# ISyE 6740 - Spring 2025
# Project Report

**Team Member Names:** Ryan Rowland

**Project Title:** Evaluation of Methods for Visual Gesture Detection

**Problem Statement**

Hand gesture recognition (HGR) is a popular area of research with technologies' increased capability to quickly process human–machine interaction. It has wide applicability across robotics, augmented reality and medicine. The two domains of gesture research revolve around the use of sensor based technologies (sensor gloves) and vision based approaches (camera). The gesture research can be further categorized into the detection of static gestures where the subject is not moving (eg. thumbs up) and dynamic gestures where the subject is in motion (eg. hand waving). Static gestures simply rely on the features from a single image and can be detected using traditional image processing techniques; however, the temporal nature of the dynamic gesture data requires a specialized set of modeling techniques to be successful. This investigation will focus on the techniques commonly used to detect and classify both static and dynamic gestures using visual data.

Several approaches are typically taken to segment the data including skin color (YCbCr, RGB or HSV), edge detection, or deep learning algorithms to identify the location of the hands in the frame.

The segmented data is then processed via a feature extraction step using principal component analysis (PCA), linear discriminant analysis (LDA), speeded up robust features (SURF), scale-invariant feature transform (SIFT), convexity defect, wavelet packet decomposition (WPD), cepstral frequency, or kurtosis position.

Classification is then performed using techniques including artificial neural networks such a multi-layer perceptron (MLP), deep neural networks (such as CNN and RNN), K-nearest neighbors, hidden markov models (HMM), dynamic time warping (DTW), finite state machine (FSM), long short-term memory (LSTM) or GRU [1].

Typical accuracies with these combinations of techniques have $85 - 95$ accuracy on a variety of datasets of which many are regional sign language sets. However, reports largely lack information in regards to processing time.

One thing is common in my review of literature and that is that the datasets that are being used are fairly large and generally have multiple individuals providing the gesture information. While public datasets are available on Kaggle for both stationary and dynamic gestures, these gestures may not fit the need of an individual [2,3]. For instance, an individual may have a difficult time moving their right hand and want a custom gesture to navigate their desktop.

This project aims to determine the minimum number of recordings that an individual would have to make using their webcam in order to meet the desired recognition of three static

and three dynamic gestures in the environment in which the recorder typically uses their personal computer. It also aims to evaluate several classification techniques to determine which provides the best performance as defined by hitting an accuracy threshold in the minimum average detection time across the three gestures.

**Methodology**

Webcam recordings of three static gestures and three dynamic gestures were collected. The three static gestures included right hand stop sign, right hand closed fist and right hand thumbs up. The three dynamic gestures included right index finger swiping up from a closed fist, pinching index and thumb together and the index and middle finger swiping down.

Static gestures were recorded in sets of 1500 frames (with every 3rd frame being collected) taken with the gesture being minimally rotated and moved in the environment to capture different angles, distances from the camera lens and artifacts created from the motion.

Dynamic gestures were initially recorded in 3 second increments with each gesture being performed 50 times.

In addition to the desired gestures, supplemental datasets were also recorded. For the static gestures, 11 additional non-desired gesture/random noise were recorded. These were used as the "other" class with the aim to eliminate misclassification errors resulting from the model trying to force unknown gestures into one of the training classes. Additionally, three testing set recordings were collected with both desired and non-desired gestures were performed.

For the dynamic gestures, other gestures were also recorded and a single large test set containing the dynamic gestures and nonsense gestures was recorded.

No transformations were required for the static gestures since it was easy to capture the different variations (rotations, distances from the camera) simply by recording them. However, in order to reduce the data collection requirements for the dynamic gestures the training sets were augmented by sampling random frames and interpolating between them. Doing this I was able to increase my data set size 10-fold. Additionally, the dynamic recordings had to be normalized to a set number of frames and were interpolated/extrapolated so all training and testing data were 60 frames in length.

An outline of the collected static and dynamic datasets is shown below in Figures 1 and 2.

