VizFit: Week 5 Milestone Documentation

Sasha Lawson Sam Donaldson
Alana Matheny Noah Buchanan

February 18, 2022

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

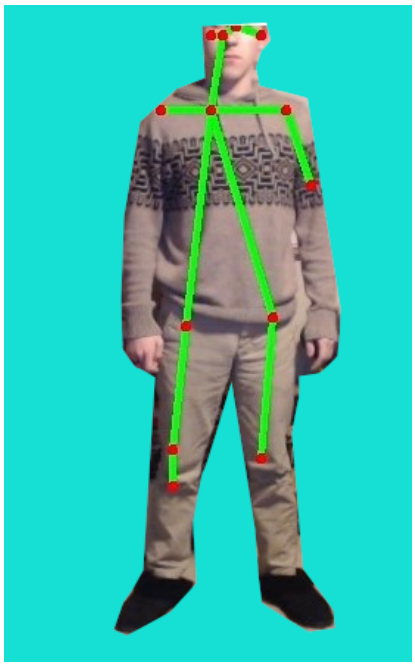
Contents

1 Summary Report

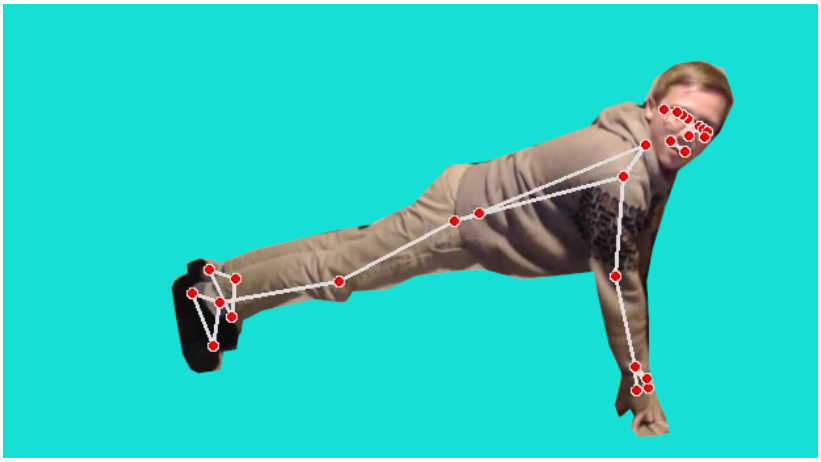
1.1 State of the Project

The state of the project so far can be seen through two lenses: back-end API and front-end Vue application connection and, more importantly, the machine learning models.

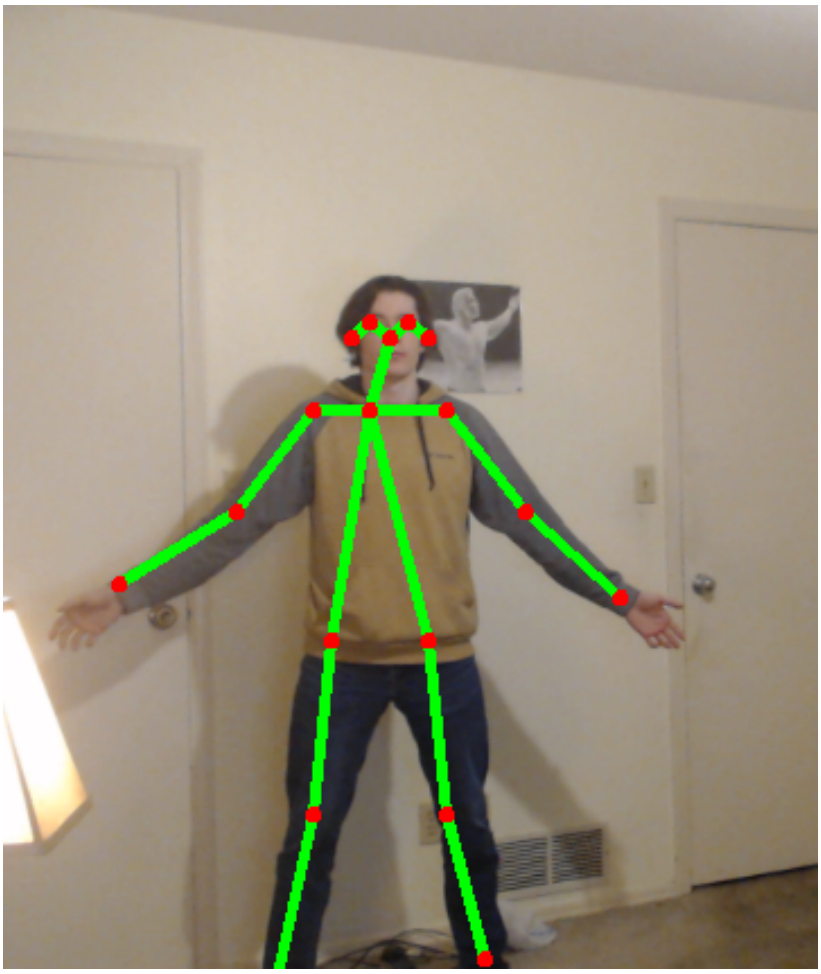
- **Back-End API:** Most of the project's first phase has been dedicated to building and researching machine learning models for wire-frame creation and exercise detection. However, an API controller template was created and using artificial methods a test was run to ensure that the controller could retrieve data from the front-end. This test was successful.
- **Machine Learning Models:** The initial phase of this project was based around research regarding standards in creating a wire-frame and exercising recognition model. With this information, new custom models will be built. These are the models that will be implemented into the overall architecture. However, the custom models will need to be and will be improved through the life of the project. In total, four models were created and/or tested. Two pre-built models and two custom models.
  - Pre-Built Model 1: This model used OpenPose; however, is believed to have been drastically altered because the wire-frame output is horrible and OpenPose normally would not produce such an output. However, this particular model was used as an example of what not to do and what to avoid. For example, the model was very inconsistent when the user was moving which is a big problem for an application that should recognize different patterns of movement. Additionally, it could not recognize profile views nor could it interpret hidden limbs which would cause issues regarding push-ups, for example. The following is an average wire-frame output. [1]



