Super Mario Game using Shoulders Position

Abstract:

There has been significant research on how Video Games cause health issues like Obesity etc. There has been negative opinions in the society about video games, especially its effects on the children on how they are making them lazy. But, several research has been done on the benefits of video games, it proved that video games can improve attention control, spatial reasoning etc. But several health risks associated with sitting and playing video games outperforms those benefits. So, in this project I propose a video game that can be played using body movements instead of sitting and controlling actions through a keyboard. Super Mario is a well-known and well-loved video arcade game that is played all over the world. This video game can be played with just three simple actions (move left, move right and jump). In this project, I propose using the shoulders position to play super Mario. We can play the game with just simple movements without using a keyboard or a mouse.

Keywords: Minimum boundary rectangle, Super mario, OpenCV, MediaPipe.

Introduction And Motivation:

Many researches have proved that excessive sitting behavior is a risk factor for many adverse health outcomes. You consume less energy when you sit than when you stand or move. Long periods of sitting have been related to a variety of health issues, according to research. Obesity is one of them, as is metabolic syndrome, a group of disorders that includes high blood pressure, high blood sugar, extra body fat around the waist, and abnormal cholesterol levels. Too much sitting in general, as well as prolonged sitting, appears to raise the risk of death from cardiovascular disease and cancer.

There is some evidence that gaming has some cognitive benefits, such as greater attention control and spatial reasoning, though it is unclear how far these benefits extend outside of the video game environment into the real world. Finally, video games have medicinal uses, such as teaching patients with degenerative disorders how to improve their balance, assisting adolescents with ADHD in improving their thinking skills, and instructing surgeons on how to perform technically difficult surgeries. One common example is carpal tunnel syndrome, which affects many gamers. Carpal tunnel syndrome, which is common in office workers, is caused by inflammation of a nerve in the wrist, resulting in discomfort and numbness.

When the tendons that move the thumb become inflamed, it causes "Gamer's thumb," also known as "PlayStation thumb" De Quervain's tenosynovitis is the medical word for this condition, which can cause swelling and limited movement. Gamers are also at risk for trigger finger, also known as stenosing tenosynovitis, which occurs when a finger becomes trapped in a bent posture as a result of chronic inflammation.

The character's movement is constrained to a limited number of predefined users actions, where the user mainly uses thumb to play. Excessive sitting in front of video games without body movement has caused many issues in many ways. In this project I propose a video game which can be played with the help of body's posture rather than only playing with thumbs which is mainly causing inflammation. While many game actions were usually produced by some combinations of mashing buttons and pushing joysticks, these quick hand moments were only a small percentage of action that would occur during an afternoon video game. It would be more meaningful if somehow, our physical actions would help the jagged hers run and jump and shoot their way through the virtual worlds they are inhabited in.

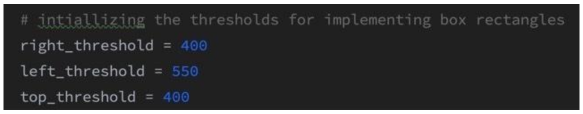
In this project, I propose a super mario game using shoulder movement. Nintendo's Super Mario platform game series is based on and starring the fictional plumber Mario. Alternatively called the Super Mario Bros. Even though super mario is a very simple game, it is highly addictive. In this project we can play super mario games using simple movements of shoulders.

Approach:

This project has __ major parts. First is where the user can play the game by using simple shoulder movements without any keyboard and mouse. To detect the pose movement I have used a media pipe package. Second is using a media pipe to detect the shoulders position, using minimum boundary rectangle algorithm and predefined thresholds to judge which action to take. Third part is using the pynput package to access the keyboard and press the keys.

Using a Media pipe I have extracted the shoulders movement. I have considered that the camera feed is working and is detecting the user. Recognizing its user is present in the game frame. Considering whether a game performs or not and whether a motion is recognized or not.

