International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

Rock, Paper and Scissors Using Opencv

Gaurav Mathur[1], Yash Gupta[2], Chaitanya Manik[3], Dr. Vijayasherly V[4]
[1,2,3,4]SCOPE, Vellore Institute of Technology, Vellore, India.
Emails: *gaurav.mathur2021@vitstudent.ac.in[1], yashgupta2021a@vitstudent.ac.in[2], chaitanya.manik2021@vitstudent.ac.in[3], vvijayasherly@vit.ac.in[4]*

Abstract
The game of Rock, Paper, Scissors is very simple. Each player picks one of the three objects and these rules are applied to see who has won that round: Paper wraps (beats) Rock, Scissors cut (beat) Paper, Rock blunts (beats) Scissors The challenge of the game is to guess what your opponent will choose and pick the appropriate object to beat them. People find it quite hard to pick a sequence of perfectly random choices, so any pattern that a player develops could be learned by the opponent and used to win the game. We will be implementing AI Based classic rock, paper and scissor game. We capture our data using OpenCV and make the dataset with 3 type of hand gesture. Once the data is collected, we train our model. At last, we play the game using OpenCV. The basis of this project was to experiment with Deep Learning and Image Classification with the aim of building a simple yet fun iteration of the infamous Rock Paper Scissors game.

Keywords: *Rock, Paper, Scissors, Opencv, Reserve Bank of India, NABARD, Rural, Finance*

1. Introduction
In the game of Rock, Paper, Scissors, each player picks one of the three objects (usually by making the appropriate hand shape on a count of three!) and these rules are applied to see who has won that round: Paper wraps (beats) Rock, Scissors cut (beat) Paper, Rock blunts (beats) Scissors The challenge of the game is to guess what your opponent will choose and pick the appropriate object to beat them. People find it quite hard to pick a sequence of perfectly random choices, so any pattern that a player develops could be learned by the opponent and used to win the game. We will be implementing AI Based classic rock, paper and scissor game. We capture our data using OpenCV and make the dataset with 3 type of hand gesture (Depicting rock, paper and scissor). Once the data is collected, we train our model. At last, we play the game using OpenCV. The basis of this project was to experiment with Deep Learning and Image Classification with the aim of building a simple yet fun iteration of infamous Rock Paper Scissors game.

2. Proposed Model
2.1. Identifying the Hand
We would try to create the area of interest and screen on which the user would we doing the gesture of hand, paper and scissors. We have narrowed down where the action is going to take place. But we still need to capture the action (meaning identify the hand). For that, there are several different approaches. We would be using histogram of oriented gradients or background subtraction. If you subtract the background of an image you are left with the foreground. And this is what we want here: the hand is (supposed to be) the only thing moving in the region of interest. Any user who wants to play, first takes a picture of 3 instances i.e., rock, paper and scissors. Then the model would automatically capture the images for training of the dataset. By this way 4 classes of data would be created. We would label them as {[s],[r],[p],[blank]}. Later on we would be running the CNN algorithm to classify the images into 4 categories.

2.2. Recognizing the Gesture
Once we have isolated the hand, we can start thinking of strategies to recognize what gesture it is making. I would be using CNN for this purpose. Once the image is captured from above, we would be creating our data set based on that image by zooming and changing different parameters of image. Which will give us 4 datasets persisting of 3 gestures i.e., rock paper scissors and a blank background. Once the system is able to recognize the gestures from liv3e video cam, we would be able to play and make a simple algorithm for playing the game with computer. [1-5]

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

2.3. Training the Model

To recognize the hand gestures from the data set of 4 types of classes we would be using a NASNET Mobile model along with transfer learning to classify the images. We would keep the hidden layers minimum as the model needs to be fast. Since a player cannot wait for a longer time in front of the screen for model to be trained. Tentatively we would be going for 712 units in starting with 2 hidden layers of dropout 40 percent in our CNN model. [6-10]

