Hand Gesture Interface

Team Members:

- Matthew Kasper (NetID: mtkasper)
- Matej Popovski (NetID: popovski)
- Aniket Patel (NetID: apatel267)

Problem we are trying to solve:

- Want to create a purely hand gesture-based system for interacting with computers as traditional high-fidelity interfaces, such as keyboards, mice, and remotes, are not convenient for some types of computers and people.

Why we want to solve it:

- A gesture-based system can provide a more personalized and comfortable method of interacting with computers to those suffering from arthritis, carpal tunnel syndrome, or other conditions that limit hand mobility.
- This system can also make Interaction with some types of computers, such as VR/AR systems and smart TVs, more convenient for a general user when compared to traditional control methods.
- Creating a gesture-based control system appears to be a highly engaging and challenging project.

Current State of the Art:

- There are many methods for hand detection and gesture classification.
  - YOLO algorithms
  - Cascade Classifier
  - ResNet
  - MediaPipe
  - TraHGR: Transformer for Hand Gesture Recognition via Electromyography - Uses electrical activity of muscles and nerves to classify gestures
  - Visual Hand Gesture Recognition with Deep Learning – survey of static, dynamic, and continuous methods, highlighting transformer models, multi-modal fusion, weak supervision, and data challenges.

- While there exists purely gesture-based interfaces in some devices, most are proprietary, inflexible, require special hardware, and are generally only for use on one system. Examples are listed below.
  - Meta's hand tracking system on the Oculus VR goggles.
  - Microsoft Kinect hand tracking for the Xbox system.

Novel or Re-Implementation

- There already exist hand detection and gesture classification systems, but our goal is to build one from scratch ourselves to better understand the underlying techniques and implementation details. This approach is challenging, but it will give us deeper insight and allow us to customize the solution as needed.

- We plan to start by developing our own code (from scratch), but if necessary, we will integrate selected external libraries to enhance functionality and improve performance.
- For example, we can use MediaPipe Hands for landmark detection (existing solution) but design your own classification logic or ML model on top of those landmarks.
- In terms of hand gesture-based interfaces for computers, many systems exist but are inflexible and require specialized, expensive hardware.
- Our system will only require a color webcam and can be run directly on the user's computer, making it much more accessible than the alternatives and a novel expansion on existing ideas.

.

Possible steps to solve the problem:

- Create an object detection system to create bounding boxes around the hand in an image. Or use existing methods to create the bounding boxes, such as YOLO or a cascade classifier.
- Create a custom neural network using pytorch or tensorflow to classify images of hands to specific gestures.
- Extract key features (landmarks, distances, angles) from the detected hand.
- Based on the classified gesture, perform certain actions (mouse movement, mouse click)
- Create a simple project webpage to host information about our project.

Performance Evaluation

- The performance of the gesture recognition will be measured using standard methods: we will have a training and testing set of hand gesture images and determine success rate on each.
- Furthermore, we will also be performing real-time tests to make sure the model predicts gestures correctly given a video feed and feels responsive and accurate to the user.

Rough Time:

- **Weeks 1–4:** Define project scope and target gestures; set up environment and libraries (OpenCV, MediaPipe, etc.).
- **Weeks 4–8 (30 Oct):** Implement object detection (bounding boxes) of hand and static gesture classification.
- **Weeks 8–13 (11 Dec):** Map hand position and gestures to actions on the computer. Create a website to present the project.