To detect the action I have used a minimum boundary rectangle, where I have used a threshold for the boundary. In figure 1, I have defined the right, left and top threshold. If the user moves left to left threshold, the system will perform the left action. If the user moves right to the right threshold, the system will perform the right action. If the user jumps beyond jump threshold, the system will perform the jump action.



```
# intiallizing the thresholds for implementing box rectangles
right_threshold = 400
left_threshold = 550
top_threshold = 400
```

Figure 1: Thresholds for box rectangles

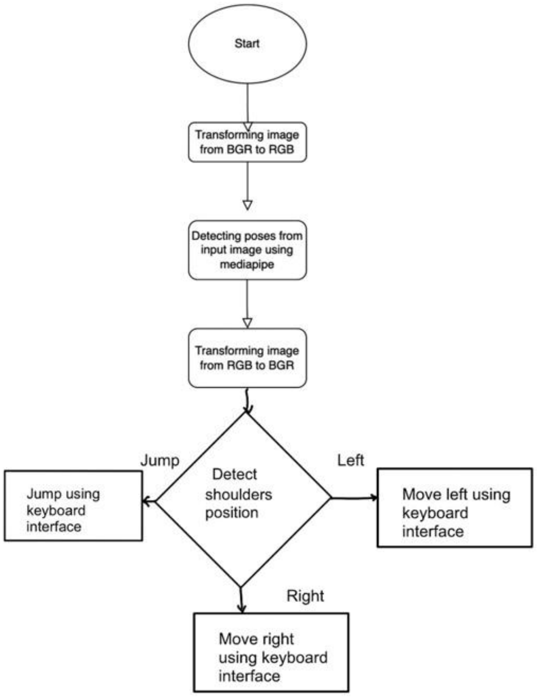

Start
Transforming image from BGR to RGB
Detecting poses from input image using mediapipe
Transforming image from RGB to BGR
Detect shoulders position
Jump
Left
Right
Jump using keyboard interface
Move left using keyboard interface
Move right using keyboard interface

Figure 2: Flow chart of the main Approach

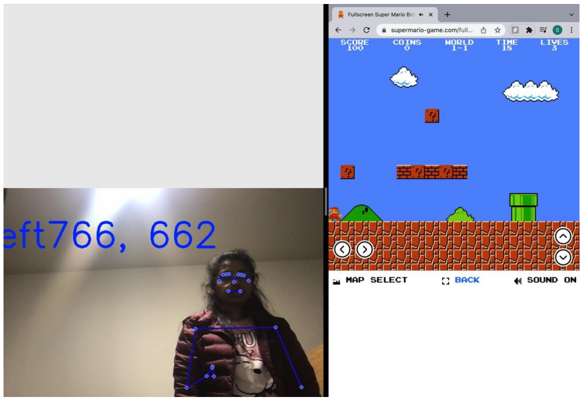


Figure 3: Working example

Figure 3, is the working example of a super mario game which portrays the mario moving to left when the user moves to the left.

Technologies used:

Anaconda:

Anaconda is the most popular python distribution platform, it is an open source platform, and it's a platform of choice for many engineers. Anaconda has many packages, called conda packages. It has a cloud-based repository of over 7,500 data science and machine learning packages. Conda also has environments, using which we can create a stream-lined software, it has utilities to build, distribute, install, update and manage software.

In this project, Conda is used to collect, install and manage several packages that have been used in this project.

Pose estimation:

Detecting the user's posture is critical for this project. I had to determine whether the user's body posture is moving left, right, or jumping. With the enormous differences owing to clothes, muscle structure, body type, illumination, and many other circumstances, determining the degree of freedom configuration of the user's body proved difficult. To handle this I have used the Mediapipe package. MediaPipe is a cross-platform open-source Machine Learning framework for creating complicated and multimodal applied machine learning pipelines. The mediapipe can be used in machine learning models like face detection, multi-hand tracking, object detection, and tracking, and many more.

The pose detector model also known as blazepose(full body) detector is inspired by Leonardo's Vitruvian man[5]. The radius of a circle circumscribing the entire person and the incline angle of the line connecting the shoulder and hip midpoints are used to predict the midpoint of a person's hips.

