VizFit: Final Documentation

Sasha Lawson Sam Donaldson
Alana Matheny Noah Buchanan

May 10, 2022

Abstract

This project sets out to solve the following problems: Lack of an insightful fitness breakdown, inconsistency in maintaining a fitness routine, and the difficulty in properly exercising due to improper technique or a lack of general fitness knowledge. We propose solutions to each respective problem. First, by utilizing personalized user analytics that we aggregate over time we will provide insightful information about a user's personalized fitness journey. Second, providing an at-home substitution to going to the gym will allow for a better incentive to users in-order to maintain fitness consistency. Finally, to decrease the difficultly of properly exercising we will implement computer vision and machine learning concepts to determine the accuracy of a given fitness activity and maintain useful statistics such as rest times between exercises.

Our goal, on a much broader scope, is to make fitness easier. This will be accomplished through various functions such as computer vision and machine learning models, a front-end webpage which will provide personalized data and feedback, a database, and an API to move data amongst these entities. Our deliverables will include the Project Proposal, User Interface Design Graphics, Database Schema and Data, Deep Learning Model Code, Webpage Application Development Code, API Development Code, and the Final Report.

Department of Computer Science and Engineering
University of Arkansas – Fort Smith

Contents

Contents **2**

1 Introduction **3**
1.1 Problem Statement . . . 3
1.2 Objective . . . 4

2 Background **4**
2.1 Overview . . . 4
2.2 Technology . . . 6
2.3 Related Work . . . 7

3 Design **9**
3.1 Architecture . . . 9
3.2 Pre-Project Risk Assignment . . . 10
3.3 Tasks and Schedule . . . 11

4 Experimentation & Development **12**
4.1 Model Experimentation . . . 12
4.2 Exercise Recognition Experimentation . . . 18
4.3 Website Development . . . 20
4.4 Python Application Development . . . 27

5 Future Work **32**

6 Project Members **32**
6.1 Team Members . . . 32
6.2 Departmental Advisors . . . 33

7 References **34**

1 Introduction

1.1 Problem Statement

Fitness is important. The benefits of proper fitness in increasing lifespan has been demonstrated in a plentiful number of prior research studies. However, nearly 80% of adults are not getting enough exercise. [1] There are three problems individuals are facing regarding fitness. The first is the lack of an insightful fitness breakdown. Insight into a fitness routine allows one to better identify weaknesses and strengths in both the physical and technical sense. Second, consistency in maintaining a fitness routine. Failure in maintaining a consistent fitness routine leads to inconsistent progression and the risk of dropping the routine all together. Lastly, exercising correctly is important but difficult. For example, repetitive, overdone exercises can cause tearing, stress, and other negative effects on one's body. [2] Each one of these problems overlap in various ways, but stand out distinctly enough to be highlighted as key points to be improved upon in this project.

Exercising is a highly versatile activity. Depending on what exercise one does they net a different result. High intensity exercises that gets the heart pumping faster lower high levels of cholesterol while a peaceful walk soothes high blood-sugar levels. [3] This level of versatility demands personalization to get the most out of one's exercising and avoid overexercising parts of the body. This can be accomplished through insight into one's fitness and health data.

One of the many traditional New Year's Resolutions is to "get into shape". However, about two thirds of those who make any resolution at all quit within the first month [4]. This demonstrates a failure to maintain a consistent fitness routine. Diving deeper into this issue suggests that this inconsistency could be caused by having to go to the gym. Out of 2,000 Americans surveyed, 50% felt working out in a gym environment to be a "daunting" task. Many cited feeling intimidated by fit individuals and even the opposite sex. [5] Fear and laziness are the enemy of consistency in this situation causing many to abandon their fitness plan before making any progress.

Properly exercising is difficult. Setting aside all the issues already listed, safely and correctly exercising is critical. This means just doing the motion of the exercise does not equate to the exercise's intended result. Take a push-up for example, what seems like a straightforward act of going up and down using the strength of one's arms can be done incorrectly. Given the correct situation doing a push-up incorrectly can lead to potential upper body pain and injury [6]. Ensuring one's safety through proper technique is key to lessening the difficulty of properly exercising.

1.2 Objective

Our project sets out to solve three problems that modern fitness applications typically have. Firstly, our application will make fitness insightful. We will keep a variety of statistics for each user to provide invaluable information about their fitness goals and their rate of progression. This makes each user's fitness journey highly personalized.

The second problem we are attempting to solve is the matter of consistency required for physical fitness. Making it to the gym multiple times a week is a daunting task for busy individuals. This is where our application comes in. With our application you can exercise from any location that you have a computer and a webcam.

The last problem for most is that exercising is difficult. Our application aims to make it easier through two key innovations. First, our application will monitor your exercises and make sure you are performing them correctly. Additionally, our application will track your live progress and rest times throughout a workout. Both of these combined can make fitness magnitudes easier for those that find fitness difficult.

2 Background

2.1 Overview

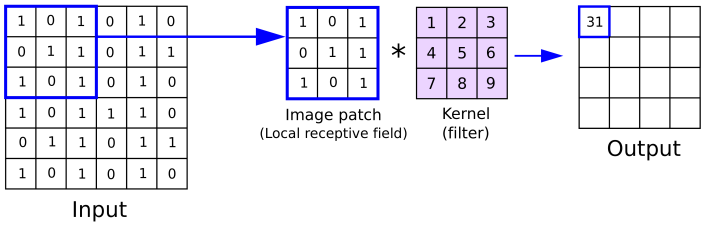
Convolutional Neural Network

The traditional neural network is a powerful tool that can be used in a variety of ways within limitations; one such limitation is the extremely costly computation of large amounts of data i.e images. Classifying multiple images in real time with millions, potentially billions, of parameters (depending on the resolution of the images and the amount of layers and neurons used) is not practical. This is where convolutional neural networks (CNNs) come in handy. CNNs do three specific things to make image classification practical:

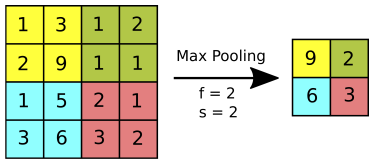
1. Reduce the number of input nodes and parameters.
2. Tolerate small shifts in where the pixels are in the image. Essentially meaning that through each filter that is applied, a band of width = $\text{floor}(\frac{\text{length}(f)}{2})$ in pixels, where f is the filter, will be removed from the edges of the image.
3. Take advantage of the correlations that are observed in complex images.

The first thing a CNN does is apply a filter to the input image. Filters are sometimes also called kernels. For this explanation the former will be used. A filter is a smaller matrix of pixels typically around the size of 3×3. The intensity of each pixel in the filter is determined by back propagation. To begin with the pixel values in the filter are randomized.

The filter is overlaid onto the image and the dot product of the image and the filter is calculated to get a value that we add a bias to and then put into a feature map.



We can say that the filter is "convolved" with the input after performing this which is what gives CNNs their name. The filter is then moved over one pixel or possibly more, this is up to the users discretion, and does that process over again. Once the feature map is finished we run it through an activation function, typically ReLU. Once the map has been run through ReLU we max pool the feature map to further minimize the size, max pooling meaning that the filter does not overlap itself, so for a 2×2 filter it would move 2 pixels. Max pooling selects the spots where the filter did the best job matching the input image; mean pool alternatively could be used where the mean of the filter is taken rather than the maximum value.



Now we convert the pooled layer into an $n \times 1$ input layer. These input nodes are then plugged into a normal neural network. [7]

2.2 Technology

MySQL MySQL is a relational database management system. MySQL remains free and open source providing high data scalability, security, and high performance speeds. [8] MySQL is widely used by companies such as Youtube, Twitter, Facebook, Netflix, Github, and Paypal. [9] Our primary method through which we query and insert data is through SQL calls via JDBC from our Java-based API.

Vue.js Vue is a self-described "progressive" framework for constructing user interfaces and front-ends. Vue's core library focuses on the view layer only, meaning one can start simple then build up in complexity from there. [10] The simplicity of Vue comes in the form of components which contain some data and/or logic. These components can be reactive based on logic within the component's script. Vue's easy to use framework was used to build the webpage dashboard where users can track their health and fitness analytics.

Vuetify Vuetify is a user interface framework built to complement Vue.js. Vuetify is easy to learn and implement into any vue app. Vuetify has "meticulously" crafted components that are built to be responsive and work "out of the box" for any type of screen. [11] Vuetify's functionality is used to allow for easy web design development in our front-end system.

