AMC ENGINEERING COLLEGE

Department of Information Science and Engineering

PROJECT PHASE-2 ON

CAMERA BASED INTERACTIVE COMPUTER FUNCTIONS USING HAND GESTURES

GUIDED BY	PROF AMUTHA R

PRESENTED BY		
SL.NO	NAME	USN
1	ABHISHEK N NAIRY	1AM20IS004
2	D N MEGHANA	1AM20IS023
3	NAMRATHA P	1AM20IS058
4	SANCHALI A	1AM20IS083

ABSTRACT

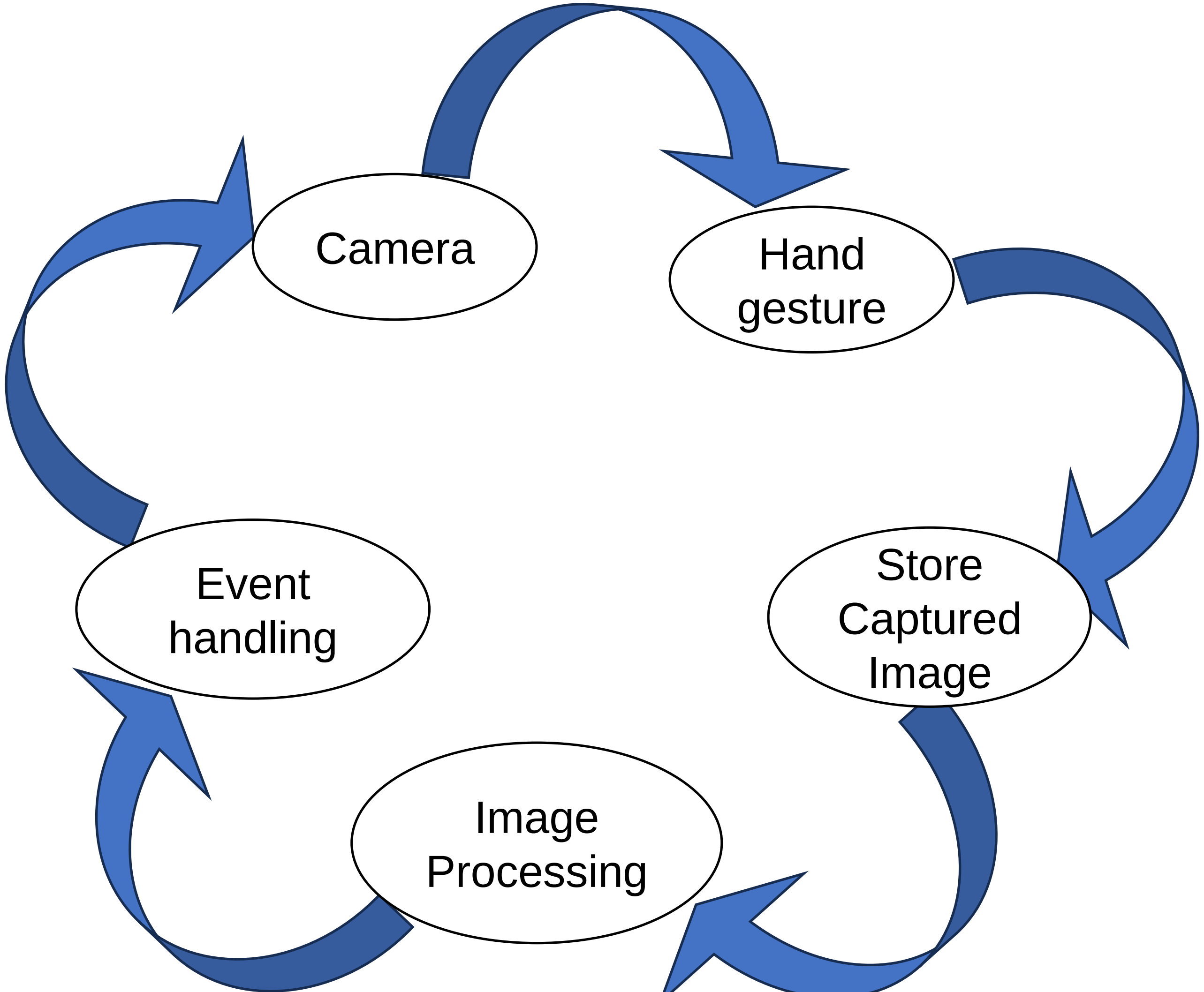
The recognition of Hand gestures has become a critical point as it is widely used in everyday applications. The challenge in this is to improve the recognition effect and develop a fast recognition method. Glove and led-based methods involve external devices in detecting and interpreting hand gestures, making human-computer interaction less natural. So, different approaches have been used previously that use purely hand gestures in many systems based on human-computer interaction. This system provides a more natural human-computer interaction; it must be made efficient processing speed of classifying the test data (images) from among the training data (database stored for gestures recognition).This speed makes gesture recognition more effective and reliable to use as compared to previously proposed methods. In this research paper, a proposed system based on a camera based interactive wall display using bare hand gestures with efficient processing speed for controlling the speed of the mouse and other functions. This system has three modules: one uses Genetic Algorithm and Otsu thresholding to identify the query images as the right or wrong gesture and perform the correct action in case of the proper motion, another module controls functions outside of PowerPoint files or Word documents, e.g., to open folders and go through drives, and the third module uses the convexity hull method for finding the number of fingers open in the user’s gesture and operates accordingly.