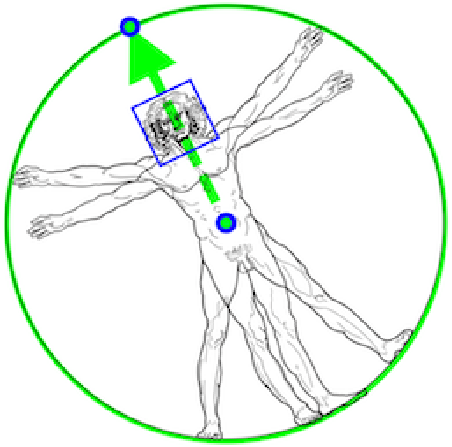


Figure 4 : Vitruvian man aligned using two virtual key points predicted by the BlazePose detector in addition to the facial bounding box.

BlazePose: BlazePose consists of two machine learning models: a Detector and an Estimator. The Detector extracts the human region from the input image, whereas the Estimator receives a 256x256 resolution image of the detected person as argument and returns the key points. The blaze pose returns 33 key points according the following order in figure 3.

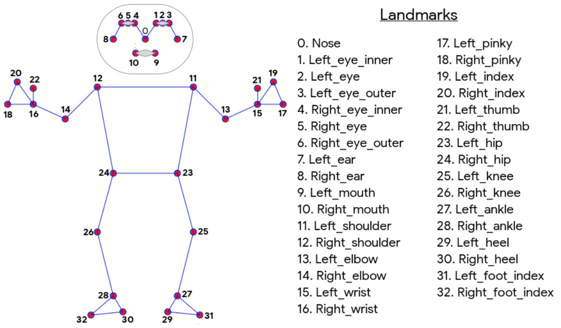


Figure 5: Sample image showing landmarks

Open CV:

Open CV stands for Open Source Computer Vision Library. It is an open-source library that contains various computer vision algorithms. There are multiple modules available in OpenCV library. OpenCV contains a modular structure, and it has shared as well as static libraries.

The below are the modules from OpenCV:

Core functionality: It has data structures including a dense multi-dimensional array.

Image Processing: This module has both linear and non-linear image filtering and transformation, color conversion, graphs etc.

Video Analysis: This module has object tracking algorithms, motion estimation etc.

Camera Calibration: It has stereo and single camera calibration, estimating of an object's pose etc.

2D Features Framework - It has functions for detecting features, descriptors etc.

Object detection - This module has functions for detecting objects and instances of predefined classes like faces, cars, people etc.

Features of OpenCV:

1. OpenCV has automatic memory handling capabilities. All the data structures can deallocate memory buffers when needed using destructors.
2. OpenCV also has automatic allocation of memory for parameters of the output function.
3. OpenCV has a Saturation Arithmetics feature, which means when dealing with 16-bit image pixels, the values may fall out of range. But, OpenCV handles this internally to prevent it from happening.

In this project, I used the video analysis module of OpenCV. I used the Videocapture method to capture the live webcam feed. I also used the Image Processing module's cvtColor method to recolor the input_image from RGB to BGR. Apart from this I used OpenCV to write action on the image for decoding purposes.

Pynput

This library contains functions and methods that can be used to control and monitor input devices. Pynput contains subpackages for various types of input devices, for example pynput.mouse contains classes that are used to monitor or control a trackpad. Pynput.keyboard has classes that are used to control the keyboard.

I used Pynput to control keyboard actions in this project. I instantiated a keyboard listener. It is a threading.Thread, and this thread invokes all the call backs.

Keyboard Listener thread:

Listener callbacks are called from an operating thread directly, especially in platforms like Windows. There is a possibility of blockage of input for all operations, this can occur due to long running threads or blockage operations. This can be handled by using a queue, which will make the operating system handle the messages by using separate threads.

Toggle Event Listening for the keyboard listener:

Main challenge of pynput in this project is that Once pynput.keyboard.Listener.stop has been called, the listener cannot be restarted, since listeners are instances of threading.Thread. So there is a need to add an internal flag to ignore events when not required.

