**Concordia University**

**FitnessRun: Real-Time Squat Detector Game for Fitness Using Computer Vision**

**Adam Atamnia and Ismail Shafique**

**Professor Yaser Esmaeili Salehani**

**April 1, 2025**

**Abstract:**

**Introduction:**  
As is well-known across many countries around the world, the rise of obesity has significantly increased over the past couple of decades. More specifically, recent data suggests that 26.6% of Canadians struggle with being overweight, and this data is consistent within both rural and urban communities alike (Public Health Agency of Canada, 2020).

A screenshot of a graph

AI-generated content may be incorrect.

Another study presents how over the years, there has also been a large percentage of adults being classified as physically inactive. The World Health Organization conducted a study where they found approximately

These statistics are extremely concerning and thus, the goal of this project is to build a computer-game application (using various concepts in Computer Vision). The main purpose of this computer game would be for the user to lose as many calories as possible (encouraging exercise while at the same time being used for weight-loss to combat obesity). The theme will be based on an endless-runner type game, while the controls will be performing exercise movements (such as performing a 150 degree box-squat which constitutes to a “jump” when going over obstacles). The goal of building this into an endless-runner application is to encourage users to beat personal records (in terms of their fitness objectives) and have a great time playing a highly interactive game! Many people struggle with exercising (as it’s physically demanding on the body) while also finding the time where they can relax themselves while playing video games. Thus, our main selling point towards this application is being able to burn calories while at the same time keeping it in a game-type theme.

To briefly go through what the objective of the game would look like

**Methods:**

In this section we will explain the exact steps that we took to accomplish the Computer Vision endless-runner game that we built. Note that this section will be divided into phases (each phase being a step that we took to reach our objective). Also take into consideration that the phases section will often cross reference Experiments/Results and are both highly connected.

Phase 1: Deciding the Primary Exercise for Fitness-Run

Knowing before-hand that we wanted to build an endless-runner game that incorporated some sort of physical exercise as the main control system (mainly the jump mechanism over obstacles), this brought up a valuable question: “which exercise to use?” Upon asking this question we decided to narrow down our exercises to three different possibilities and state the pros/cons. Thus, being able to find the right physical movement for the endless-runner game.

Option 1: Standard Push-Up

In terms of the pros upon first glance, a standard Push-Up was ideal to be implemented in our game since it was a very standard and well-known exercise. The greater reason which inclined us to possibly use Push-Ups was due to its ease of being implemented with Computer Vision. More specifically, we imagined using a model (or training one) which could detect certain landmarks within the human body. Those landmarks would specifically be the wrist, elbow and shoulder respectively. Using that landmark detection in those certain areas, we can then test and manipulate a variety of different experiments. The main one being, finding the angle in between the ankle, wrist and shoulder which we can then use to determine if someone has indeed performed a Push-Up or not (if the angle between these limbs were 90 degrees or under, the program can record that as a successful Push-Up etc.) We can also use these landmarks to detect different variations of Push-Ups (such as diamond Push-Ups, wide-arm Push-Ups etc.) which would increase versatility to our game. Another advantage to using this exercise was the fact that its movement is quite slow, thus being easily detectable for various Computer Vision Landmark detection systems (note that this specific point will be expanded on further when comparing models, refer to the Experiments/Results section). Although there were many positive aspects when it came to the Push-Up, after further examination this exercise had many problems that were initially overlooked. Amongst them was the fact that a Push-Up was more of a strength exercise rather than being cardio-oriented (focus on building muscle rather than fitness-oriented which focuses on burning calories). This is further supported when studying how many calories are burnt, which is approximately 7 calories per minute (Push Ups Calories Burned Calculator, 1) (these numbers are approximate considering the various variables that come into play, this is assuming a person who is 150 pounds). This number is relatively low when it comes to a fat-burning exercise, and thus this was definitely a con when it came to our specific game. It’s also well-known that the average person (considering that a vast majority of the population is inactive in the first place) cannot perform more than 10 Push-Ups consecutively! Thus, the Fitness-Run game would only last 15-20 seconds per run which is much too short for duration. Since the Push-Up is an exercise which is performed in a plank position on the ground, as well as your face and entire frontal body facing the floor, this posed many problems to our game design. Most notably, not being able to see the screen when playing the game (since your eyes are constantly staring downwards). This presented a large problem as the user will not be able to even see the obstacles in the first place, and thus, this exercise was quickly discarded from being used as the jump mechanism in our game.