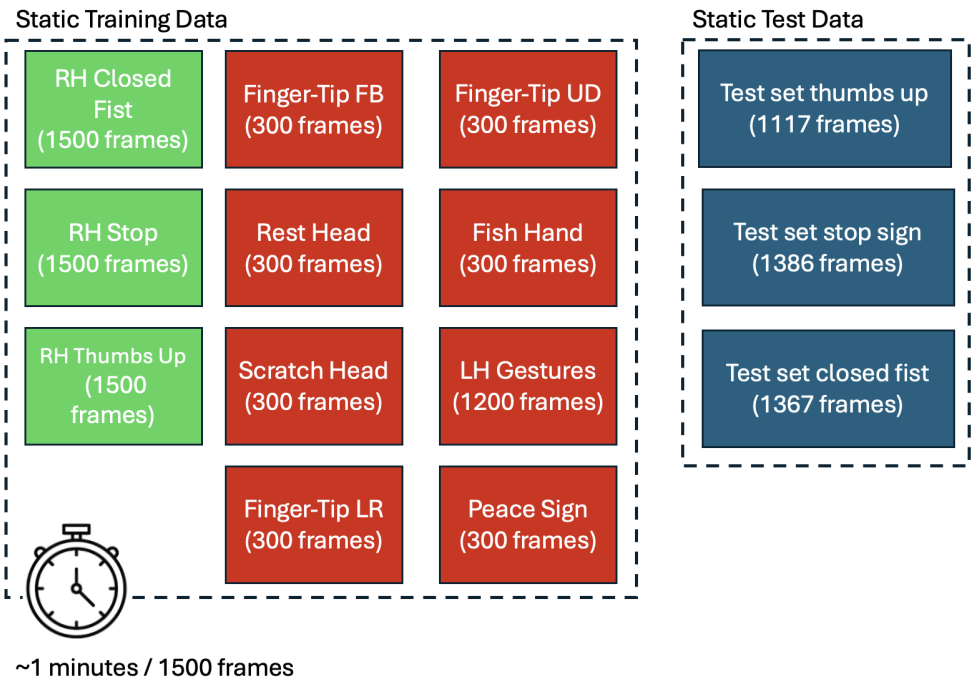


Figure 1: Static Gesture Data Sets

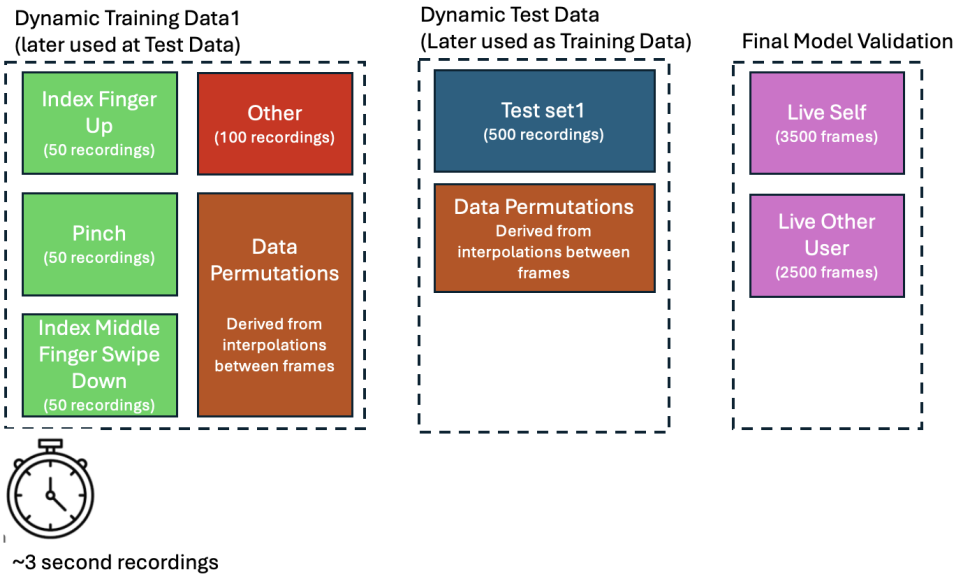


Figure 2: Dynamic Gesture Data Sets

In order to minimize the data segmentation and feature extraction requirements, Google Mediapipe's hand landmark algorithm was used to identify the x,y and z coordinates of the 21 points that can be used to describe a hands orientation [4]. The mapping for the hand landmarks are shown in Figure 3 below.

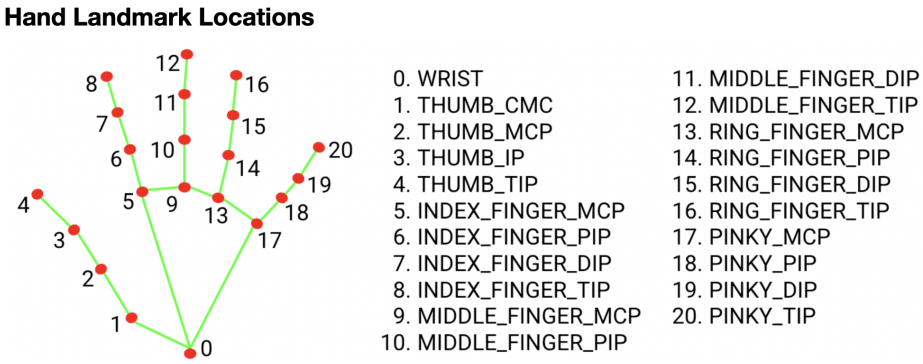


Figure 3: Google Mediapipe Hand Landmark Locations

Not only does Mediapipe eliminate the computational requirements and time to preprocess the gestures, it also makes the dataset significantly more compact reducing the storage requirements. The use of this model reduces the need to record in different environments, with varying lighting conditions aligning with the overall goal of minimizing the required data collection. Example usage of this is shown below in Figure 4.