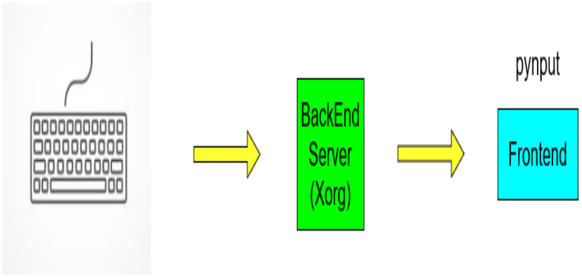


Figure 6: Keylogger - Pipeline

Media Pipe

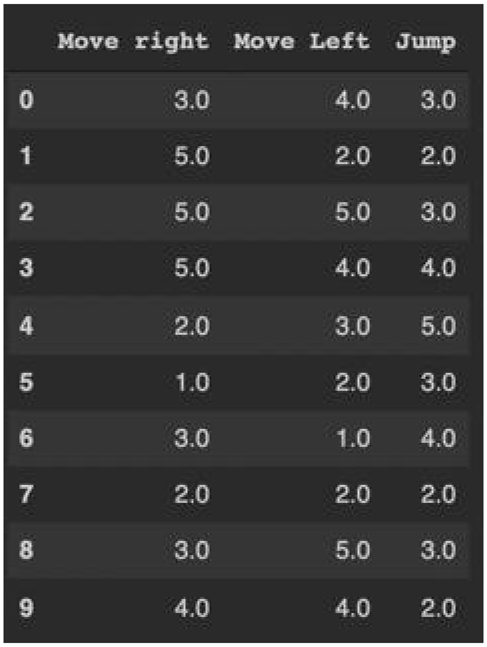
Media Pipe offers end to end acceleration for fast ML inference solutions. It uses cutting-edge technologies to develop ready-to-use ML solutions.

ML pipeline used in this project:

The media pipe library utilizes a two-step detector-tracker ML pipeline; it has already proved to be effective in MediaPipe Hands and MediaPipe Face Mesh solutions. This pipeline first detects a region of interest within a given frame. This can be either a person's face or nose, in this project it is used to detect shoulder's position. This tracker predicts the pose landmarks and segmentation mask within the region of interest. In the case of videos, a tracker is only called when needed. For example, when there is no region of interest in the video frame, the tracker no longer needs to call the tracker to identify pose position. This feature improved the efficiency and reaction time in the project.

Evaluation:

To evaluate the super mario game, I have carried out within subject evaluation, where I have asked 10 users. To avoid counter-balance between the user's review I asked users to play the game 3 times randomly when they wanted to. I have taken the average on users rating and got the below table. In the below table, we have an average rating given by each user for specific action. With the help of table from figure 6, I have done quantitative analysis.



	Move right	Move Left	Jump
0	3.0	4.0	3.0
1	5.0	2.0	2.0
2	5.0	5.0	3.0
3	5.0	4.0	4.0
4	2.0	3.0	5.0
5	1.0	2.0	3.0
6	3.0	1.0	4.0
7	2.0	2.0	2.0
8	3.0	5.0	3.0
9	4.0	4.0	2.0

Figure 7: 10 users rating for each action

From the figure 7, which gives the mean ratings for each action. By looking at the plots we can say that the mean for each action is almost the same and there isn't much difference in the mean ratings. So we can say that on average these actions performed almost the same and is around 3.0.

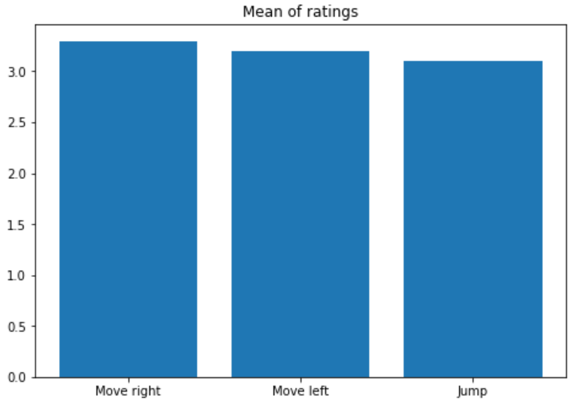


Figure 8: Mean ratings for each action

From figure 8, which gives the median of each action performed by the user. From the below graph we can say that the median of the action move left is more when compared to other two actions.