Spring Spring is a RESTful API built on Java. Spring has "battle-tested" security that protects against attacks. Spring is commonly used so documentation is plentiful. Since Spring is built on Java, which is a language each of our team members is familiar with, each team member is able to help develop the API. [12]

OpenCV-Python OpenCV-Python is an open-source python library package that provides various computer vision functionality. [13] OpenCV-Python was integral to the application as it provided the connection between the webcam and the machine learning program.

PyQt5 PyQt5 is a Python binding used to implement and simplify GUI development. One such advantaged provided is that the user use drag and drop functionality to create user interfaces. [14] This package was used to create the Python Application interface in a streamline manner.

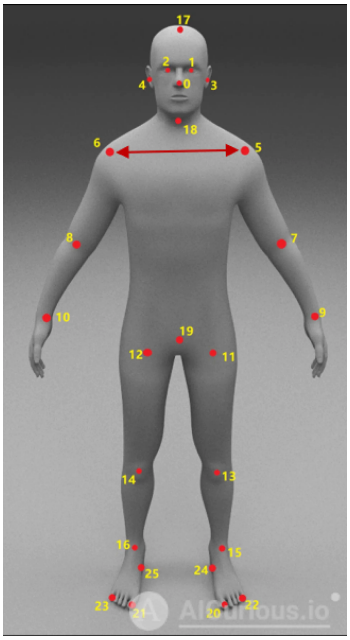
MediaPipe MediaPipe is a cross-platform package that offers customizable machine learning solutions. These machine learning functions can be used on live video feedback and recordings. [15] MediaPipe was useful in creating a target to aim for regarding the wireframe model.

TensorFlow TensorFlow is a library package with various machine learning models that is used by both beginners and experts alike. [16] TensorFlow was used to create neural networks that were critical in the training process. Additionally, TensorFlow offered various training data that was useful.

2.3 Related Work

Push-up Counter

In [17] a utilization of OpenCV and Deep Learning was demonstrated to complete the task of counting push-ups. Their specific approach involved human keypoint detection (or human pose estimation). They used the MPII Human Pose Dataset and various push-up videos from Facebook as their training dataset. They used a unique metric to measure the quality of these keypoints, Percentage of Correct Keypoints (PCKs): "A detected joint is considered correct if the distance between the predicted and the true joint is within a certain threshold. The threshold here is chosen as 0.25 times of the distance between 2 wrists (or the distance between point 5 and point 6 in the image)."



This model for key point extraction is separate from the exercise recognition model. Signal processing from the keypoints is what is used to count repetitions. In their conclusion they stated they felt they could have achieved better results with more keypoints and leveraging the keypoints to correct wrong push-up poses. We will attempt to achieve both of these. The "sweet spot" of keypoints will likely be found through experimentation so we cannot report on that as of now. As for bad exercise poses, we intend to measure and fix these through the angles between specified keypoints (the best keypoints to measure will also likely be apparent through experimentation).

Recognizing Exercises and Counting Repetitions in Real Time

In [18] pose tracking and exercise recognition is used to count repetitions on pull-ups, push-ups, and squats. Their method involves three phases; pose tracker to identify and track users, exercise recognition to detect the name of the appeared exercises, and a counter to count and indicate the correct and incorrect repetitions. For pose tracking they use OpenPose. For exercise recognition however, the authors of this paper did not use a convolutional neural network rather a normal neural network; it is our teams' belief that we can achieve better results through a convolutional neural network. For repetition counting they first use a formula and the predictions of their neural network to predict the exercise as follows:

$$\text{performed\_exercise}(f) = most\_common(y(f-9), y(f-8), ..., y(f))$$

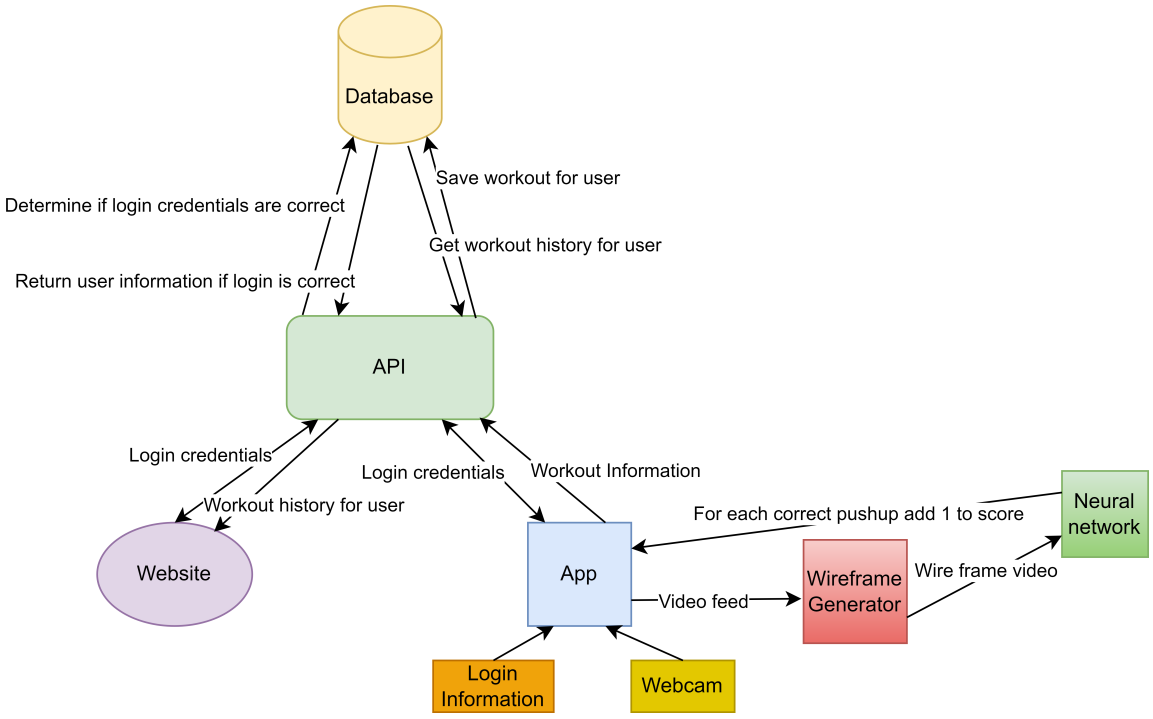
Where, f: frame, and y: predicted label from the neural network.

Once the exercise is detected, there are pre-selected parameters such as exercise range of motion, the major joint, and type of motion (push or pull) that will be used to count repetitions. We will likely use the same formula for detecting exercise as them. However, as stated before, they did not use a convolutional neural network; we will attempt to implement this and achieve better results.

3 Design

3.1 Architecture

Our application is a basic program that captures video feed from a user's device. The application sends that video feed to an application that generates wire frames based on a human body. The application then sends the video of the wire frames to our state-of-the-art neural network. The network then determines if the wire frames, based on the user, are moving in such a way that the user is doing a correct push-up. If the network determines that a user has done a correct push-up it increases the user's score. We keep track of the number of correct push-ups the user has done and save that information based on the user's credentials to our database. The database is behind an API that allows us to easily select and insert information securely. Our website is also connected to this API and will display a user's information based on their login credentials.



3.2 Pre-Project Risk Assignment

Building a fitness application comes with many risks. Given below are some of the common risks and the steps we are taking to mitigate them.

1. **Data Security** - Data security for users is very important to us. To assure privacy for our users' data, we will implement appropriate and thorough encryption to their data through a secure API and database. We will also make the data we collect as anonymous as possible, and the user will own their data. For example, the data we collect will not be sold or given to anyone and we will not save any recordings we take of a user doing their exercises.

2. **Issues With The Systems** - Many things could go wrong when implementing the systems that would prevent users from using the application. For example, if the server goes down, the API can not connect to the database, and if the API can not connect to the application and/or website the user would not be able to save their exercise data or retrieve their saved data. To prevent this issue from occurring the database, API, application, and website will be thoroughly tested for bugs, constructed properly with dependability in mind, and the appropriate security will be implemented to ensure privacy for the user. One other issue that would prevent users from using the application would be if the machine learning model is constructed and integrated into the application incorrectly. This issue could lead to not being able to detect the user or correctly interpret the data it collects. To ensure this is prevented, we will do extensive testing when integrating the model into the application and have the model inspected by every team member.

3. **Bias Datasets** - When using neural networks to learn to identify humans and the specific movements they do, bias or incorrect datasets while training or after training could result in unfair results depending on the user. To prevent bias datasets, we will implement well-tested, pre-trained datasets to track users and the specific movements they do. In looking for datasets we will attempt to find datasets where the set will include different body types, ages, genders, etc. By using different groups of people. This will help standardize results and mitigate bias. One of the other ways we will implement a bias-free model will be to create a stick figure representation of a human to train the model to so differences in users body type, clothing, and other factors will be mitigated.

