# SCREEN CONTROL USING GESTURES

A PROJECT REPORT

#### Submitted by

AMAN GOEL RA2211003011296

AYUSHI MISHRA RA2211003011344

#### Under the Guidance of

## Dr. TYJ NAGA MALLESWARI

(Associate Professor, NWC)

### *in partial fulfillment of the requirements* *for the degree of*

## BACHELOR OF TECHNOLOGY

## in

## COMPUTER SCIENCE ENGINEERING

## with specialization in (CSE Core)

## 

## DEPARTMENT OF COMPUTING TECHNOLOGIES

## COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## KATTANKULATHUR- 603 203

### MARCH 2024

### Annexure II

Department of Computing Technologies

##### SRM Institute of Science & Technology

##### Own Work Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

##### Degree/ Course : B. Tech CSE Core

**Student Name : Aman Goel, Ayushi Mishra**

##### Registration Number : RA2211003011296, RA2211003011344

**Title of Work : Screen Control Using Gestures**

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

* Clearly referenced / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| Aman Goel Ayushi Mishra |



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

## BONAFIDE CERTIFICATE

Certified that 21CSE292P project report titled “**SCREEN CONTROL USING GESTURE RECOGNITION**” is the bonafide work of “**AMAN GOEL RA2211003011296, AYUSHI MISHRA RA2211003011344”** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

##### DR. TYJ NAGA MALLESWARI DR. M. PUSHPALATA

|  |  |  |
| --- | --- | --- |
| **SUPERVISOR**  Associate Professor  NWC |  | **HEAD OF THE DEPARTMENT**  COMPUTING TECHNOLOGIES |

**ABSTRACT**

In this project, we aim to develop an innovative gesture recognition system that enables users to control their computer screens and simulate mouse and keyboard inputs using hand gestures. Leveraging the power of computer vision and machine learning, our system utilizes PyAutoGUI, MediaPipe, and OpenCV to detect and interpret hand movements in real-time. We have designed and trained a neural network on a custom dataset to achieve high accuracy in gesture recognition, ensuring a seamless and intuitive user experience.

The system is capable of recognizing a variety of gestures, each mapped to specific keyboard or mouse functions. For instance, users can navigate their screens, click, scroll, and even type using predefined hand movements. This approach not only enhances accessibility for individuals with physical limitations but also offers a novel and efficient way for users to interact with their computers.

Our project stands out due to its emphasis on accuracy and user-friendliness. By creating a custom neural network and dataset, we have tailored the system to accurately recognize a wide range of gestures. Additionally, we have focused on developing a user-friendly interface that allows for easy customization and adaptation to individual needs.

### ANNEXURE IV

TABLE OF CONTENTS

[**ABSTRACT**](#_bookmark1) **iii**

**TABLE OF CONTENTS iv**

[**LIST OF FIGURES**](#_bookmark2) **v**

**LIST OF TABLES vi**

[**ABBREVIATIONS**](#_bookmark3) **vii**

[**1 INTRODUCTION**](#_bookmark4) **1**

* 1. About Gesture Recognition 2
  2. About Dataset and Neural Network 3
  3. Software Requirements Specification 4

[**2 LITERATURE SURVEY**](#_bookmark20) **5**

2.1 Survey 5

**3** [**SYSTEM A**](#_bookmark30)**RCHITECTURE AND DESIGN 15**

3.1 System Overview 15

3.1.1 Gesture Detection 16

3.1.2 Neural Network Model 17

3.2 Design of Modules 18

**4 METHODOLOGY 21**

4.1 Data Collection and Preprocessing 21

4.1.1 Data Annotation 23

4.1.2 Data Augmentation 25

4.2 Model Training and Evaluation 28

**5 CODING AND TESTING 30**

**6 RESULTS AND DISCUSSIONS 40**

6.1 Model Performance 41

6.2 User Experience Evaluation 43

**7 CONCLUSION AND FUTURE ENHANCEMENT 45**

**REFERENCES 46**

**APPENDIX**

**A CONFERENCE PUBLICATION 50**

**B JOURNAL PUBLICATION 51**

**C PLAGIARISM REPORT 52**

**LIST OF FIGURES**

##### [2.1 Thresholding segmentation in action on the skin lesion image input](#_bookmark12) . . 4

[2.2 **Computer Vision Pipeline**](#_bookmark16). . . . . . . . . . . . . . . . . . . . . . . . . . 5

* 1. [**ROC curve CNN and dermatologists**](#_bookmark22) . . . . . . . . . . . . . . . . . . . . 8
  2. [**confusion matrix with CNN vs doctors**](#_bookmark26) . . . . . . . . . . . . . . . . . . . 10