2.4. Game Deployment

Before deployment the game we need to perform a really important step. While capturing the data we only captured 100 images from the user which quite less to train he model on these images. For that we use image augmentation technique to increase our dataset to 1000 images per class. By this way we increase our training dataset making it more fool proof. To deploy the game, we would capture the images from live feed the scores of user and computer would be maintained on the screen. The computer would randomly pick out of 3 images and the according to the user's gesture the respective scores would be added. By this way you can play with your computer in an interactive manner and try to beat the machine. [11]

2.5. Architecture Diagram

This is the Architectural diagram of our project – Rock, Paper, Scissors using Machine Learning and OpenCV. (Refer Figure 1)

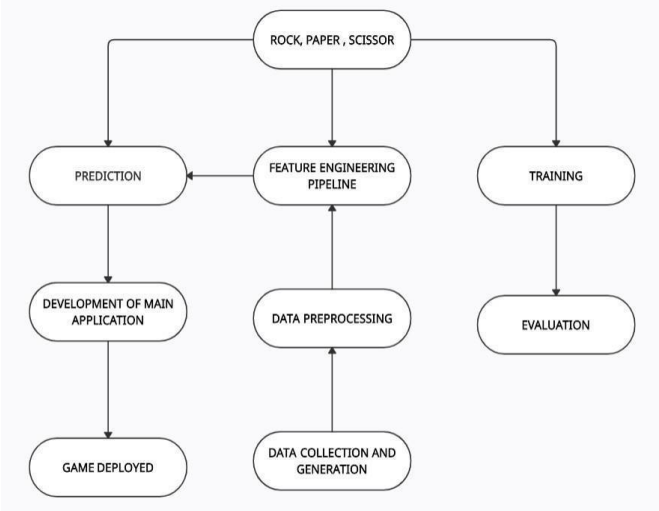

Figure 1 Architectural Diagram

3. Literature Survey

The skin model [1] to separate human hands using face detector that is based on ada-boosting algorithm including OpenCV library [1]. General mesh feature and three layers back- propagation algorithm is used for their algorithm. Images of human face and head regions are captured. It detects the face region excepting color information, and extracts the skin color model from face region for detecting hand region. Detect a human face, and make a skin model from detected face region, to get the robust skin information according to the illumination variance. Owing to play this game, human feels that robot has intelligence and smart. And robot can get user's faces and voices to enroll for author identification. The algorithm has three steps: matrix allocation, frame processing and gesture extraction and assimilation stage. Used standard matrix allocation algorithm to form x-y coordinates where motion feature was used to extract the air-swipe gestures. It detects the gesture of the hand when placed on the front camera of the device [2]. Therefore, the dataset consisting of 4 gestures (left, right, up, down) of 5 different people was taken. When a specific gesture is detected then the application creates a toast specifying the gesture that has been detected. This approach is able to recognize the given set of four gestures with both recall and precision over 96%. They have majorly implemented and compared camshaft algorithm and haar cascades. And used different mechanisms to detect face such as via color, and motion. Face recognition takes an image from a video or a camera as input and they basically focused on finding images which have a monochromatic background and removing portions of images which are very dark or bright [3]. It identifies Haar cascades as the most efficient way of face detection as it gives better accuracy in facial expression whereas cam shift algorithm is very expensive and not very effective in performance. It involves: Query image frame from web camera, Outline detection, and Creation of Box2D world, Function to iterate through contour tree, Creation of object from outer contour and Creation of objects [4].They have used a web camera as a source and Open Computer Vision (OpenCV) library as processing tool of the images which is the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