3.3 Tasks and Schedule

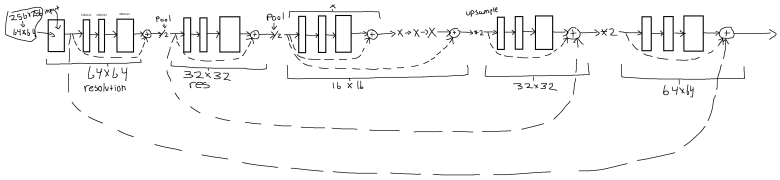
Task	Dates	Status
Create, revise, and finish proposal	1/10 - 1/22	Completed
Research and find good exercise datasets	1/17 - 1/30	Completed
Implement code for initial dashboard webpage design	1/24 - 1/30	Completed
Implement code for initial machine learning application GUI	1/24 - 1/30	Completed
Setup and configure database	1/24 - 1/30	Completed
Create and implement database schema	1/24 - 1/30	Completed
Implement wireframe manipulation	1/31 - 2/6	Completed
Create first neural network model	1/31 - 2/6	Completed
Create presentation for Week 5	1/31 - 2/6	Completed
Create preliminary back-end API	1/31 - 2/13	Completed
Create controller, readers, and writers classes for back-end API	1/31 - 2/13	Completed
Connect dashboard webpage API to back-end API	2/7 - 2/13	Completed
Create functionality for users to create/delete their account	2/7 - 2/27	Completed
Improve, rebuild, and train neural network model	2/7 - 2/27	Completed
Improve visual design of dashboard webpage	2/14 - 2/20	Completed
Finish GUI for machine learning application	2/21 - 2/27	Completed
Create user authentication for both front-end applications	2/21 - 2/27	Completed
Create presentation for Week 8	2/21 - 2/27	Completed
Improve, rebuild, and train neural network model	2/28 - 3/13	Completed
Finish front-end application designs	2/28 - 3/6	Completed
Ensure back-end API sends/receive data from machine learning application API	3/7 - 3/13	Completed
Attempt to finalize neural network model	3/14 - 3/20	Completed
Fix issues with front-end applications' designs	3/14 - 3/20	Completed
Create presentation for Week 11	3/14 - 3/20	Completed
Review all system code and correct bugs as needed	3/21 - 3/27	Completed
Discuss adding extra features	3/21 - 3/27	Completed
Implement complete user testing and fix bugs as needed	3/28 - 4/10	Completed
Finalize any project paperwork	4/4 - 4/10	Completed
Receive project review and make adjustments as needed	4/11 - 4/17	Completed
Final presentation	4/18 - 4/24	Completed
Project defense	4/24 - 5/1	Completed

4 Experimentation & Development

4.1 Model Experimentation

- **Experiment 1:** Our first attempt at a pose estimation model involved a relatively small network consisting of skip connections and a direct regression of key point coordinates. The architecture leading up to the dense layer outputs follows the pattern of hourglass modules and the same can be assumed for all experiments past this point.

  A rudimentary sketch of the model architecture is shown below:



  The flatten layer to the final dense output is not included in this specific illustration.

  As a result of our outputs being direct regression of coordinates, the target value was a vector of size 32, 16 x and y values flattened into one vector. At the time of this experiment our data augmentation consisted only of data normalization(scaling pixel values 0 - 1). The result of training our first model on the data was not deterministic in its reasons for failure. However, given the nature of the relatively unchanged data we trained on and the smaller scale of the first architecture we felt that we had plenty of room to improve.

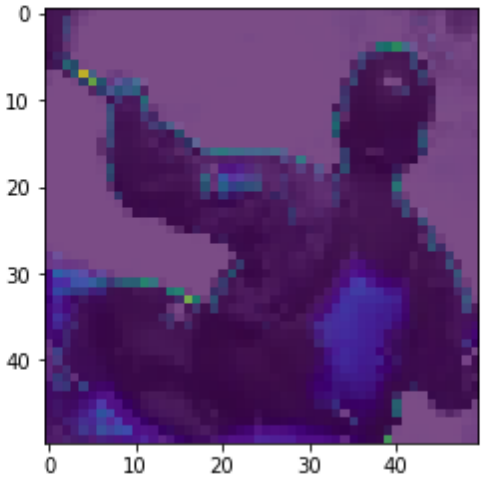
- **Experiment 2:** Our second major experiment we performed involved augmenting the data according to best practice techniques in the various research papers we have been using for inspiration. The data augmentation consisted of cropping the images around the annotated person in each image as well as removing all images where all key points were not visible; occlusion of joints is a common problem researchers attempt to solve but for our purposes it is unnecessary. Images prior to this in many occasions contained multiple people with only one of them being annotated. At the time of this experiment we were still using an unchanged architecture and found minimal improvements with regards to the data changes.

- **Experiment 3:** After we saw minimal improvement from the prior data augmentation it was time for us to take a look at our model. After some further research we were led to the decision of creating heat maps instead of direct key point regression predictions from dense layers. These types of convolutional neural networks are typically referred to as fully convolutional networks. With this change of our model further data augmentation was required alongside this. In order for our network to output heat maps our target outputs that we are calculating the loss of our network from must also be heat maps. Our data augmentation now consists of normalizing pixel values to 0 - 1 and converting the key point coordinate targets to heat map targets. The process of creating the heat maps is relatively simple: Each key point is one 256×256 matrix with a single pixel with a value of 1 while the rest is 0, the image then undergoes Gaussian blurring and we have our first key point heat map, do this for the

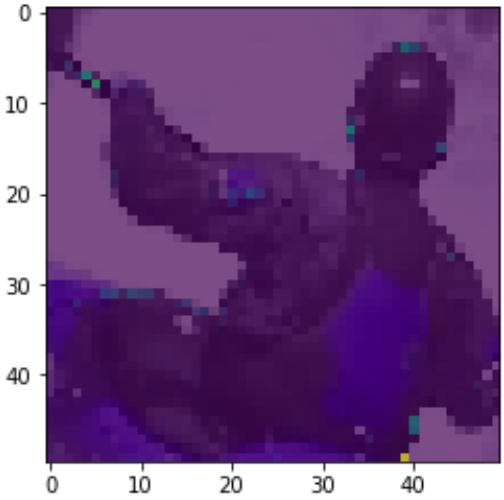
other 15 key points in each target output and we are ready to train. Each target value now has the shape $16 \times 256 \times 256$, 16 denoting each key point and 256×256 denoting the size of each heat map. The output was now generated from two subsequent convolution layers that output 16 filters each one for each key point. Unfortunately this data augmentation had major drawbacks: our data was now vastly more expensive to hold in memory considering instead of holding 32 floating point values we now have 256 * 256 * 16 floating point values. As a result of this the amount of images we could train on at a time reduced drastically. We developed a workaround of training on roughly 100 images at a time for 10 epochs each and then moving on to the next 100 in the data set. Regardless of any attempted fix this affected our results heavily resulting in low accuracy. Some results of the model after minimal training and roughly 30 minutes of training are included below:

```
plt.imshow(imgn, cmap='jet', alpha=0.3)
```

```
t[199]: <matplotlib.image.AxesImage at 0x1dc180d4400>
```

