Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2022) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

An Enhanced Approach in Developing Hand Gesture based Stone Paper Scissors Game Using Artificial Intelligence

Varsha Naik*, Abhishek Chebolu, Prajakta Chaudhari, Snehalraj Chugh, Ahbaz Memon, Janhavi Chavan

Dr. Vishwanath Karad MIT World Peace University, Survey No: 124, Paud Rd, Kothrud, Pune, 411038, Maharashtra, India

Abstract

This project uses machine learning, computer vision, and deep learning to build human-versus-computer game of rock, paper, and scissors. The Convolutional neural network, Mediapipe, Haar Cascade Classifiers, OpenCV, Matplot, and other object identification algorithms are used to recognise player's gestures. In this research, we've developed a hand-locating frame that incorporates Mediapipe's hand gesture recognition system and 21-axis skeleton projections.

Players can be tracked at 30 frames per second using our generalised detection approach, which doesn't require a specific background or illumination setting. In this project, players can not only play with other players present physically with them but also devise an AI to substitute the need for an actual human player. Furthermore, the AI works by analysing the opponent's previous movements in context to estimate their most likely future move. With this approach, the algorithm has a better chance of winning over human players than if it relied just on probability.

When playing against an opponent in a game of rock-paper-scissors, we have used three different models over our object detection neural network: an improved Markov chain framework, multi-label classification using SVM, and an RNN model incorporating LSTM and GRU. An entirely new dataset of approximately 6000 images was created from scratch in order to differentiate between rock, paper, and scissors with a 99.999% accuracy rate. The primary goal of this study was to detect the gesture and make the artificial intelligence (AI) capable of defeating a human opponent while simultaneously detecting the hand movements of the player in real-time.

Keywords: Convolutional Neural Network, Recurrent Neural network, Artificial Intelligence, Support Vector Machine, Mediapipe, Haar Cascade

*Corresponding author.

Email addresses: varsha.powar@mitwpu.edu.in (Varsha Naik), ahbazmemon0@gmail.com (Ahbaz Memon), snehalchugh2016@gmail.com (Snehalraj Chugh), prajakta.p.chaudhari@gmail.com (Prajakta Chaudhari), abhishek.chebolu@gmail.com (Abhishek Chebolu), janhavi.a.chavan@gmail.com (Janhavi Chavan)

1. Introduction

The world-famous hand game Rock-Paper-Scissors gives a fascinating setting to study human cognitive processes and artificial intelligence [1]. Simple in rule structure yet deep in psychological depth, this game is a great testbed for more advanced computational models. Our work uses the game's simplicity to test complex machine learning and computer vision technologies to bridge the gap between human unpredictability and machine precision. This technique aims to improve our knowledge of human-computer interaction, gesture recognition, and predictive analytics.

Our work introduces a new approach to understand Rock-Paper-Scissors. This method uses CNNs, Google's Media Pipe, Haar Cascade Classifiers, and OpenCV. Unlike other gesture recognition systems, ours does not need environmental factors like lighting or background. This flexibility is a major improvement over current method. These old approaches have strict setup requirements, limiting their use in real-world situations. Our objective is to create a high-performing real-time system that can compete with humans. This sets our system apart:

- Our technique relies on a cutting-edge hand-locating framework for intuitive gesture recognition. MediaPipe's cutting-edge hand gesture detection technology and 21-axis skeleton projections enable precise hand gesture tracking at 30 frames per second. This greatly improves accuracy and ensures that the system will remain responsive regardless of its surroundings.
- Our approach is unaffected by lighting or backdrop. Traditional systems, which require regulated settings to work, lack this flexibility.
- The system reacts and predicts. It can predict an opponent's actions by examining their past activities. Combining novel techniques including an improved Markov chain, SVM multi-label classification, and an RNN with LSTM and GRU helped foresee this. Due to this diverse combination, our system can apply complex prediction algorithms that go beyond typical models' basic patterns.
- We created a unique collection of over 6000 photos to improve our artificial intelligence. These photos were meticulously sorted into rock, paper, and scissors. This dataset's 99.999% accuracy rate is unmatched in the industry, making it a game-changer compared to other datasets.
- Our technology engages instantly by recognizing and responding to human actions. This factor keeps the game lively and distinguishes our approach from slower, turn-based ones.

This technique has many advantages over the most popular and successful methods in the research. Contrary to traditional systems, which need certain environmental setups, our method is more adaptable. Because it works well in a variety of lighting and background conditions, it's easy to use in daily life. Our efforts have improved input processing accuracy and speed. Our well selected dataset and a harmonious blend of complicated analytical methodologies led to this accomplishment. Due to increased efficiency and response speed, the industry has set a new benchmark, improving consumer experience.

Diagnostic Framework for Proactive Analysis: Our analytical component uses past data trends to predict the future. Our method gives strategic foresight due to this rare trait in conventional models. The system can predict and adapt to future occurrences, giving a proactive response. In development, we prioritized a user-friendly and straightforward interface. This interface, which requires no equipment or technical abilities, makes our system more accessible. Doing so ensures that our solution is accessible and user-friendly for a broad spectrum of users.

Our original mix of cutting-edge data processing approaches to promote user-system interaction, notably in interactive gaming, has set our research apart. We enhance Rock-Paper-Scissors game interaction dynamics using sophisticated sequential data processing models [2]. Historically, predictive analytics used these models. A complex understanding and response mechanism that closely mimics human decision-making is achievable with such an approach. Another distinguishing feature of our work is the development and use of a full dataset. This dataset contains several circumstances and inputs to ensure the system's outputs are correct and contextual. Technology for interaction-based applications has advanced greatly with this study.

This study breaks beyond standard research methods and enters an area where technology and intuition coexist in interactive systems. Beyond interactive gaming, our work might change user interfaces in other sectors. Our innovative technique prepares for future research on intuitive, human-behavior-aligned systems. We have developed concepts and methods that might enhance technology usage in educational tools, assistive technology, and other fields. This research is a big step in creating more understanding, compassionate, and human-needs-responsive systems. We are creating a future where technology is intuitive, responsive, and effortlessly interwoven into our lives. Technical advances and a deep understanding of human interaction dynamics achieve this.

2. Motivation

This study was inspired by the need to combine human intuition with machine learning in basic yet strategic games like Rock-Paper-Scissors [3]. This game, a model of decision-making under uncertainty, offers unique opportunity to study and advance human-computer interaction. The goal is to enhance human gesture identification using sophisticated machine learning methods and to investigate artificial intelligence's prediction abilities in emulating and exceeding human strategic thinking. Our novel approach to gesture detection and predictive modeling in games sets it apart from other contemporary methods.

- We use Convolutional Neural Networks, Google's Mediapipe, and Haar Cascade Classifiers to improve gesture recognition. This allows more precise and real-time monitoring of complex hand motions than current methods.
- Our method's ability to work in many environments is a major improvement. This universality is a major improvement over earlier systems, which need customized parameters for perfect performance.
- Our system uses improved Markov chains, support vector machines, and recurrent neural networks (RNN) with LSTM and GRU to predict and adapt to human methods better than standard AI models.

Our work offers a new real-world application in early childhood education. Consider integrating our advanced gesture recognition technology into interactive educational platforms for kids. Rock-Paper-Scissors, a fun game, might change learning when combined with our artificial intelligence's gesture recognition [4]. In this simple game, kids may enhance their cognitive skills, learn pattern recognition, and practice decision-making in a fun and engaging way. This artificial intelligence-enhanced game might help educators and parents track children's progress. It combines study and play. Because of its flexibility and predictive skills, artificial intelligence (AI) keeps the game challenging yet accessible, making it a flexible and creative instructional device. This accommodates everyone's learning pace.

3. Literature Survey

Dynamic interaction between people and machines has long been a focus of interest and rigorous research, especially in games, which are wonderful venues for testing and improving AI and ML models. Especially with games. The simple yet unexpected Rock-Paper-Scissors (RPS) game [5] is ideal for studying gesture detection, human-computer interaction, and predictive artificial intelligence. This literature review examines much major research. Each provides a unique perspective and technique to these subjects. Frameworks for human-robot collaborative gaming, digital gaming hand gesture recognition systems, high-speed robotic hand teleoperation mechanisms, analyses of human behavioral patterns in RPS games, strategies for accelerating multi-agent reinforcement learning in zero-sum games, and chaotic dynamics in asymmetric RPS games are included in this compilation. This survey summarizes the most relevant findings from these studies, analyzes their methodology, and compares their strengths and weaknesses. To strengthen artificial intelligence's ability to understand and anticipate human actions in real-time interactive situations, these insights should be combined to identify trends, gaps, and future research areas.

Research frequently discusses the interaction of AI and gesture detection in gaming. According to "Developing a Lightweight Rock-Paper-Scissors Framework for Human-Robot Collaborative Gaming," [6] the framework is detailed. In the RPS game, this system uses machine learning architectures and hand gesture detection to enable intuitive social robot interactions. As in "Playing Games Using Hand Gesture Recognition", [7] MediaPipe detects hands well. By offering excellent precision and reactivity, MediaPipe may improve gameplay. These studies emphasize the need of entirely seamless human-computer interfaces in the gaming sector, where gesture detection systems must be precise and real-time. The paper "Teleoperation of High-Speed Robot Hand with High-Speed Finger Position Recognition and High-Accuracy Grasp Type Estimation," [8] emphasizes teleoperation. Proper grip type estimation and fast finger position recognition are stressed in the article. Despite the distinct scenario, real-time image processing and machine learning are comparable to gaming gesture recognition.

According to the literature, predictive analytics is another important foundation. "Detecting Human Behavioral Patterns in Rock Paper Scissors Game Using Artificial Intelligence" [9] predicts RPS behaviors using neural networks. This method exploits human players' past behavior to outperform them. "Accelerate Multi-Agent Reinforcement Learning in Zero-Sum Games with Subgame Curriculum Learning" [10] introduces the Subgame Curriculum Learning (SACL) framework to speed up Nash Equilibrium learning in zero-sum games. These studies highlight the growing complexity of artificial intelligence in understanding and forecasting human behavior, citing predictive analytics' successes. The latest study on RPS games' unpredictability is "Chaotic Dynamics in Asymmetric Rock-Paper-Scissors Games" [11]. Extending the typical game model to incorporate asymmetric dynamics reveals the complex behaviors and probable chaos in such systems. This study illustrates the variety and complexity of approaches and concepts utilized in simple games.

