YODA:

Yoga pOse Detection & Analysis

- Alexandra Savelieva
- Bryan Morgan
- Pritesh Patel

1.

Problem statement:

- Sedentary lifestyle
- Limited access to gyms
- Virtual sessions are challenging
- Unguided practice is risky

Yoga can help, but ...

6'x 2'2" x 1'4"
6'x 2'2" x 1'4"

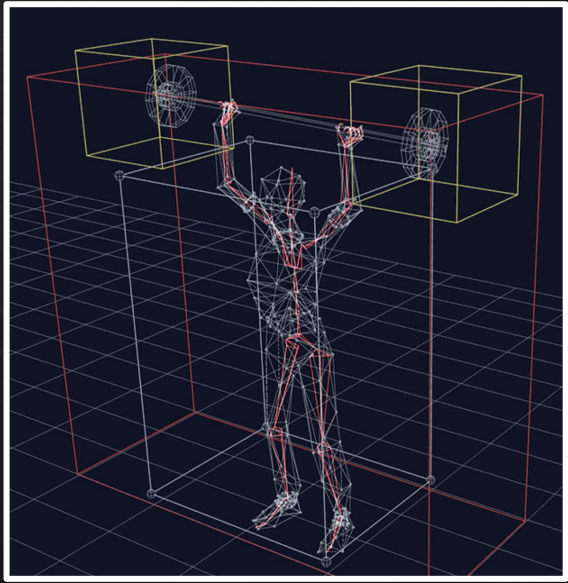
What's on the market

ally

- ✘ iPhone only
- ✘ No Yoga practice
- ✘ No live instructor support
- ✘ Sends video to the cloud

TEMPO

- ✘ Very expensive ($1995)
- ✘ Setup required ($250)
- ✘ Bulky (6'x 2'2" x 1'4")
- ✘ No Yoga practice



MIRROR

PELOTON

- ✘ Mirror ($1499 + setup, no real-time feedback)
- ✘ Peloton ($2499+, no real-time feedback)

Proposal

- A privacy-preserving solution that facilitates virtual guided yoga practice
- Inference about yoga poses on the device and / or in the cloud
- Provide feedback to participants
  - Real-time (correction of poses)
  - Trends (FitBit-like approach, except that we track how many asanas and how well you did, progress, day challenge, etc.)
- Provide feedback to the instructor (on individual students and the group)
  - Real- time (are people following or out of sync?)
  - Trends (how many participants follow, do they improve week to week, etc.)

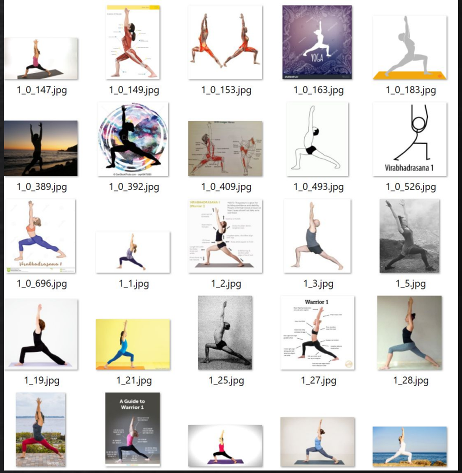
2.

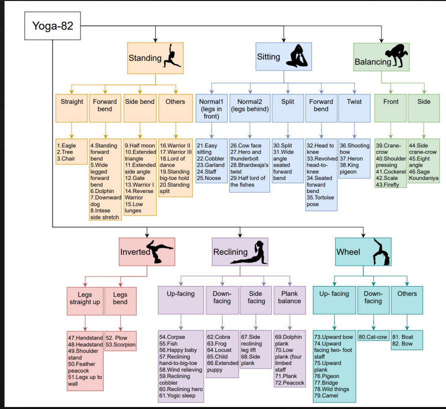
DATASET

Collection, exploration, curation, preprocessing

Exploratory data analysis of Yoga-82 dataset

- Tried out Cobra, Dolphin, Warrior 1
  - (1121, 113, 293 samples)
- Data quality:
  - ~7.5% URLs are unavailable
  - ~11% images are corrupted
  - ~20% images are drawings, not photos
  - ~7% are repeats (in different zoom / cropped /etc.)
  - Other issues (wrongly labeled, junk, poorly executed, variations, text, multi-person, mostly female)
- Down to [at best] 915, 75, 242 samples for each pose respectively (<80%)





DATASET

- Public datasets
  - Yoga-Poses-Dataset
  - Yoga 82
- Public datasets were very limited and mostly stock images so did not provide enough data to effectively train a model to do inference on live stream video
- Created our own datasets to augment the public images by capturing frames from live feeds of volunteers performing different yoga poses.
- Created 13000+ images in addition to public dataset of ~3000