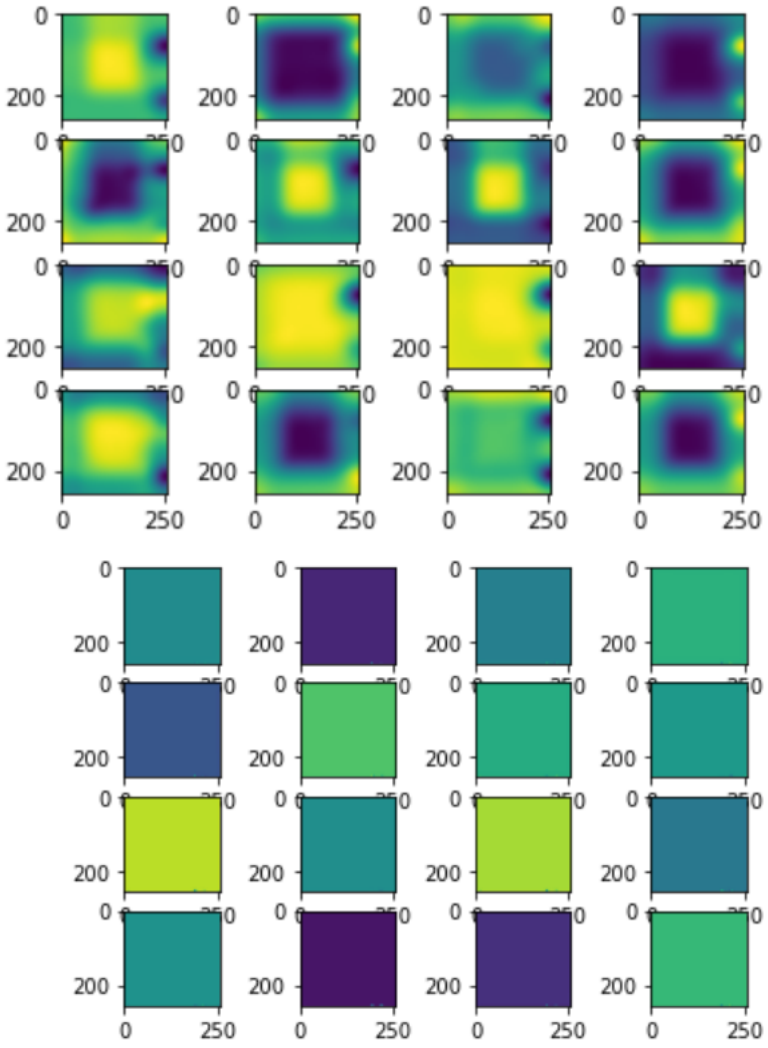


t[215]: <matplotlib.image.AxesImage at 0x1dc18de35b0>



As you can see after some time training the parts it seems to be learning are vanishing, this problem was reoccurring and if the model was left to train for longer periods the heat maps would become completely blank.

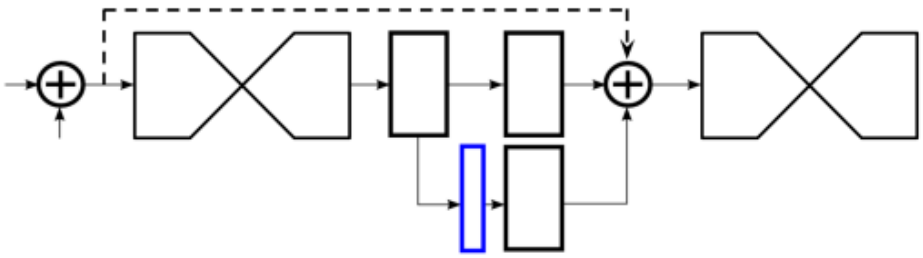
- **Experiment 4:** In an attempt to combat, what we thought at the time to be a vanishing gradient problem, we implemented Leaky ReLU in place of ReLU as well as changing the architecture of our skip modules to implement batch normalization and added more skip modules overall increasing the size and amount of parameters of the architecture. Below are the results of minimal training and extensive training:



Once again it is evident that our models training time is not synonymous with increased accuracy. However, the result of this model was particularly perplexing considering the heat maps were no longer zeros but one, extremely close, if not the same, pixel value for all pixels.

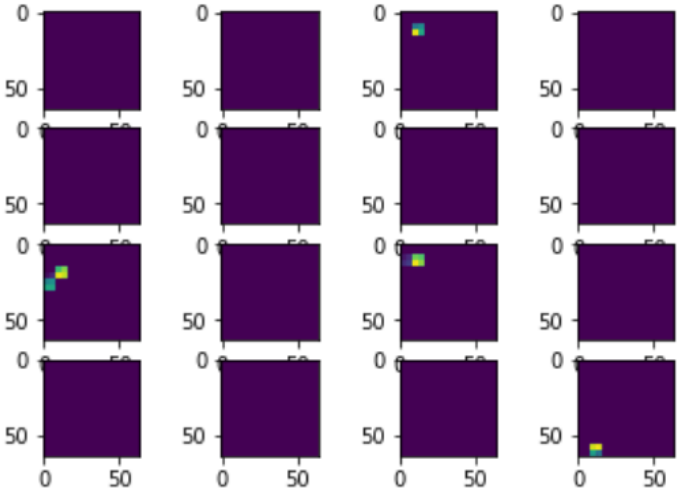
- **Experiment 5:** After the past experiments' failures we began further modifying the architecture. First, we would adopt the full architecture of the hourglass architecture, and instead of beginning and ending at a resolution of 256×256 for each hourglass module we would first downsize to 64×64 before being fed into any hourglass module. This also meant that our target outputs would take up significantly less memory since the dimensions now consisted of $16 \times 64 \times 64$. This allowed us to have a much larger amount of data during training time which resulted in an instant increase in accuracy, however it was still not quite accurate enough.
- **Experiment 6:** The final experiment performed involved implementing intermediate su-

pervision and utilizing the new found space in memory we had to work with. The latter was a simply relatively process: instead of just having a single hourglass module we would stack them back to back, specifically 8. This is where intermediate supervision comes in. In between each hour glass module the output is mapped to 16 output filters so that we can apply loss to it, once this is done it is mapped back to 128 output filters so it can be added with the residual connection from before the current hourglass and the original output from the current hourglass. An illustration of this process is included below:

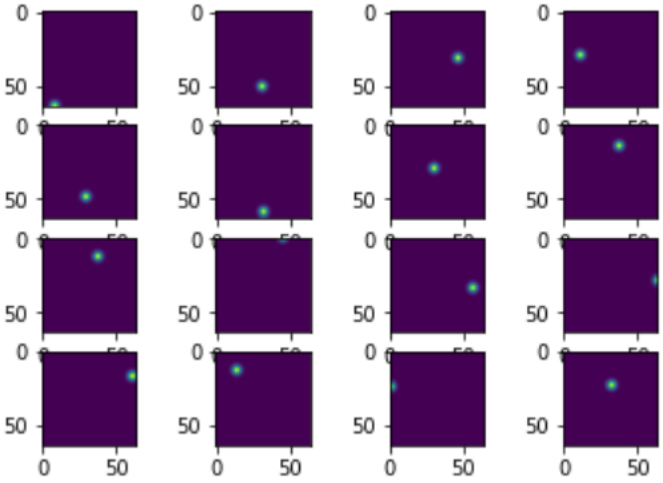


[19]

The result from these changes made it appear that the model was actually learning something, finally reaching 30% accuracy after not particularly long periods of training. However, the model did not reach satisfactory results so that we could use it for the production application. Below are some example outputs from the model:



Compared to the target output:



While we cannot say with complete confidence that if we had more resources and more time we could continue to train this model and it would be accurate, we can say that it did appear to begin localizing certain joints and possibly could have continued to learn with more training.

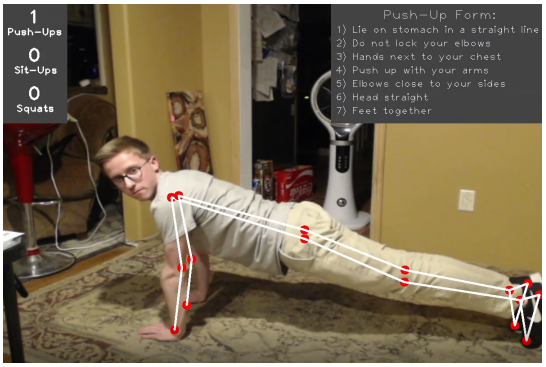
4.2 Exercise Recognition Experimentation

The approach the team took to how the application detected and ensured correct exercising was the use of angles and trigonometry functions. Given the points generated by the wireframe model, trig functions were used to determine the angles at certain locations on the body. Certain angle ranges at the correct locations can help determine the exercise and how well that exercise is being done. Additionally, some exercises require the distance between two points being checked. The application checks three exercises: push-Ups, sit-Ups, and squats. Each exercise is checked in its own method; however, each method is called at the same time so each exercise is checked at the same time. If one of the three exercises is identified then a counter is increased for that particular exercise.

- **Push-Ups Recognition:** The first version just focused on one angle. This angle used three points on the right side of the body. Point one being at the hand, point two being at the elbow, and point three being at the shoulder. The angle in question is formed at the elbow when the individual pushes up or goes down.

  It was not until the second official version that the left side of the body was put into the play. An average was used between the right angle and the left angle to determine a 100% value for going up and down. Additionally, it was in the second version that a "straight body" checker was implemented. This checked the straightness of one's body from their head to their feet. This ensured that no one could simply move their arms while sitting and gain points.

  It was not until the third official version that two critical checkers were added. Number one, a "Is Standing" checker was added. This, of course, ensured that the individual was not just simply standing straight and moving their arms in a push-up fashion. Number two, the position of the hands in comparison to the head was checked. Through discussion with others it was realized that an individual could lay on their back and do a push-up motion with their hands and the system would count that as a proper push-up. This, of course, is wrong and was corrected.

