# Rock Paper Scissors Game

The Rules

Rock, Paper, Scissors is an iconic game which (almost) everyone knows how to play. After a countdown, two players make a hand gesture of either "Rock", "Paper" or "Scissors". If you both pick the same gesture, it's a draw. Otherwise:

- *Rock* beats *Scissors* (✊ -> ✌️)
- *Scissors* beats *Paper* (✌️ -> 🖐️)
- *Paper* beats *Rock* (🖐️ -> ✊)

My Game

My website uses AI models and computer vision to allow for playing of the game. The game uses your webcam to track you hand motion. It uses Mediapipe Hands, an AI model that tracks the motion of your hand to locate your hand on screen. The geographic hand data is passed to a TensorFlow convolutional neural network that I trained to classify the hand shape. It's important to note that if more than one hand is present on screen, only one hand's data will be used. So please, only have one hand in frame. You may wish enable hand tracking, which visualises the hand that is being tracked.
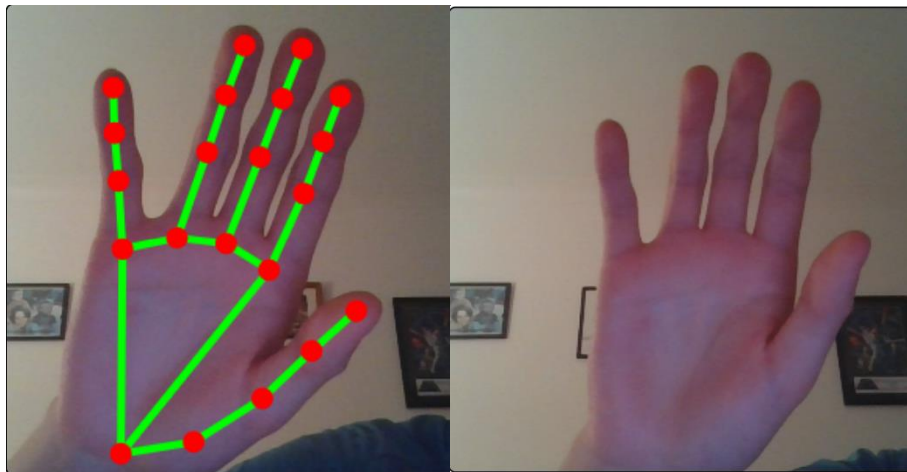
| Hand tracking enabled | Hand tracking disabled |

You can choose to play against a computer (who simply makes random moves), or against a friend (or your other hand!) as long as they are in frame of the webcam. Whenever, you make a move, a bar chart is displayed, which shows my AI model's confidence that your hand is the specified pose. I think the rest of the game is self-explanatory so please enjoy!

Training the Model

I gathered images of people playing rock paper scissors from a wide range of sources and combined them with images I had taken of myself playing the game to produce a dataset of 9,394 image. I passed these images through the Mediapipe Hands model to convert the images into geographical data. I then applied various data augmentation techniques to increase the size and diversity of the dataset. This led to ~60,000 datapoints for the model to train with. I trained a convolution neural network on this dataset which achieved an accuracy of 98.1% on the augmented data and 99.5% on non-augmented data. This model is used by the website using tensorflow.js.

If you wish to train your own model, you may download the python code I used. Simply place any images you wish to use in the folder titled "Raw Images". Then run the "main.py" script (after having installed the relevant dependencies). The newly trained model will be automatically transferred to the website folder as required.

As with the training code, the code for the website is also open source. To run the website code simply run "npm start" (after running "npm install" to get all dependences). The website was created using react.

Please find the code to both the website and python training scripts here: https://github.com/philipmortimer/Rock-Paper-Scissors

Example Inputs

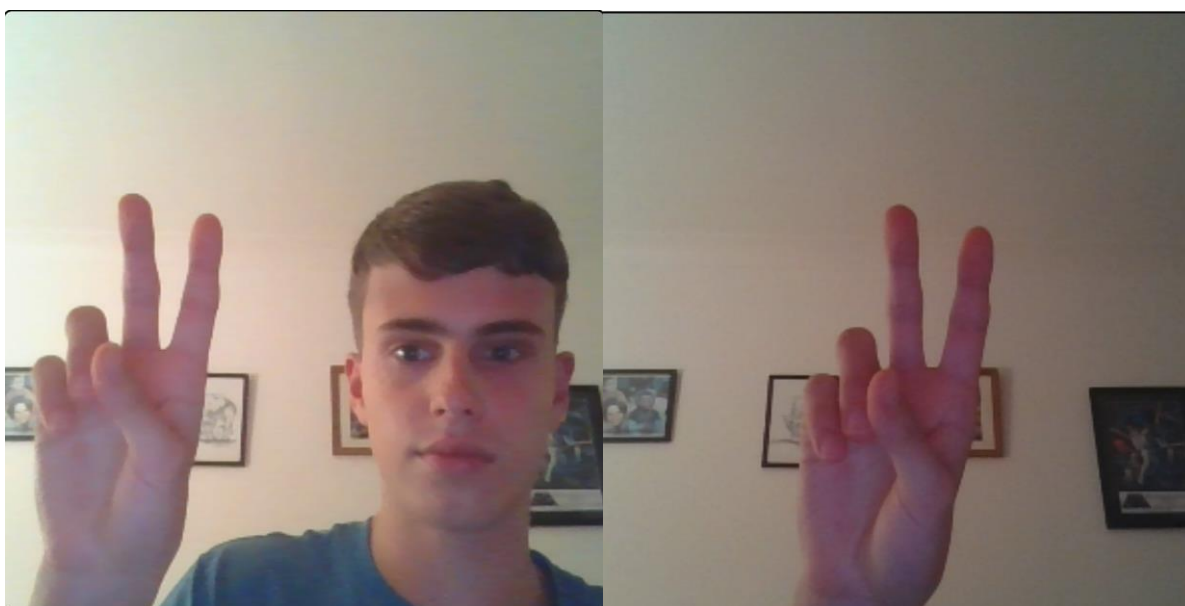Here are some example images which work very well for the website:

Rock:

Paper:

Scissors:

Credit

I made use of datasets made by Laurence Moroney [CC 2.0], Julien de la Bruère-Terreault [CC-BY-SA 4.0] and Roman Glushko [CC-BY-SA 3.0]. Additionally, the following projects inspired me to produce my own website: https://www.romaglushko.com/lab/rock-paper-scissors/ , https://trekhleb.dev/machine-learning-experiments/#/experiments/RockPaperScissorsMobilenetV2 and https://medium.com/geekculture/rock-paper-scissors-with-hand-gesture-recognition-841297a7d915 . My website supports similar functionality to these sites but builds on them in many ways; My website has a very high classification accuracy, is far less susceptible to bias and supports two human players. My work here is licensed as CC-BY-SA 4.0. I hope you have fun with my website and find my project insightful!! :)

Me

This work was all produced by me (Philip Mortimer). Please check out my GitHub: https://github.com/PhilipMortimer