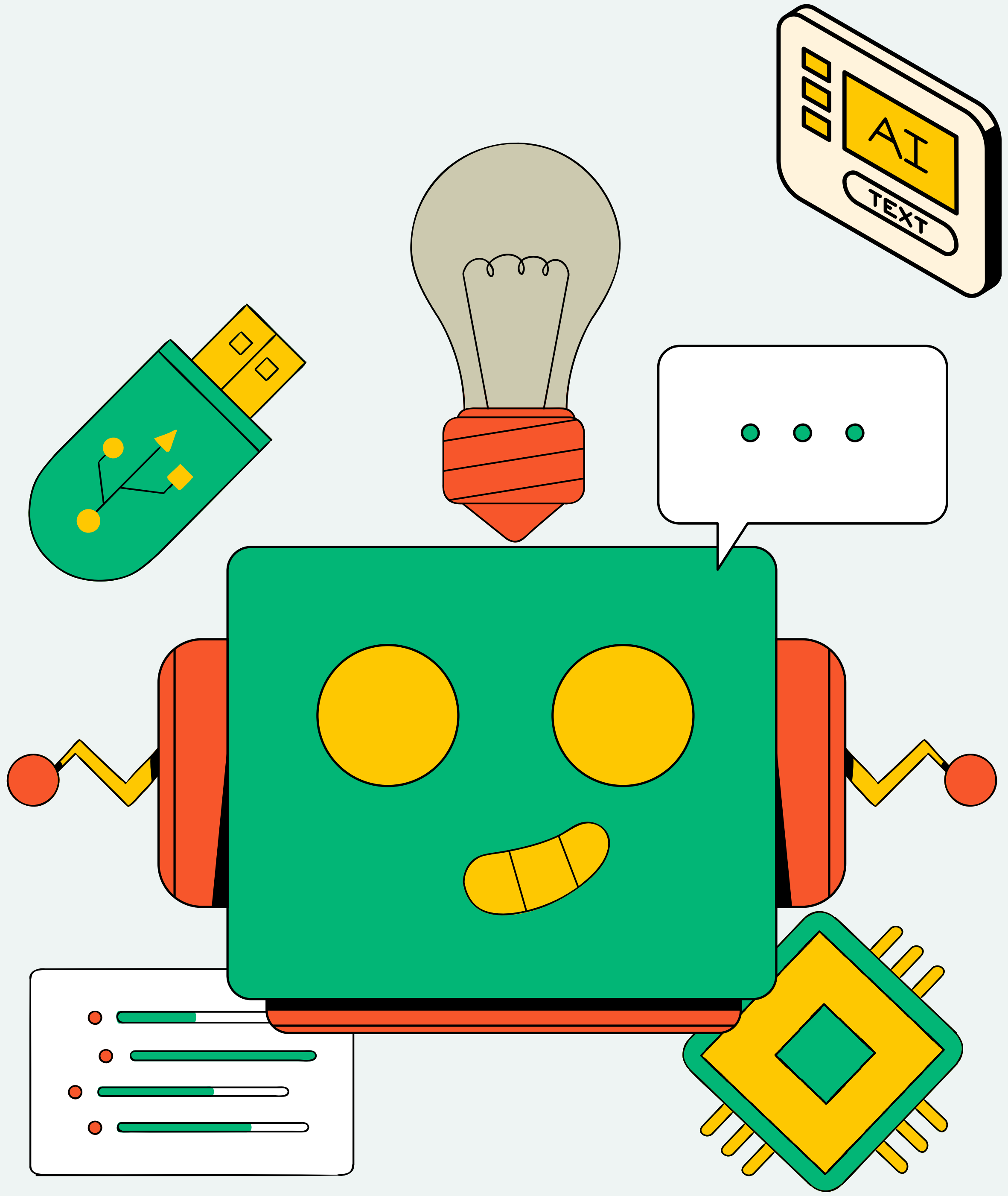
NATURAL LANGUAGE PROCESSING - CV(UCS672)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

WE LEARN FOR THE FUTURE



VIRTUAL MOUSE

PRESENTATION

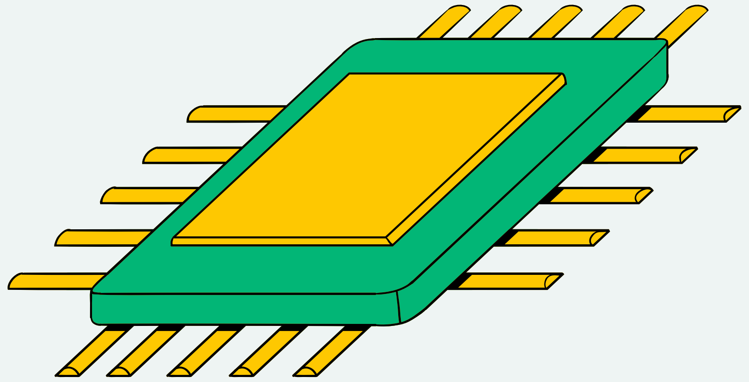
PRESENTED BY:

MAYANK GUPTA(102103247)