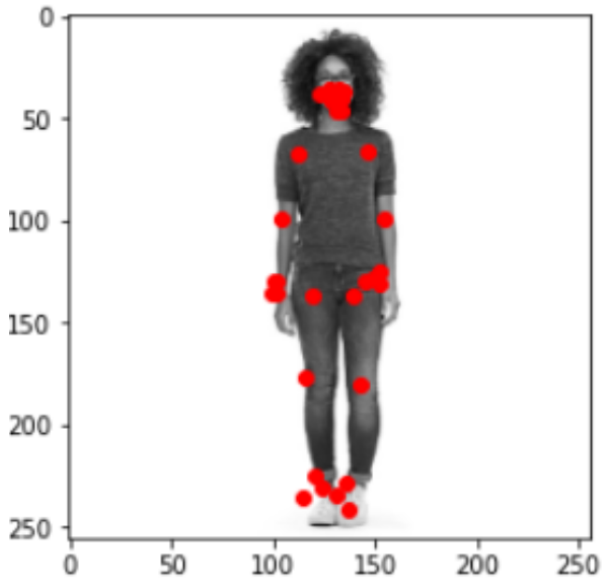
– Pre-Built Model 2: This model used MediaPipe, and unlike Model 1, was used as an example of what to do and what a good output wire-frame should be capable of doing. Features to highlight include, profile view recognition, substantially increased speed and accuracy, and was more reliable with movement. Model 2 could also interpret hidden limbs for the most part and design more appropriate wire-frame based on the hidden limbs. [2]



– Pre-Built Model 3: This model uses Google's MoveNet.SinglePose.Lighting and is highly modular and could use any Tensorflow model for pose estimation that follows the same format as the model used, which we plan to follow this format. The model itself is pre-built but we have added functionality to plot and connect the points that are the output of the model. This means that once we have our own Tensorflow convolutional neural network working we can simply swap out the model it is loading to our custom one given everything goes smoothly. The pre-built model itself here is not extremely impressive but it is a model mainly used for speed and as stated before it is a highly modular system and any Tensorflow model can be swapped in which is the main reason for its inclusion in the report. [3]



– Custom Model 1: The team's current approach to the first custom model is a very accurate 2D estimation for the wire-frame. Once the 2D estimation is to an acceptable degree, a 3D version will be implemented. Additionally, the current model is trained off a very small data set which has made it very apparent that finding an appropriate data set is priority number one. The following image is what the custom model output. The key-points of this model are based off of the pre-built model 2 using media pipe.



– Custom Model 2: The team's second custom model focuses on exercise detection and is still in a very experimental phase. However, the testing that has been done so far shows potential. For example, this model demonstrates excellent speed.

1.2 Tasks Completed

To summarize, the following are completed tasks.

- Created, revised, and presented the Project Proposal presentation.
- Begun research into wire-frames, exercise detection, and best data sets.
- Began working with multiple pre-trained models to get a feel of what best practices there are.
- Created first neural network model.
- Create back-end API controller template and confirmed front-end connection.
- Create and presented Week 5 Milestone presentation.

2 Technical Report

2.1 Individual Accomplishments & Performance

- Alana:
  - Created the 'Risk' and part of the 'Design' sections of the project proposal, along with the mock-ups of the application and website
  - Assisted in making the presentation of the project proposal
  - Created the UAFS informational website
  - Begun server and database setup
  - Researched exercise datasets
- Noah:
  - Found and tested Pre-Built Model 1
  - Found and added functionality to Pre-Built Model 3
  - Researched different metrics and methods of building wire-frame poses
  - Created the prototype version of our first custom model
  - Worked with Sasha on the Objective statement, created the Related Work section, and performed research to gain background knowledge in our database technology and in convolutional neural networks in our proposal
  - Aided the team in the creating our proposal presentation
- Sam:
  - Created API controller
  - Created Vue.JS project
  - Created prototype model for detecting pushups
  - Connected the Vue.JS project to the API via Axios
  - Researched exercise datasets
  - Created the architecture design for the proposal
- Sasha:
  - Created the Abstract, Problem Statement, portions of the Technology, and Tasks & Schedule in the proposal
  - Worked on the proposal and milestone presentations
  - Found and tested Pre-Built Model 2
  - Researched wire-frame and exercise detection models/methods in order to help develop the architecture of how our models interact

2.2 API Deliverables

- controller.java
- vizfitapiapplication.java

2.3 Machine Learning Model Deliverables

- Pre-Built Model 1 (OpenPose)
- Pre-Built Model 2 (MediaPipe)
- Pre-Built Model 3 (Tensorflow MoveNet Lightning)
- Custom Model 1 (Pose Estimation)
- Custom Model 2 (Exercise Recognition)

3 References

[1] Athtripathi, "Using openpose library in python with openpose tensorflow and opencv," Sep 2021. [Online]. Available: https://medium.com/geekculture/using-openpose-library-in-python-with-openpose-tensorflow-and-opencv-10a5496c359a

[2] Mhproductionhouse, "Latest pose estimation realtime (24 fps) using cpu — computer vision — opencv python 2021," Apr 2021. [Online]. Available: https://www.youtube.com/watch?v=brwgBf6VB0I

[3] "movenet/singlepose/lightning." [Online]. Available: https://tfhub.dev/google/movenet/singlepose/lightning/4