[3.3 **Adam optimizer error rate**](#_bookmark29). . . . . . . . . . . . . . . . . . . . . . . . . 11

[4.1 **Sample lesions from each class**](#_bookmark32). . . . . . . . . . . . . . . . . . . . . . . 13

[4.2 **File structure of dataset**](#_bookmark34). . . . . . . . . . . . . . . . . . . . . . . . . . . 13

[4.3 **Diagnosis Techniques Plot**](#_bookmark36) . . . . . . . . . . . . . . . . . . . . . . . . . . 15

* 1. [**Number of data points of each class**](#_bookmark38). . . . . . . . . . . . . . . . . . . . 16

##### [Effect of SMOTE and number of variables on KNN (Euclidean Distance)](#_bookmark39) 17

[4.6 **YOLO CNN Layers**](#_bookmark43). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

[4.7 **ANN sample architecture**](#_bookmark48). . . . . . . . . . . . . . . . . . . . . . . . . . 23

[4.8 **Convolution Layer simulation**](#_bookmark52). . . . . . . . . . . . . . . . . . . . . . . 25

[4.9 **Pooling layer simulation**](#_bookmark54) . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

[4.10 **CNN complete architecture**](#_bookmark56). . . . . . . . . . . . . . . . . . . . . . . . . 26

* 1. [**CNN pre assigned weights for each class**](#_bookmark58) . . . . . . . . . . . . . . . . . . 27
  2. [**Sample layers of our CNN model**](#_bookmark59). . . . . . . . . . . . . . . . . . . . . . 28

[5.1 **Overall architecture of Product.**](#_bookmark67). . . . . . . . . . . . . . . . . . . . . . 32

[5.2 **Difference between UI & UX**](#_bookmark69). . . . . . . . . . . . . . . . . . . . . . . . 33

[5.3 **Ionic Framework**](#_bookmark71). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

[5.4 **Ionic Framework**](#_bookmark73). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

[5.5 **Back end Architecture**](#_bookmark75) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

[5.6 **Components of API**](#_bookmark77). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

[5.7 **REST API Architecture**](#_bookmark78). . . . . . . . . . . . . . . . . . . . . . . . . . . 38

[5.8 **SOAP API Architecture**](#_bookmark79). . . . . . . . . . . . . . . . . . . . . . . . . . . 39

v

**LIST OF TABLES**

##### [2.1 Thresholding segmentation in action on the skin lesion image input](#_bookmark12) . . 4

[2.2 **Computer Vision Pipeline**](#_bookmark16). . . . . . . . . . . . . . . . . . . . . . . . . . 5

* 1. [**ROC curve CNN and dermatologists**](#_bookmark22) . . . . . . . . . . . . . . . . . . . . 8
  2. [**confusion matrix with CNN vs doctors**](#_bookmark26) . . . . . . . . . . . . . . . . . . . 10

**vi**

**ABBREVIATIONS**

**AI** Artificial Intelligence

**API** Application Programming Interface

**AUC** Area Under the Curve

**CNN** Convolutional Neural Network

**CV** Computer Vision

**DL** Deep Learning

**FPS** Frames Per Second

**GPU** Graphics Processing Unit

**GUI** Graphical User Interface

**IoT** Internet of Things

**ML** Machine Learning

**NN** Neural Network

**RGB** Red, Green, Blue (color model)

**ROC** Receiver Operating Characteristic

**RNN** Recurrent Neural Network

**SVM** Support Vector Machine

Vii

**CHAPTER 1**

**INTRODUCTION**

In recent years, gesture recognition has emerged as a transformative technology in the field of human-computer interaction, offering a more natural and intuitive way for users to interact with digital devices. This project aims to develop an advanced gesture recognition system that allows users to control their computer screens and simulate mouse and keyboard inputs using hand gestures. By leveraging the power of computer vision and machine learning, our system provides a seamless and efficient interface for various applications, from accessibility enhancements to gaming and virtual reality.

* 1. **About Gesture Recognition**

Gesture recognition is a technology that interprets human gestures, such as hand movements, to perform specific tasks or commands. Its importance lies in its ability to bridge the gap between humans and machines, making interactions more natural and intuitive. This is particularly beneficial for individuals with physical disabilities, as it provides an alternative method of controlling devices without the need for traditional input devices like keyboards and mice.

Additionally, gesture recognition can enhance user experiences in gaming, virtual reality, and augmented reality, where natural and fluid interactions are crucial for immersion and engagement.