dataset. Because of restrictions of Box2D they had to do approximation and scalingof outlines and tessellation of objects with Delaunay triangulation algorithm. It is done by using color-based detection technique of the physical sticks' color and rotated bounding box technique to get the angle of the physical sticks rotated by the player. The other methods that are involved are: Multi-Object Tracing Method, Kernel based method and Camshafts algorithm [5]. The dataset is produced by the live video input from the camera of the mobile device. It was found that the performance of the device during the object detectionprocess depends on the device's specifications. The research paper focuses on classification and infected area estimation of Frogeye, Downy mildew and Bacterial Pustule disease of Soybean. In this proposed approach, an Image enhancement technique for enhancing the image quality is used. [6] Then the k- means segmentation algorithm is applied to separate the infected cluster from leaf. Neural Network used to classify Frogeye, Downy mildew and Bacterial Pustule. Accuracy of 93.3 % is achieved for 30 images. After classification area estimation of infected area is performed. Images taken as input are downloaded from the internet. Images are captured from a digital camera and stored in standard jpeg format. This algorithm consists of steps like segmentation using K means, classification of images using neural network and finally grading of disease using region propfunction of MATLAB. The proposed algorithm achieves good accuracy in classification. In the task, the number and the locations of hands in RPS pictures are first estimated by c-Means-type clustering in conjunction with cluster validity measures After preprocessing of extracting hand region coordinates from RPS pictures, Hard c-Means and Fuzzy c-Means is applied in conjunction with cluster validity measures for searching for the optimal number and locations of hands, and the applicability of noise rejection is compared in several experiments [7]. The dataset involves RPS pictures which are preprocessed into object data to be clustered. The experimental results imply that FCM with noise rejection used with cluster validity measures works better in RPS game judgment without following complex image processing. First, raw EMG signal measured using MYO arm band. Then Levine bilinear model in [4] is applied to obtain the normalized muscle activation. a multi-layer perceptron's (MLPs) composed of one input, two hidden and one output layers are utilized and updated using gradient descent [8]. Three postures of the hand wearing MYO band and data glove was maintained and recorded for training resulting in 859 datasets. MLPs were trained on the recorded dataset and was evaluated with Cross validation method is and showed the gradually increasing generalization accuracy of MLPs. Second, finger movements are measured by using VMG30 data glove. At first, this paper used the zhongxing-micro ZC301P cameras to build a binocular stereo vision system for recording images. After the camera calibration and binocular calibration, the three-dimensional data of facial images were extracted using the functions of OpenCV computer vision library, and then 3d face model were reconstructed preliminary by DirectX. [9] According the reconstruction process, the human face three-dimensional reconstruction software was designed and developed. 3D facial information is retrieved via OpenCV which is based on 3D vision, and 3D data is reconstructed using DirectX for form a surface model of human face, so as to reconstructa vivid 3D facial surface. The design proposed and produced a preliminary 3D facial model, in addition, facial image shall be scanned automatically to realize real-time modeling combined with pre-calibrated parameters. At last, point cloud is reconstructed using DirectX library, thusproducing 3D facial model with strong visuality. CNN with a stable linearization are discussed in terms of image processing capabilities and pattern formation properties. Particularly, it is shown that infinite impulse response (IIR) zero-phase spatial linear filtering can be performed both at the network equilibrium and in a time-varying manner. This interpretation can aid in understanding some conventional templates, such as diffusion, as well as offer improvements and design methods for linear filtering on the CNN. [10] Then they move onward to CNN with unstable linearization's. When the input variable is not used for data, these systems can be

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

considered as spatial frequency generalizations of the classical averaging template. No dataset as the paper is solely research on CNN utility for image processing. They have given many examples and design procedures that take advantage of a modal representation of the linearized CNN dynamics. Some classic results from the field of pattern formation have been demonstrated in the simple CNN with first-order cells. In addition, some approaches to designing CNN templates for both linear and nonlinear spatial filtering were given.

3.1. Novelity in our Project Methodology

The methodologies described in the research papers and our methodology for the ROCK, PAPER, SCISSORS project have several differences in terms of approach, techniques used, and objectives.

3.1.1. Data Collection and Processing Techniques

The research papers primarily focus on using computer vision techniques like matrix allocation, frame processing, gesture extraction, color-based detection, contour tree iteration, etc., along with algorithms like camshaft, hear cascades, etc., for specific tasks such as object detection, gesture recognition, and disease classification.

Our methodology, on the other hand, focuses on manual data collection through user interaction with a webcam, using OpenCV for video capture and labeling. This approach is more interactive and requires user involvement for data generation.

