



Inha University in Tashkent

School of Computers and Information Engineering

Spring semester 2020

Multimedia Computing SOC4020

Ping-Pong with control by hand gesture recognition and motion tracking

Akmal Karimov

U1610016

CIE 16-6

Abstract

Video games have become global industry. Nowadays this industry is developing in very high speed. It is supposed that video games will never leave the leading places of economic sector. Its development started decades ago. Now there are a lot of types of this industry. Today the latest console games provide a realistic gaming experience with interactive features. This means that gaming process is related to webcam or console camera. This project is Ping-Pong game which is connected to a laptop webcam. The game is written in Python using libraries such as Pygame, OpenCV, TensorFlow and NumPy. In this game a player can control the racket by showing his palm and moving it up and down.

Introduction

The current century has experienced a great revolution in ICT. Game development has revealed as one of the really trendy forms of entertainment. Video games are evolved in enormous scale. There are a lot of genres and fields in game industry. Games with interaction type through a webcam, or a console camera, is a type of Augmented Reality (AR). Such games are the integration of game visual and audio content with the user's environment in real time. So, the video game market has embraced augmented reality video games, which is going on developing as new technologies develop.

On the scale of this project AR technologies were taken as a base idea. In this project simple game - Ping-Pong was chosen to integrate it with AR. For this purpose, four main libraries were used:

- Pygame for game code;

- OpenCV for video capturing and other multimedia processing;
- TensorFlow for work with trained models to recognize palm gesture;
- NumPy for complex mathematic functions and arrays.

Pygame is a set of Python modules designed for writing video games. It allows you to create fully featured games and multimedia programs in the python language. [\[1\]](#)

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception. [\[2\]](#)

TensorFlow is a free and open-source software library for dataflow and differentiable program ming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. [\[3\]](#)

NumPy is the fundamental package for scientific computing with Python. It is used for a powerful N-dimensional array object, sophisticated (broadcasting) functions, useful linear algebra, Fourier transform, and random number capabilities and so on. [\[4\]](#)

The purpose of the project was to make a simple game in Python and modify its gameplay. It had to be merged with code written with the help of other libraries mentioned above. The next paragraphs describe the method used to come to the solution and results in the end.

Method

The first step to do was to create the game itself. It was done with the help of Pygame. The following steps describe how the game was created in this project.

Settings:

- import Pygame and initialize all its modules;
- create an object to help track time;
- set up the main window by giving size (width, height) and setting its name;
- set up game rectangles (figures) by giving their sizes and position;
- set up colors for figures and background;
- define and initialize main variables like ball speed on 2 axis, player and opponent speed (rackets), scores;
- set up font for game text.

While Loop:

- handle inputs (quit, key up, key down);
- make ball animation by adding coordinates per frame;
- handle ball collisions and score counting if ball goes out;
- each time score is counted get timer value to set up interval before start of game;
- make player's and opponent's racket animation by adding coordinate of y axis, because the movement of the racket is possible only up and down;
- make logic for opponent's racket by moving it up if ball's y-coordinate is greater than opponent's top point, and down if ball's y-coordinate is less than opponent's bottom point;
- draw background and all rectangles;

- call function starting the ball (in the beginning and every time ball is out);
- display score text;
- update the contents of the display and set number of frames per second.

The second step was the main idea that had to be implemented – adding feature of controlling the racket by hand gesture (palm). The steps below describe this point:

- import OpenCV (cv2), NumPy, TensorFlow (main libs);
- get video capture object by setting screen size;
- read capture and frame, and convert frame to RGB;
- get coordinates and scores of a hand through detect_objects function in utils.py file. This function uses frame, loaded frozen inference graph and its tensorflow session. TensorFlow allows to save trained models. So there is a frozen_inference_graph.pb in label_map folder. In addition, 2 files that are used for label mapping are from official library of TensorFlow;
- draw a rectangle around a palm after detection and receiving the coordinates, and define center of the y-axis of the palm;
- integrate game code with object detection code by adding logic to motion of a player racket with center of a detected palm (y-axis), and remove key handling of up and down buttons;
- pause the game if no hand (palm) is detected.

This was description of logic and algorithm of this project. Source code and necessary files are provided with comments.

Experiment result

The result was as expected. A player runs the code and game display, and camera display are visible. On game display the player can control racket on the right side by hand (palm gesture). The camera display uses web camera and the detected palm is surrounded by green rectangle, so player can see whether he is in the game.

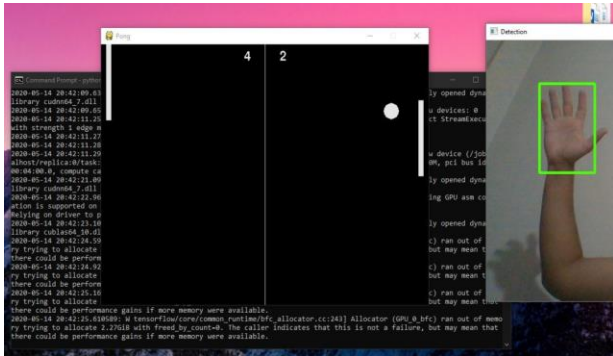


Figure 1.

As you can see, this is an example of running the project code. There is a game display in the center (it has scores on the top). On the right there is a camera display where the palm is detected and surrounded with rectangle.

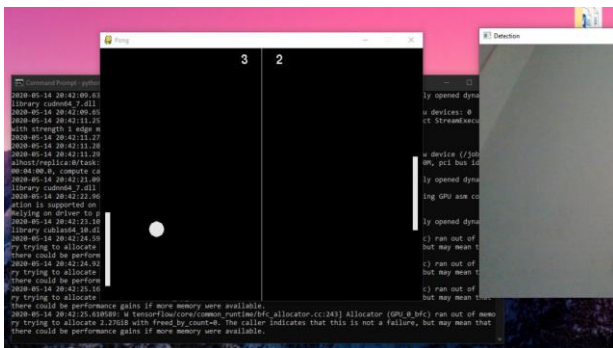


Figure 2.

Here is a case when there is no hand detection. The game is paused.

References

https://github.com/floydhub/object-detection-template/tree/master/models/ssdlite_mobilenet_v2_coco_2018_05_09

https://github.com/tensorflow/models/blob/master/research/object_detection/utils/label_map_util.py

https://github.com/tensorflow/models/blob/master/research/object_detection/protos/string_int_label_map.proto

<https://www.youtube.com/watch?v=Qf3-aDXG8q4>

<https://www.youtube.com/watch?v=E4lh9mpn5tk>

<https://towardsdatascience.com/object-detection-with-less-than-10-lines-of-code-using-python-2d28eebc5b11>

<https://leimao.github.io/blog/Save-Load-Inference-From-TF-Frozen-Graph/>

<https://blog.metaflow.fr/tensorflow-how-to-freeze-a-model-and-serve-it-with-a-python-api-d4f3596b3adc>

<https://www.toptal.com/machine-learning/tensorflow-machine-learning-tutorial>

<https://www.geeksforgeeks.org/opencv-python-tutorial/>

[1] <https://www.pygame.org/wiki/about>

[2] <https://opencv.org/about/>

[3] <https://en.wikipedia.org/wiki/TensorFlow>

[4] <https://numpy.org/>