Despite each research's unique contribution, similarities and differences are obvious. The focus on real-time, precise gesture identification is like collaborative gaming and hand gesture detection frameworks. These systems are used in human-robot interaction, not digital gaming, which differs in user experience and technical demands. The prediction models used to investigate human behavior in RPS games and develop tactics in zero-sum games are employed using historical data and learning algorithms. Unlike the neural network technique, which focuses on human behavior pattern recognition, the SACL framework addresses AI agent strategy learning [12]. Different techniques have different scopes and usefulness. An intriguing alternative to conventional research is studying chaotic dynamics in asymmetric Rock-Paper-Scissors (RPS) games. This research switches the emphasis from deterministic prediction models to understanding and accepting human-computer interaction unpredictability and complexity. This move promotes a more nuanced approach to artificial intelligence development in the game business and emphasizes nuances and complexity over expected accuracy.

These studies employed several ML and AI methodologies, resulting in many approaches [13]. However, deploying these technologies in many real-world situations is difficult. These systems are having restricted scalability and flexibility when they rely on MediaPipe and need advanced high-speed image processing. The biases in training data and the challenges of understanding complex human decision-making processes must be addressed. Additionally, predictive algorithms can read and predict human behavior. Chaos dynamics provides a useful perspective and shows the complexity and unpredictability of human-computer interactions [14], which calls into question predictive analytic approaches.

The gaming literature shows tremendous development in gesture recognition, predictive artificial intelligence, and human-computer interaction. Machine learning frameworks and hand gesture recognition may make gaming more spontaneous and dynamic. Meanwhile, predictive analytics may help artificial intelligence understand and predict human behavior. However, the study highlights some key issues. These systems' dependency on particular technologies and demand for high-speed processing raise concerns about their scalability and adaptability. The study of chaotic dynamics highlights the complexity of human decision-making processes and the possibility for unpredictability in human-computer interactions, making it harder to build trustworthy prediction models.

Future research should concentrate on improving gesture recognition systems' universality and versatility. This effort should also aim for more advanced and robust prediction models that properly understand human behavior. Also study the balance between predictability and adaptability in AI. Addressing these shortcomings will help us create artificial intelligence systems that are responsive, precise, sensitive, and adaptable to human behavior [15]. Such advances would be quantum leaps in human-computer interaction.

4. Data Generation

We were trying to build a hand tracking-based Rock-Paper-Scissors that would play as an opponent for this project. With the help of images classification and object processing methods, researchers are trying to find solutions to such issues by dealing with real-time hand motion detection. During a search on google for "the Rock-Paper-Scissors" game, it brings up plenty of web pages where you may play these games. Figure 1 showcases our machine learning pipeline for hand gesture recognition in Rock-Paper-Scissors. We pre-processed 100 images, then augmented them to create 1,000 variations for MediaPipe to generate skeletal images. Although individual models like multilabel classification and RNN had their specific strengths, they also had limitations in class differentiation and pattern variability. By combining model outputs and refining through activation functions, we enhanced overall performance. Figure 1 presents the project's full workflow, from data collection to model integration, leading to a real-time system that predicts hand gestures with high accuracy, as Figure 2 illustrates in detail.

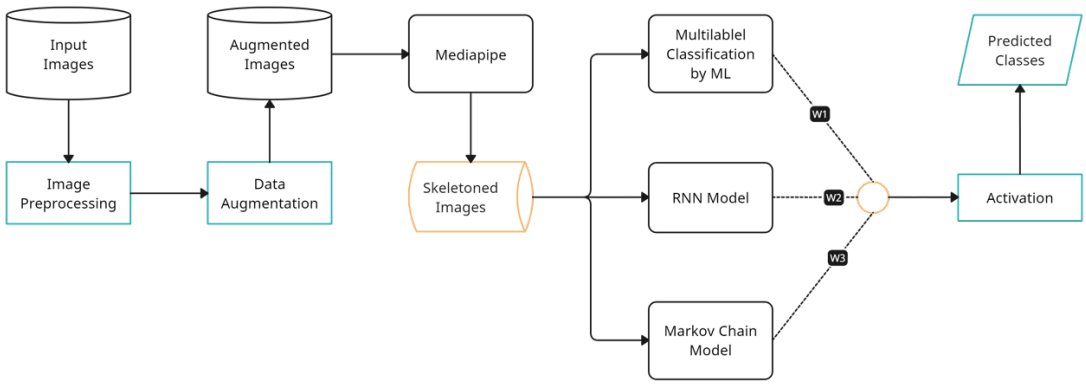

Figure 1: Overview of the end-to-end gesture recognition and AI predictive modelling workflow for Stone Paper Scissor Game.

Playing versus an AI opponent or different human gamers is possible in several of these games. Afiniti.com is an example of either of such web pages. Distant gamers may now duel against one another in a virtual arena, for each hand sign a graphic is clicked by each participant once they have connected to play. Game results and the opponent's moves are shown to the gamers after each has picked their respective moves for the round. Getting icons to press makes it easy for people to be using the software inside an internet browser with little processing capacity. As a consequence, this strategy necessitates a whole new manner of gameplay. Using a computer to game Rock-Paper-Scissors is a very different experience from enjoying the game [16]. We looked at object detection techniques since we wanted gamers to be able to employ their basic bodily motions in conjunction with our effort to classify images. Object detection differs from image classification in that it focuses on detecting distinct component regions together in a picture. Observing the player's wrists & detecting his motion is similar to how a person might compete. Therefore, among the most significant decisions taken throughout this research was indeed the selection of the detecting technique, follow Figure 2.

- Gesture-based systems typically just have a palm picture as an input, but a truly human interface situation presents an image with a great quantity of additional info in the background as well as the movements.
- To fulfil the demands of HCI, where certain motion might send multiple messages depending on how it is performed, current gesture detection and classification only provide classification systems and ignore the geolocation data of the movement.

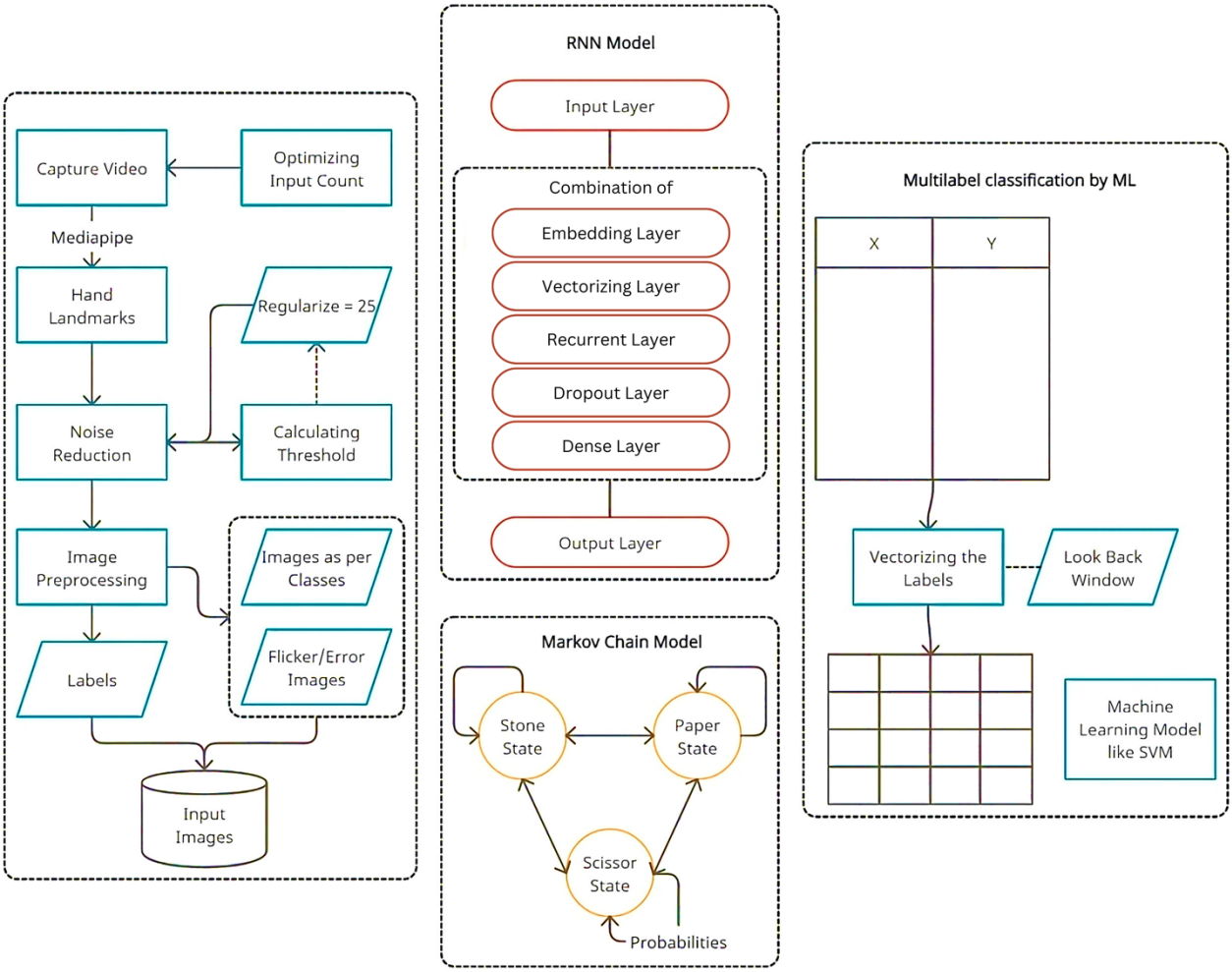

Figure 2: Images are collected, preprocessed, sent through the CNN model, and the output is a refined picture with projections, as shown by this figure.

This work, considering the aforementioned constraints, provides a quick and easy data collecting approach that is unrelated to complexion luminance while demonstrating high precision in low-light circumstances, hence increasing the variety of samples and resolving the information shortage.