* 1. **About Gesture Recognition**

For our gesture recognition system, we have developed a custom dataset that is tailored to our specific needs. This dataset includes a diverse range of hand gestures, each mapped to corresponding control commands. By personalizing the dataset, we ensure that our system can accurately recognize and interpret the intended gestures.

We employ a neural network to classify these gestures, which has been meticulously designed and trained to achieve high accuracy. The neural network architecture is optimized to extract relevant features from the input images, enabling precise and efficient gesture recognition.

* 1. **About Gesture Recognition**

Our system is developed using Python 3.10, chosen for its speed and extensive library support. We utilize PyAutoGUI for simulating mouse and keyboard actions, MediaPipe for robust hand tracking, and OpenCV for real-time image processing. TensorFlow and Keras are used for implementing the neural network. The software is designed to be compatible with various operating systems, ensuring that it can be widely used across different platforms.

The choice of Python 3.10 and these libraries ensures that our system is both fast and reliable, providing a smooth user experience.

**CHAPTER 2**

**LITERATURE SURVEY**

In the initial phase of our project, we undertook a comprehensive review of around ten academic papers related to gesture recognition. This review was essential to understand the current landscape of the field, including the methods and technologies commonly employed. By studying these papers, we aimed to identify the various approaches to gesture recognition, their strengths, and their limitations.

For each paper, we focused on understanding the specific methodologies used for detecting and interpreting gestures. This involved examining the algorithms, techniques, and tools employed by researchers to achieve accurate gesture recognition. Through this analysis, we gained valuable insights into the different ways in which gesture recognition can be implemented and the challenges associated with each approach.

The information gleaned from this literature survey was instrumental in shaping our project. It provided us with ideas for our own system and helped us determine the best practices to follow. We learned about the significance of accurate hand tracking, the various neural network architectures that could be employed, and the importance of a high-quality dataset for achieving reliable recognition. These insights guided our decisions regarding the selection of technologies and the development of our methodology, ensuring that our approach to gesture recognition was informed and effective.

* 1. **Survey**

Gesture recognition is a technology that interprets human gestures, such as hand movements, to perform specific tasks or commands. Its importance lies in its ability to bridge the gap between humans and machines, making interactions more natural and intuitive. This is particularly beneficial for individuals with physical disabilities, as it provides an alternative method of controlling devices without the need for traditional input devices like keyboards and mice.

Additionally, gesture recognition can enhance user experiences in gaming, virtual reality, and augmented reality, where natural and fluid interactions are crucial for immersion and engagement.

**CHAPTER 3**

**SYSTEM ARCHITECTURE AND DESIGN**

The architecture of our gesture recognition system is meticulously designed to provide a seamless and efficient user experience. It comprises several interconnected modules, each performing a specific function to ensure accurate gesture recognition and translation into computer commands.

* 1. **System Overview**
     1. **Gesture Detection**

The core components of our system include:

* **Camera Input:** This module is responsible for capturing the real-time video feed from the user's webcam or an external camera. It ensures that the video stream is consistently available for processing by other modules.
* **Hand Tracking:** Utilizing the advanced capabilities of the MediaPipe library, this module detects the user's hand in the video feed and tracks its movements. It provides crucial information about hand landmarks, which are key points on the hand used for gesture recognition.
* **Gesture Recognition:** This is the heart of our system, where the magic happens. The gesture recognition module analyzes the hand landmarks provided by the hand tracking module using a neural network specifically trained for this purpose. The neural network classifies the hand's posture into one of the predefined gestures, such as a swipe or a click.
* **User Interface:** To ensure user-friendliness, we provide a graphical interface that allows users to interact with the system, customize gesture mappings, and view real-time recognition results. The interface is designed to be intuitive and easy to navigate.
* **Command Execution:** Once a gesture is recognized, this module translates it into a corresponding computer command. For example, a swipe gesture could be mapped to a scroll action, while a pinch gesture could simulate a mouse click. We use the PyAutoGUI library to execute these commands, allowing for seamless interaction with the computer.
  + 1. **Neural Network**

The core components of our neural network are:

* **Custom Dataset:** We have created a personalized dataset for sign language detection, which includes images of hand gestures representing alphabets, numbers, and four specific functions. This dataset is tailored to our project's requirements, ensuring that the neural network can accurately recognize the intended signs.
* **Data Processing:** The dataset images are processed using the MediaPipe library to extract hand landmarks, which are then used as input features for the neural network. Each image is converted into a set of coordinates representing the position of key points on the hand.
* **Neural Network** **Model:** A Random Forest Classifier is employed as the neural network model for this project. It is trained on the processed data to classify each gesture into its corresponding sign (alphabet, number, or function).
* **Model Training:** The model is trained using a split of the dataset, with 80% of the data used for training and 20% for testing. This ensures that the model is well-validated and can generalize well to new, unseen data.
* **Accuracy Measurement:** The accuracy of the model is measured using the test data, providing a quantitative assessment of the model's performance in recognizing sign language gestures.
* **Real-time Inference:** In the live application, the trained model is used to predict the sign language gestures in real-time, providing immediate feedback to the user.