DATA FLOW DIAGRAM

Level-0

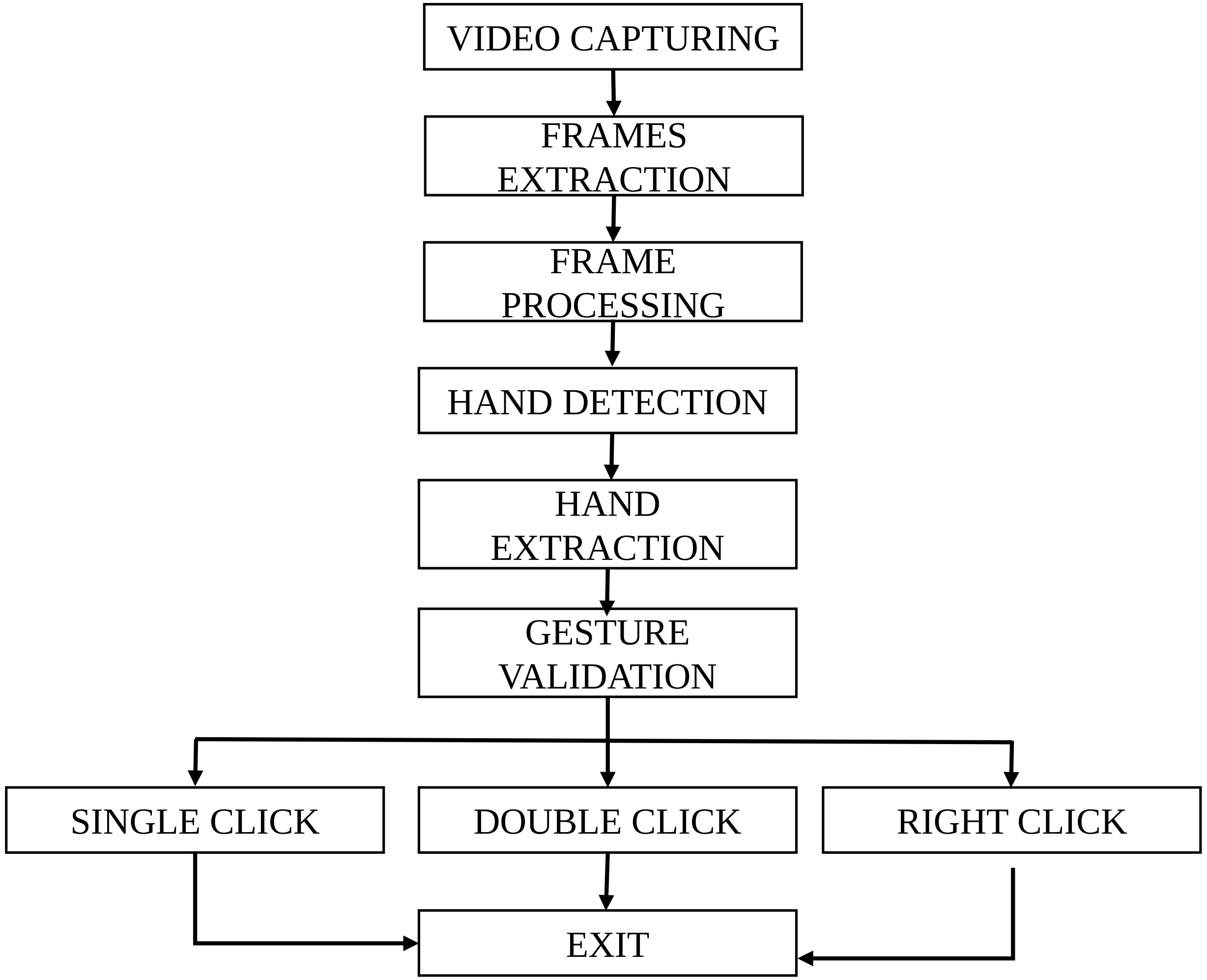
DATA FLOW DIAGRAM

Level-1



DATA FLOW DIAGRAM

Level-2



MODULES SPLITUP

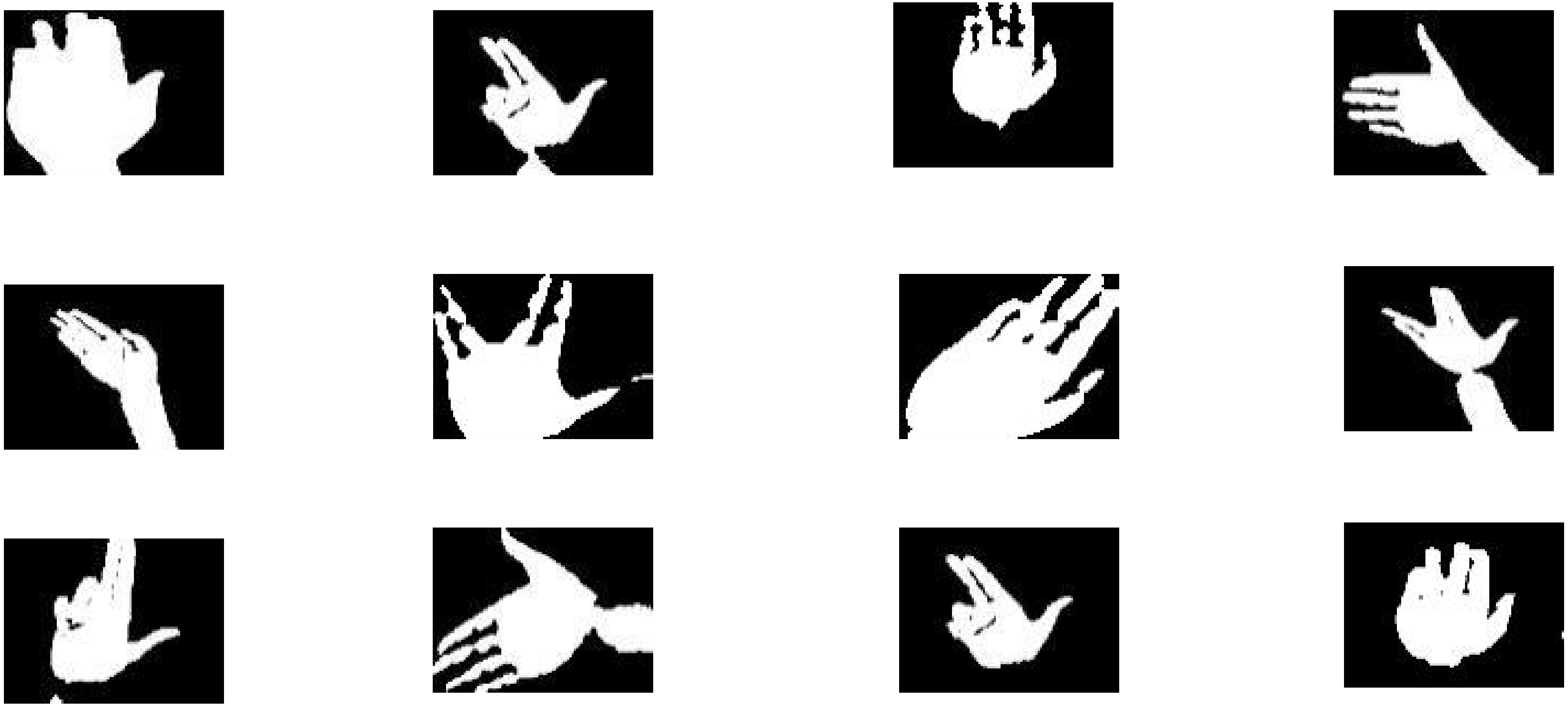
- Dataset Gathering
- Preprocessing dataset
- Feature extraction
- Algorithm modules
- Gesture recognition and validation
- Getting the accuracy of the algorithm

