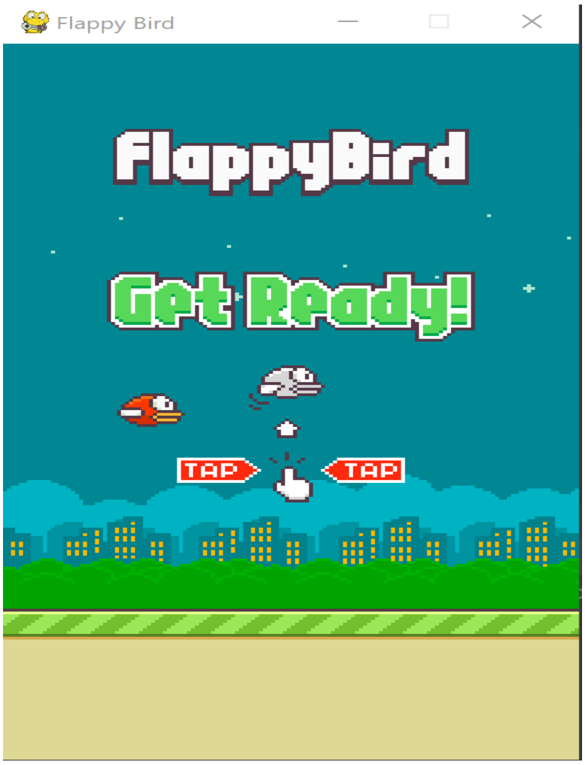
Flappy Birds Game using Mouth Opening or Eye Blinking, as well as using speech

1. Abstract

Flappy Birds is a game where a bird flies when we tap the screen (smartphone) or press space bar on the laptop. The goal is to cross as many pipes as possible without crashing the bird (bird either hits the pipe or the ground). In this paper, I have proposed a novel way of playing this game by keeping in mind the limitations and challenges faced by specially abled people. Specially abled people cannot play this game in such a fashion. So, in this novel idea, they can control the bird by opening their mouth or by blinking their eye instead of using the space bar. They can also control the menu by saying commands like start the game, return to menu and quit the game. In short, I have made a complete hands-free experience so that users don't have to either click the mouse or press a key on keyboard. In short, the resultant goal of my project is to make handicapped people play this game, which I have successfully achieved.

2. Introduction

Flappy Bird is an arcade-style game in which the player controls the bird Faby, which moves persistently to the right. The player is tasked with navigating Faby through pairs of pipes that have equally sized gaps placed at random heights. Faby automatically descends and only ascends when the player taps the touchscreen. Each successful pass through a pair of pipes awards the player one point. Colliding with a pipe or the ground ends the gameplay. During the game over screen, the player is awarded a bronze medal if they reached ten or more points, a silver medal from twenty points, a gold medal from thirty points, and a platinum medal from forty points [1]. So, instead of playing this game on a smartphone, we shall be playing this game on a computer. We have changed the original task of this game to adapt it to computer program by flapping the bird when space bar is pressed on the keyboard. The rest of UI and sounds and everything remains the same as it was there in devices having a touch-screen application. Fortunately, I did not have to develop the entire game from scratch as I found this game readily available on the web [2]. How great it would be if specially abled people could also play this game without having to hit spacebar on the keyboard? However, since I needed to adapt this game for handicapped people, I thought of using Computer Vision and Speech Recognition to give the users a completely hands-free experience of playing the game. We can also make the experience completely with speech but the obstacle in that was it takes some time to convert speech to text, as it is not done natively on the machine, but we use cloud services for the same. To minimize the latency and response time, we shall be controlling the bird flap using Computer Vision. The primary work used in



completing the project will be done by using Computer Vision. I have used the OpenCV library in python to give the users a hands-free experience of playing the game.

The tasks are broken down as:

1. Detect the person's face.
2. Detect the opening of mouth.
3. Detect the blinking of eyes.
4. Minimize the response time between action performed and bird flew up.

The above-mentioned tasks happen almost instantly by using cascades provided by OpenCV which are very fast. This is a novel approach as I am giving options to the users to either make the bird flap by opening mouth or blink eyes. Incorporating OpenCV in a game is an interesting idea.