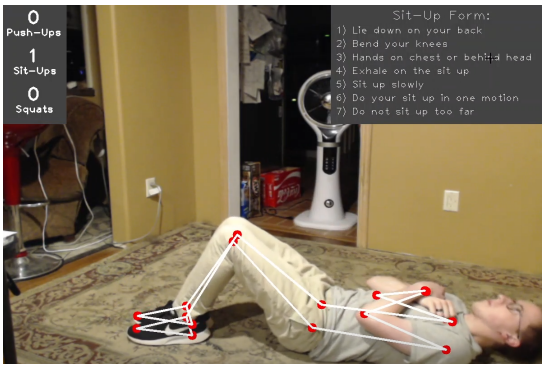


- **Sit-Ups Recognition:** The initial version took into account three keypoints on the left side of the body. One at the shoulder, another at the torso, and the last at the knee. As the sit-up motion is taken the angle produced at the torso between the leg and the upper body

is measured.

Once again, it was not until the second official version that the other side of the body was factored in via an average. Another checker was also implemented. This time it checked the angle at the knee to ensure that the user was not straightening their legs at any point.

For the most part, the sit-up was quite straight forward as it is a hard exercise to "cheat" at. As such, fewer checkers were required. With this said, another "Is Standing" checker was added just in case. Additionally, a feedback checker was added to help notify when to give feedback regarding incorrect leg or back positioning.



- **Squats Recognition:** The squat proved to be very challenging as it was easily confused with people doing actual squats and people just getting on the ground. The original exercise recognition function was very simple. It looked at three keypoints on the right side of the body. The first point was at the torso, the second was at the knee, and the last point was right at the ankle. The only angle that was being checked was the angle formed when the knee bent during a squat.

  In the second official version, the left side was added and averaged together with the right side. Additional, an angle check was added to both shoulders. This was intended to check whether or not one's arms were out perpendicular to the body. Not only is this a proper form, but it helped eliminated the confusion the the function had with individual simple getting on the ground.

  It was not until the third official version, that the standard "Is Standing" checker was added. However, in this case it was actually checking if the user was standing as opposed to on laying on the ground unlike in the previous two exercises. In addition to this, the "Arms Out" checker was reworked as it was too inconsistent on whether it would catch the arms true angle. As apart of its reworking, the angle approach was dropped in favor for the point-to-point comparison approach that was first seen in the push-up recognition. The points in question are the hand points in comparison to the hip points. If one's hands are next to their hips the exercise will not detect a squat. For the most part, this solved the issue.

- **Other Highlights:**
  - *Left Facing vs Right Facing:* One issue that popped up quickly was that the individual would not also being facing the camera the perceived "right way". Since the angles relied heavily on which side of the body was facing the camera an opposing set of angles had to be created to accommodate either side of the body being faced at the camera.
  - *Duration System:* To ensure that once a correct action was recognized the counter would not continuously add up, a duration system was put in place. Simply put each of the exercises can be broken into two parts. For example, the push-up is partly pushing up and partly going down. Once the first part is correctly identified the duration is set to 1. Then once the second part is correctly identified the duration is set back to 0. Once both of these actions are seen the counter increase only once. What enables the counter to work properly is its reliance on the the duration going back and forth.
  - *In Frame:* The "isInFrame" method ensures that the body is well enough in frame for the wireframe to properly map onto the body. This function is quite handy as it significantly reduces the amount of errors that are possible. This due to the fact that because the whole body must be in frame before the wireframe maps on, the wireframe is less likely to misidentify apart of the body as something it is not.

4.3 Website Development

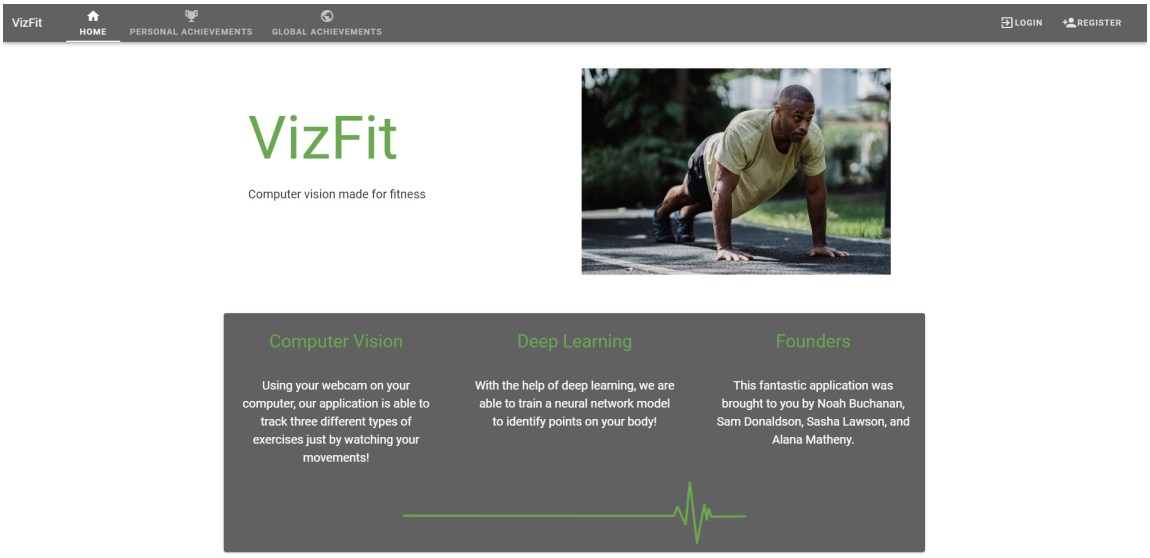
- **Registration:** Registration was accomplished by storing a unique username and password hash in the database. If the provided passwords during registration does not match the website will not allow the user to register until they do match. If the username already exists on the database the website will notify the user that that username is already taken.

- **Login:** Login was accomplished by taking the entered username and password from the user, passing the password to a hash function, and checking to see if there is a row on the table with that username and password hash. If there is no result from the query the website will display "username and/or password is incorrect" to the user. If there exists a user on the table with the given username and password hash, the API will return a User object with the username and user ID and a Session object with the username, user ID, and session hash. The session information will be inserted into the database for future use. The website will write the returned session object to local memory to use for future automatic login attempts.

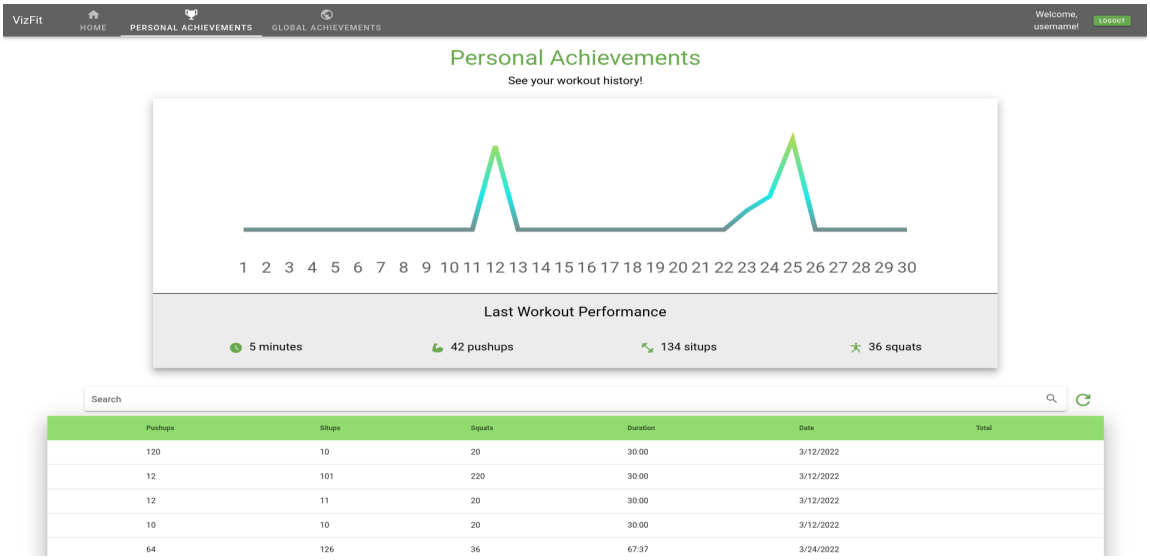
When the website is closed and reopened, if a session exists in local memory, the app will send a request to attempt to login with that token. If the database contains a session with the provided user ID, username, and session hash, that will be considered a valid login. The application will then generate a new session object, with a new session hash, username, and user ID, that will be sent back to the website. The website will then update the session object stored in memory with the new session information. Logging out will clear the saved session object so there will be no automatic login attempt once the page is re-opened.