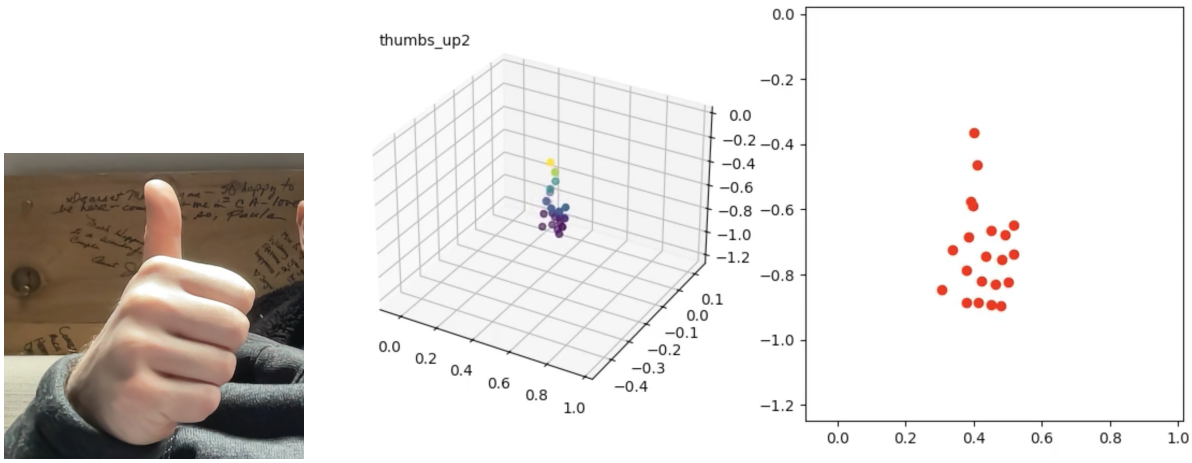


Figure 4: Example Usage of Mediapipe

After collecting the necessary data and doing the required processing, the static data was classified using KNN (with and without PCA), SVM (with and without PCA), MLP and MLP with both dropout and batch normalization. The dynamic data was classified using a simple RNN, LSTM and GRU.

### Evaluation and Final Results

Each of the multi-class classification models were evaluated against their test sets and the average F1 score, recall and precision of the model was reported using all of the data. The tabulated results for the static model are available in Appendix 1 and plotted below in Figure 5. It can be seen that from a base performance standpoint that the MLP models performed the best with precision and recalls between 85% and 90%. This is followed closely by the SVM model with a slightly lower precision.

