

Final Project

AI & Machine Learning

Arne Samson

3XD

Table of Contents

Introduction	2
Use cases	3
Personalized Pose Correction.....	3
Progress Tracking	3
Technologies used.....	4
PoseNet	4
Image Recognition	4
Teachable Machine.....	4
Training the model.....	5
TM for Pose Analysis	5
TM for Image Recognition	5
The Road of Development.....	6
1. Using the Yogism Feedbackdag Demo.....	6
2. Finding the right data	6
3. Training the model.....	7
4. Making the screens	8
5. Online Deployment.....	9
Conclusion	10

Introduction

Yogism – an innovative app that marries ancient practices with cutting-edge technology. Powered by TensorFlow, this app not only assesses your yoga poses in real-time through PoseNet but also identifies the pose you're performing using image recognition. In this paper, we will explore the various facets of Yogism, from its diverse use cases and the challenges encountered while integrating TensorFlow, to the carefully crafted styling of the app and the user experience across different screens. Join us as we delve into how Yogism elevates your yoga practice, addressing both the technical intricacies and the user-centric aspects that make this app a holistic companion on your path to well-being.

Use cases

Personalized Pose Correction

YogaMate can analyze users' real-time poses, offering instant feedback on their alignment and suggesting corrections. This use case is beneficial for practitioners looking to improve their form and maximize the benefits of each pose.

Progress Tracking

Users can track their yoga journey over time by recording and comparing their performance in different poses. Yogism could provide insights into their flexibility, balance, and overall progress, motivating them to achieve their fitness goals.

Technologies used

PoseNet

PoseNet is instrumental in Yogism's ability to analyze yoga poses in real-time. By tracking key points on the user's body, it provides instant feedback on alignment and balance during each pose, contributing to a more insightful practice.

Image Recognition

Yogism employs image recognition to identify the specific yoga pose being practiced. This technology enhances user understanding by providing information about each pose, facilitating a more informed and guided yoga journey.

Teachable Machine

Yogism leverages Teachable Machine, a user-friendly platform developed by Google, to train its machine learning models. This intuitive tool facilitates the training process for both pose analysis and image recognition, allowing the models to learn and adapt from diverse examples. Teachable Machine's accessibility enhances Yogism's capabilities, contributing to the app's commitment to providing accurate and personalized feedback during real-time yoga practice.

Training the model

TM for Pose Analysis

Yogism's pose analysis model underwent training on Teachable Machine using a dataset carefully crafted with images of different stages of pose execution. This dataset included examples ranging from fair to decent to excellent representations, captured collaboratively with my pookie wookie. The intuitive interface of Teachable Machine facilitated efficient model learning and adjustment.

TM for Image Recognition

For identifying specific yoga poses, Yogism harnessed the capabilities of Teachable Machine in training an image recognition model. The training dataset included images representing various yoga postures, ensuring the model's proficiency in accurately classifying poses during real-time usage.