- **Personal Statistics:** Personal statistics are determined by the user ID of the currently logged in user. Personal statistics are automatically gathered and displayed if the user is logged in. If the Workouts table contains workouts belonging to that user ID, the API will return those rows to the website to be displayed.
- **Global Statistics:** Global statistics is a table with a user on each row. The table has the total number of each exercise each user has done and is sorted in descending order of total exercises completed. This table is automatically refreshed when the website it opened.

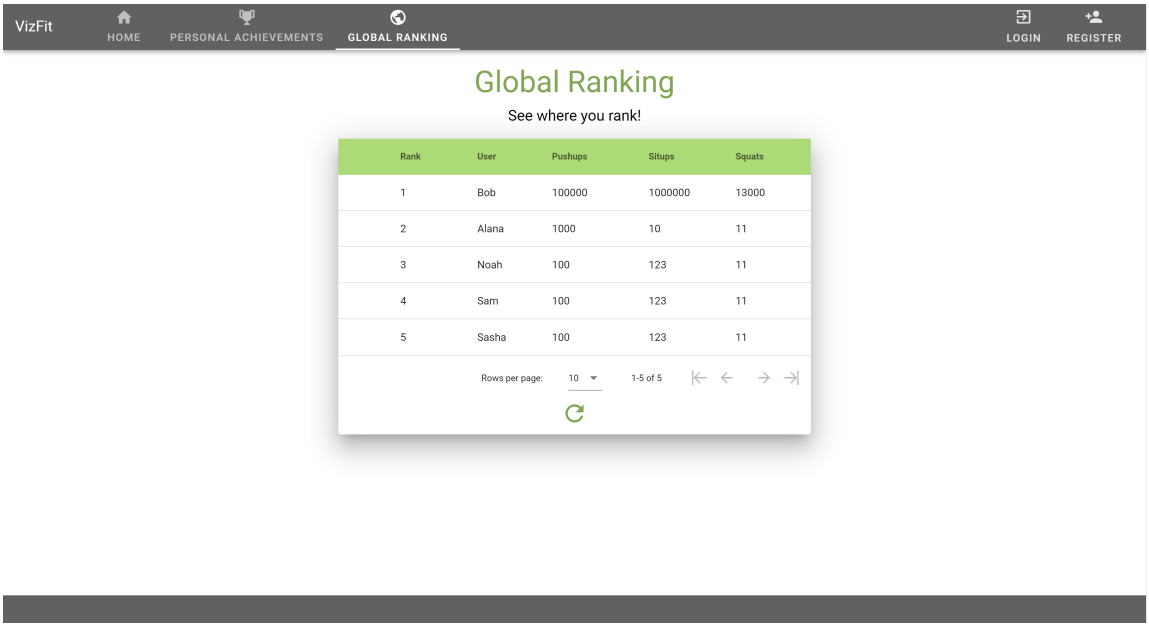
Website Homepage



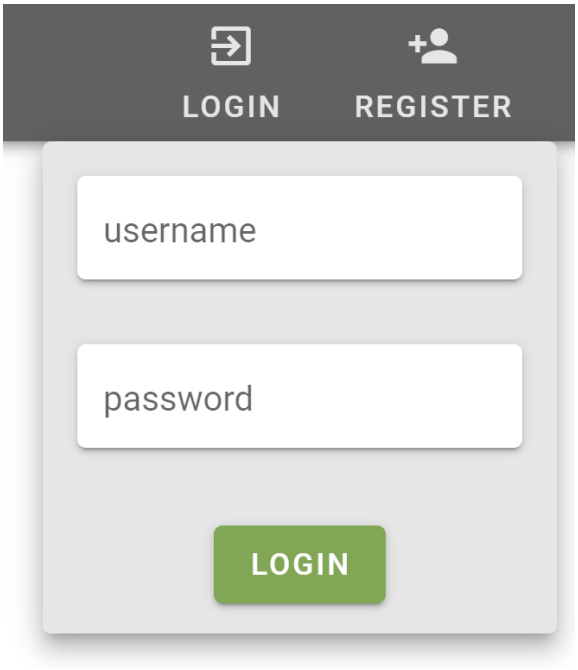
Website Personal Analytics Page



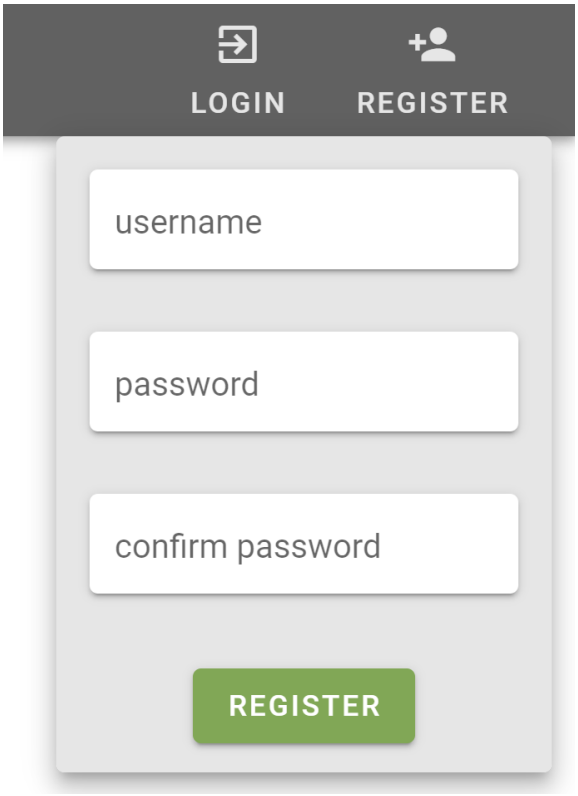
Website Global Ranking Page



Website Login



Website Register



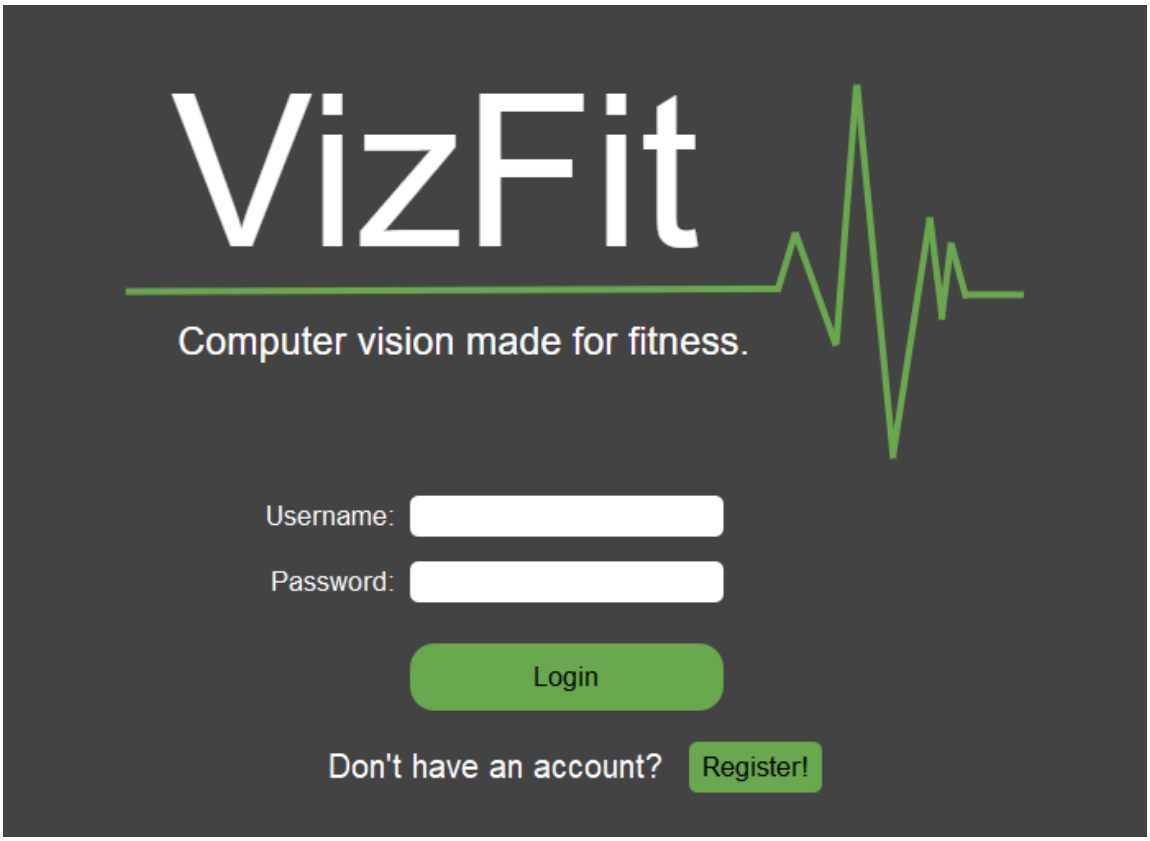
4.4 Python Application Development

The Python Application is the user's front-end to the wireframe model and exercise recognize functions. Additionally, the user is able to register, login, and connect to the website using this GUI.