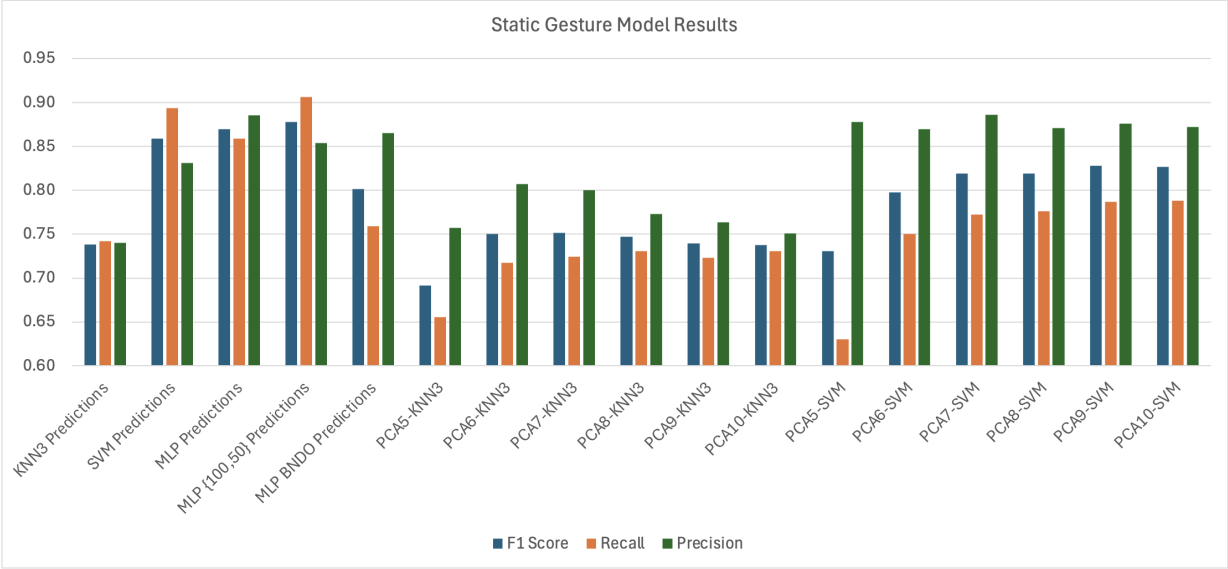


Figure 5: Enter Caption

As it turned out, a lot of necessary information was removed in the principal component analysis. I expected a lot of correlation to exist in the landmark data; however, the data that was removed appears to have been significant to identifying the gesture since there were a large number of false negatives. It really didn't provide any benefit in terms of model training time since the SVM and KNN models converged very quickly.

On the basis of the highest F1-score– the best performing static gesture detection model was determined to be the MLP(100,50). Using this as the basis, I wanted to assess how much data was required to get reasonable model performance. Starting with all the data, I randomly sampled 1000, 500, 250, 100, 50 and 25 samples from each of the categories "thumbs up","closed fist", "stop sign" and "other". Using this data I trained an MLP model with 100 and 50 neurons per layer. The performance of these models is shown below in Figure 6. From the plot, it can be seen that the F1 Score really doesn't drop too much until after the data set size drops below 100 samples for each class. There is a trade off between recall and precision with the precision dropping and the recall increasing slightly. This means the model will be slightly less sensitive to detecting the gesture which is what you would expect from having fewer samples. It is very fast to record 100 samples from each class with the camera recording up to 30 frames per second using OpenCV. Even if only sampling every third frame, in less than a minute you can make all of the recordings necessary to have a good performing static gesture detection model.

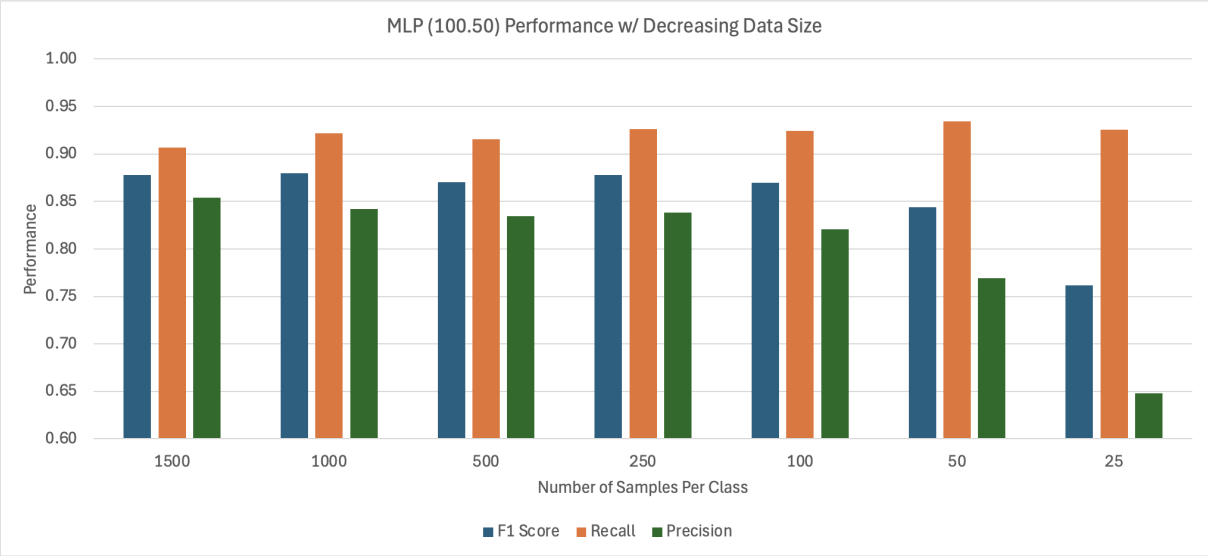


Figure 6: MLP (100,50) Model Performance w/ Reduced Data Set Size

Dynamic gesture detection was initially trained using a dataset containing 50 recordings of each dynamic gesture and 100 recordings of "other" gestures. 9 permutations were taken from each recording to make a larger dataset. A test set with 500 total recordings was also created. During the initial modeling efforts, it was found that model accuracy was fairly poor using the assigned test and train data for the RNN, LSTM and GRU models. These accuracies ranged between 37 and 47 percent. In all cases the models were able to distinguish between the three desired gestures but struggled with falsely flagging "other" gestures as the known gestures.

