of layers and nodes was probably not perfect enough to accurately translate the side view to top view and vice versa, this resulted in our large number of false positives.