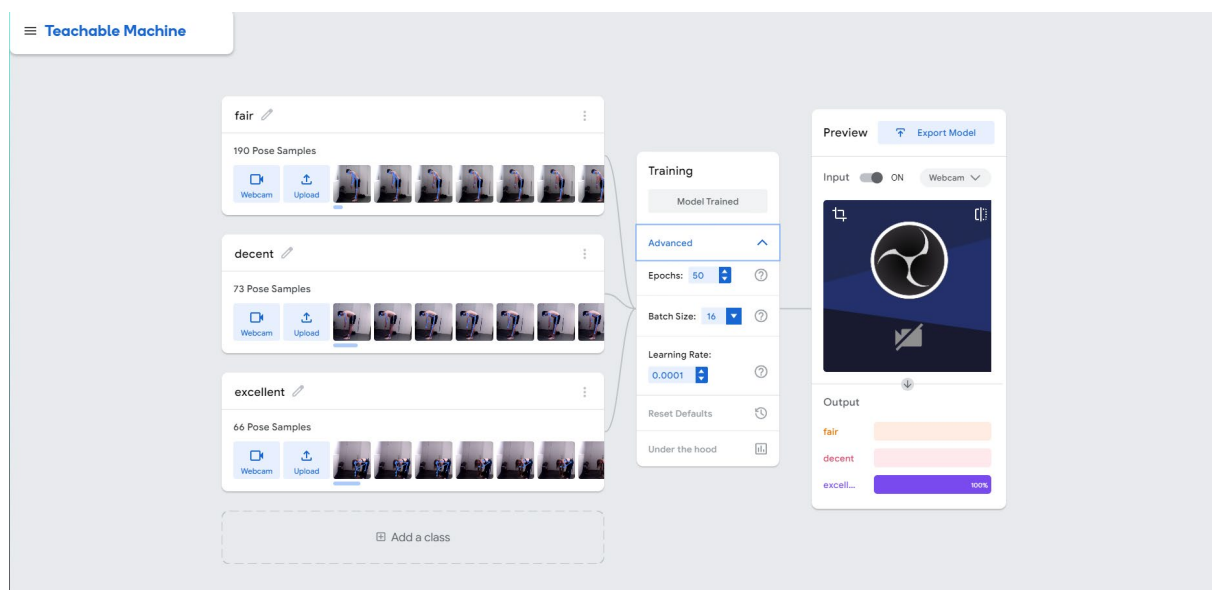
The Road of Development

1. Using the Yogism Feedbackdag Demo

For this demo I first went and grabbed our working demo for Lab3 which recognises how well you're performing a specific yoga pose. This uses PoseNet and is also trained via TM so it was a great starting platform.

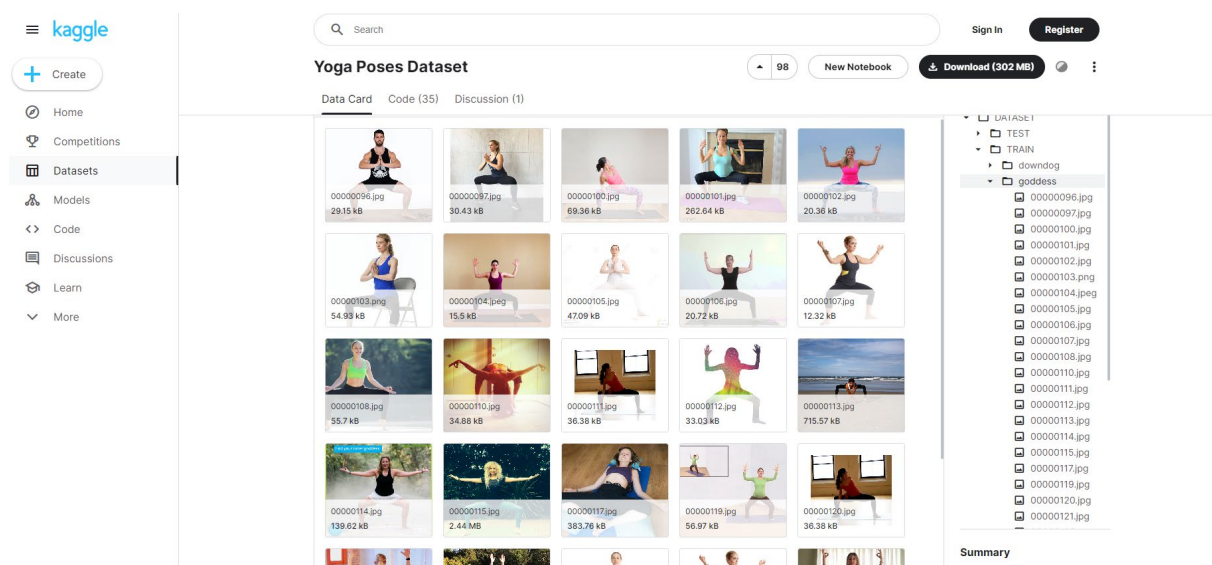
I first tried to optimize the pose detection because we already had to leave out 1 class because of the lack of a qualitative webcam and good lighting.

However this didn't really work because of a lack of quality data and time so I went for the image recognition route.