It was chosen to use OpenCV for pattern recognition due to the obvious limits on the information and pictures accessible. We recorded hand gestures datasets comprising the user's palm movements all combining static and flickering photos. To prevent the system from mistakenly not recognising the palm, we include flickering classes so to ensure that perhaps the algorithm understands hand motions that aren't even any signals but merely just motion. Our method uses the three motions of Rock, Paper, Scissors to play. There are set no's of images taken at a specific time interval for every move category. For each picture, the user's palm is rotated throughout all 3 axes and also sidewards in almost all the distinct viewpoints. The dataset collection process makes use of OpenCV [17] as well as Matplotlib.

Mediapipe, which is Google's open-source tech, is used to detect palm. These include palm recognition, human supervision, facial recognition or item detection. It was presented in 2019 as a cutting-edge technological advancement. One must use Mediapipe [18], which grips the hands and impresses spots on every joint using its libraries. As an open-source programme, OpenCV uses its libraries for certain of its surveillance and image-processing functions.

Neural networks, such as CNN's, are particularly useful for image analysis. The following explanation will emphasize picture processing because this is the sole usage for this feature underneath the MediaPipe structure. For our model, TensorFlow, a deep learning architecture was incorporated, as with all the necessary modules including Keras & TensorFlow. We also utilise the whole matplotlib toolkit to display as well as build static, dynamic, and live visuals, while CV focuses on inputs that help us display but also evaluate the information.

To build a dataset of real-time pictures at 30 fps, we use OpenCV and a computer's 720p camera for gathering 2,000 photos, which would be the ideal amount of training images. In total, we have 6000 photos in our database, divided into three categories, each with a total of 2000 pictures. Detecting pictures in complicated backgrounds seems challenging again for the algorithm, thus in order to prevent the framework from incorrectly detecting the gestures, we feed on blurry photos. This loop continues until the user pushes a key to quit or until we attain the desired number of photos for a category. Afterwards, each group is separated by a time interval of 50 seconds.

4.1 Object Tracking, Classification and Detection

Palms gestures are detected using a combination of Object detection, tracking and classification techniques. In object detection, "articles" are clusters of pixels that have been grouped. We want to recognise those pixel groups as things that move not just in the x or y-axis, but then in time as well. Motion detection is very effective when it relies on the temporal link among frames of a video. The procedure of identifying the palms inside the pictures is made easier by identifying the locations within every subsequent frame.

The method whereby the identified items inside the picture are categorised as to what the element is known as "object classification. It boils down to determining whatever the object is. Various factors, such as form, movement, coloration, and surface, could be used to identify objects [19]. An entity's movement, stance, pace, and orientation may all be determined by monitoring it over time, as can its relationship to other objects in the scene. Frame-by-frame monitoring of a recognised item in actual time is a huge and challenging subject, for monitoring to be reliable, it is essential to know and comprehend the movement and change of the item in question.

Consequently, we've put photos into a system that can handle temporal features in our method. We may use our algorithm, which is built on a wide range of palm movements in actual time, to follow the item at a pace of 30 fps instead of concentrating on it throughout the test phase. Using Haar Cascade and Mediapipe, we can place our palms at any level giving optimum perception while enhancing accuracy.

4.2 Haar Cascade

Haar Cascade is a machine-learning-based object recognition technique. It operates using many 'features' These 'features' are in fact contrast values calculated over continuous rectangular areas at picture points [20]. Real-time detection is possible by quickly computing these properties using a 'integral image' concept. The Haar Cascade algorithm is based on this. It is good at distinguishing faces and other simple things because it concentrates on intensity differences across surrounding regions in an image.

By nature, the Haar cascade is slow as compared to today's cutting edge technology, therefore, in order to train its Haar features we used Multiscale Processing methods so that our algorithm efficiently analyses images at different scales. Integral photos greatly reduce processing load for real-time object identification, which is necessary for gaming gesture detection. This speed is achieved without losing the accuracy needed to identify desired items. In the training phase, an efficient convergence or reduced training time, highlight these aspects. Efficient model training is a significant breakthrough, especially with this large dataset.

Our work uses the Haar Cascade technique to recognize hand gestures in Rock-Paper-Scissors. Our algorithm dynamically selects Haar Features for real-time input, instead of relying on a fixed set of features, to obtain better accuracy. The OpenCV library's cv2 module simplifies this technique's implementation. The Haar Cascade is stable and effective for real-time hand gesture detection in controlled environments like gaming [21], despite its limits in more complex circumstances.

4.3 Mediapipe

Google designed Microsoft MediaPipe, a complex framework. It aids multimodal (audio and video) machine learning pipeline building. Graph-based architecture underpins MediaPipe. Each graph in this design represents a connected processing node group. These nodes may modify photos and run complex deep learning models. MediaPipe's versatility and real-time data stream management are its biggest advantages. This is particularly useful in real-time hand movement monitoring and categorization systems like ours. One of its biggest virtues is that it can handle video frames at high speeds (up to 30 frames per second in our case) without delay. MediaPipe also works well with TensorFlow and OpenCV to develop sophisticated deep learning models and computer vision algorithms [22]. MediaPipe has limitations. Complex graph-based design makes learning difficult, one of the biggest problems. This architecture may take time and practice to understand. Even though it processes and categorizes real-time data well, lighting, background clutter, and camera quality might affect its performance.

Our analysis found that MediaPipe's hand gesture recognition technology is essential for real-time Rock-Paper-Scissors hand gesture detection and classification [23]. High accuracy and speed provide an interesting and responsive gaming experience and the application's smooth operation. MediaPipe's vast capabilities allow us to manage dynamic hand gesture detection, a crucial component of our study methodology. It is a multimodal framework, which means it applies to a variety of mediums, including sound/video. With MediaPipe, a programmer can fixate upon that method and design features of the proposed, and afterwards assist the software with results that can be reproduced throughout various devices, that also have a few benefits with utilising features just on MediaPipe framework. To deploy output-ready deep learning, the MediaPipe framework was created. It contains released code that accompanies studies and is used to create technological demonstrations. Information manipulations, multimedia processing framework and functional reasoning models also make up the graph modular components of MediaPipe. PyTorch, TensorFlow, PyTorch, OpenCV, etc. all employ a graph of processes as an ML tool.

We used the MediaPipe framework for output-ready deep learning. Which facilitates the development of algorithms with pre-built functionalities, reducing the complexity of background luminescence of an image and promoting exact geometric parameters such as lengths [distance between (0 - 4), (0, 8), (5, 17)]. This distance provided an additional feature set for Deep Learning’s Input layer by leveraging different planes of an image. Augmentation is a great method to simulate different scenarios for different images, but it can be an accuracy eater for some classes of image. This behaviour is because the output of Medipipe leads to skeleton images with different planes, and when we pass them to the Augmentation Image generator, some augmented images can look the same, despite them being coming from different classes or categories. This is because of augmentation parameters such as rotation, alignment, zoom, etc. In our algorithm we fixed the values of such parameters for different classes by reversing this effect.

Two methods are operating together in the Mediapipe’s hand gesture recognition optimiser, one is Hand Landmark Model and the other is Palm Detection Model [24]. The Palm Detection Method delivers an appropriately clipped palms picture, which is then handed onto the Landmarks System for further processing. There is a lot less usage of data augmentation (such as flips, inverting, resizing) in the ML algorithms because of this procedure. Palm detection and landmark placement are traditionally done by detecting the hands and overlaying that on the consecutive frames. A new method is used in this Hand Analyzer to deal with the difficulties of ML pipelines. Recognizing palms is a time-consuming technique that requires picture analysis and thresholding, as well as working across a wide range of palm lengths. There is an easier way of recognising palms besides simply identifying them out of a bounding box: the Palms detectors are learned, which predicts coordinates surrounding immovable items such as the palms and knuckles. This is followed by the employment of a decoder-encoder, which extracts a larger scene background. Hands Feature modelling comes into the equation as soon as hand recognition has been applied to the entire captured image.

We gave the raw images to MediaPipe in different batches, where the images were shot by different resolution, and each batches had four types of images first original image (I), Enhanced Image (E), Preprocessed image (P), Preprocessed + Enhanced Image(PE). This gave an algorithm to have varieties of images. For example

- Batch 01: Resolution: 32 x 32, I, E, P, PE
- Batch 03: Resolution: 64 x 64, I, E, P, PE
- Batch 01: Resolution: 92 x 92, I, E, P, PE, and so on

3-dimensional hand-knuckle coordinates are used to accurately locate 21 important spots inside the identified hands areas by regression, which generates the coordinated projection straight from the palm highlight framework in MediaPipe. There are three values for every hand-knuckle marker: two for the picture widths and heights (x, y), and the other one for the hand's knuckle's depth (z). Value decreases when the hand gets nearer to cam. Drawing.utils is used to display the points over the hands and palms, and holistics is used to combine them [25]. The holistic category contains five criteria, all of which have been outlined below:

- **Static Image Mode:** This method, which defaults to False, handles the incoming images as though they were part of a video stream. If it detects palms in the initial frames it receives, it will attempt to locate the landmarks.
- **Upper Body Only:** Holistic modelling includes facial recognition, eye, posing, and palms among others. For action identification, the parameter is disabled by the standard. When it comes to a holistic approach, hands and posture modelling get a lot in common with each other.
- **Smooth Landmarks:** The lag is reduced by setting this value to true which is the default.
- **Min Detection Confidence:** There is a range of [0.0 to 1.0]. The standard value is 0.5, which results in the rapid detection of a certain element. In our research, we don't want to identify any randomised item with a probability level lower than 0.5.
- **Min tracking confidence:** By default, the value is set to 0.5.