* 1. **Design of Module**
* **Camera Input Module:**
  + Captures live video feed at a specified resolution and frame rate.
  + Converts the feed into individual frames for processing by subsequent modules.
* **Hand Tracking Module:**
  + Employs MediaPipe's hand tracking algorithm to identify key hand landmarks in each frame.
  + Outputs a set of coordinates representing the position of each landmark.
* **Gesture Recognition Module:**
  + Processes the landmark coordinates to extract features relevant for gesture recognition.
  + Utilizes a neural network model, trained on a custom dataset, to classify the hand posture into predefined gestures.
* **Command Execution Module:**
  + Maps each recognized gesture to a specific keyboard or mouse action based on predefined settings.
  + Executes the mapped actions using PyAutoGUI, allowing for direct control of the computer.
* **User Interface Module:**
  + Provides a graphical interface for system settings and real-time feedback.
  + Allows users to customize gesture-to-command mappings and view the system's status.

**REFERENCES**

[1] Hernández, E.; Sanchez-Anguix, V.; Julian, V.; Palanca, J.; Duque, N. Rainfall prediction: A deep learning approach. In International Conference on Hybrid Artificial Intelligence Systems; Springer: Cham, Switzerland, 2016; pp. 151–162.

[2] Goswami, B.N. The challenge of weather prediction. Resonance **1996**, 1, 8–17.

[3] Nayak, D.R.; Mahapatra, A.; Mishra, P. A survey on rainfall prediction using artificial neural network. Int. J. Comput. Appl. **2013**, 72, 16.

[4] Kashiwao, T.; Nakayama, K.; Ando, S.; Ikeda, K.; Lee, M.; Bahadori, A. A neural network-based local rainfall prediction system using meteorological data on the internet: A case study using data from the Japan meteorological agency. Appl. Soft Comput. **2017**, 56, 317–330.

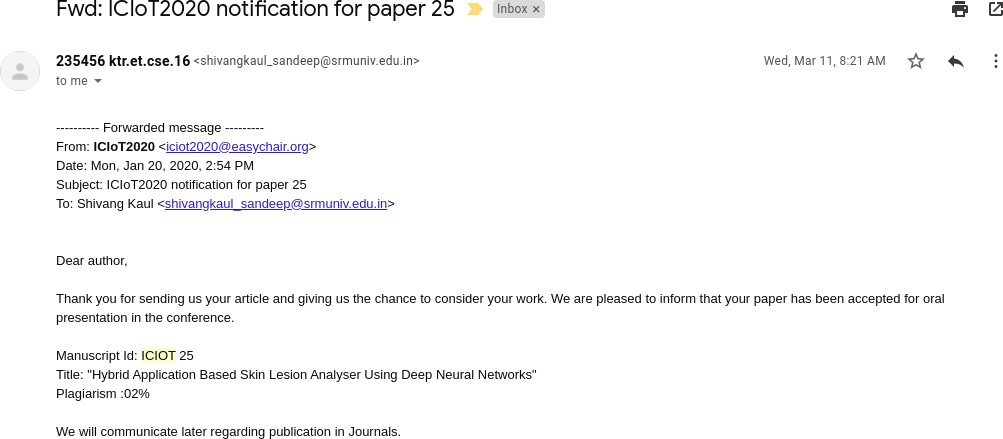
[5] Mislan, H.; Hardwinarto, S.; Sumaryono, M.A. Rainfall monthly prediction based on artificial neural network: A case study in Tenggarong Station, East Kalimantan, Indonesia. Procedia Comput. Sci. **2015**, 59, 142–151.

[6] Muka, Z.; Maraj, E.; Kuka, S. Rainfall prediction using fuzzy logic. Int. J. Innov. Sci. Eng. Technol. **2017**, 4, 1–5.

**APPENDIX A**

**CONFERENCE PRESENTATION**

Our paper on **Hybrid application based skin lesion analyzer using deep neural networks** was presented at ICIOT 2020 conference held at SRM. 200+ shortlisted teams presented their papers on various fields in the conference. Our paper got accepted as paper id : 25 with a plagiarism of just 2 %.



##### Figure A.1: ICIOT 2020 Acceptance

On presenting the paper in this international conference held at SRM KTR campus, we received positive remarks and suggestion from the judging panel. We were then awarded the best paper award at the same conference.