3.1.2. Machine Learning and Model Training

The research papers employ various machine learning algorithms like neural networks, clustering algorithms, and deep learning techniques for tasks such as image classification, gesture recognition, and disease diagnosis. They also discuss model training, accuracy evaluation, and optimization techniques. Our methodology involves transfer learning, where you leverage pre-trained models for image classification tasks related to ROCK, PAPER, SCISSORS gestures. This simplifies the training process and reduces computational resources compared to training a model from scratch.

3.1.3. Application and Deployment

The research papers often focus on developing specific applications like gesture based game control, disease diagnosis tools, or object detection systems. They may discuss deployment challenges and real-world applicability. Our methodology focuses on developing an end-to-end application for ROCK, PAPER, SCISSORS game, including live testing on a webcam feed. The emphasis is on creating a user-friendly and interactive application rather than a specialized tool for a particular task.

3.1.4. Technology Stack and Tools

The research papers mention using specific tools and libraries like OpenCV, DirectX, MYO arm band, CNN architectures, etc., tailored to their respective tasks and research objectives. Our methodology extensively uses OpenCV for video capture, data preprocessing, and model training, making it accessible for developers familiar with Python and computer vision libraries. In summary, while the research papers focus on specialized tasks and techniques using computer vision, machine learning, and specific algorithms, our methodology takes a more practical and user-centric approach for developing a game application, leveraging transfer learning and interactive data collection through webcam interaction. Both approaches have their merits based on the intended application and project goals.

4. Methodology

- Step 1: Gather Data, for rock, paper scissor classes.
- Step 2: Visualize the Data.
- Step 3: Preprocess Data and Split it.
- Step 4: Prepare Our Model for Transfer Learning.
- Step 5: Train Our Model.
- Step 6: Check our Accuracy, Loss graphs & save the model.
- Step 7: Test on Live Webcam Feed.
- Step 8: Create the Final Application.

4.1. Gathering Data

To gather the data we use OpenCV (Video Capture). The idea is simple: We Create a box of interest in the video capture where the user will do the hand gestures. We would record 4 type of gestures i.e. Rock, Paper, Scissors and Nothing. As the code runs a Video Capture window of OpenCV opens up. Click

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

's', 'r', 'p' or 'n' to record hand gesture for Scissor, rock, paper and nothing respectively (Refer Figure 3, 4, 5, 6). Once we select the label the webcam captures the gestures (for starting we have taken 100 samples each) and stores the data on RAM. Later on this captures data is divided in labels and used to train the model. (Refer Figure 2)