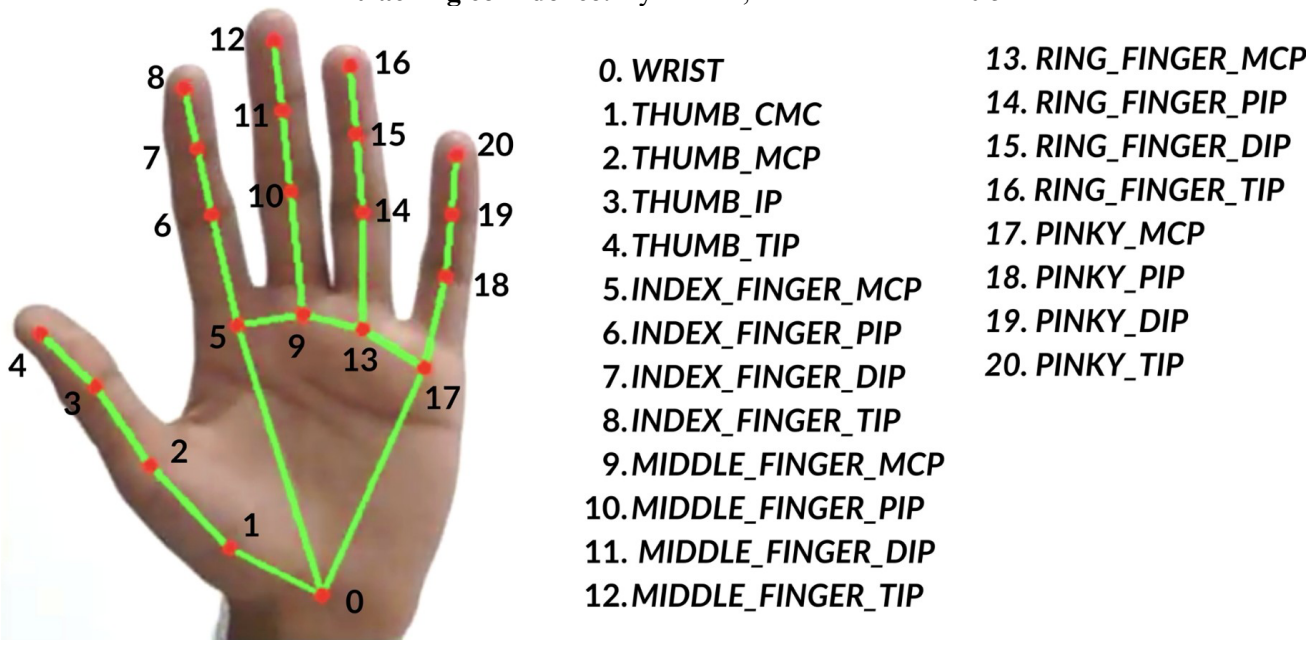

Figure 3: These are all the given mappings for hand landmarks marked by MediaPipe

We employ the Right-Hand Features to classify our scenario since the systems approach incorporates palm connections. For example, point 5 represents the beginning of the index finger (Index Finger MCP), and it is linked to point 8 represents the Index Finger TIP, point 9 represents the Middle Finger MCP, having the points 6 and 7 are also linked, Figure 3. In just about every other location on the picture, it displays exactly the same. Since all of the recognition is dependent on dots, there is no need to worry over coloration or reflecting characteristics in our approach. This method recognizes palms in photos with only partly visible dots too, due to its outstanding range and extensive hand-detection.

4.4 Pre-processing of Images

Currently, our project uses OpenCV, a sophisticated and open-source computer vision and machine learning package. OpenCV, or Open Source Computer Vision Library, helps us pre-process images. OpenCV has several real-time computer vision programming features. OpenCV's fundamentals include image processing and pattern recognition. It converts images to pixel arrays that may be manipulated and explored using its many methods. The optimized C/C++ library supports multi-core processing. Since it performs well in real-time applications, it is excellent for our gesture detection system.

For our project, OpenCV's ability to analyze images quickly is crucial for real-time gesture detection. Due to its wide range of tasks, it can do anything from reading and writing photographs to identifying objects and changing

images. We can adjust our approach to fulfill the Rock-Paper-Scissors game's hand motion recognition requirements. The free and open-source program OpenCV has a large developer community and a lot of documentation and tools for development and troubleshooting. OpenCV excels in controlling lighting and backdrops, however, real-world situations may prevent this. Beginners may find the library challenging to use because to its extensive functionality and image processing requirements.

Our work is mainly on OpenCV for initial image capture and processing. The images are initially captured and evaluated utilizing OpenCV, and afterwards, the analysed outputs are passed onward for further process. After capturing a picture within the window, we use OpenCV to identify the specific palm gestures from the different data kinds by detecting the important areas. The highlighted dots represent these crucial spots, and when meshes are found, we must link them together. As a result, the mediapipe library contains a utility feature called drawing utils, which is used to join the points upon the hands.

In order to proceed, we need two frames, one for obtaining pictures, and the other for plotting the results of those pictures. A black backdrop is required for the second frame. OpenCV utilizes a BGR video channel, whereas Matplot, as well as MediaPipe, are using the RGB video channel, thus the colour channel is changed [26]. Since OpenCV is just a one-shot method, no more computations are required once the colored stream has been converted. Additional data is processed and thus the output is maintained. For the second frame's markers to be plotted, a series of computations are required.

We ought to create a border all around our hand so that we can trim it and save it in a certain directory, Figure 4. Other approaches use a fixed-size rectangular frame to locate the hand's region, which is not the case here. Gaming motion capture doesn't always need accurate placement of the player's hands since it implies an unreasonable increase in the player's effort. Thus, we can see that in earlier research, the precision of a camera's entire visual field has not been addressed.

Figure 4: The right-hand image includes iterations and trials that were performed over dataset photos in order to create a perfect optimised rectangle around the right hand, as well as a mirror image. The left and middle image projections depict the variations tried to get the appropriate border around the hand.

In this case, if the frame we are receiving is shorter than the one we previously surrounded the hand with, we would substitute it. Finally, we get a frame that's nearest to the points on the hand. This frame is then saved in the directory. Once the rectangular border is plotted around the hand, and when the movements were recognised with the pictures being scaled to the same dimensions, we pass this information to the CNN model for later processing. Flipping the image is done using the cv.flip() function with the image as well as the direction it is inverted, horizontally as well as vertically.

We experimented with a variety of resolution ranges while capturing these images. We started with 28x28 pixels, which was too little, and it detected anything that it wasn't meant to identify. Images couldn't be seen there either. Taking 128*128 pixels, the computing effort was far more since the image was larger than required. In the end, we landed on 96x96 pixels since it was the most comprehensive form we could come up with.

The images on the black screen were then stored in a specified directory in our training phase program. All inputs used are set to release, as well as all frames that are deleted when the images were collected and saved.

5. Data Preparation

A GPU, instead of the CPU, Figure 5, is utilised in the computations for speedier execution. This is because a GPU is built to generate high-resolution pictures and video all at the same time, and that they are optimal for developing AI and neural network models since they can handle numerous calculations instantly. Since the CPU acts as a portal between the sources and the GPU's components, we can't eliminate anything from the ML setups. Quicker the data flows, the faster the computations are. We use the Nvidia GTX 16 series of GPUs (1660 TI) with 6 GB of graphics in all computations, even though the efficiency varies from one GPU to another. We tried utilising Nvidia's GTX 1650 Ti, which has 4 GB of graphics, though the earlier works brilliantly with 6 GB of graphics and has superior overall effectiveness.

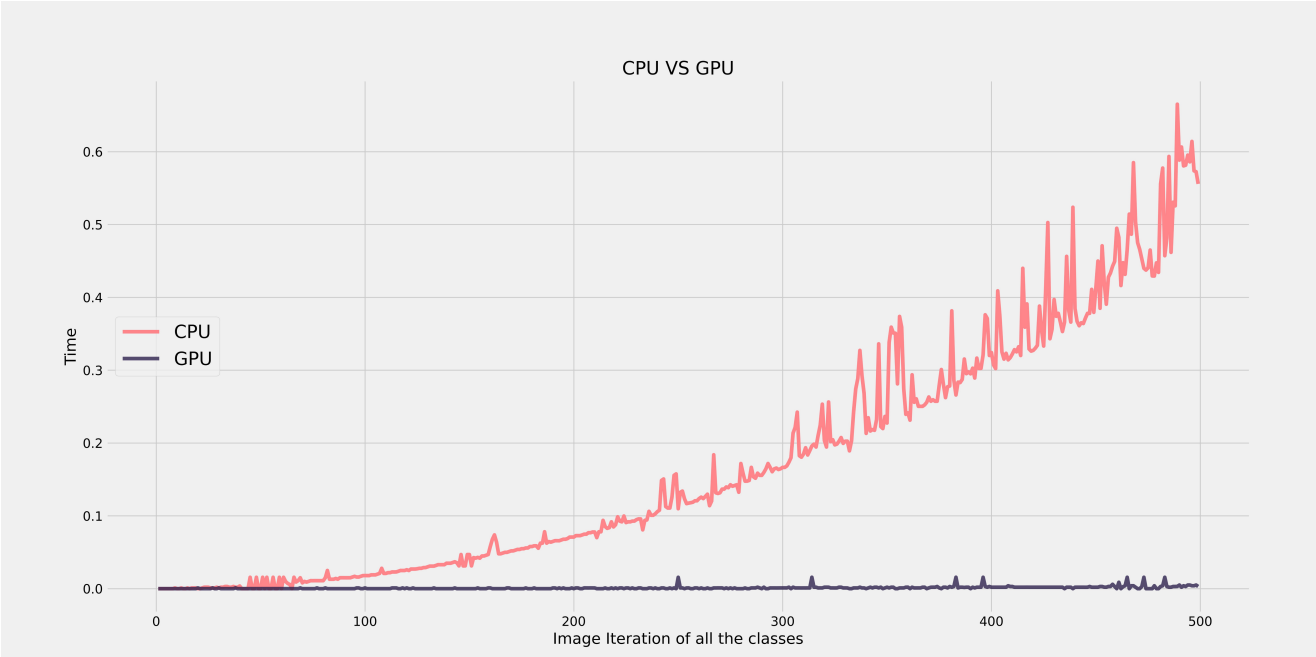

Figure 5: The CPU and the GPU are compared in terms of time.