Using the original test set to train the model and the original training set to test the model, I was able to get significantly better performance with the RNN, LSTM and GRU models having accuracies of 77.5%, 90.1% and 91.2% respectively. The precisions, recalls and F-1 scores for each of the gestures are recorded below in Figure 7.

| Model | Metric | Gesture | | | | |
|---|---|---|---|---|---|---|
| | | *Index-Middle Swipe Down* | *Pinch* | *Index Finger Up* | *Other* | *Avg* |
| RNN | Recall | 76% | 68% | 88% | 79% | 78% |
| | Precision | 75% | 66% | 75% | 92% | 77% |
| | F1- Score | 75% | 67% | 81% | 85% | 77% |
| LSTM | Recall | 100% | 83% | 95% | 85% | 91% |
| | Precision | 81% | 83% | 98% | 98% | 90% |
| | F1- Score | 89% | 83% | 96% | 91% | 90% |
| GRU | Recall | 95% | 98% | 97% | 80% | 93% |
| | Precision | 96% | 73% | 95% | 99% | 91% |
| | F1- Score | 96% | 84% | 96% | 89% | 91% |

Figure 7: Initial Dynamic Gesture Detection Results

I believe these improved results are due to the original test set being twice the size of the training set and it having a more comprehensive "other" gesture class included. By reversing their roles, the model was able to detect many more variations (spatial differences) of the same gesture.

In order to evaluate the impact of data set size on the dynamic GRU model performance, data was randomly sampled from the training data and the precision, recall and F1 scores were calculated. The results of this study are shown below in Figure 8. Clearly training data set size has a much larger impact on the performance of the model compared to the static gesture detection and 250 samples per class provides an F-1 score of at least 85%. The quality of the data also appears to have an impact with 500 samples actually having lower performance than 250 samples. This may suggest that the permutations could be adding little value. If using permutations it appears that you would need greater than 50 recordings for each class. At 3 seconds per recording this means it should take about 10 minutes of recording to get a good performing model (which is far longer than the 1 minute it takes to make a performant static model).

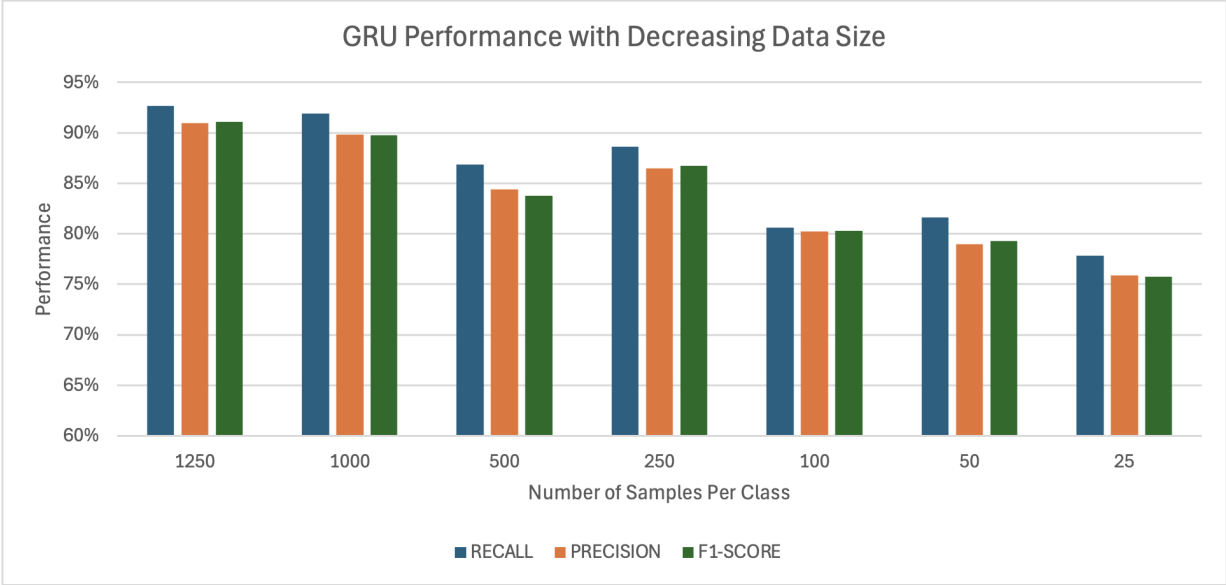


Figure 8: GRU model performance w/ reduced data set size

After finding that the GRU model performed the best for the dynamic data and the MLP model performed best for the static data, I did some user testing to determine how well the models worked in practice and their usability.

I overlayed the model outputs with my openCV recording script and had it detect the live static and dynamic gestures. Screenshots from the test are shown below in Figure 9. The top text is the dynamic gesture detected (along with its probability) and the bottom text is the static gesture detected.

