CS-594 Project report:

Prof. Glavic Boris
Leonardo Borgioli

9th December 2022

1 Note

All the plots can be seen in the GitHb repository the online preview version of the Notebook (code Gesture_prediction_provenance), has all the plot already loaded. For the Lime plots please refer to the Notbook plots as the function explainer.show_in_notebook() shows a much more comprehensive plot.

2 Introduction

The project aims to address an issue in a sensory glove-based interface designed to control a robotic-assisted surgical robot. This innovative interface enables surgeons to intuitively manipulate robotic arms using their hand movements. We propose a system that integrates an HTC Vive tracker, a Manus Meta Prime 3 XR sensory glove, and SCOPEYE wireless smart glasses. The system is designed to control one arm of a da Vinci surgical robot. In addition to moving the robotic arm, the surgeon can use their fingers to control the end-effector of the surgical instrument. Hand gestures are also employed to implement functions such as clutching.

Currently, however, gesture recognition lacks stability, occasionally causing the model to fail in accurately perceiving the intended gestures. This issue adversely impacts the user experience. The goal of this project is to utilize explanation techniques, specifically SHAP and LIME, to better understand and interpret the model's decision-making process.

This project summary will outline the steps implemented in the associated Jupyter notebook and provide an overview of the methodology.

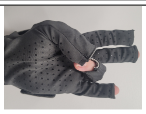
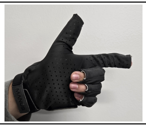
Hand Gesture	**Representation**	**Hand Gesture**	**Representation**
Ring		**Clutch**	

Table 1: Gestures trained for distinct functionalities. **Clutch** invokes the clutch mechanism (of the dVRK robot) and **Ring** turns the robot control on or off. If neither of these two gestures is recognized, the system returns **None**.

3 Preprocessing

This section of the code focuses on loading the data and creating the DataFrame that will be used to train the models. The data generated by the da Vinci Research Kit (the surgical robot used) is in the '.bag' format. This data can be easily transformed into a DataFrame (using the Pandas library) and saved as a '.csv' file. For this project, we work by loading the '.csv' file instead of directly reading the '.bag' files, as reading '.bag' files requires the use of the ‘bapy‘ library, which depends on the Robot Operating System (ROS). Note that ROS is only compatible with Ubuntu systems.

The sensory glove generates a skeleton mesh composed of 21 nodes. It is important to note that the available data represents the relative pose of each node to a virtual base defined at the start of the glove. For this reason, we chose to work exclusively with relative rotation, as it exhibits significantly less variance across different runs compared to relative translation. To achieve this, we dropped the translational features from the dataset. Additionally, other preprocessing steps were performed, such as renaming features to make them more user-friendly. Please refer to the Jupyter notebook for visualizations of the resulting DataFrame.

4 Data Visualization

As the name suggests, the purpose of this section is to visualize the data. In the Jupyter notebook, an example of the skeleton mesh generated by the sensory glove is provided. Additionally, the different fingers have been labeled to enhance clarity. This section also serves to verify data integrity, ensuring consistency throughout the various import and export processes.

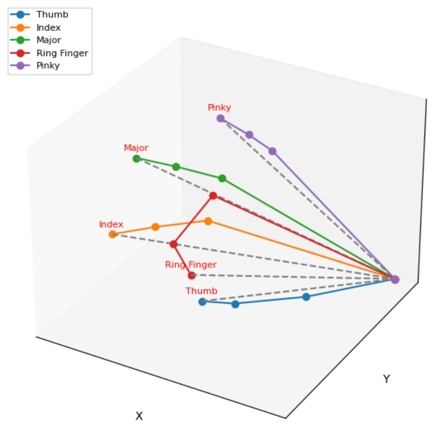
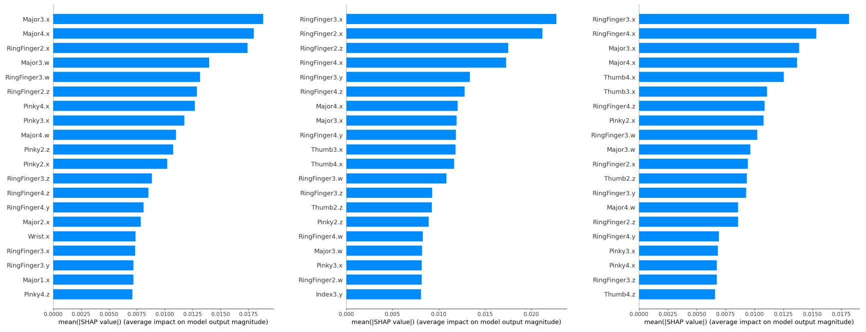