In our network, we used TensorFlow release 2.5.0, and we also added a number of critical modules, including the Nvidia Graphics, the Cuda Toolkit (v.11.0), cuDNN (v.8.2), and Tensorflow (v.2.5.0). The latest release of Tensorflow that is compatible with that of the Cuda library is 2.5.0, and this release provides the best overall accuracy. Our final dataset comprises three input types, namely rock, paper, and scissors, and 2000 photos each, Figure 6. The dataset is saved in a particular class list that includes just about all the JPEG images that were gathered when the process ended.

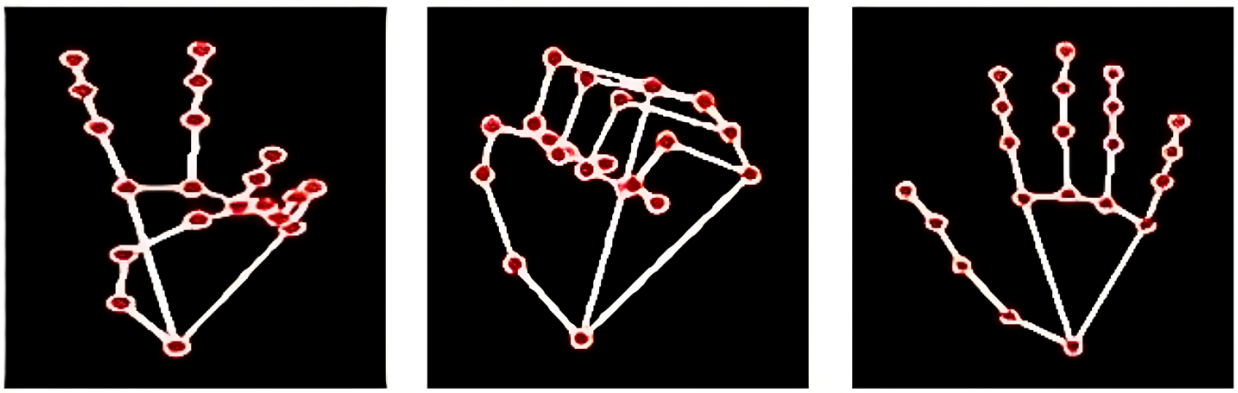

Figure 6: Resized three classes Scissor, Rock, Paper used in processing.

We give particular emphasis to the data's format, which is (3, 2000, 96, 96, 3), with 3 denoting the total classes, 2000 denoting the number of photographs in each, 96x96 denoting the image's dimension, and 3 denoting the RBG stream. The photos are therefore stored in a variable named "X" that contains a collection, while the class indices are stored in "Y", which are two separate list variables. The entire arrays are constructed, with the said form of "X" being 6000, 96, 96, 3, and the size of 'Y' being 6000. If we attempt to examine the 2456th picture in 'Y,' for instance, we will see the 456th picture of the 2nd class, i.e., 'Y' = '1' = Paper. If we divide it by 255, it also helps us to feed the computational model with a normalised "X".

6. Splitting Data

The training system data should make up 80 percent of the database's total size, and the test data should make up 20 percent of the database's total size, which is currently the optimum situation presented for our dataset. We randomised the entire database using rounds to acquire all of the information groups in a generic pattern to guarantee that the photos remain randomized. Randomness is fixed at 2, as well as the shuffling is set to true, so we can select any phase we want. If it had been set to negative, pictures wouldn't be jumbled, as well as the epoch would only learn to recognise pictures based on paper.

7. Training Stage

An ANN is a complicated intelligent network that modifies its inner architecture depending upon the data it receives. Changing the weight of the network is indeed the key to achieving this effect. Animals' neurological systems are mimicked by artificial neural networks (ANNs). Animals' intellect is the consequence of multiple cells' learning through perceptions and surroundings. ANNs also train by seeing as well as interpreting the patterns of inputs and outputs.

The weight of every input varies. The output signal is transmitted towards the succeeding neuronal when the aggregate of these data exceeds the limit. For instance, the ANN is a part of supervised methods. Components of an ANN are linked to each other to form connections of information. Getting this information out of the brain is a challenge. For the same reason, mining techniques have been driven to derive principles for classifying data. The database is the first step in the prediction process. The training data as well as the test sample are separated from the rest of the information. To build a system, training data is utilised, whereas test data is used to evaluate the classifier's precision.

ANN is a less common alternative for visual analysis due to its dependency on reliable data sources. The spatial properties of a picture are lost while using ANN. In photography, spatial characteristics are the organisation of pixel dots. CNN, on the other hand, is able to use pictures as input properly [27]. Images may be filtered to produce extracted features. When producing a network, CNN doesn't capture the information in a forward-facing manner but instead references the very same information over and over again.

The CNN classification is now regarded as among the most cutting-edge techniques for ML. Visual categorization difficulties may be solved with CNN and in numerous datasets, CNN-based study increased the greatest result. It excels in analysing patterns both locally and globally in images. In this way, basic characteristics like borders and slopes may be merged to create increasingly complicated elements like angles and shapes, and ultimately entities. CNNs are made up of levels of neurons. Numerous 2D vectors (or channels) are fed into the convolution operation, and many two-dimensional matrix formed as the result of object detection operations. Each matrix has an incoming and outgoing number that varies, and a single control matrix is computed as follows:

$$A_j = f\left(\sum_{i=1}^{N} I_i \times K_{i,j} + B_j\right) \quad (1)$$

An initial kernel grid (Ki, j) is used to complicate the input data (Ii). The total of each convoluted grids is again calculated, while each member of the resultant grid is given a biased factor, Bj, Equation 1. Lastly, every component of the preceding matrices is subjected to an "f" (activation function — Non-linear one) in order to generate a single output, Aj. Kernels matrix contains a localised feature extraction technique that gathers geographic characteristics from source grids. The goal of the training technique is to identify groups of kernels matrix K that can be utilised for picture categorization. Kernel matrix, as well as biases, may be trained using the back-propagation learning approach, which is used to improve artificial neural network values.

CNN employs filters and many tiers of pictures to evaluate visual information. Among these levels are the math tier, a corrected linear module, as well as a completely linked overlay. In order to interpret the connections which, the networks "see," to analyze the input, as well as to generate an n-dimensional vectors result, several layers are necessary. When looking at the picture source, the n-dimensional result is what's utilised to link the many characteristics that may be seen. The categorization result may then be sent to the recipient. Building pieces of this Convolutional Neural Network sequential model may be found at various levels, Table 1.

Table 1: Overview of parameters for Convolution Neural Network and Configurations.

Layer Type	Parameter	Range	Description
Convolutional Layer	Filters	32	Pattern-finding filter used to enhance organic appearance of photos.
	Kernel Size	(3,3)	Grid size for convolution operation.
	Strides	(1,1)	Number of pixel dots displaced across the picture.
	Padding	'same'	Padding type to ensure output layer is the same size as input neurons.
	Activation	ReLU	Rectified Linear Unit (ReLU) activation function for improved gradients propagation and faster functioning.
	Regularization	L2 (0.01)	L2 regularization applied to convolutional weights to prevent overfitting.
	Dropout	0.2	Dropout rate to minimize overfitting by randomly dropping out neurons during training.
Batch Normalization	Momentum	0.9	Momentum term for updating moving mean and variance in batch normalization.
	Epsilon	1e-5	Small constant added to variance to avoid division by zero in batch normalization.
Max-Pooling Layer	Pool Size	(2,2)	Size of the pool for minimizing variables in a pooling layer.
	Strides	(2,2)	Number of pixel dots displaced across the picture in the pooling layer.
	Padding	'valid'	No padding to reduce spatial information quickly.
	Pooling Type	Max	Max pooling to automatically select the largest feature from a set of features in the pool.
Global Average Pooling	-	-	Global Average Pooling to reduce spatial dimensions to a single value per feature map, aiding in model generalization.

Fully Connected Layer	Neurons	128	Number of neurons in the fully connected layer.
	Activation	ReLU	Rectified Linear Unit (ReLU) activation function for the fully connected layer.
	Regularization	Dropout (0.3)	Dropout rate for fully connected layer to minimize overfitting.
Output Layer	Neurons	3	Number of neurons in the output layer for three classes (stone, paper, scissors).
	Activation	Softmax	Softmax activation function for multiclass classification, providing probabilities for each class.
Optimization	Optimizer	Adam	Adam optimizer for updating weights during training.
	Learning Rate	0.001	Learning rate for controlling the step size during optimization.
	Decay	1e-5	Learning rate decay to gradually reduce the learning rate during training.
Loss Function	Loss	Categorical Cross entropy	Categorical cross-entropy loss function for multiclass classification.
	Metrics	Accuracy	Accuracy metric to evaluate the performance of the model during training and testing.

We use two layers of convolutional data that affect the amount of data to improve efficiency. The outcome of the preceding layers is consolidated into a unified layer before even being utilised in the outputs of another level, which is a flattening one. Hidden layers and dropouts are also included again in the architecture to make it more general. The parameters supplied are passed on to the output layer, which does the actual unit conversion, Table 1. Since SoftMax offers us the likelihood of categorising a picture into distinct classes, we utilise ReLU activating again for hidden nodes and SoftMax activation for such outputs.

Compiling the methodology with the hyper-parameter, measurements and loss function is the next step after the framework has been created. Here, CNN graphics have been optimised using Adam optimiser. Categorically sparse cross-entropy produces the error between the measured and predicted values. As required, weights are increased and lowered once they have been determined. In this case, the measures are based on prediction performance. Setting the number of epochs to 10 allows for quicker and more efficient model training. Changing the epoch minimizes the quantity of processing required. When we adjust the epoch lower, it reduces the amount of crunching. When we attempted to increase the number of epochs from 15 to 25, we concluded that the model wasn't generalizing because of a lack of data, Table 1. There's only a maximum of 50 seconds spent on each period, and the outcomes were credited and due to speedier GPUs and more processing. We were able to attain 100% correctness in our testing predictive performance. There was one class with 100% accuracy and the remaining two classes with 99% and 98% of accuracy in the classification reports, making it an exceptional achievement. We implement a predictive model that shows the picture's action and classification instantaneously at 30 fps to test the real-time correctness.

CNN was used as a major model in the NN architecture instead of the more standard ANN, which has a larger quantity of units and a greater danger of overfitting [28]. The filters are learned dynamically by CNN, and they do not need to be mentioned specifically [29]. Input data may be filtered to focus on the most important elements. Such spatial characteristics are collected using the images that are fed into a CNN. This makes CNN great for extracting a large number of characteristics. As a result, CNN doesn't have to spend time calculating each and every characteristic.