MODULE DESCRIPTION

Dataset Gathering

- Creating a dataset for camera-based hand gesture recognition in machine learning involves several critical considerations. The dataset should comprise a collection of images or videos capturing diverse hand gestures, ensuring a range of hand poses, orientations, and lighting conditions for the model's robustness. This diversity is crucial to enhance the model's ability to generalize effectively in real-world scenarios.
- In terms of features, the dataset must include high-resolution image data annotated with corresponding gesture labels. Additionally, environmental factors such as lighting conditions and background should be considered and documented to account for potential variations during recognition.
- Ethical and legal standards play a pivotal role in the data collection process. Adhering to ethical guidelines, respecting privacy, and obtaining appropriate consent are imperative. Furthermore, compliance with data protection regulations is necessary to ensure the responsible and lawful use of the collected data.
- Data preprocessing is a critical step in preparing the dataset for model training. Standardizing images to a common resolution, augmenting the dataset to introduce variability, and normalizing and cleaning the data contribute to consistency and noise reduction.
- Collaboration with researchers in computer vision and gesture recognition is recommended to leverage their expertise. Seeking advice on model architecture, training strategies, and evaluation metrics specific to hand gesture recognition can significantly enhance the overall effectiveness of the project.

SAMPLE DATASETS



NUMERICAL DATASETS

Gesture Name	Input Images	Validated Images	Recognition Rate=(input/validated)*100	Function Against Gesture
Gesture with all fingers closed	60	56	93.33	System shutdown
Gesture with index finger open	60	55	91.66	Mouse cursor control
Gesture with index and middle finger open	60	57	95	Single click
Gesture with index,middle and ring finger open	60	58	96.66	Double click
Gesture with index,middle,ring and little finger open	60	55	91.66	Right click
Gesture with all five fingers open	60	54	90	Exit

Table 1

Gesture Name	Input Images	Validated Images	Recognition Rate (%)	Function Against Gesture
Gesture with only index and small finger open	60	59	98.33	Show slide sow of PowerPoint
Gesture with all fingers closed	60	54	90	Scroll down to the next slide or page
Gesture with only thumb open	60	55	91.66	Scroll up to the previous slide or page
Gesture with all five fingers open	60	57	95	Close the currently open file or explorer window

Table 2

Preprocessing Dataset

- **Labeling:** Annotate each data point with corresponding gesture labels to create a supervised learning dataset.
- **Data Cleaning**: Remove irrelevant frames, noise, or outliers to enhance the quality of the dataset.
- **Normalization**: Normalize pixel values or depth information to ensure consistent input features for the model.
- **Grayscale Conversion:** Convert the images to grayscale to simplify processing.
- **Noise Reduction:** Apply filters (e.g., Gaussian blur) to reduce noise and enhance feature visibility.
- **Thresholding:** Convert the image to binary form for easier feature identification.