3. Motivation

The huge companies investing in designing the game and having various ideas on how cool or different the game should be often forget to think about handicapped people. Only two in five (42%) gamers think that developers do enough to accommodate gamers with mental and physical disabilities according to a new study [3]. However, 30% of US gamers identify as disabled, as do 20% of gamers in the United Kingdom, per research firm Newzoo [4]. So, disabled people take up a large proportion of gamer population. And game developers often forget to keep their requirements in mind. As a result, I have extended the game functionality for the specially abled people so that they can also enjoy playing the game. In this paper, I have also myself played the game for 5-10 times and I could notice that I can play the game as good by blinking my eyes or opening my mouth as I would be playing by hitting the spacebar, so the program is quite efficient in detecting if the bird needs to flap or not.

4 Technologies Used

1. Python 3.10

This is the programming language used to code the project. I have used PyCharm IDE and organized all the files in a single folder. We have assets folder, with audio, sprites, and cascades as sub-folder. Audio and Sprites are from flappy birds game [1] and cascades folder contain .xml files for haarcascades [5]. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. [6]. Thus, I have used the following types of cascades.

A. haarcascade_eye_tree_eyeglasses.xml (since I have glasses and need to detect my eyes with glasses),

B. haarcascade_frontalface_alt.xml (to detect my face)

C. haarcascade_mcs_mouth.xml (to detect if mouth is opened or not)

2. OpenCV

This is a very famous library aimed at real time computer vision. We use the web-cam and perform various insights and analysis on the video feed. The cascades used for face, eye, and mouth detection are supported by OpenCV.

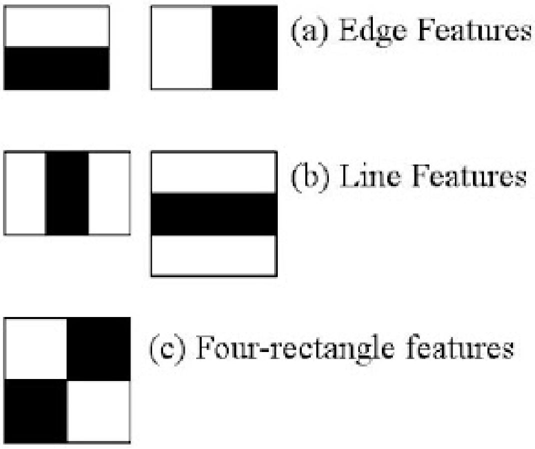
3. Pygame

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language. [7]. So, here in this project, most of the pygame functionality was already implemented from the github link, and I have just changed the logic for flapping the bird that instead of pressing the spacebar, I will be opening my mouth or blinking my eyes.

4. Speech Recognition

This is a famous library to convert speech to text. I have used this library to interpret my voice commands by using Google Cloud Speech API. This can be used to control the menu of our game.

Object Detection Using Haar Cascades in OpenCV:



We shall use haarcascades to detect the important relevant features of our project, that is, face, eyes, and mouth. For these popular tasks, they already exist on the net. [5]

1. Face Detection:
First, we will detect the face of the person playing the game. This is used for other subsequent tasks. Now we have two options. We can either make bird fly on eye blink or mouth opening.

2A. Eyes Detection:
We will detect both the eyes and if both are blinked, we shall make the bird fly.

2B. Mouth Detection:
We will detect the mouth and if it is opened, we shall make the bird fly.

How to play the game:

1. Press Space key or say “start” to start the flappy birds game.
2. Face your computer's camera and blink your eyes or open your mouth to make the bird fly. Keep playing till the bird either hits the ground or the pipe. The score updates dynamically after each pipe is successfully crossed. Once we hit the ground or a pipe, the game is over.
3. Now the user has two options, he can either say “return” to return back to main menu screen, or he can say “quit” to quit the game. Pressing space key returns to main menu.
4. Press ESC key or say “quit” from main menu (Fig. 1) or when game is over to exit the game.

Procedure to run the game:

1. Install the necessary dependent libraries like cv2, speech_recognition and pygame. A simple pip install command should do the needful.
2. Run python run_game.py with 2 command line parameters. The first one will be used to specify if the bird should flap using mouth opening or eye blinking. By default, if nothing is passed, it will do mouth opening. The second one will be used to specify if we should enable speech recognition or not. Enabling this makes our system a bit slower so by default, it is kept disabled.
3. Press ESC key to exit the game

Code Analysis:

1. flappy.py: I have downloaded a Flappy Bird Clone[2]. The modifications made here are changing spacebar press to eye blink or mouth open. Also, since disabled people will be playing this game, I have added speech recognition where user can say start the game, return to menu and quit the game.