2. Finding the right data

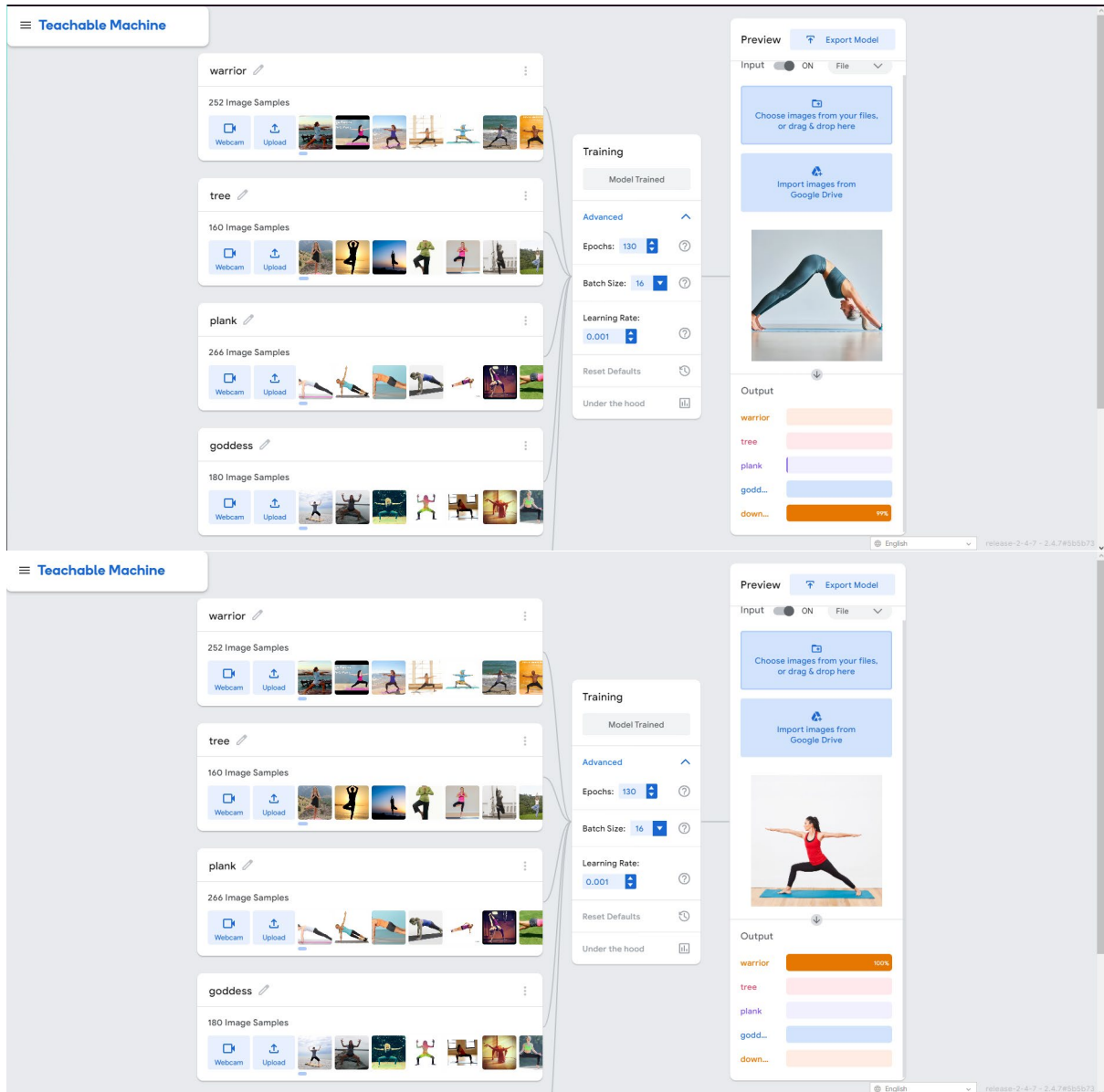
Evi told me about Kaggle, a site where you can find datasets. Here I [found this dataset](#) which has tons of pictures for 5 yoga poses.



3. Training the model

With a lot of data I went and trained a model with Teachable Machine.

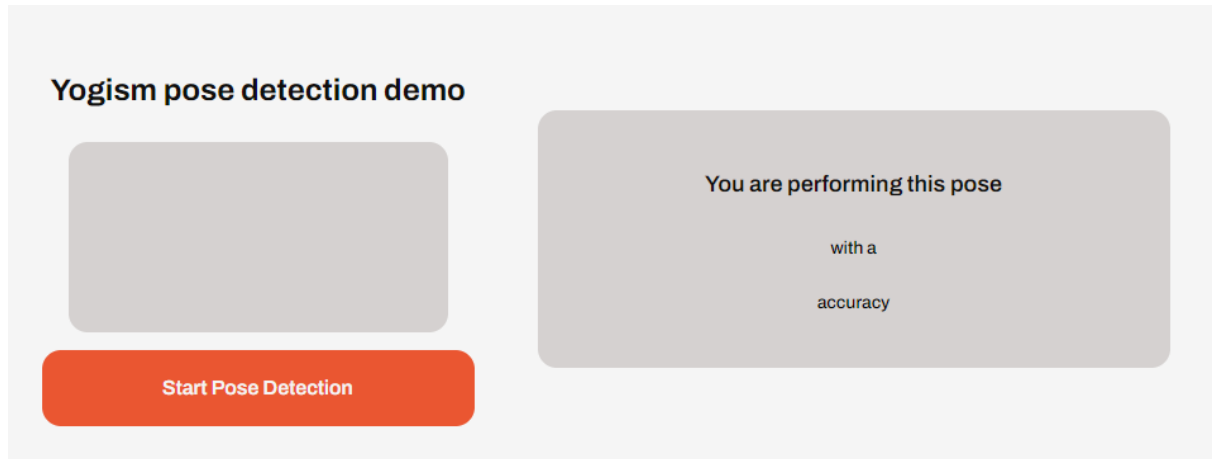
There are 5 poses and thus 5 classes. I changed the epochs to 130 to get a more accurate result.