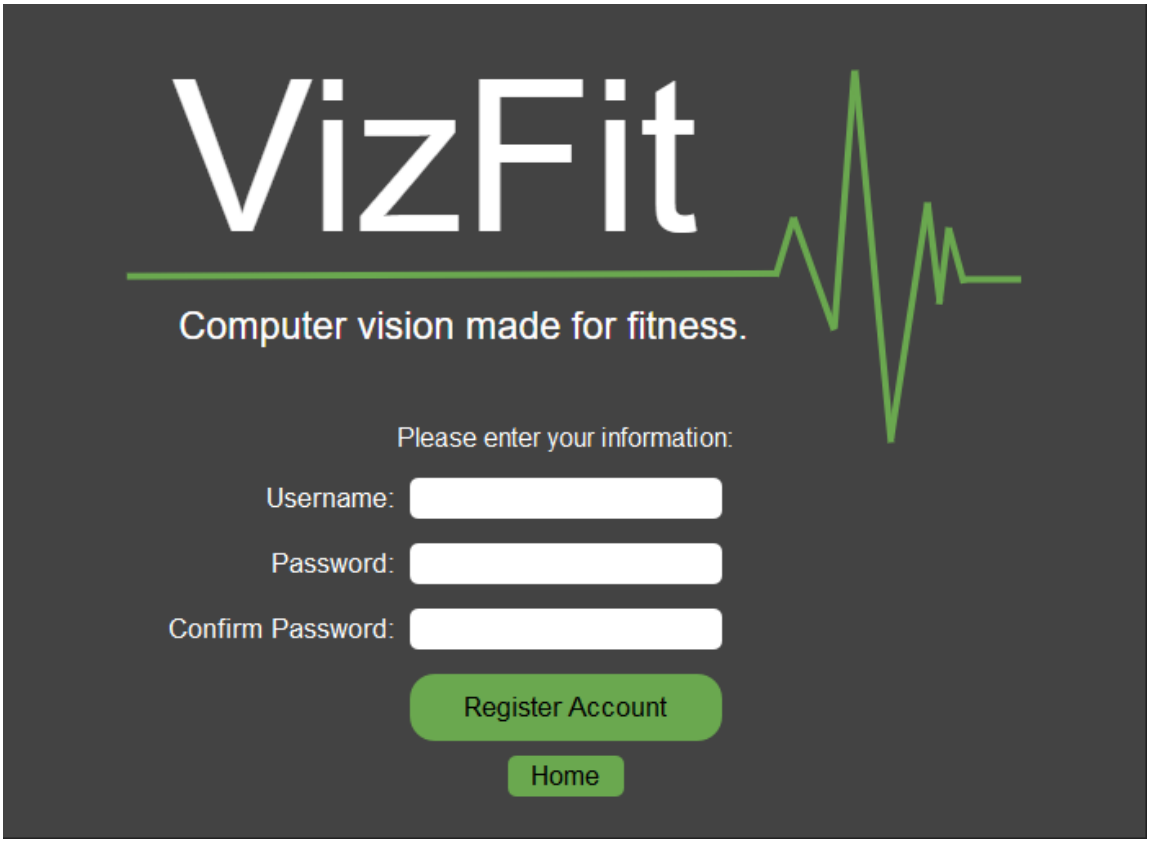
The application is a fixed $700x510$ because there is not too much for the user to interact with so taking the full screen would only created unused real estate on the page. Additionally, the application takes advantage of the stacked widgets. A widget represents a front-end page. Each widget can be swapped in and out for viewing/interacting purposes. In total there are five widgets: HomeWindow, RegisterWindow, WelcomeWindow, ResultWindow, and Result2Window. The reason for their being two result windows is that there is a function within the result window that can immediately lead back to itself. To ensure that everything is properly reloaded (i.e. exercise scores) it is sent to a different, yet same result window. These two windows bounce back and forth for as long as the user is using the exercise application. Each widget connects and transitions smoothly with each other.

The registration, login, and sending exercise data functions all rely on the Python API which has specifically setup methods to deliver and receive any information that is required. All error handling is done locally on the application as opposed to the back-end API or elsewhere within the process.

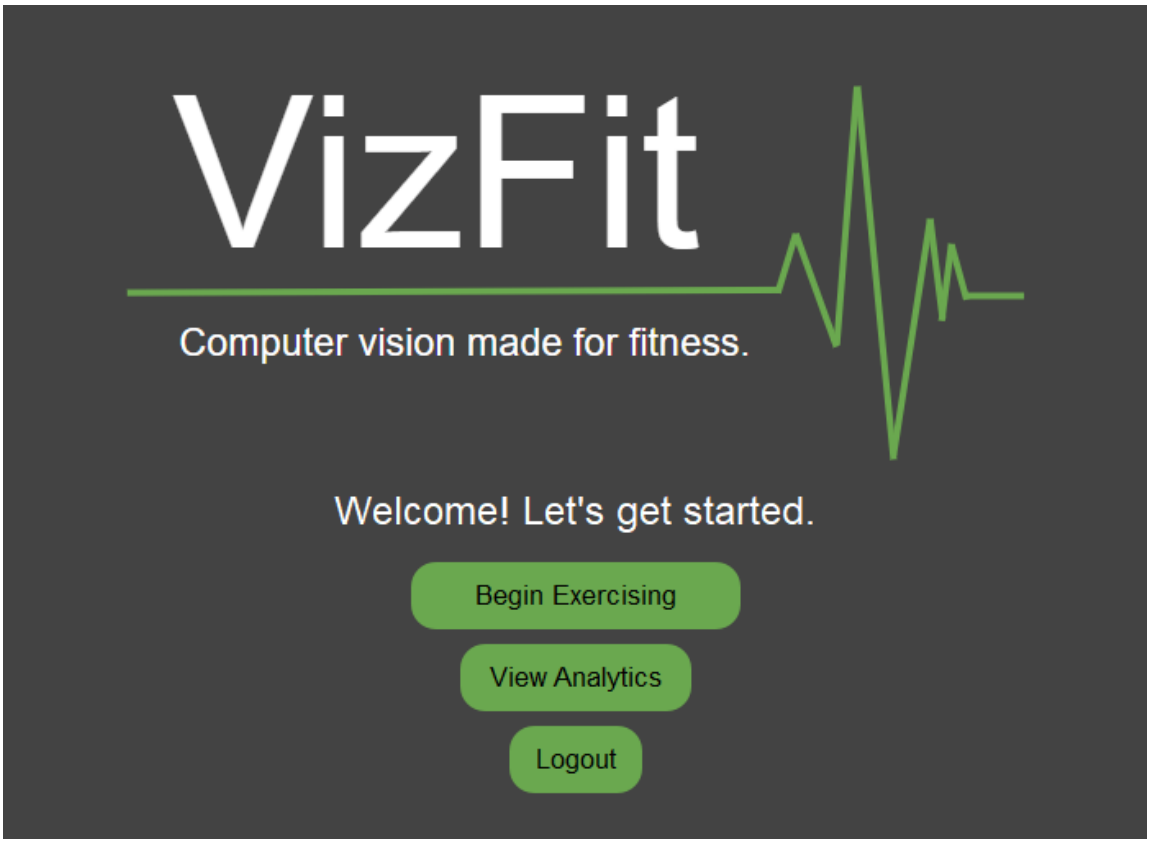
Home Window


VizFit
Computer vision made for fitness.
Username:
Password:
Login
Don't have an account?
Register!