Feature extraction

Extract relevant features from the segmented hand region. Features are characteristics that distinguish one gesture from another. Common hand gesture features include:

- **Contour Analysis:** Extract the contour of the hand to identify its shape and size.
- **Finger Count:** Count the number of fingers, which can be determined by finding peaks in the contour.
- **Hand Orientation:** Analyze the orientation of the hand in relation to the camera.
- **Movement Direction:** Track the direction of hand movement over consecutive frames.

Gesture recognition and validation

- To create a reliable camera-based interactive system using hand gestures, start by gathering a diverse dataset of relevant hand movements.
- Clean, normalize, and enhance this dataset for quality. Choose an appropriate deep learning model, like a pre-trained, and train it with your prepared dataset, fine-tuning parameters for optimal performance. Establish a mapping between recognized gestures and specific computer interactive functions.
- Implement the model for real-time detection in your application and use a separate dataset for validation. Conduct user testing to gather feedback on responsiveness and accuracy. Fine-tune probability thresholds based on user input and ensure adaptability to various hand sizes and movements.
- Implement a user-friendly feedback mechanism for recognized gestures, address security concerns, and continuously improve by gathering new data for periodic model retraining. Provide clear documentation on recognized gestures and associated functions to enhance user understanding.

Getting the accuracy of the algorithm

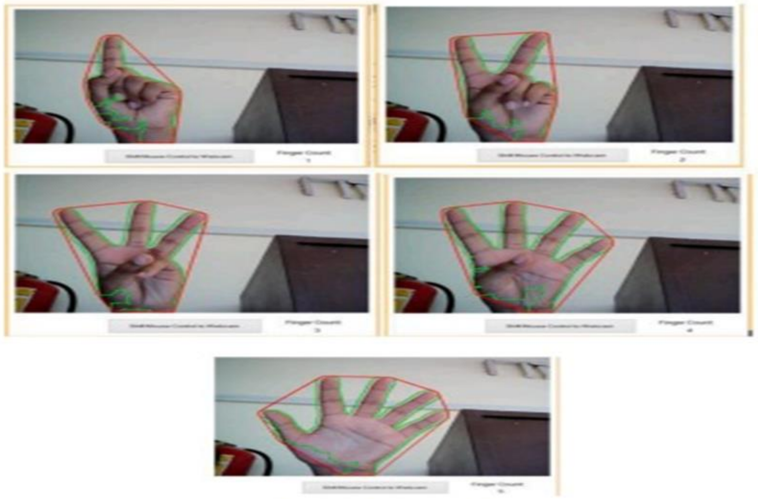
- Data Collection and Labeling:
  - Gather a diverse dataset of hand gesture images or videos that represent the range of actions your system needs to recognize.
  - Ensure each sample is accurately labeled with the corresponding gesture.
- Training the Model:
  - Use a suitable machine learning or computer vision algorithm to train your model on the labeled dataset.
  - Split the dataset into training and testing sets to evaluate the model's performance.
- Validation and Fine-Tuning:
  - Validate the model on a separate validation set to identify potential overfitting or underfitting issues.
  - Fine-tune the model parameters based on validation results to optimize its performance.
- Performance Metrics:
  - Monitor these metrics during training and testing phases to assess overall performance.
- Cross-Validation:
  - Perform cross-validation by training and testing the model on different subsets of the dataset.
  - This helps ensure that the model generalizes well to unseen data and isn't overly dependent on a specific set.
- Real-world Testing:
  - Evaluate the model's accuracy in real-world scenarios by deploying it in the intended environment.
  - Test the system with various users and lighting conditions to assess its robustness and reliability.