As 2-dimensional pictures must be reduced to 1-dimensional vectors before they can be used in ANN, classifying images becomes more complex. This greatly expands the number of variables that may be used to train models. Memory and computational power are required to increase the number of learnable parameters as well. To put it another way, it'd be prohibitively costly. CNN's key benefit over its competitors is that it can immediately identify the most essential elements despite the need for human intervention. For our machine vision and image analysis studies, CNN has proven a great answer.

8. Player 1 vs Player 2

After defining the hands and movements used in the game, we can now develop a function that allows two players to compete at the same time. As a result of this feature, two people may compete against one another with two parameters being provided to each of their hands at any one time.

Only when both hands are provided to the camera does the frame in the game begin operating, preventing the game from being started incorrectly. 0 is for stone, 1 is for paper, and 2 is for scissors in this function, which is used to forecast outcomes between two players. You might think of it like this: If there are two players with stones on either side, then it's an equal match. The same rules apply to all other circumstances. If there is a tie, 0 gets returned, while if the left player wins, then -1 gets returned, and if the right player wins, then 1 gets returned.

This cell also contains the variables that will be used in the game, such as keymax, thickness, regularization, and res count, and the start is set to false. The trigger code is equivalent to a stone and paper, which has to be shown through hands by each player at the beginning of the game to start. For the following one minute, the images are taken, and the maximum count of wins and losses for each player is returned, as well as the results of which players win. The code was restricted to a minute, which could be altered at any time, to prevent the loop from going on indefinitely.

9. Player vs AI

AI is a cutting-edge technology in the game industry. Deep learning as well as machine learning techniques have allowed games to become more intelligent. In current games, players are encouraged to think about their own patterns and then respond accordingly. AI is helping developers come up with new ways of doing things.

We've thought about the possibility of having two players, such as Player 1 and Player 2 above. However, there may be a situation when a person is both online and offline and has no one to play with. In order to play Rock-Paper-Scissors with a computer, we've built an AI model of the game. In our search, we came across a variety of models as well as algorithms, from which we chose the most often used ones in games, which were:

- Firstly, the Markov chain model
- Second, SVM with multi-label classification
- Finally, RNN, along with LSTM and GRU

The neural network model was previously trained to classify the gesture as either rock-paper-scissors before this. Here, we're using the model's current state to forecast its future state, or sequence. Among the most common examples of this sort of application is the weather forecast model, which includes parameters such as temperature, humidity, and climate change. In light of this information, the model forecasts what the weather will be like on the next day. However, in our model, the elements of the model—such as rock-paper-scissors—serve as the model's characteristics. As a result, we may anticipate the model's future state/sequence using an existing sequence of rock-paper-scissors.

9.1 Markov Model

We employed the Markov Chain, a wonderful probability theory notion, to anticipate the order of rock-paper-scissors choices in our analytical inquiry [30]. This paradigm holds that a system's future state is solely affected by its present state, not its history. In circumstances when immediate antecedents directly affect outcomes, the approach is very beneficial. Rock-paper-scissors has internal dynamics, and this feature reflects them. Based on this structure, the Markov Model cycles between states that represent options like rock, paper, or scissors. A key part of this process is the Markov Matrix, which painstakingly calculates the likelihood of each possible outcome based on an individual's most recent choice. This model's strength is its sequential dependency recognition and utilization. Consequently, our artificial intelligence can make informed forecasts regarding the player's next move, improving its future prediction ability.

This model is not constraint-free, It makes the erroneous assumption that the choice preceding it completely affects subsequent options. It entirely overlooks long-term trends or intricate methods a player may use. Due to this underlying assumption, the model's prediction accuracy may be restricted, especially against tactically clever opponents. In the context of our game, the Markov Model is essential for real-time AI improvement. Consider a rock-paper-scissors sequence with rock having a higher probability. Instead of guessing, the model predicts "rock." The probability from the continually evolving Markov Matrix were used to calculate this prediction. The Markov Model in our artificial intelligence system helped us create a dynamic and strong opponent. This foe can strategically respond to human techniques. This integration shows probabilistic models' ability to create interactive, AI-driven interfaces. The game's experience tapestry improves, and strategic predictability expands.

As a statistical technique for modelling and forecasting the impacts of future modifications, the Markov chain is used to calculate the likelihood of transfers among various states. Decisions made by humans are constantly influenced by their thoughts and feelings. In the game of stone-paper-scissors, this is also the case. Using this method, the outcome of the earlier games impacts the player's moves in the following game. The Markov matrix is a distinctive characteristic of such a clever system that incorporates the game outcomes into it.

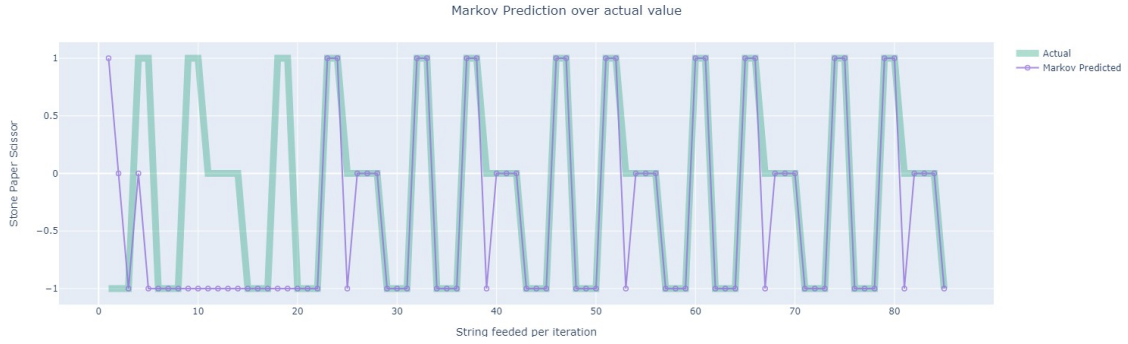

Figure 7: Strings are fed per iteration, predicting Markov over actual value.

Stone, paper, and scissors are the three states of play in the game. All three states are regulated by a total of 9 possibilities. Based on the previous move, the Markov model predicts the player's future move, Figure 7. For example, the previous sequence of the game is stored as Stone, Paper, and Scissors. Since the likelihood of stone occurring is larger than that of paper and scissors, the Markov model will verify the previous sequence of events when forecasting the player's next move and, based on that, predicts the next move as S, i.e., stone.

9.2 Multi-label Classification Via SVM

We employed the Support Vector Machine (SVM) approach, a well-established supervised machine learning method, to classify multi-labels in our study [31]. We studied how to improve a neural network's ability to anticipate a player's next rock, paper, scissors move. The support vector machine (SVM) excels in data classification [32]. The hyperplane is placed in the best multidimensional space to clearly separate data points.

In our research, the SVM model processes the player's past hand motions, which the neural network previously categorized as rock, paper, or scissors, in a sophisticated way, which lets the model anticipate player behavior. A player's hand gesture starts the operational dynamics, which cycle through the motions. This approach dynamically modifies the sequence count to fine-tune the prediction model output. A one-dimensional array catalogs movements during the procedure. After a thorough sequence analysis, this array is separated into two multi-label groups, X and Y. For example, if the window size is 5 and the sequence is Stone, Stone, Stone, Paper, Paper, Scissors, Scissors, then the x would be Stone, Stone, Stone, Paper, Paper, and y, i.e., the prediction would be Scissors.

As it becomes a retrieval framework, SVM encodes gesture sequences and divides them into training and testing subsets. This showcases the program's excellence. This division is fundamental because it allows the model to forecast based on previous trends. Our main benefit of adopting support vector machines (SVM) is its constant performance in managing large-dimensional data and amazing abilities in binary classification, Figure 8. This trait is crucial for predicting a player's gesture. It's crucial to know that SVM's computational needs are high and that its efficacy may diminish with a big number of datasets. We carefully managed this constraint throughout our investigation.

Figure 8: Strings are fed per iteration, predicting Multi-Label SVM over actual value.

9.3 RNN, LSTM & GRU

Our research processes sequential data in our human-versus-computer rock, paper, scissors game using an advanced blend of RNN, LSTM, and GRU. RNNs are crucial to our method, they are best for sequential tasks like ours because they can keep and reuse input via internal loops [33]. Our technique relies on the typical RNN structure, which transmits data between nodes. Its capacity to evaluate sequences by retaining a "memory" of former inputs with the current input makes it ideal for tasks that need context from earlier data points. Due of their poor short-term memory, classic RNNs struggle with long-range dependencies.

Our RNN design uses LSTM units to bypass this restriction. Internal gates in LSTMs let the model retain longer data sequences. These gates are excellent at determining what information should be saved and what should be discarded, allowing the network to store relevant information throughout extended periods. This trait of LSTMs is similar to how the brain prioritizes information. This is necessary to precisely predict the opponent's next move in our game. Long short-term memory (LSTM) networks manage long-term dependencies well, but their complexity requires a large number of units to execute well [34].

Figure 9: Strings are fed per iteration, predicting RNN over actual value.

A Gated Recurrent Unit (GRU) is a type of RNN. Like LSTM, but with a smaller number of parameters, it is learned quicker. Hyper-parameters such as input size and batch size, layers and classes are all configured in the RNN model at the start of the training process. A function is developed, and constraints are added in order to evaluate movements. The model is constructed, and the RNN model is fed a series of inputs. Additional movements may be predicted based on the user's prior actions using an RNN model. GRUs may be learned faster than LSTMs, balancing model complexity and performance, Figure 9.

Our method deconstructs recurrent neural networks (RNNs) into connected units with the same architecture but different weights to efficiently analyze sequential input. Since it allows the model to include both the player's current and past moves, this arrangement is necessary to properly estimate the opponent's future move. To provide accurate model outputs, hyper-parameters like input size, batch size, layers, and classes must be configured from the start of training. Based on the player's prior actions, our model employs the RNN architecture to anticipate future movements. Define a function and define restrictions to evaluate.