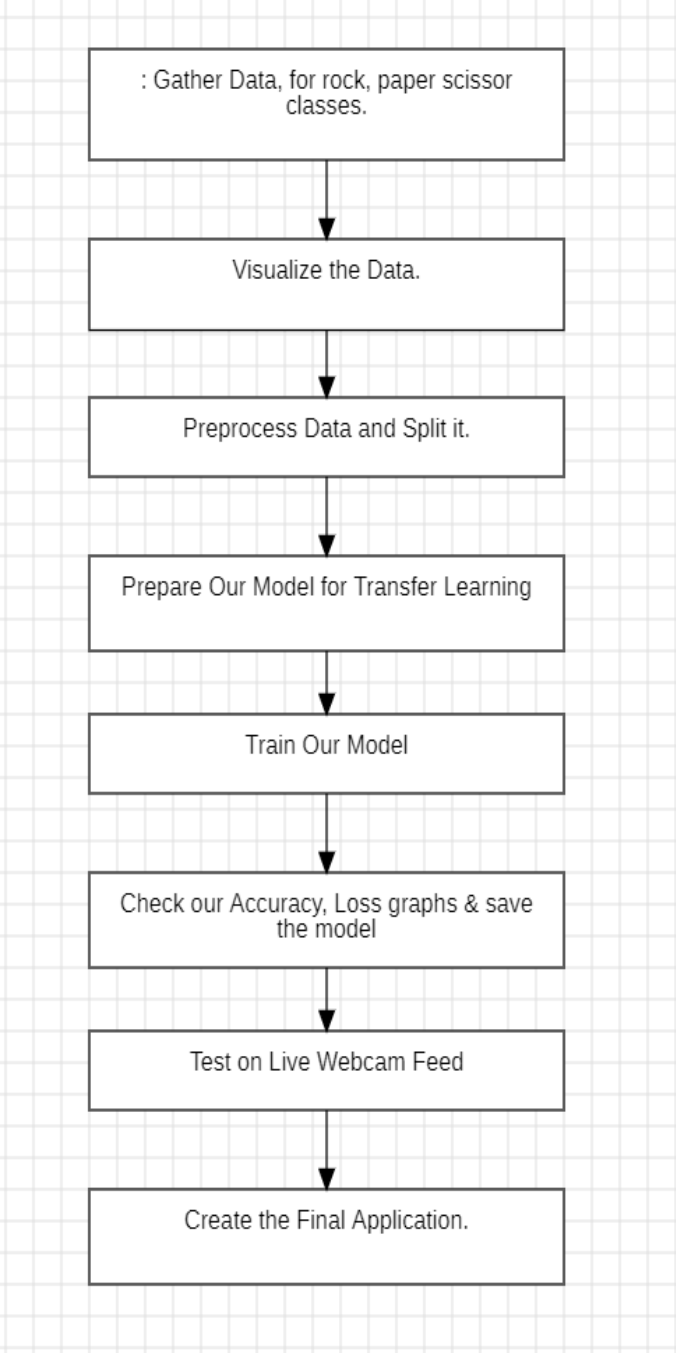

Figure 2 Methodology Flowchart

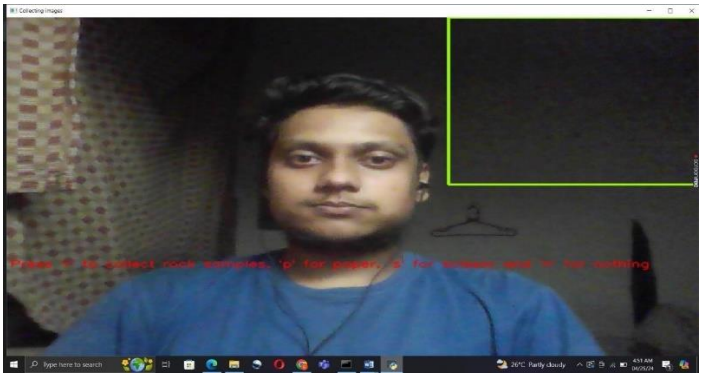

Figure 3 Nothing

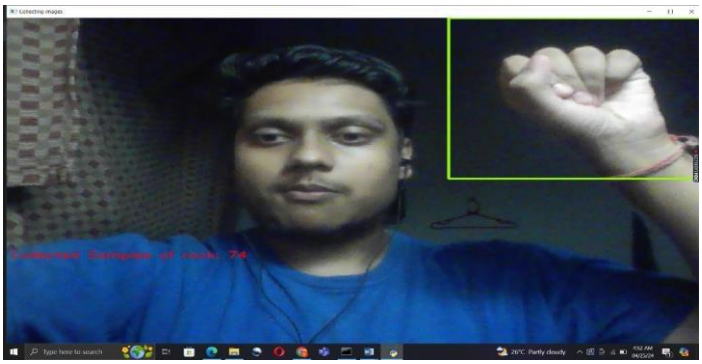

Figure 4 Stone

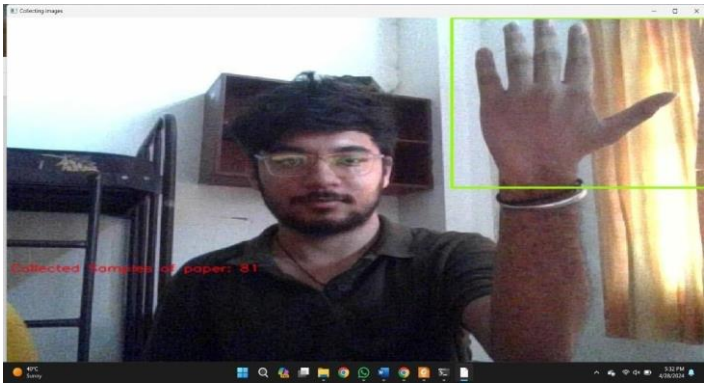

Figure 5 Paper

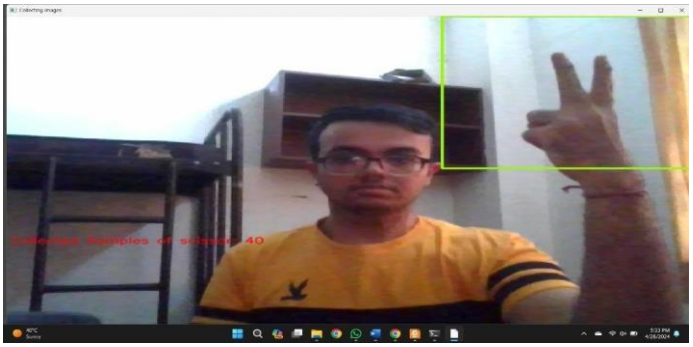

Figure 6 Scissors

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

4.2. Visualizing

Since we are not saving the images in the disk, we need to see we have captured the right data. (Refer Figure 7)

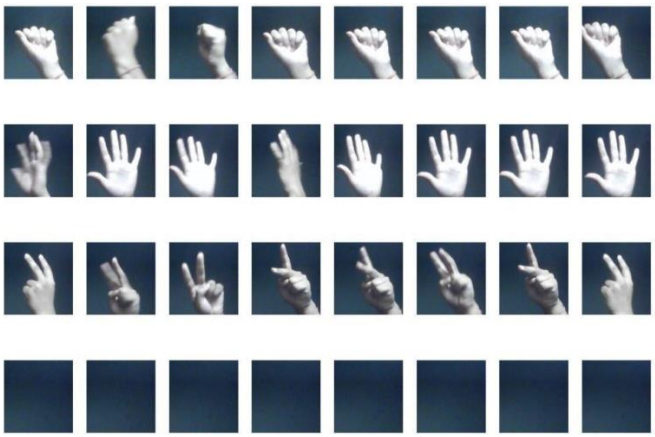

Figure 7 Hand Gesture Pictures

4.3. Processing the Data

In this step, we have combined all the images and labels in a single list and then preprocess them as required by the network. After preprocessing is done we will split them into train and test sets.

ML Model

1. In this we have decided to go ahead with NASNETMofbile model (considering the speed and accuracy)
2. We loaded NASNETMobile without the head, because it was trained on 1000 ImageNet classes and we have to predict just 4 classes so we don't need the head of the model.
3. We did Transfer Learning so we added a few necessary layers on top of the base model to create our custom head in which the final layer will contain the number of nodes equal to the number of classes which in our case is 4.
4. The images below show the layers which we added on NASNETMobile just to make the final output layer of 4 nodes (Rock, Paper, Scissors and nothing).

4.4. Training of Model

Before compiling and training our model, We augmented our dataset to add some random images to the data. Since we just recorded 100 images augmentation helps us to increase our training examples.

4.5. Accuracy

Figure 8 show accuracy of our model. After running some 15 epochs on the 20 batch size we were able to reduce our loss significantly, making this model suitable for our project.