Algorithm Modules

- Otsu Thresholding : The Otsu method is an image thresholding technique used to automatically find the optimal threshold value for image segmentation. It calculates the threshold by maximizing the inter-class variance between the background and foreground pixels. By iteratively finding the threshold that minimizes intra-class variance, it efficiently separates objects from the background in grayscale images, aiding In image processing and analysis tasks.
- Generic Algorithm :A generic algorithm is a search technique that mimics natural evolution to find optimal solutions for a given problem. It works by creating and evolving a population of individuals, each representing a possible solution. The individuals with the highest fitness scores are more likely to survive and reproduce, passing on their genes to the next generation. The process repeats until a termination condition is met.

Predict result and analysis

- The predicted results and analysis of hand gesture detection for computer interactive functions depend on the effectiveness of the trained model and its integration into the interactive application. Upon deployment, the system should be capable of accurately recognizing and interpreting various hand gestures, translating them into specific computer commands or functions.
- The analysis of predicted results involves evaluating key performance metrics such as accuracy, precision, recall. High accuracy indicates the model's overall correctness in recognizing gestures, while precision and recall offer insights into the system's ability to minimize false positives and false negatives, respectively.



Test results of hand gestures for mouse cursor control.

Gesture Name	Function Against Gesture
Gesture with all fingers closed	Show slide show of PowerPoint
Gesture with all five fingers open	Scroll down to the next slide or page
Gesture with only index and small finger open	Scroll up to the previous slide or page
Gesture with only thumb open	Close the currently open file or explorer window

Test results of hand gestures for functions of powerpoint and word files.

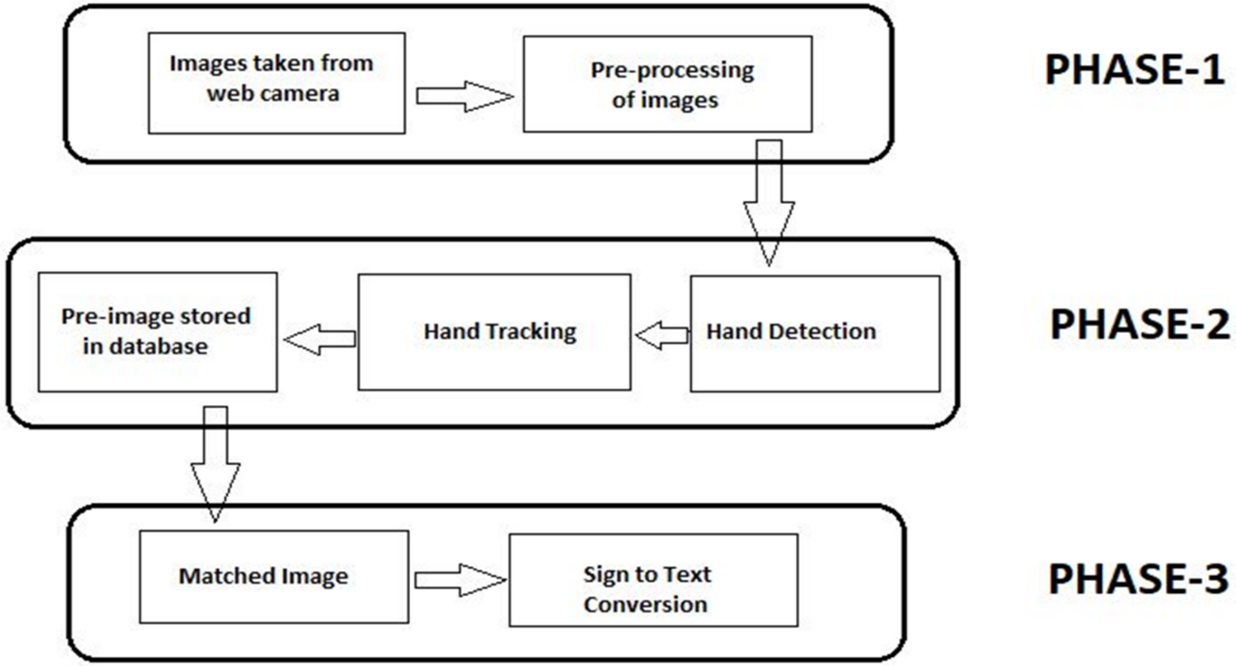
Gesture Name	Function Against Gesture
Gesture with all fingers closed	System Shut down
Gesture with index finger open	Mouse cursor control
Gesture with index and middle finger open	Single click
Gesture with index, middle and ring finger open	Double click
Gesture with index, middle, ring and little finger open	Right click