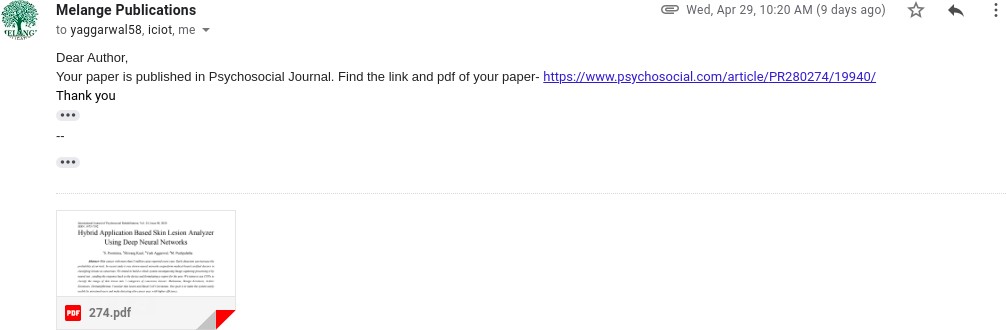
##### Figure A.2: ICIOT 2020 Best Paper award

83

**APPENDIX B**

**PUBLICATION DETAILS**

We submitted our research paper for publication at IJPR publication house puducherry. We had selected the journal **International Journal of Psychosocial Rehabilitation (ISSN: 1475- 7192)**. We got the acceptance notification from the IJPR stating our paper has been published in the April Issue of the same journal. Proof of publication is attached in figure [B.1](#_bookmark123) The research



##### Figure B.1: Publication Notification

paper cover page has been attached below.

International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 08, 2020 ISSN: 1475-7192

Hybrid Application Based Skin Lesion Analyzer Using Deep Neural Networks

1S. Poornima, 2Shivang Kaul, 3Yash Aggarwal, 4M. Pushpalatha

***Abstract--****Skin cancer with more than 5 million cases reported every year. Early detection can increase the probability of survival. In recent study it was shown neural networks outperform medical board certified doctors in classifying lesions as cancerous. We intend to build a whole system encompassing Image capturing processing it by neural net , sending the response back to the device and formulating a report for the user. We intent to use CNNs to classify the image of skin lesion into 7 categories of cancerous lesions: Melanoma, Benign Keratosis, Actinic Keratoses, Dermatofibroma, Vascular skin lesion and Basal Cell Carcinoma. Our goal is to make the system easily usable by untrained users and make detecting skin cancer easy with higher efficiency.*

***Key words--****Neural Networks, Image Processing, Convolu-tional Neural Networks, Skin Cancer Detection, Skin Lesion Imaging, App Development, Localization Algorithms, Cloud Computing, GCP, Compute Engine, App Engine.*

1. **INTRODUCTION**

Skin Cancer is a major kind of cancer with around 5 million reported cases worldwide every year. The major cause of skin cancer is exposure to UV rays. Diagnosing skin cancer generally included the skin lesion being examined by a doctor. Recent studies have shown neural networks to be more efficient in classifying lesion as cancerous as compared to trained doctors. Misdiagnosing or late detection of cancer can lead to a higher mortality rate and less chance of cure. The goal of this project is making detection and classification of lesions on the skin easier. Not all the marks on skin are a matter of concern but early detection and treatment of cancer can save lives. So this gives the user a way to check if there’s a chance of the mark on your skin being cancerous. The aim of this project is to detect and analyse such a correlation using neural networks. It is expected that the outcome of this project will lead to automated classification of skin lesions.

1. **LITERATURE SURVEY**

The following papers were read and analysed for the refer-ence of this paper. A brief image has been presented here.

1) Andre Esteva et al. 2017,” Dermatologist-level classification of skin cancer with deep neural networks.” Contribution: Claimed to classify skin lesions at par with board trained dermatologists. Methodology used:

*1Assistant Professor, CSE Department, SRMIST, Chennai, India*

*2Assistant Professor, CSE Department, SRMIST, Chennai, India,* [*shangkaul@gmail.com*](mailto:shangkaul@gmail.com) *3Assistant Professor, CSE Department, SRMIST, Chennai, India,* [*yaggarwal58@gmail.com*](mailto:yaggarwal58@gmail.com) *4Assistant Professor, CSE Department, SRMIST, Chennai, India*

**DOI: 10.37200/IJPR/V24I8/PR280274**

**Received: 21 Jan 2020 | Revised: 08 Feb 2020 | Accepted: 14 Mar 2020 2545**

**APPENDIX C**

**PLAGIARISM REPORT**