2. run_game.py: This is the starting file of the game. It contains the welcome animation screen and starts the game.

3. detect.py: This is OpenCV code to detect 3 things.

A. Face:

First, we will detect the face of the person playing the game. This is used for other subsequent tasks.

B. Eyes:

This is OpenCV code to detect if eyes have blinked or not. I have loaded the face and eyes cascade and the function here returns True or False based on eye blinking.

C. Mouth:

This is OpenCV code to detect if mouth is opened or not. I have loaded the face and mouth cascade and the function here returns True or False based on mouth opening.

5 Evaluation

After developing the system, I needed people who could test the system and evaluate and tell me how good and efficient it is. As per qualitative analysis I performed in evaluation stage, I assumed I am the expert and looked for problems. Thus the code will not fail or throw an exception if face is not detected, but the bird will continuously flap, making it lose the game. In quantitative analysis, played the game several times. I noticed that my eye detection is perfect, but my mouth detection is not. That means when I open my mouth, it is not getting detected in the code. Likewise, if my mouth is closed, still it sometimes gets detected as mouth is open. I have told my 5 friends to play the game 10 times each.

Player	Blink	Open mouth	Accuracy
1	9	7	80%
2	10	8	90%
3	10	6	80%
4	10	5	75%
5	9	7	80%
Average:	96%	66%	81%

Thus, the total accuracy is 81%. It means, on an average, the studies performed over both the commands over all 5 users is 81%.

We can perform a **t-test** on hypothesis that blinking eyes to flap bird is significantly better than opening mouth. That is, we can statistically prove with a certain confidence level that accuracy of 96% is better than 66%. This is a **Within-Subjects** Design as each user plays game on both the designs, with **Independent Variables** as blinking eye and opening mouth in each design case, and **Dependent Variable** as accuracy in both.

6 Future Work

1. Instead of passing True/False in command line arguments for both the parameters, vis, controlling bird via mouth/eye blink and enabling speech recognition or not, we can integrate it with the Operating System by telling a command to automatically execute the python file, and use Natural Language Processing so that users can say “Play Flappy Birds using Eye Blinking and control the Menu using Speech” instead of just interpreting True/False.
2. We can incorporate flapping of bird using only speech. However, when I tried using just that, it was giving a response after a few hundred milliseconds. This is too much, and we need to process the flap immediately else the bird will hit the pipe or the ground. In future, if the speech recognition becomes faster, we can control the bird almost immediately by using speech recognizer.
3. Since mouth opening detection is not as efficient as reported by several users, we can do something to improve the haarcascade_mcs_mouth.xml. However, there is a better approach to do object detection and that is via using dlib library. The dlib can be used to extract the mouth region via facial landmarks. [8]. This is a more modern and better way of object detection using computer vision using dlib but it is slow, heavy-weight and takes a long time to install on the system. So, we can extend the project by coding the logic in dlib as well, so that users with a powerful CPU can work more efficiently with our software.
4. If face is not detected, we can notify the user via a sound that he needs to clearly show his face in the webcam. Additional python libraries might be needed for that.

References

[1] Wikipedia: https://en.wikipedia.org/wiki/Flappy_Bird

[2] https://github.com/sourabhv/FlapPyBird

[3] Why Games Are Still Letting Down People With Disabilities (forbes.com) https://www.forbes.com/sites/mattgardner1/2021/1

2/03/why-games-are-still-letting-down-people-with-disabilities/?sh=5982e52a7be0

[4] Gamers With Disabilities On Issues In Accessible Gaming https://www.refinery29.com/en-us/2021/12/10711964/gamers-with-disabilities-accessible-video-games

[5] haarcascades at master · opencv/opencv https://github.com/opencv/opencv/tree/master/data/haarcascades and https://github.com/AlexeyAB/OpenCV-detection-models/tree/master/haarcascades

[6] OpenCV: Cascade Classifier https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[7] Pygame - Wikipedia https://en.wikipedia.org/wiki/Pygame

[8] Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python - PyImageSearch https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/