Dataset we Created (13,000+ Labelled images)

10 participants (primarily seasoned yoga practitioners and certified trainers), videos recorded from a variety of devices and perspectives, 12 yoga poses: cobra, tree, cat+cow, downward dog, mountain, lotus, side plank, eagle, scale, warrior 1, warrior 2

Main Goals

On the Bleeding Edge

- ✘ Run without internet
- ✘ Privacy preserving
- ✘ Can be used in almost any home environment

User Centered

- ✘ Near real time
- ✘ Direct user feedback
- ✘ Able to view % probability of categorized pose
- ✘ Able to view skeleton used for classification

Not just for Kids

- ✘ Work on different people
- ✘ Work for different genders
- ✘ Work for different body types

Design & Experiments

	Approach 1	Approach 2	Approach 3
Training	Cloud	Cloud, Edge	Edge
Inference	Cloud	Edge	Edge
Video stream	Edge	Edge	Edge

3.

APPROACH 1:

YODA pose detection on the edge and in the cloud with OpenPose

YODA using OpenPose in the Cloud and at the Edge

- OpenPose showed better results than PoseNet on our benchmarking experiment
- Action detection on top of pose detection: sliding window of 5 frames
- Extended https://github.com/felixchenfy/Realtime-Action-Recognition
- Ran multiple classifiers on the keypoint data and picked best (DNN with 3 layers, 50x50x50), inference classified at probability of > 85%

Cloud + Edge Solution

Edge: capture video;
Cloud: training & detection

- Leverage OpenPose cmu (accurate, but slower).
- Images size: 656x368
- Data set: >3800 images for 4 classes
- Training time: ≅ several minutes
- Average frame rate: 25-30fps

Goals

- Evaluate feasibility of real-time yoga pose detection with OpenPose
- Evaluate accuracy of running on the edge w/ first solution
- Identify limitations of the approach
- Identify problems with data feed

DEMO 1

```
Accuracy on testing set is 0.9426229508196722
Accuracy report:
              precision    recall  f1-score   support

       eagle       0.92      0.88      0.90       148
       lotus       1.00      1.00      1.00       202
   sideplank       0.94      0.94      0.94       282
        tree       0.91      0.94      0.92       222

    accuracy                           0.94       854
   macro avg       0.94      0.94      0.94       854
weighted avg       0.94      0.94      0.94       854

Time cost for predicting one sample: 0.00001 seconds
```

Edge device: Lenovo ThinkPad X1 Yoga (4th Gen): CPU: Intel Core i7 | GPU: Intel UHD 620 | RAM: 16GB | Storage: 1TB | Display: 14-inch, 1080p

Cloud server: GPU-enabled (Gpu AC2.8x60, 8 vCPU, 60 GB RAM, 1 x V100 GPU) Gpu AC2.16x120

4.

APPROACH 2

On the Edge

On The Edge Solution

All Jetson, all the time

- Leverage OpenPose Mobile_Thin arch.
- Images size: 656x368
- Data set: >8500 images for 3 classes
- Training time:
  - ≅2 hours (edge)
  - ≅15 min (cloud V100)
- Average frame rate: 2.4fps

Goals

- Evaluate efficacy of local training
- See what is possible / not with local training
- See whether local updates are possible
- Evaluate efficacy of running on the edge w/ first solution

On The Edge Solution

5.

APPROACH 3

YODA pose detection at the Edge with trt_pose

YODA USING TRT_POSE ON JETSON

NVIDIA AI/IOT project to accelerate pyTorch Models using TensorRT (trt_pose)

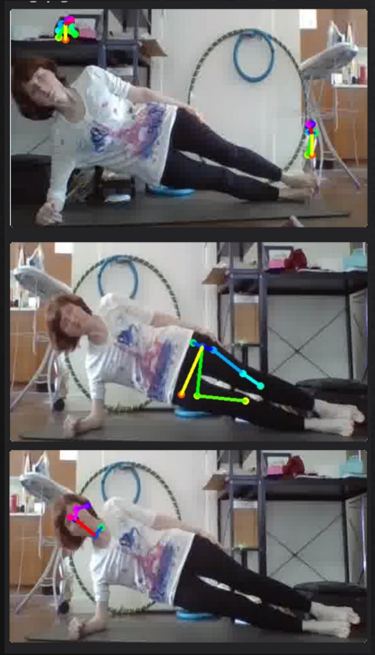
- ➔ Training via Microsoft COCO formatted keypoint data
  - ◆ Generated MS COCO keypoints from images and video frames for 3 poses (6500+ images @resolution 224x224 as required by Resnet18)
- ➔ Based on Resnet18-baseline model optimized to run on TensorRT (torch2trt)
- ➔ Ran multiple classifiers on the keypoint data and picked best (Random Forest)
- ➔ Inference classified at probability of > 60%