Detailed Design

- In designing a hand gesture detection system for computer interactive functions, the process begins with a meticulous analysis of the specific interactive requirements. Subsequently, a diverse and annotated dataset is collected, undergoing preprocessing steps like cleaning, normalization, and augmentation to enhance its quality.
- The data representation is then determined, choosing between image frames, depth maps, or keypoint coordinates. The model selection involves choosing a deep learning architecture, with the option of leveraging transfer learning for improved performance. Following model training, a comprehensive gesture mapping is established, linking recognized gestures to distinct computer interactive functions.
- The integration of real-time detection into the interactive application is implemented, and a separate validation dataset is utilized to fine-tune the model and assess its accuracy. Continuous user testing and feedback collection inform adjustments to parameters like probability thresholds, ensuring a balance between precision and recall. The adaptability of the model to diverse users and its capacity to handle variations in hand sizes and shapes are crucial considerations.
- The implementation of a feedback mechanism, security measures to prevent unintended actions, and comprehensive documentation for developers and users contribute to a well-rounded system. Continuous improvement through periodic updates and scalability considerations further enhance the robustness of the hand gesture detection system, providing a seamless and user-friendly interactive experience.

ARCHITECTURE

The research was carried out in three phases:



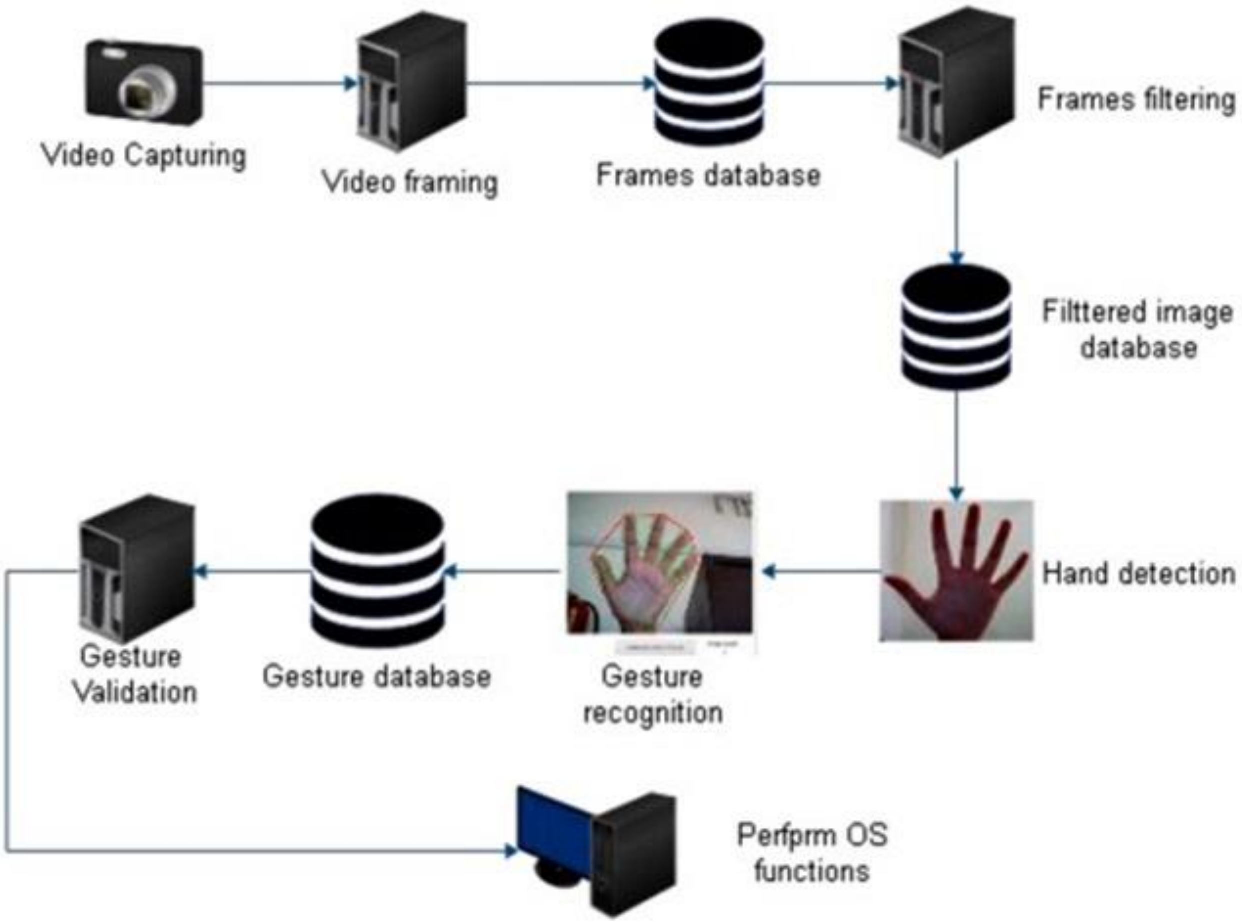


Figure : System Architecture

SYSTEM ARCHITECTURE

As shown in the following flowchart fig 1:

- Hand gestures are captured and recorded as single frames, which undergo preprocessing such as smoothing and resizing.
- The hand detection module receives the preprocessed image, performs hand extraction to crop the hand region, and prepares the image for gesture validation.
- Gesture validation involves comparing the test image's gesture with dataset images, all represented as binary images. If a match is found, the corresponding OS operation is executed.
- The system uses a camera compliant with USB 2.0 Interface and MS Windows XP Service Pack 3, with features such as autofocus, up to 2-megapixel resolution, and video capture up to 1600 × 1200 pixels (HD quality).
- The camera handles varying lighting conditions, utilizing autofocus for optimal results. Dim lighting may introduce noise and blur, while bright light is preferred over sunlight. The system comprises two major modules: Gesture Validation and Cursor Control, with the former's steps detailed in a flowchart.

REFERENCE

[1] Ning-Ning Xie, Fang-Fang Wang, Jue Zhou, Chang Liu, Fan Qu, “Establishment and Analysis of a Combined Diagnostic Model of Polycystic Ovary Syndrome with Random Forest and Artificial Neural Network”, Hindawi BioMed Research International Volume 2020.

[2] A. Saravanan, S. Sathiamoorthy, “Detection of Polycystic Ovarian Syndrome: A Literature Survey,” Asian Journal of Engineering and Applied Technology, 2018.

[3] Goodman, Neil F., et al. "American Association of Clinical Endocrinologists, American College of Endocrinology, and androgen excess and PCOS society disease state clinical review: guide to the best practices in the evaluation and treatment of polycystic ovary syndrome-part 1." Endocrine Practice 21.11 (2015): 1291-1300.

[4] Polycystic ovary syndrome (PCOS). Johns Hopkins Medicine. (n.d.). Retrieved December 25, 2021.