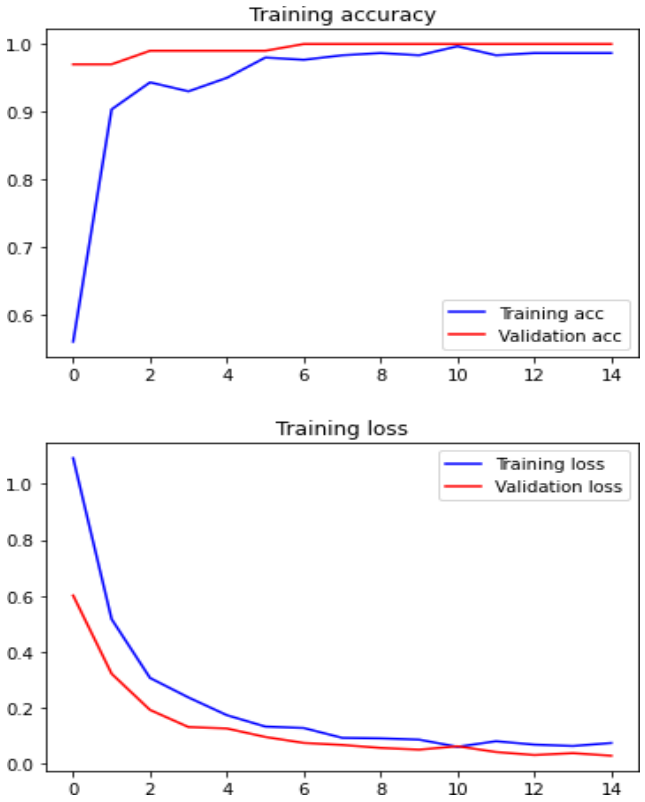

Figure 8 Accuracy

4.6. Testing on Webcam

Finally, we tested our model on live video feed, to see how well our model is able to predict the gestures from video. (Refer Figure 9-13)

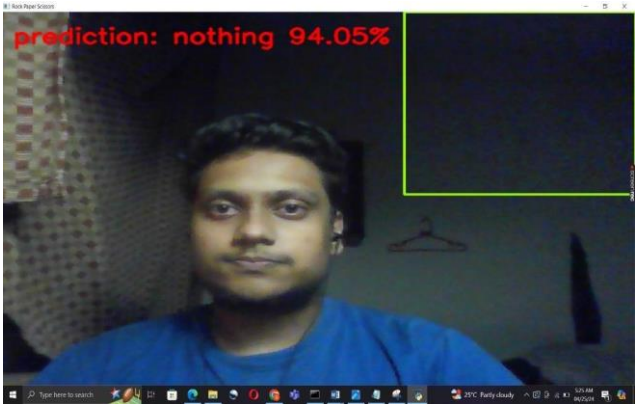

Figure 9 Results as Nothing

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

Figure 10 Results as Scissors

Figure 11 Results as Paper

Figure 12 Results as Rock

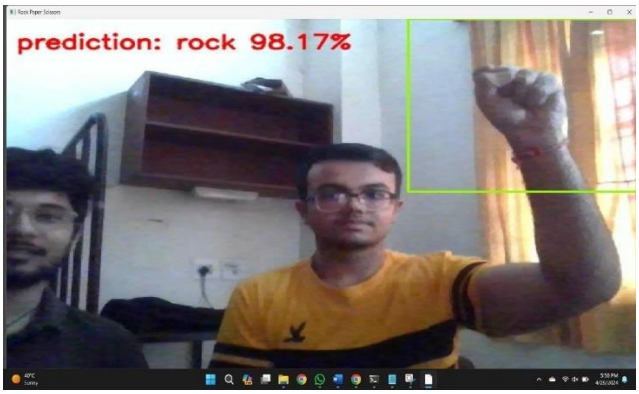

Figure 13 Results as Rock

4.7. Final Application

Finally, we combine all the models and functions together and created an application that can play with you anytime. Beat it if you can.

4.7.1. Testbed

- The Project was built and simulated on a Windows Operating system x86
- 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space
- Software:
- Modern Operating System:
- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)

4.7.2. Libraries Used

tf.keras.applications.NASNetMobile used for making our ML Model Python Packages and modules used: OS, cv2, numpy, matplotlib, time, tensorflow. We have used submodules like from tensorflow.keras.preprocessing.image we imported ImageDataGenerator, from tensorflow.keras.models we imported Model, load_model, from tensorflow.keras.layers we imported Dense, MaxPool2D, Dropout, Flatten, Conv2D, GlobalAveragePooling2D, Activation, from tensorflow.keras.optimizers we imported Adam, from tensorflow.keras.utils we imported to_categorical, from sklearn.model_selection we imported train_test_split, from sklearn.preprocessing we imported LabelEncoder, from random. we imported choice,shuffle , from scipy we imported stats as st and from collections we imported deque for making responsive and user friendly component for our Graphical User Interface and training our Machine learning model