Figure 1: Hand Skeleton Mesh

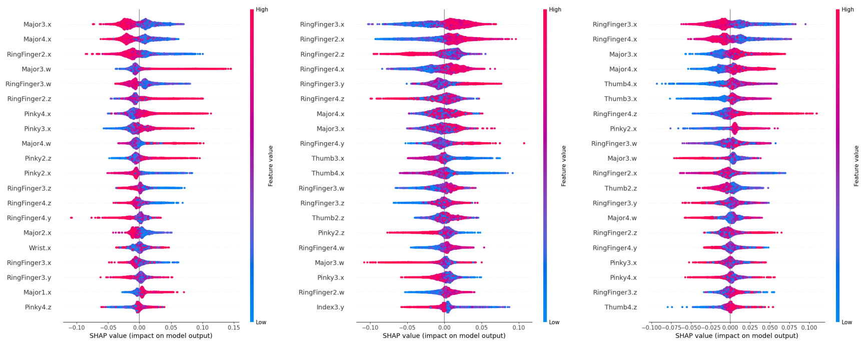
5 Random Forest

Random Forest is the first model trained; the hyperparameters have been selected using a GridSearch (3 different values for each parameter and a CV of 5). The model on paper performed well (Accuracy 0.968, F1 score 0.968, and Recall 0.968); however, when tested in real cases, depending on the angle of the user, it would perform much poorly. Firstly the Shap model has been used, the results are consistent; we can see that for the RF gesture (second plot), the decision was made by looking at the major and Ring Finger, followed by the other fingers. In contrast, for the clutch gesture, a broader analysis of the fingers is performed, as executing a clutch requires closing the RF, major, and pinky fingers. The None class will more do a complementary analysis.



(a) Class 0 (b) Class 1 (c) Class 2

Figure 2: SHAP Summary Plots for Classes 0 (Clutch), 1 (Ring Finger), and 2(NONE)



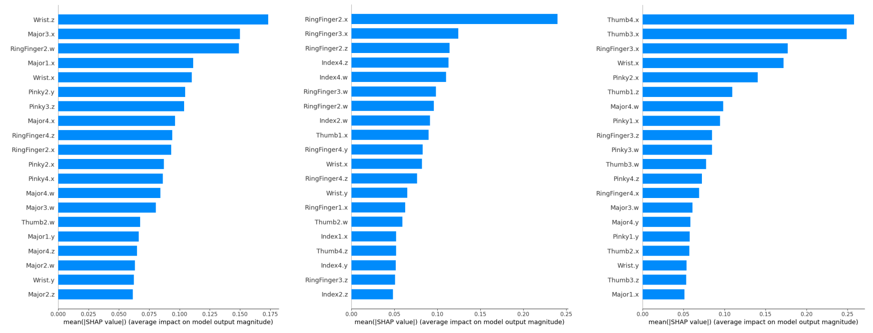
(a) Class 0 (b) Class 1 (c) Class 2

Figure 3: SHAP Summary Plots making visible the positive and negative classes for Classes 0 (Clutch), 1 (Ring Finger), and 2(NONE)

After the Lime model has been used. In the case of Lime, we can see that the considerable size of the dataset firstly causes an issue in a correct visualization. Also, the result is less consistent (pointing mainly the pinky as explanation of a clutch motion) than in the Sharp model. Here below we will try to train the model with a much less size of training data to see the difference, we do a randomsapling of the original data by using the sklearn train_test_split functionality (first to define a training size of 10% of the original and the to split it in train teest sets).

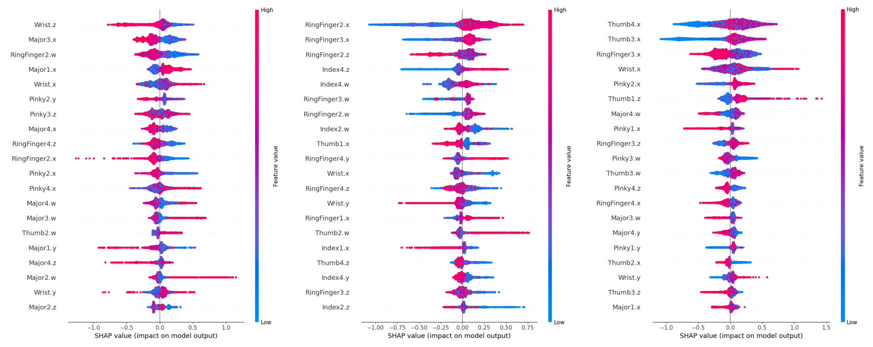
6 Light Gradient Boosting Machine

The second model is is the LGBM, currently used in the project. Still some poses were causing issues and flickering in the prediction.



(a) Class 0 (b) Class 1 (c) Class 2

Figure 4: SHAP Summary Plots for Classes 0 (Clutch), 1 (Ring Finger), and 2(NONE)



(a) Class 0 (b) Class 1 (c) Class 2

Figure 5: SHAP Summary Plots making visible the positive and negative classes for Classes 0 (Clutch), 1 (Ring Finger), and 2(NONE)

As observed earlier, the LIME model differs from SHAP but starts aligning with it as the sample size decreases. In this case, we see that the wrist plays a significant role in the prediction. This explains why the model is more stable: when the user shifts their hand poses, considering only the ring finger itself can result in incorrect predictions (recall that every pose is provided as a relative transformation to a virtual base set at the beginning). This also highlights why the RF model was unstable, as it focused only on the relative rotations of the nodes of specific fingers. However, this also reveals the limitation of the current model. Ideally, for the selected gestures, only the positions of the fingers should be considered (which is currently not possible as the pose is computed to a virtual base). To address this, we are working on adapting the point cloud by computing the relative pose of each node to its preceding one.

7 Conclusion

The application of explanation techniques like SHAP and LIME, which provide insight into the decision-making processes of the gesture recognition models. These techniques have been instrumental in identifying weaknesses in the models, particularly in how they sometimes fail to recognize gestures correctly. By analyzing the importance of different features in the model's predictions, SHAP and LIME helped discover patterns and inconsistencies in the data. A comparison with different dataset sizes was performed to evaluate the performance of both models. It was shown that LIME struggled to maintain coherent explanations when the dataset was too large, and the same issue occurred with the computation time (as seen in the notebook cells).