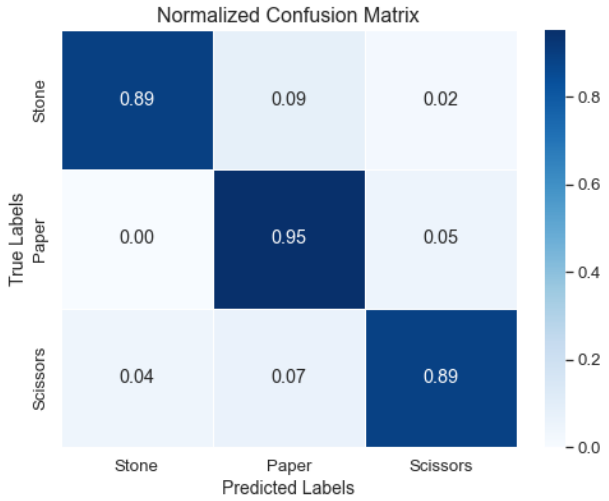

Figure 10: Analysis of gesture classification accuracy: stone, paper, and scissors predictions assessed.

The Figure 10 displays normalized confusion matrix quantitatively evaluating our classification model's accuracy for the Rock-Paper-Scissors game. High accuracy rates—89% for stone, 95% for paper, and 89% for scissors—demonstrate the model's efficacy, especially for paper. Controlled data augmentation refined the model's ability to generalize, preventing misclassification of augmented images. By combining models and optimizing parameters, we achieved a balanced accuracy across all classes. This matrix not only confirms the robustness of our tailored approach but also shows promise for adaptation to new datasets, ensuring a reliable and versatile gesture recognition system.

Three advanced neural network technologies—RNN, LSTM, and GRU—provide a solid foundation for our game by effectively assessing and learning from sequential gesture data. This paradigm lets us accurately forecast opponent moves. This integration emphasizes the benefits of each strategy while compensating for its drawbacks, creating a more accurate and reliable prediction model.

10. Results

Ultimately, the objective of this research was to develop an intellectual in-game algorithm for rock, paper, and scissors who would be able to distinguish and overcome her/his opponent's patterns of natural gameplay. As a result, a dataset was developed in tandem with the study using CNN's multiple ML and deep learning models. Images from OpenCV and also Matplot are transitioned from Blue-Red-Green (BGR) to Red-Green-Blue (RGB). Then the MediaPipe and Haar Cascade were used as effective tools to identify complicated hand gestures accurately. With the use of 6,000 preprocessed pictures, three-hand gestures were employed in the game to pit not only one player against another but also a person against artificial intelligence (AI).

We were able to get the best results by utilising real-time hand monitoring and adaptive magnification of the hand frame. Our model recognised when the hand went forward or backwards, and all photos were scaled to the specified dimensions before being captured and saved. All of this contributed to a more precise identification of the hands. During this particular brainstorming session, the only topics discussed were the pictures presented through Mediapipe and the scaling and storing of the dark background image.

As a result, we didn't have to create a CNN from the ground up, which reduced the amount of time it needed to gather and utilise information. A vast number of layers and several variables were used in training CNNs: filters, kernel size, strides, padding, and activation having values of 32, (3, 3), (1, 1), "same" and "ReLU" respectively. The use of minimum computer resources resulted in a model which was trained in less time than current and existing best practices. The use of blurry and flickering images of the hands helped recognise their motions better, i.e., rock-paper-scissors. Our results were ideal at 30 fps because of our extensively trained model, which attained 99 percent accuracy for the classes, exceeding state-of-the-art performance, and precisely describing hand movements. One class achieves the highest f1 value of 1.00, while the other classes have f1 values of 0.99 and 0.98, respectively. Precision and recall were identical to those of the f1 score.

We can see in Table 2, that Model 3 employing RNN coupled with LSTM and GRU achieved the greatest prediction performance, not only in training but in the testing sample for our RNN gaming prediction models too. Contrary to any other algorithm, our ultimate accuracy of 0.94 is the best we've ever achieved. As a machine learning model, model

2—Multi-label Classification through an SVM—did pretty well, as demonstrated in the classification report. It doesn't perform as well as RNN, which was due in part to anomalies detected in SVM, but after eliminating them, it effectively predicted, seen in Table 2. By far, greatest results from ML model are achieved by machine learning model, which has an efficiency of 81 percent on the training dataset and 60 percent on the test dataset.

Table 2: Comparison in Scores of Markov, SVM ML, RNN on training and testing data

			-1	0	1	Accuracy	Macro Avg	Weighted Avg
Precision	Training	Markov	0.71	0.88	0.95		0.85	0.83
		SVM ML	0.75	1.00	0.86		0.87	0.84
		RNN	0.88	1.00	1.00		0.96	0.96
	Testing	Markov	0.44	0.60	0.50		0.51	0.51
		SVM ML	0.53	1.00	1.00		0.84	0.83
		RNN	0.70	1.00	1.00		0.90	0.90
Recall	Training	Markov	0.95	0.62	0.75		0.77	0.80
		SVM ML	0.93	0.48	0.90		0.77	0.81
		RNN	1.00	0.78	1.00		0.93	0.94
	Testing	Markov	0.73	0.38	0.25		0.45	0.48
		SVM ML	1.00	0.25	0.83		0.69	0.68
		RNN	1.00	0.67	1.00		0.89	0.89
F1-Score	Training	Markov	0.81	0.73	0.84	0.80	0.79	0.80
		SVM ML	0.83	0.65	0.88	0.81	0.79	0.80
		RNN	0.94	0.88	1.00	0.94	0.94	0.94
	Testing	Markov	0.55	0.46	0.33	0.48	0.45	0.46
		SVM ML	0.70	0.40	0.91	0.68	0.67	0.65
		RNN	0.82	0.80	1.00	0.86	0.87	0.86
Support	Training	Markov	37	24	24	85	85	85
		SVM ML	43	21	21	85	85	85
		RNN	38	23	24	85	85	85
	Testing	Markov	11	8	8	27	27	27
		SVM ML	8	8	6	22	22	22
		RNN	7	9	6	22	22	22

First, we notice an accuracy of 80% on the training set and a little over half that, 48% on the testing set for our first model, the Markov chain model that combines preceding series based on its preceding series, Table 2. There is a significant research gap, which we want to address with this project in the future, and we expect to develop a new kind of ML model with an accuracy comparable to that of an RNN model. For the training and testing data, the precision for predicting the "left player's wins" is above 88% and 70%, respectively, while the "right player's wins" and "the gameplay ties" are predicted to be over 100% for both training and testing samples. Moving forward, we want to work on increasing the recognition accuracy and adding dynamic gestures like swiping up and down, as well as exploring alternative hand-held or finger-game possibilities.

11. Advantages and Disadvantages to Society

In artificial intelligence and human-computer interaction, our paper discusses a study that helps society but admits its limitations compared to current research.

11.1 Advantages

- Our system's accuracy in a variety of environmental situations is a huge step toward making sophisticated gesture recognition technology more accessible and suitable to daily usage. Educational tools, interactive games, and rehabilitative treatment may benefit from this technology.
- Our approach uses predictive analytics from improved Markov chains, support vector machines, recurrent neural networks, LSTM, and GRU, making it unique. Advanced artificial intelligence may predict human behavior. This might be valuable in user interface design, behavior analysis, and healthcare prediction models.
- Our gesture recognition system can revolutionize interactive learning [35], allowing users to navigate educational content through intuitive hand movements. Imagine an application where children use gestures to learn sign language or control interactive stories, making education a kinesthetic and engaging experience, catering to diverse learning preferences and potentially enhancing engagement and retention.

11.2 Disadvantages

- Compared to traditional AI models, our technique requires a lot of visual data and computing resources for training and implementation. This may cause data storage, computing power, and energy consumption issues for applications in resource-constrained contexts.
- The method may be less adaptable in smaller circumstances where less complex machine learning models may be used due to the difficulties of mixing CNN, RNN, and SVM. This is due to implementation difficulty. Because of this complexity, developers, or researchers with minimal knowledge in these systems may confront admittance issues.

In conclusion, this work has illuminated artificial intelligence and its human interactions. It has highlighted the significant education and healthcare prospects it creates. Along with the difficult stance of implementing these sophisticated technologies, we are aware of the obstacles we will encounter, such as the massive demand for data and resources. Instead of impediments, these challenges will guide our future research and advances in this intriguing field. They will challenge us to develop clever, general, and flexible AI solutions.

12. Conclusion

We have made considerable strides in merging powerful machine learning algorithms with real-world human-computer interaction, according to this study. Through our Rock-Paper-Scissors system, we have pioneered gesture recognition and predictive analysis while establishing an engaging platform for human connection with technology. This lets us use technology beyond entertainment. We achieved success by using cutting-edge technologies like CNNs, Markov models, SVM multi-label classification, and RNNs with LSTM and GRU. We succeeded thanks to this technology. Combining these technologies with MediaPipe and Haar Cascade has helped recognize Rock-Paper-Scissors hand gestures. It's a huge achievement in computer vision since our system can operate without interruptions in real time, recognize hand gestures at 30 frames per second, etc. A major issue was the absence of datasets that met our needs. Our solution was a unique dataset with several hand gestures. This dataset was crucial to our model's training and will be used in future studies in data-poor locations.

Our research shows technology can understand and interact with human behavior. Given our algorithms' accuracy and capacity to adapt to and anticipate human behaviors, we may be able to design systems that smoothly integrate into human-centered environments. Our technique works in low-light and different skin tones, proving its adaptability. No extra hardware is needed to utilize it. Our technology's sophisticated gesture recognition extends into the realm of smart device interaction, offering a seamless way to control various applications and devices through natural movements. For instance, it could be integrated into smart kitchens, where chefs can navigate recipes, control kitchen appliances, or even adjust environmental settings without having to touch surfaces, thereby maintaining hygiene and efficiency [36]. This approach not only streamlines complex tasks but also enriches the user experience with hands-free convenience, bringing a new level of interaction to smart home ecosystems.