The MLP static gesture detection performance was exceptional, with the gestures even being detected at appropriate transition points. For instance, when rotating a thumbs up, it detected the gesture only when the hand had an angle between roughly 40-140 degrees. This is shown below in Figure 10. This same effective transitioning was noted when moving to and from the closed fist and stop sign as well as with their rotations.

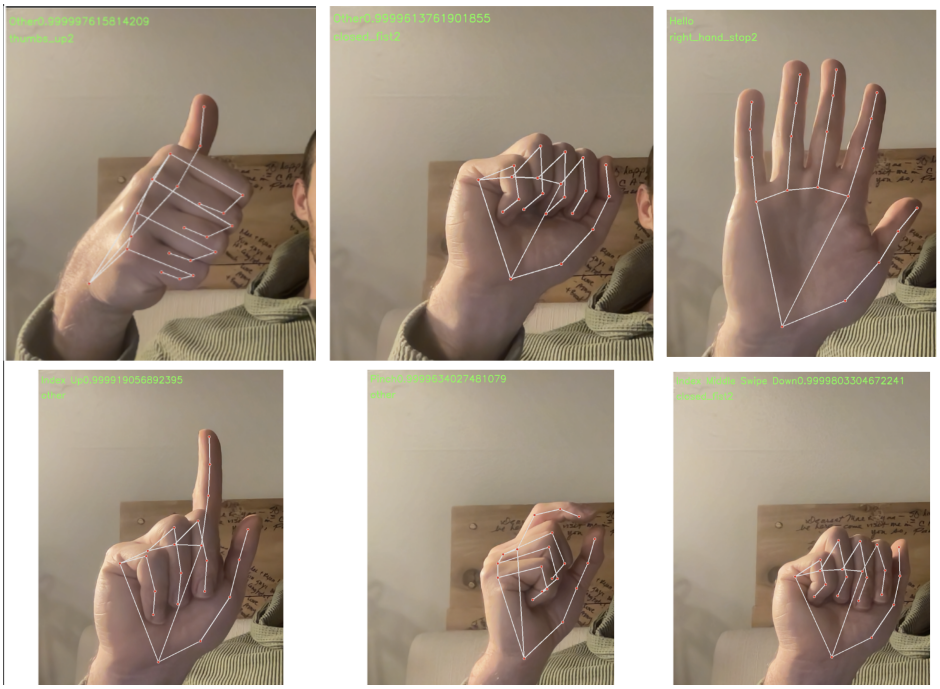

Figure 9: Results of Live Test (top images are static gestures, bottom images are the final state of dynamic gestures

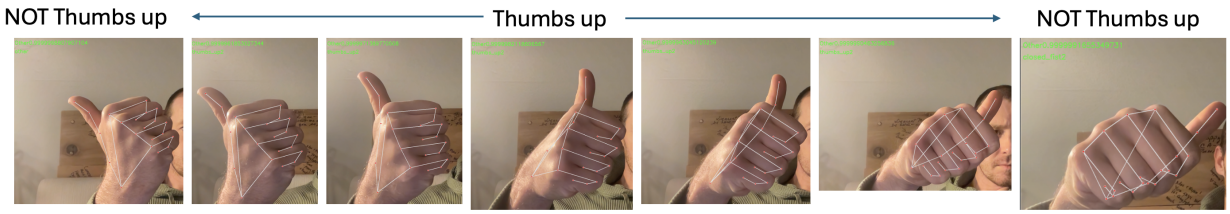


Figure 10: Enter Caption

The GRU dynamic gesture model's live detection was not as good as its precision, recall and accuracy would suggest. While it did detect the gestures, it seemed to struggle when the gesture speed was too fast. This was particularly notable with the index finger up gesture. This issue is inherent to the training data all requiring the same number of frames. Given this requirement, the biggest challenge of getting good dynamic gesture performance is balancing the speed of the gesture in the training data with the weighting of the end state of the gesture. For instance if I need a 3 second recording of index finger up and I want to do it very quickly then then the last 2 seconds of the recording may be the index finger being fully extended. This provides an unequal weighting that may cause the model to predict an index finger up dynamic gesture when in reality the index finger is just pointed up and never actually made the movement from the downward position to pointing up. Or in my case, the model doesn't recognize the fast gesture.

A final live test recording was prepared that contains all of the gestures. The static gestures had a 100% accuracy while the GRU dynamic gestures model results were not quite as good and are shown below in Figure 11.

Despite the dynamic model not being perfect I do think it could provide adequate performance for use in controlling devices.

To further understand how specific the models are to my hands, I had a second person record their selves performing the gestures and the model was applied to their recording. After evaluating the video, the static model performed very well on their hands as well.

