In the creation of medical equipment, usability testing is critical. It ensures patient safety, checks planned functioning, and gives a technical edge. Its functionality is evaluated to determine if it is appropriate for usage in the rapidly expanding field of eHealth applications.

The performance of the former algorithm using HMM which is basically a •Prediction based model that uses the previous and current state and an ensemble of probabilities to achieve predictions
yielded excellent results as it achieves a 85 % accuracy in task sequencing . However we still want to improve it , as it is still facing bugs sometimes where it jumps from action to action. The data is an experiment carried out on multiple participants who had to perform 7 different actions using a pen.

EXPLAIN THE DATA.

The main remaining challenge is now to develop an automated task sequencing framework in order to automate the whole process. In order to do so, we are using camera glasses which record everything in the user's field of vision .

The initial idea was to use MTL:
In Machine Learning (ML), we typically care about optimizing for a particular metric, whether this is a score on a certain benchmark .
In order to do this, we generally train a single model or an ensemble of models to perform our desired task. We then fine-tune and tweak these models until their performance no longer increases. we sometimes ignore information that might help us do even better on the metric we care about. Specifically, this information comes from the training signals of related tasks. By sharing representations between related tasks, we can enable our model to generalize better on our original task. This approach is called Multi-Task Learning (MTL).
A layer would be subdivided into n different layers and each of them would only focus on learning a single action . For example, sub_lay 1 would focus on Cap on/off, layer 2 on flipping the pen and so on and so forth .
However, this has never been done before this way, and due to the lack of documentation and knowledge, we decided to try a different approaches.

Therefore we Decided to try two different approaches . Unsupervised and supervised learning. The idea was to try to use clustering on the data, using KMeans algorithm, to end up with numerous clusters and interpretable data. The second approach was to give raw data as an input to a NN and to try to predict the action , label.

Each one has its own pros and cons . Unsupervised learning does not need as much resources as supervised.No risk of overfitting . Easy go find how many classes are there before giving data for training.

As we can see on the results , the first try wasn't very convincing as we only achieved a 15% accuracy. Thus we tried to find the optimal number of clusters for the Kmeans algorithm to achieve a reasonable accuracy.
We tried to use silhouette coefficients: **Silhouette** refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified.[1]
The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

As we can see on the graph, the optimal number of clusters should be 16. As we couldn't understand what it represents, we decided to try to cluster rows 2 by 2.
Future ==> ??

We only achieved 20% accuracy using neural Networks . Therefore we had to try a new approach.
MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc.
It has multiple applications such as Face detection , face mesh etc… . We will be interested in hand detection

This is how it works : the framework is a google model trained over 1M images and it basically detects a bunch of landmarks on your hand, which are classified and identified as shown in the figure.
Our algorithm also presents several features as it prints all the landmarks on the video and detects whether the detected hand is left or right and outputs the x and y coordinates of each landmark in the image frame, the origin being top left corner. As you can see we have different landmarks and the main goal is to connect the hand data (left right, coordinates of different landmarks) with the glasses data to end up with a proper prediction.

To achieve such a project, we have to choose a high performance metric. We have to choose between object-gaze, gaze-hand or hand-object and we could also combine all three of them. I have the intuition that hand object distance is the most interesting and most promising metric, but we still have to show it in a proper way.
Now we still have to come up with a new neural network which combines both the object/gaze information and the hand detection information and outputs a prediction on the action.