3COE9

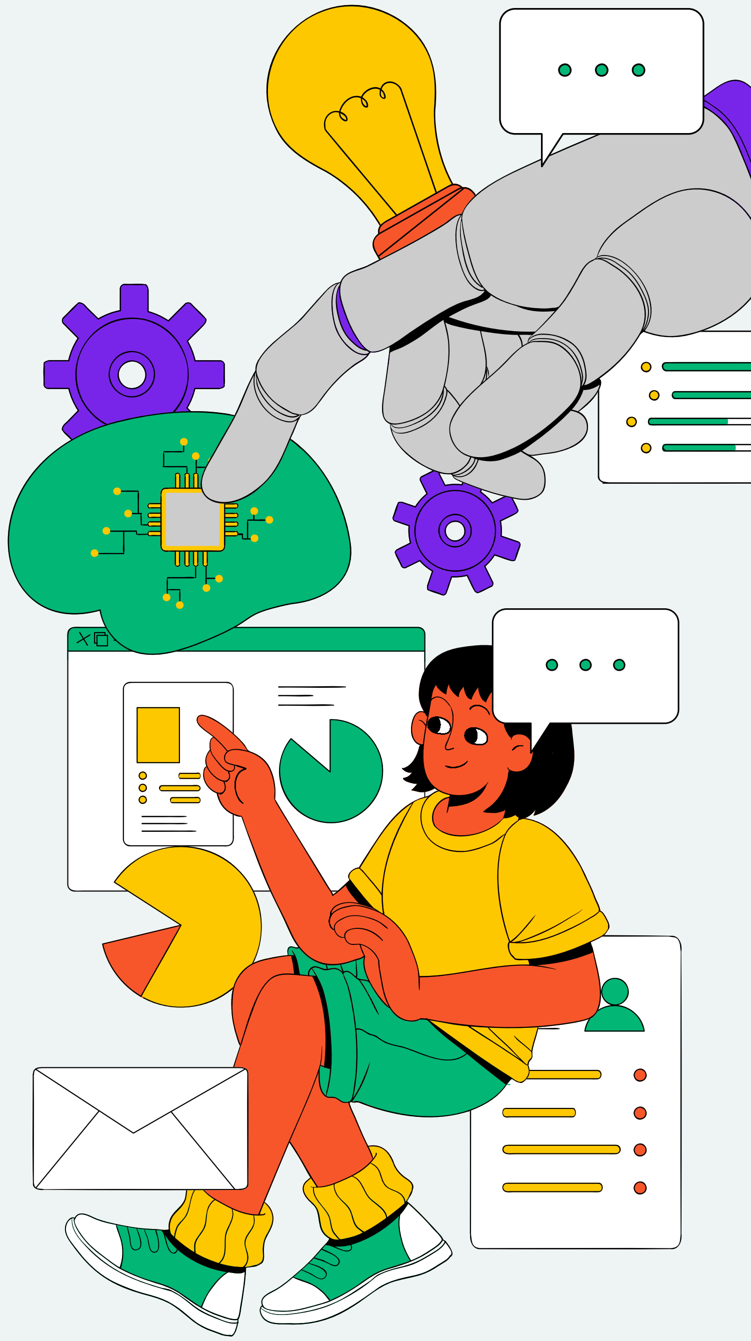
PRESENTED TO:

DR. SEEMA WAZARKAR



OBJECTIVES

- Develop a real-time hand gesture recognition system using MediaPipe for accurate detection.
- Create virtual mouse control functionality enabling users to interact with their computer screens via hand gestures.
- Optimize system performance for real-time operation through frame size reduction and model optimization.
- Enhance user experience with customizable gesture sensitivity and cursor speed options.



METHODOLOGY

- Capture video frames from the webcam using OpenCV's VideoCapture module
- Utilize MediaPipe's Hands library to detect hands and track landmarks in each frame.
- Map the coordinates of detected hand landmarks to the screen resolution for accurate cursor positioning.
- Analyze the spatial relationship between key landmarks (e.g., index finger and thumb) to determine mouse actions, such as clicking or cursor movement.
- Simulate mouse actions (e.g., clicking, moving) using PyAutoGUI based on the detected gestures and mapped coordinates.

IMPLEMENTATION

SETUP AND CONFIGURATION:

- Install necessary libraries such as OpenCV, MediaPipe, and PyAutoGUI.

VIDEO FRAME CAPTURE AND PREPROCESSING:

- Initialize the webcam and capture video frames using OpenCV's VideoCapture module.
- Preprocess each frame, including flipping for correct orientation and converting to RGB color space.

IMPLEMENTATION

HAND DETECTION AND LANDMARK TRACKING:

- Utilize MediaPipe's Hands library to detect and track hand landmarks in each frame.
- Draw landmarks on the frame for visualization and debugging purposes.

GESTURE RECOGNITION AND ANALYSIS:

- Identify relevant hand gestures, such as index finger pointing and thumb positioning, for mouse control actions.
- Analyze the spatial relationship between key landmarks to determine the intended mouse action (e.g., clicking, cursor movement).

IMPLEMENTATION

COORDINATE MAPPING AND CURSOR CONTROL:

- Map the coordinates of detected hand landmarks to the screen resolution for accurate cursor positioning.
- Simulate mouse actions (e.g., clicking, moving) using PyAutoGUI based on the detected gestures and mapped coordinates.

USER INTERACTION AND CUSTOMIZATION:

- Provide visual feedback to the user by overlaying detected landmarks and cursor movements on the webcam feed.
- Allow users to customize settings (e.g., gesture sensitivity) for personalized interaction.

CONCLUSION

Our virtual mouse control system successfully enables intuitive interaction with computer screens through hand gestures. With smooth responsiveness and customizable options, it enhances user experience while ensuring robust performance and reliability. Looking ahead, there's potential for further innovation and collaboration to expand gesture-based interaction technologies, shaping the future of human-computer interaction.