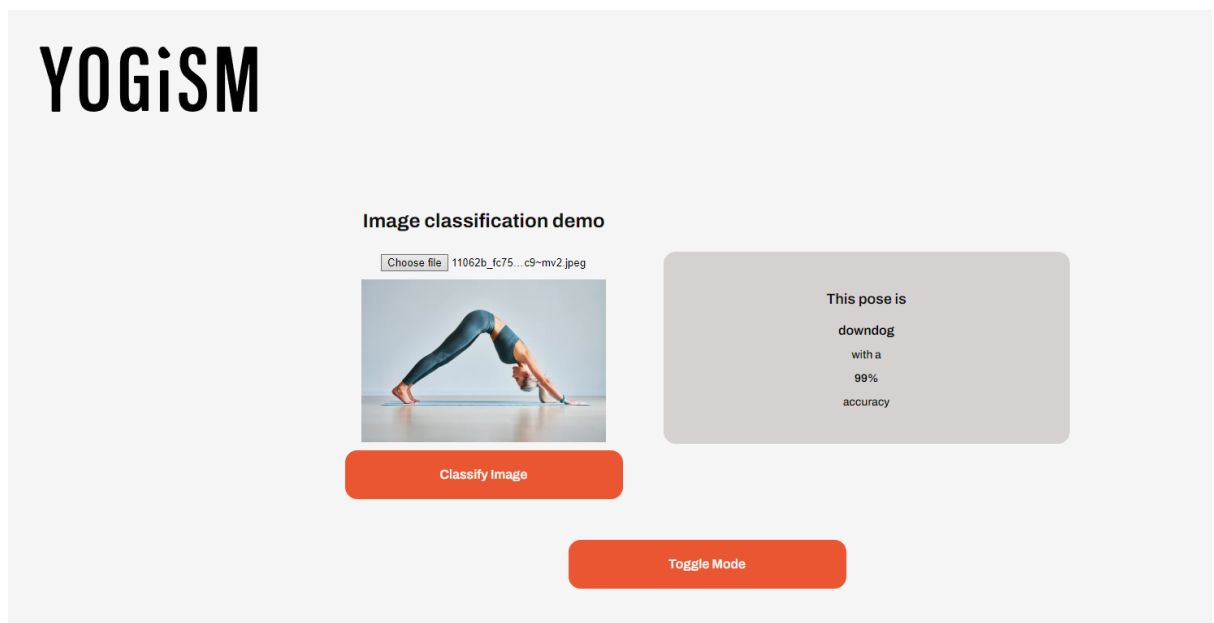
4. Making the screens

Next I had to make the webpage to use the model. This was fairly easy because I already had a basis with the existing Yogism demo.

I just made a toggle for 2 screens, one for poses and one for images.



(started with this)



(ended with this)

5. Online Deployment

I also deployed my GIT repo on Vercel. This was mainly so other people with webcams could test. You can check it out via [this link](#).

Do note. You have to reload the page if you want to use either of the modes.

(reload → use pose, or, reload → use image)

Conclusion

If I had more time I would have like to actually perfect the pose detection.

Anyways, Yogism is a fusion of TensorFlow's machine learning capabilities, specifically through PoseNet and image recognition trained on Teachable Machine, empowers Yogism to provide users with real-time pose analysis and identification. The project not only showcases technical proficiency but also reflects a commitment to enhancing the holistic well-being of individuals through the innovative use of AI.

Skills:

- **Machine Learning Expertise:** The project demonstrates a solid grasp of machine learning concepts, particularly in the implementation of TensorFlow for pose analysis and image recognition.
- **Data Collection and Collaboration:** The collaboration with a partner to collect diverse pose data showcases effective teamwork and the ability to curate datasets that capture variations in pose execution.
- **Problem Solving and Optimization:** Overcoming challenges in integrating TensorFlow into Yogism highlights problem-solving skills.
- **User Experience Design:** The attention to styling, user interface, and the development of different screens within Yogism reflects an understanding of user experience design principles.
- **Staying Conventional:** From the start I used conventions like: GIT for codebase, BEM for styling, vars in CSS and a well-structured HTML.

Yogism is not merely an app; it is a testament to the transformative potential of AI and machine learning in enhancing personal growth and mindfulness. The project showcases a diverse set of skills and the ability to address both technical and user-centric aspects, making it not only a valuable contribution to the intersection of technology and holistic wellness but also to myself as a web developer.

I believe this project shows well how I went from an innocent student who didn't even know what HTML is to a skilled Digital Experience Designer who uses AI and the latest web technologies to deliver awesome experiences.