The dynamic gesture model results for the others hands are displayed below in Figure 12. It is worth noting that the precisions are lower partially as a result of the user not completely executing the gesture initially and partially as a result of not waiting for the delay before moving on to the next gesture.

| | index middle swipe down | pinch | index up |
|---|---|---|---|
| recall | 82% | 94% | 92% |
| precision | 94% | 83% | 100% |
| average delay (frames) | 13.3 | 8.9 | 4 |
| average gesture length (frames) | 16.1 | 8.5 | 11.37 |
| average detection period ( frames) | 16.4 | 14.7 | 26.9 |

Figure 11: Live Test Results- SELF

| | index middle swipe down | pinch | index up |
|---|---|---|---|
| recall | 90% | 79% | 65% |
| precision | 62% | 100% | 95% |
| average delay (frames) | 8.0 | 15.0 | 4.64 |
| average gesture length (frames) | 10.1 | 12.4 | 4.9 |
| average detection period ( frames) | 13.5 | 20 | 12 |

Figure 12: Live Test Results- OTHER

**Conclusions and Future Application**

Overall, the detection of static gestures was much easier than detection of dynamic gestures. It requires far less data to build a suitable model and much less effort in preparing the recordings. For the static modeling, the MLP model far outperformed the KNN and slightly outperformed the SVM models. In all it took around 100 images of each image to have good MLP performance (F1 score $> 85\%$). For the dynamic modeling the size of the training set had to be quite large ($> 500$ recordings) to get good performance using a GRU or LSTM model.

There are two additional avenues I'd be interested in evaluating in regards to dynamic gestures. The first would be to test using a Graph Neural Network to see if the model would require less data to train and have better performance because of its inherent understanding of the relationship between the hand landmarks as opposed to considering them independently.

Another alternative that could very easily address the speed and data requirements of the training data is to actually train two static gestures (one representing the initial position of the dynamic gesture and one representing the final position of the dynamic gesture). The dynamic gesture could be recognized if these two static gestures are recognized within a defined number of frames.

After evaluating the aforementioned GNN and dual static gestures techniques, I am interested in developing an basic application that allows for the control of a web browser. This would include opening a new browser window, opening a new tab, scrolling up and down in the browser, switching between tabs, and zooming in and out on the website.

This basic application would allow me to have a better understanding of how well the model performs, its limitations and where to focus further improvements. After this works, I'd be interested in testing and incorporating pupil tracking to extend the list of potential applications.

Ultimately, I'd like to make this type of technology accessible to non-technical persons allowing them to record the data for their own personal gestures. This study is a good initial investigation into some of the challenges present in doing so.

## References


[1] Rahman, Md Mijanur, et al. "A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems." Expert Systems with Applications, vol. 266, Mar. 2025, p. 125929, https://doi.org/10.1016/j.eswa.2024.125929.

[2] Danish. "20bn-Jester." Kaggle, 16 Jan. 2020, www.kaggle.com/datasets/toxicmender/20bn-jester/code.

[3] Rahman, Md Mijanur, et al. "A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems." Expert Systems with Applications, vol. 266, Mar. 2025, p. 125929, https://doi.org/10.1016/j.eswa.2024.125929.

[4] "Hand Landmarks Detection Guide Google Ai Edge Google AI for Developers." Google, Google, ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker.Accessed25Apr.2025.