Option 2: Jumping-Jack

This was another strong consideration for us in terms of a possible exercise for jumping over obstacles in the game. That being said, this exercise too had many advantages which inclined us to possibly use this within our application. The first strength was the fact that Jumping-Jacks are a very beneficial exercise when it comes to cardio and fitness. The average person (even if they’re very inactive) can perform much more Jumping-Jacks (comfortably) than Push-Ups since it focuses more on body balance rather than strength. Jumping-Jacks also burn around 9 calories per minute which is an increase when comparing to how many calories are burnt from Push-Ups in a minute (Calories Burned From Jumping jacks vigorous, 1) (these numbers are approximate considering the various variables that come into play, this is assuming a person who is 150 pounds). Another up-side to Jumping-Jacks was the fact that they are an exercise that can be performed while staring at a computer screen (unlike push-ups where you are constantly staring at the floor), thus, being aware of the actual game-play. Despite these positive aspects we ultimately decided to discard Jumping-Jacks at the end due to a number of reasons which were not compatible with our Computer Vision design and the theme of our game. The largest reason for moving on from this exercise was the fact that performing a Jumping-Jack is a relatively quick-moving action. Not only this, but you have multiple key joints within the body that are moving all at one time in this fast motion. This was a realization that came to us after trying to code the Landmark Detection Model with Computer Vision. We found that fast movements often disrupted the pose detection landmarks (since the model constantly had to recalculate because of the movements being increasingly fast (a more in depth answer will come later in this report). Another obstacle that we faced (which was another reason for us to abandon this exercise from the game) was due to scalability issues. A Jumping-Jack needs the entire body to be visible from the camera-lens, thus involving more key points to be detected by the Computer Vision Model. This was an issue that was very difficult to avoid, and it hindered the performance of our application if we were to use it (in combination to the fact that these points will be moving extremely fast considering how Jumping-Jacks are performed quickly).

Option 3 (picked exercise): Bodyweight Squat

This was ultimately the exercise that we ended up choosing for our game (specifically when the character jumps over obstacles). The Bodyweight Squat was ideal for our needs in the game and it functioned well with Landmark Detection. More specifically, it was easy for any model that we used (whether a pre-trained model or a model that we trained ourselves using CNN) to detect certain landmarks within the squat since its detection points are very apparent within the human body. For example, identifying body parts like the ankle and knee were very simple since they usually protrude even when wearing pants, thus making it simple to detect these key features using a Computer Vision Model. Another point being that a squat is a relatively slow exercise to perform and thus, most models will not have to constantly calibrate to refind certain points (as for example the ankle is always in a constant position, while the knees and hip are moving in a slow motion). This is vital for accuracy of the model since this is something we struggled with when trying to test out exercises that involved faster ranges of motion which ultimately caused Landmark Detection to be difficult to identify smoothly. Continuing upon the last point, we realized that it would be very easy to detect a correctly performed squat since we can simply calculate the angle (between the hip, knee and ankle) to identify a standard squat (we can set the values to whatever we want, whether it be 90 degrees or under 150 degrees etc). Finally, bodyweight squats were excellent when it came to the number of calories burnt per minute, it’s estimated that around 10 calories can be burnt per minute (if done with high intensity and assuming the person weighs 150 pounds) (Leventon, 1). After seeing the benefits this exercise brings both in terms of the Computer Vision aspect and game-performance, we decided to move forward with this action (and for it to be our main control system for the jumping of the character).

Phase 2: MediaPipe Model for Landmark Detection and Squat Counter

Phase 3: Using

Phase 4: Training the CNN Model with a Unique/Custom Dataset

Phase 5: Model Comparison

Phase 6: Implementation of the Squat Counter Using MediaPipe

**Experiments/Results:**

**Conclusion:**

**References:**

*"Obesity in Rural and Urban Canada."* Public Health Agency of Canada, 4 Nov. 2020, health-infobase.canada.ca/datalab/canadian-risk-factor-atlas-obesity-blog.html. Accessed 29 Mar. 2025. (Ignore Images and Ads)

​Leventon, Scott. "Calories Burned Squats Calculator." *Fitness Volt*, updated by Tom Miller, CSCS, 5 Sept. 2023,<https://fitnessvolt.com/squats-calories-burned/>. Accessed 30 Mar. 2025 (Ignore Images and Ads)

"Push Ups Calories Burned Calculator." *BizCalcs.com*, [www.bizcalcs.com/push-ups-calories-burned/](http://www.bizcalcs.com/push-ups-calories-burned/). Accessed 29 Mar. 2025 (Ignore Images and Ads)

"Calories Burned From Jumping jacks, vigorous." *MyFitnessPal*, [www.myfitnesspal.com/exercise/calories-burned/jumping-jacks-vigorous-421094](http://www.myfitnesspal.com/exercise/calories-burned/jumping-jacks-vigorous-421094). Accessed 29 Mar. 2025 (Ignore Images and Ads)