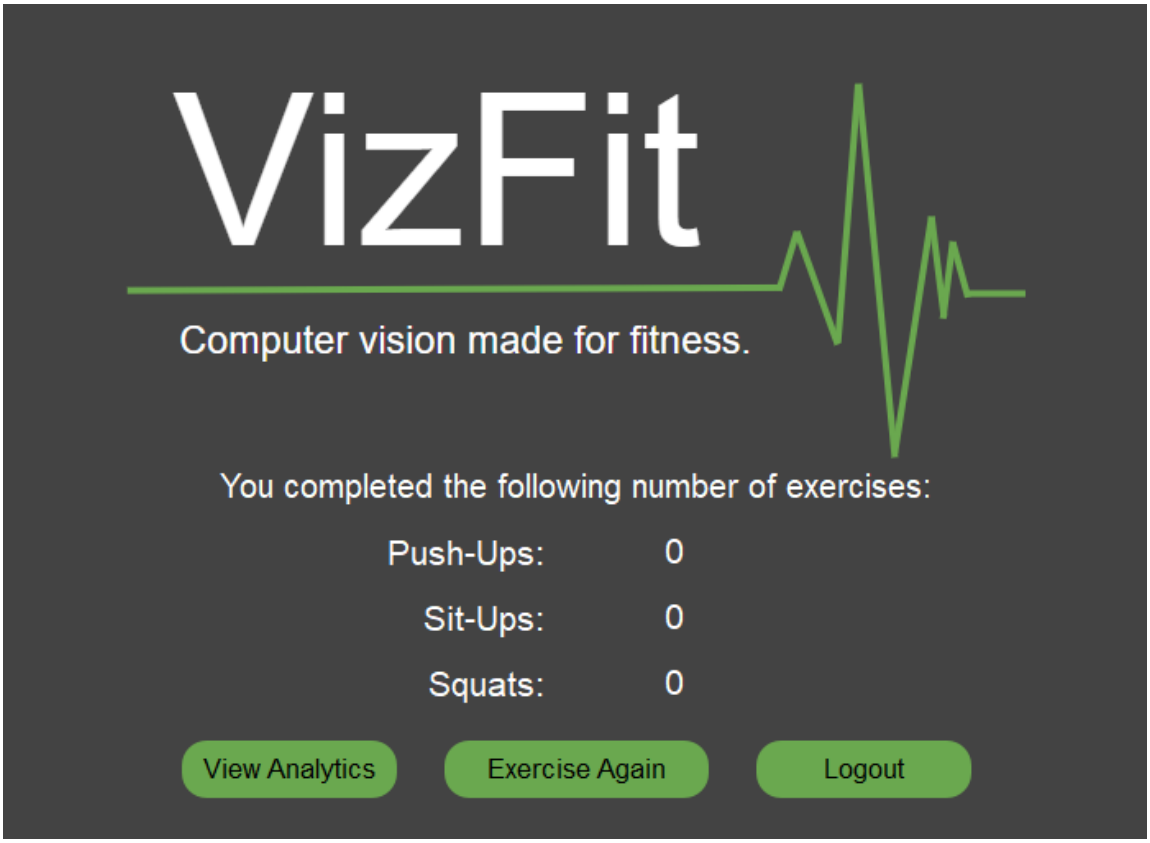
Register Window



Welcome Window



Result/Result2 Window



5 Future Work

- Further Improve Exercise Recognition
- Add Additional Exercises
- Improve Feedback
- Improve Wireframe Model
- Compatible Mobile Development

6 Project Members

6.1 Team Members

Alana Matheny (4023) Alana Matheny is Computer Science major with concentration in Data Science and Artificial Intelligence and a minor in Mathematics in the department of Computer Science and Engineering at the University of Arkansas - Fort Smith. She has completed relevant coursework for the proposed project by completing CS 3113 - Artificial Intelligence, CS 2033 - Web Systems, CS 2043 - Database Systems II, CS 3003 - Distributed Systems, CS 4003 - Software Engineering, CS 4033 - Ethics in Professional Practice, CS 3323 - Computer Graphics, CS 4043 - Formal Languages, CS 4323 - Data Analytics, and CS 4373 - Information Retrieval. Her responsibilities will include the development of a machine learning model, front-end development of the website, front-end development of the application, back-end development of the database, and back-end development of the API.

Sasha Lawson (4023) Sasha Lawson is a Computer Science major with concentration in Data Science and Artificial Intelligence and a minor in Mathematics in the department of Computer Science and Engineering at the University of Arkansas – Fort Smith. He has completed relevant coursework for the proposed project by completing CS 3113 – Artificial Intelligence, CS 3323 Computer Graphics, and CS 4333 – Machine Learning. He also obtained relevant experience through his internship at ArcBest Technologies as a Software Developer Intern. His responsibilities will include the development of a machine learning model to learn the proper actions for a given exercise. Additionally, he will develop required back-end systems to support the dashboard webpage.

Sam Donaldson (4023) Sam Donaldson is a Computer Science major with concentration in Data Science and Artificial Intelligence and a minor in Mathematics in the department of Computer Science and Engineering at the University of Arkansas – Fort Smith. He has completed relevant coursework for the proposed project by completing CS 3113 – Artificial Intelligence, CS 4343 – Natural Language Processing, CS 2033 – Web Systems, CS 4003 – Software Engineering, CS 3323 - Computer Graphics, and CS 2043 – Database Systems II. He also obtained relevant experience through his internship at ArcBest Technology as a Software Developer Intern. His responsibilities will be the development of a machine learning model, front-end development of the website, and back-end development of the database and API.

Noah Buchanan (4023) Noah Buchanan is a Computer Science major with concentration in Data Science and Artificial Intelligence and a minor in mathematics in the department of Computer Science and Engineering at the University of Arkansas – Fort Smith. He has completed relevant coursework for the proposed project by completing CS 3113 – Artificial Intelligence, CS 4333 – Machine Learning, CS 2033 – Web Systems, CS 4003 – Software Engineering, and CS 2043 – Database Systems II, CS 4373 – Information Retrieval, CS 4363 – Internet of Things Development. His responsibilities will include the development of our proposed model, front-end development, back-end development, and development of our API.

6.2 Departmental Advisors

Our project was supervised by Professor Israel Cuevas and Professor Andrew Mackey in the department of Computer Science and Engineering at the University of Arkansas – Fort Smith.

7 References

[1] "The physical activity guidelines for americans." [Online]. Available: https://jamanetwork.com/journals/jama/article-abstract/2712935

[2] "7 ways exercise can go wrong for your mind and body." [Online]. Available: https://www.psychologytoday.com/us/blog/fulfillment-any-age/201409/7-ways-exercise -can-go-wrong-your-mind-and-body

[3] "National institutes of health," Aug 2020. [Online]. Available: https://newsinhealth.nih.gov/2020/07/personalized-exercise/

[4] J. M. Dickson, N. J. Moberly, D. Preece, A. Dodd, and C. D. Huntley, "Self-regulatory goal motivational processes in sustained new year resolution pursuit and mental wellbeing," *International Journal of Environmental Research and Public Health*, vol. 18, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1660-4601/18/6/3084/

[5] "Gymtimidation is real: Half of americans afraid to work out around others," Sep 2021. [Online]. Available: https://swnsdigital.com/us/2019/03/gymtimidation- is-real-half-of-americans-afraid-to-work-out-around-others/

[6] A. Malaythong, "Proper pushup form and technique: Nasm guide to push-ups." [Online]. Available: https://blog.nasm.org/nasm-guide-to-push-ups/form-and-technique/

[7] A. H. Reynolds, "Convolutional neural networks (cnns)," Oct 2017. [Online]. Available: https://anhreynolds.com/blogs/cnn.html/

[8] T. Branson, "8 big advantages of using mysql," May 2021. [Online]. Available: https://www.datamation.com/storage/8-major-advantages-of-using-mysql/

[9] "Mysql." [Online]. Available: https://www.mysql.com/

[10] Vue.js, "Introduction - vue.js." [Online]. Available: https://vuejs.org/v2/guide/

[11] "Why you should be using vuetify." [Online]. Available: https://vuetifyjs.com/en/introduction/why-vuetify/

[12] "Building rest services with spring," 2022. [Online]. Available: https://spring.io/guides/tutorials/rest/

[13] "Opencv." [Online]. Available: https://docs.opencv.org/4.x/d1/dfb/intro.html

[14] "Pyqt5." [Online]. Available: https://pypi.org/project/PyQt5/

[15] "Media pipe home." [Online]. Available: https://google.github.io/mediapipe/

[16] "Tensorflow 2 quickstart for beginners : Tensorflow core." [Online]. Available: https://www.tensorflow.org/tutorials/quickstart/beginner/

[17] V.-A. Nguyen, "Build a pushup counter app with opencv and deep learning," Feb 2021. [Online]. Available: https://aicurious.io/posts/2021-02-15-build-a-pushup-counter/

[18] T. Alatiah and C. Chen, "Recognizing exercises and counting repetitions in real time," May 2020. [Online]. Available: https://arxiv.org/abs/2005.03194/

[19] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016. [Online]. Available: https://arxiv.org/abs/1603.06937