ppendix A

| | KNN3 Predictions | SVM Predictions | MLP Predictions | MLP {100,50} Predictions | MLP BNDO Predictions | PCA5-KNN3 | PCA6-KNN3 | PCA7-KNN3 | PCA8-KNN3 | PCA9-KNN3 | PCA10-KNN3 | PCA5-SVM | PCA6-SVM | PCA7-SVM | PCA8-SVM-8 | PCA9-SVM | PCA10-SVM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Thumbs Up** | | | | | | | | | | | | | | | | | |
| Total Count | 352 | 481 | 409 | 432 | 262 | 231 | 242 | 250 | 314 | 327 | 339 | 251 | 263 | 284 | 345 | 387 | 392 |
| Total TP | 242 | 404 | 389 | 393 | 244 | 204 | 213 | 217 | 234 | 236 | 239 | 232 | 247 | 263 | 290 | 321 | 322 |
| Total FP | 110 | 77 | 20 | 39 | 18 | 27 | 29 | 33 | 80 | 91 | 100 | 19 | 16 | 21 | 55 | 66 | 70 |
| Total FN | 188 | 26 | 41 | 37 | 186 | 226 | 217 | 213 | 196 | 194 | 191 | 198 | 183 | 167 | 140 | 109 | 108 |
| Precision | 0.69 | 0.84 | 0.95 | 0.91 | 0.93 | 0.88 | 0.88 | 0.87 | 0.75 | 0.72 | 0.71 | 0.92 | 0.94 | 0.93 | 0.84 | 0.83 | 0.82 |
| Recall | 0.56 | 0.94 | 0.90 | 0.91 | 0.57 | 0.47 | 0.50 | 0.50 | 0.54 | 0.55 | 0.56 | 0.54 | 0.57 | 0.61 | 0.67 | 0.75 | 0.75 |
| F1 | 0.62 | 0.89 | 0.93 | 0.91 | 0.71 | 0.62 | 0.63 | 0.64 | 0.63 | 0.62 | 0.62 | 0.68 | 0.71 | 0.74 | 0.75 | 0.79 | 0.78 |
| **Closed Fist** | | | | | | | | | | | | | | | | | |
| Total Count | 446 | 463 | 431 | 469 | 390 | 423 | 418 | 411 | 420 | 417 | 431 | 318 | 405 | 404 | 396 | 381 | 382 |
| Total TP | 297 | 344 | 344 | 356 | 303 | 267 | 286 | 288 | 291 | 288 | 291 | 240 | 303 | 320 | 319 | 313 | 312 |
| Total FP | 149 | 119 | 87 | 113 | 87 | 156 | 132 | 123 | 129 | 129 | 140 | 78 | 102 | 84 | 77 | 68 | 70 |
| Total FN | 91 | 44 | 44 | 32 | 85 | 121 | 102 | 100 | 97 | 100 | 97 | 148 | 85 | 68 | 69 | 75 | 76 |
| Precision | 0.67 | 0.74 | 0.80 | 0.76 | 0.78 | 0.63 | 0.68 | 0.70 | 0.69 | 0.69 | 0.68 | 0.75 | 0.75 | 0.79 | 0.81 | 0.82 | 0.82 |
| Recall | 0.77 | 0.89 | 0.89 | 0.92 | 0.78 | 0.69 | 0.74 | 0.74 | 0.75 | 0.74 | 0.75 | 0.62 | 0.78 | 0.82 | 0.82 | 0.81 | 0.80 |
| F1 | 0.71 | 0.81 | 0.84 | 0.83 | 0.78 | 0.66 | 0.71 | 0.72 | 0.72 | 0.72 | 0.71 | 0.68 | 0.76 | 0.81 | 0.81 | 0.81 | 0.81 |
| **Stop Sign** | | | | | | | | | | | | | | | | | |
| Total Count | 488 | 454 | 419 | 482 | 458 | 430 | 451 | 471 | 472 | 469 | 478 | 358 | 428 | 425 | 414 | 403 | 405 |
| Total TP | 413 | 414 | 382 | 432 | 413 | 350 | 398 | 401 | 407 | 402 | 407 | 342 | 403 | 403 | 397 | 392 | 394 |
| Total FP | 75 | 40 | 37 | 50 | 45 | 80 | 53 | 70 | 65 | 67 | 71 | 16 | 25 | 22 | 17 | 11 | 11 |
| Total FN | 73 | 72 | 104 | 54 | 73 | 136 | 88 | 85 | 79 | 84 | 79 | 144 | 83 | 83 | 89 | 94 | 92 |
| Precision | 0.85 | 0.91 | 0.91 | 0.90 | 0.90 | 0.81 | 0.88 | 0.85 | 0.86 | 0.86 | 0.85 | 0.96 | 0.94 | 0.95 | 0.96 | 0.97 | 0.97 |
| Recall | 0.85 | 0.85 | 0.79 | 0.89 | 0.85 | 0.72 | 0.82 | 0.83 | 0.84 | 0.83 | 0.84 | 0.70 | 0.83 | 0.83 | 0.82 | 0.81 | 0.81 |
| F1 | 0.85 | 0.88 | 0.84 | 0.89 | 0.88 | 0.76 | 0.85 | 0.84 | 0.85 | 0.84 | 0.84 | 0.81 | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 |
| | | | | | | | | | | | | | | | | | |
| Wtd Avg Precision | 0.74 | 0.83 | 0.89 | 0.85 | 0.86 | 0.76 | 0.81 | 0.80 | 0.77 | 0.76 | 0.75 | 0.88 | 0.87 | 0.89 | 0.87 | 0.88 | 0.87 |
| Wtd Avg Recall | 0.74 | 0.89 | 0.86 | 0.91 | 0.76 | 0.66 | 0.72 | 0.72 | 0.73 | 0.72 | 0.73 | 0.63 | 0.75 | 0.77 | 0.78 | 0.79 | 0.79 |
| Wtd Avg F1 Score | 0.74 | 0.86 | 0.87 | 0.88 | 0.80 | 0.69 | 0.75 | 0.75 | 0.75 | 0.74 | 0.74 | 0.73 | 0.80 | 0.82 | 0.82 | 0.83 | 0.83 |

Figure 13: Results from static gesture modeling