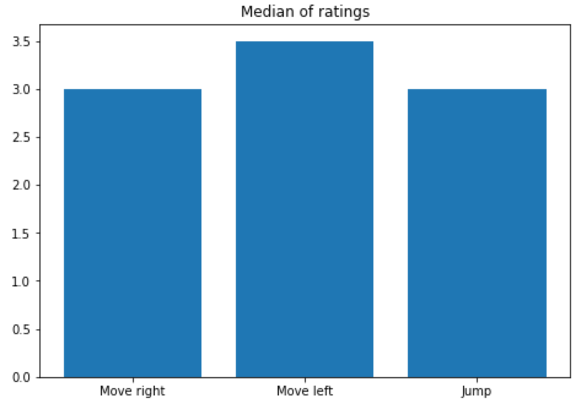


Figure 9: Median ratings for each action

From the figure 9, which gives the mode of each action performed by the user. From the figure we can say that move left action is high mode when compared to other actions. But from medians we have seen that the median of move left is more.

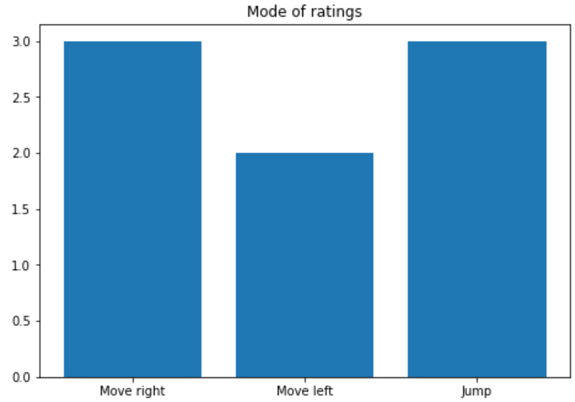


Figure 10: Mode ratings for each ratings

On taking the dispersion for the table, I got the following results. For the move right action it is seen that there is large variance, and mean is 3.0. So we can say that for this specific action, the average rating given is 3.0. It is also seen that few users gave 5.0 rating, based on this we can say that few users like this feature. From the dispersion of move left action we can say that this action got better ratings when compared to move right action.

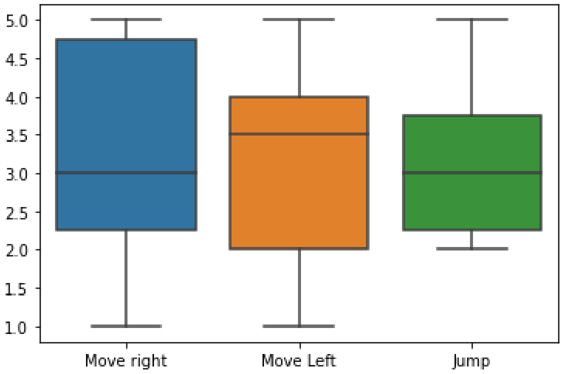


Figure 11: Median ratings for each action

From the figure 11, the jump action has less spread. It means that users were happy with this feature but the mean is almost the same as move right action. Mean of move left action is high when compared to other two actions, we can say that move left action on average has done better when compared to other actions.

Future Work and Conclusion

As discussed above, this project tries to use libraries like openCV to make the video games advantageous by eliminating the health risks of no physical activity while video gaming. I also performed qualitative analysis and presented the results in the above sections. In the future, I wish to implement this technology to create powerful use cases for controlling more complex video games. In that process, I also would like to explore more advanced libraries.

References:

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4831859/

2. https://venturebeat.com/2019/11/04/your-posture-is-killing-you-what-gamers-should-know-about-ergonomics/
3. https://google.github.io/mediapipe/solutions/pose
4. https://www.geeksforgeeks.org/python-facial-and-hand-recognition-using-mediapipe-holistic/#:~:text=MediaPipe%20is%20an%20open%2Dsource,and%20tracking%2C%20and%20many%20more.
5. https://en.wikipedia.org/wiki/Vitruvian_Man