Conclusion

After training the model we are done with user side. Now we have to focus on computer's side. We need to define the moves of computer and declare what is considers as winning and what is considered as losing point. All of these are done in the following functions. (Refer Figure 14-19) The following shows the results of the Proposed System.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

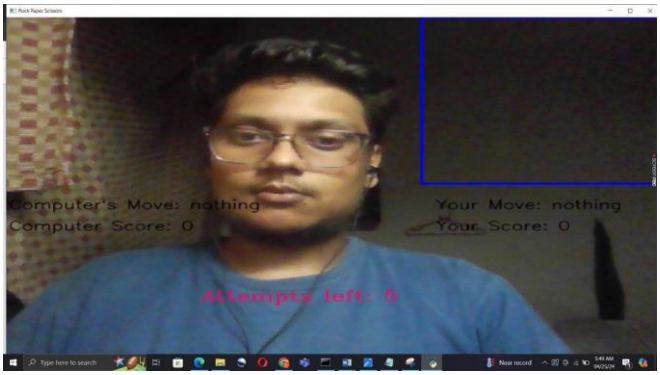

Figure 14 Computer's Move Nothing

Figure 18 Computer's Move Rock

Figure 15 Computer's Move Rock

Figure 19 Game Result

Figure 16 Computer's Move Rock

Figure 17 Computer's Move Rock

References

[1]. Yoon, H.S. and Chi, S.Y., 2006, October. Visual processing of rock, scissors, paper game for human robot interaction. In 2006 SICE-ICASE International Joint Conference (pp. 326-329). IEEE.

[2]. Sharma, T., Kumar, S., Yadav, N., Sharma, K. and Bhardwaj, P., 2017, February. Air- swipe gesture recognition using OpenCV in Android devices. In 2017 international conference on algorithms, methodology, models and applications in emerging technologies (ICAMMAET) (pp. 1-6). IEEE.

[3]. Goyal, K., Agarwal, K. and Kumar, R., 2017, April. Face detection and tracking: Using OpenCV. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (Vol. 1, pp. 474-478). IEEE.

[4]. Sedlák, M., 2010. Simulation of 2D physics of objects captured by web camera using OpenCV and Box2D. IEEE Transaction.

[5]. Izhar, F.N.M. and Sunar, M.S., An Interactive

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0448
e ISSN: 2584-2854
Volume: 02
Issue: 09 September 2024
Page No: 3034-3042

Real World Virtual Puzzle Game using OpenCV.

[6]. Gharge, S. and Singh, P., 2016. Image processing for soybean disease classification and severity estimation. In Emerging research in computing, information, communication and applications (pp. 493-500). Springer, New Delhi.

[7]. Matsumoto, Y., Yamamoto, T., Honda, K., Notsu, A. and Ichihashi, H., 2012, June. Application of cluster validity criteria to Rock-Paper-Scissors game judgment. In 2012 IEEE International Conference on Fuzzy Systems (pp. 1-5). IEEE.

[8]. Gang, T., Cho, Y. and Choi, Y., 2017, June. Classification of rock-paper-scissors using electromyography and multi-layer perceptron. In 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) (pp. 406-407). IEEE.

[9]. Yin, J. and Yang, X., 2016, July. 3D facial reconstruction of based on OpenCV and DirectX. In 2016 International Conference on Audio, Language and Image Processing (ICALIP) (pp. 341-344). IEEE.

[10]. Crounse, K.R. and Chua, L.O., 1995. Methods for image processing and pattern formation in cellular neural networks: A tutorial. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 42(10), pp.583-60

[11]. Divya, U., Sirisha, T., Durgesh, B., Avinash, T., Teja, E. N., & Gnanender, U. (2024). Accident Avoiding and Monitoring Using Open CV. International Research Journal on Advanced Engineering Hub (IRJAEH), 2(06), 1639-1645. https://doi.org/10.47392/IRJAEH.2024.0225