13. Future Scope

Future applications and modifications to our Rock-Paper-Scissors artificial intelligence system are broad and comprehensive. Starting with our current achievements, we wish to explore other pathways to growth. First, the user interaction model must be much improved. More intuitive interaction models and natural language processing (NLP) may allow players to start, control, and interact with the game by voice commands in future editions. This would make the game more accessible, particularly for players with physical limitations that prevent proper hand movements.

The dataset must be diverse and rich in content for the system to develop. Our dataset is beneficial, but expanding it to encompass more gestures, user demographics, and environmental conditions will improve the system's robustness and accuracy. Data from a larger variety of age groups, ethnicities, and hand shapes might make our artificial intelligence more adaptable and ubiquitous. Another option is to investigate sophisticated machine learning methods and systems. Generative Adversarial Networks (GANs) and Transformer models provide more complicated and efficient pattern detection and predictive analytics. Our approach relies on convolutional neural networks, but these methods provide additional possibilities. We wish to research mobile Rock-Paper-Scissors development. This real-life application is intriguing. This mobile app might make our artificial intelligence system more accessible by taking use of smartphones' ubiquitous availability and higher capabilities. The game may strengthen cognitive and motor abilities while being fun. It offers a fun way to improve reaction speeds and hand-eye coordination.

We also wish to examine fields other than gaming systems. Our technology might be used in remote physiotherapy. In this therapy, patients do guide hand movements while the artificial intelligence checks their performance and gives feedback. This might reinvent home-based rehabilitation programs, making them more effective and appealing. Finally, our Rock-Paper-Scissors artificial intelligence system has many development and application possibilities. The future holds opportunities to test artificial intelligence's limits in games and more realistic real-world settings. These include expanding the dataset, enhancing user interaction, and testing new machine learning approaches.

References

[1] M. Zink, P. Friemann, and M. Ragni, "Predictive systems: The game rock-paper-scissors as an example," in PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I 16, Springer International Publishing, 2019, pp. 514-526.

[2] E. Brockbank and E. Vul, "Formalizing opponent modeling with the rock, paper, scissors game," Games, vol. 12, no. 3, p. 70, 2021.

[3] T. Komai, H. Kurokawa, and S. J. Kim, "Human Randomness in the Rock-Paper-Scissors Game," Applied Sciences, vol. 12, no. 23, p. 12192, 2022.

[4] J. Pacheco, J. Ferreira, H. Tavares, and M. Miranda, "Machine Learning Tool for Kids: A Contribution to Teaching Computational Thinking in Schools." Note: If this is from a specific conference, journal, or book, please include that information for a more precise citation.

[5] F. Ahmed, W. A. Khan, M. Iqbal, A. R. A. Abazeed, H. Alrababah, and M. F. Khan, "Rock-Paper-Scissors Image Classification Using Transfer Learning," in 2023 International Conference on Business Analytics for Technology and Security (ICBATS), IEEE, Mar. 2023, pp. 1-6.

[6] H. Brock, J. P. Chulani, L. Merino, D. Szapiro, and R. Gomez, "Developing a lightweight rock-paper-scissors framework for human-robot collaborative gaming," IEEE Access, vol. 8, pp. 202958-202968, 2020.

[7] A. Gokul, A. Dixit, A. Tripathi, S. R. Srivastava, and P. Malathi, "Playing Games Using Hand Gesture Recognition."

[8] Y. Yamakawa and K. Yoshida, "Teleoperation of High-Speed Robot Hand with High-Speed Finger Position Recognition and High-Accuracy Grasp Type Estimation," Sensors, vol. 22, no. 10, p. 3777, 2022.

[9] M. Ghasemi, G. H. Roshani, and A. Roshani, "Detecting Human Behavioral Pattern in Rock, Paper, Scissors Game Using Artificial Intelligence," Computational Engineering and Physical Modeling, vol. 3, no. 1, pp. 25-35, 2020.

[10] J. Chen, Z. Xu, Y. Li, C. Yu, J. Song, H. Yang, ... and Y. Wu, "Accelerate Multi-Agent Reinforcement Learning in Zero-Sum Games with Subgame Curriculum Learning," arXiv preprint arXiv:2310.04796, 2023.

[11] W. Hu, G. Zhang, H. Tian, and Z. Wang, "Chaotic Dynamics in Asymmetric Rock-Paper-Scissors Games," in IEEE Access, vol. 7, pp. 175614-175621, 2019, doi: 10.1109/ACCESS.2019.2956816.

[12] T. W. Cenggoro, A. H. Kridalaksana, E. Arriyanti and M. I. Ukkas, "Recognition of a human behavior pattern in paper rock scissor game using backpropagation artificial neural network method," *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2014, pp. 238-243, doi: 10.1109/ICoICT.2014.6914072.

[13] B. Kim and K. Lee, "Study on hand gesture recognition with CNN-based deep learning," International Journal of Computational Vision and Robotics, vol. 11, no. 6, pp. 571-579, 2021.

[14] P. Tu, J. Li, H. Wang, T. Cao, and K. Wang, "Non-linear chaotic features-based human activity recognition," Electronics, vol. 10, no. 2, p. 111, 2021.
[15] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," Artificial Intelligence, vol. 172, no. 6-7, pp. 716-737, 2008.
[16] E. Kayhan, T. Nguyen, D. Matthes, M. Langeloh, C. Michel, J. Jiang, and S. Hoehl, "Interpersonal neural synchrony when predicting others' actions during a game of rock-paper-scissors," Scientific Reports, vol. 12, no. 1, p. 12967, 2022.
[17] A. Sharma, J. Pathak, M. Prakash, and J. N. Singh, "Object detection using opencv and python," in 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), IEEE, Dec. 2021, pp. 501-505.
[18] A. K. Singh, V. A. Kumbhare, and K. Arthi, "Real-time human pose detection and recognition using mediapipe," in International Conference on Soft Computing and Signal Processing, Singapore: Springer Nature Singapore, June 2021, pp. 145-154.
[19] S. Xu, J. Wang, W. Shou, T. Ngo, A. M. Sadick, and X. Wang, "Computer vision techniques in construction: a critical review," Archives of Computational Methods in Engineering, vol. 28, pp. 3383-3397, 2021.
[20] A. P. Ismail, F. A. Abd Aziz, N. M. Kasim, and K. Daud, "Hand gesture recognition on python and opencv," in IOP Conference Series: Materials Science and Engineering, vol. 1045, no. 1, p. 012043, IOP Publishing, Feb. 2021.
[21] H. C. I. Interface Using, J. T. George, and M. J. George, "Human-Computer Interaction in Game Development with Python.
[22] A. K. Singh, V. A. Kumbhare, and K. Arthi, "Real-time human pose detection and recognition using mediapipe," in Proc. International Conference on Soft Computing and Signal Processing, Singapore: Springer Nature Singapore, June 2021, pp. 145-154.
[23] F. Delogu, F. De Bartolomeo, S. Solinas, C. Meloni, B. Mercante, P. Enrico, ... and A. Zizi, "The Morra Game: Developing an Automatic Gesture Recognition System to Interface Human and Artificial Players," in Proc. International Conference on Image Analysis and Processing, Cham: Springer International Publishing, May 2022, pp. 243-253.
[24] G. H. Samaan, A. R. Wadie, A. K. Attia, A. M. Asaad, A. E. Kamel, S. O. Slim, ... and Y. I. Cho, "Mediapipe's landmarks with rnn for dynamic sign language recognition," Electronics, vol. 11, no. 19, p. 3228, 2022.
[25] V. Naik, A. Chebolu, J. Chavan, P. Chaudhari, S. Chugh, and A. Memon, "The evolution of military operations: artificial intelligence to detect hand gestures in defence," International Journal of Computational Intelligence Studies, vol. 11, no. 2, pp. 94-112, 2022.
[26] V. Harini, V. Prahelika, I. Sneka, and P. Adlene Ebenezer, "Hand gesture recognition using OpenCv and Python," in New Trends in Computational Vision and Bio-inspired Computing: Selected works presented at the ICCVBIC 2018, Coimbatore, India, pp. 1711-1719, 2020.
[27] D. Bhavana, K. K. Kumar, M. B. Chandra, P. S. K. Bhargav, D. J. Sanjanaa, and G. M. Gopi, "Hand sign recognition using CNN," International Journal of Performability Engineering, vol. 17, no. 3, p. 314, 2021.
[28] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, ... and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, pp. 1-74, 2021.
[29] V. Naik, A. Memon, A. Chebolu, P. Chaudhari, and S. Chugh, "Recognition of Struck Out Words Using a Deep Learning Approach," in ICT Analysis and Applications: Proceedings of ICT4SD 2022, Singapore: Springer Nature Singapore, 2022, pp. 585-591.
[30] Y. Li, G. Lei, L. Bui, and C. Lei, "A Hidden Markov Model Based Intelligent Platform for Characterizing Behaviors," in 2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS), IEEE, Nov. 2022, pp. 1-7.
[31] P. Achenbach, P. N. Müller, T. A. Wach, T. Tregel and S. Göbel, "Rock beats Scissor: SVM based gesture recognition with data gloves," *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Kassel, Germany, 2021, pp. 617-622, doi: 10.1109/PerComWorkshops51409.2021.9430962.
[32] J. Ghorpade-Aher, A. Memon, S. Chugh, A. Chebolu, P. Chaudhari, and J. Chavan, "DASS-21 Based Psychometric Prediction Using Advanced Machine Learning Techniques," Journal of Advances in Information Technology, vol. 14, no. 3, 2023.
[33] N. M. Rezk, M. Purnaprajna, T. Nordström, and Z. Ul-Abdin, "Recurrent neural networks: An embedded computing perspective," IEEE Access, vol. 8, pp. 57967-57996, 2020.
[34] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," Procedia CIRP, vol. 99, pp. 650-655, 2021.
[35] L. Guo, Z. Lu, and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," IEEE Transactions on Human-Machine Systems, vol. 51, no. 4, pp. 300-309, 2021.
[36] V. Van Wymelbeke-Delannoy, C. Juhel, H. Bole, A. K. Sow, C. Guyot, F. Belbaghdadi, ... and M. Paindavoine, "A cross-sectional reproducibility study of a standard camera sensor using artificial intelligence to assess food items: the Foodintech project," Nutrients, vol. 14, no. 1, p. 221, 2022.