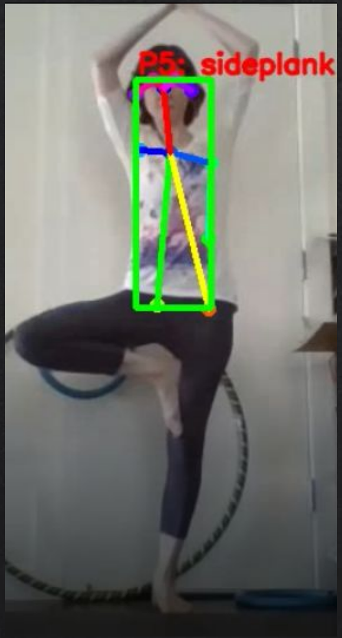
6.

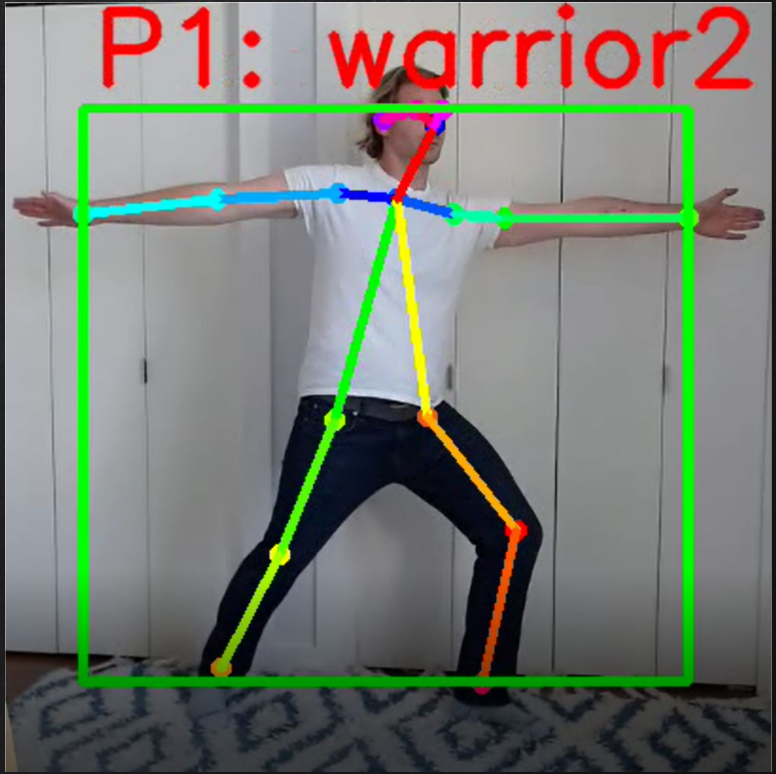
FUTURE WORK

NOT EVERYTHING WENT WELL…

- Alexandra: having fun with planks







- Bryan: invisibility socks

FUTURE WORK

Detection improvements

- ✘ Two-step classification
- ✘ Improved dataset
- ✘ Troubleshooting why some visible joints are consistently not detected
- ✘ Cleanup of "false positives" prior to training and in real time
- ✘ OpenPose skeleton + TRT FPS

Feature development

- ✘ D&I: bootstrapping the model for people with disabilities
- ✘ Evaluation of transitions between poses
- ✘ Support more poses
- ✘ Trends analysis (quality of poses in the morning vs. 8-hr work day)
- ✘ Emotion tracking

7.

SUMMARY

Main Goals

On the Bleeding Edge

- ✓ Run without internet
- ✓ Privacy preserving
- ✓ Can be used in almost any environment

User Centered

- ✓ Near real time
- ✓ Direct user feedback
- ✓ Able to view % probability of categorized pose
- ✓ Able to view skeleton used for classification

Not just for Kids

- ✓ Work on different people
- ✓ Work for different genders
- ✓ Work for different body types

Summary

Overview

- ✘ We were able to accomplish an MVP of all of our goals
- ✘ We were able to get solid + consistent results with different people
- ✘ We wished we could go further (but we'll keep going :-)))

What we learned

- ✘ Data quality is king/queen
- ✘ Just because you can train in the cloud doesn't mean it will run on the edge
- ✘ Considering human engagement is almost as important as the model
- ✘ Pushing the envelope often requires an ensemble of different techniques