Yoga Pose Classification

Yogi Bears

Margaret Nesi,
Madison Lease,
Whitney Hannallah

The Problem

01

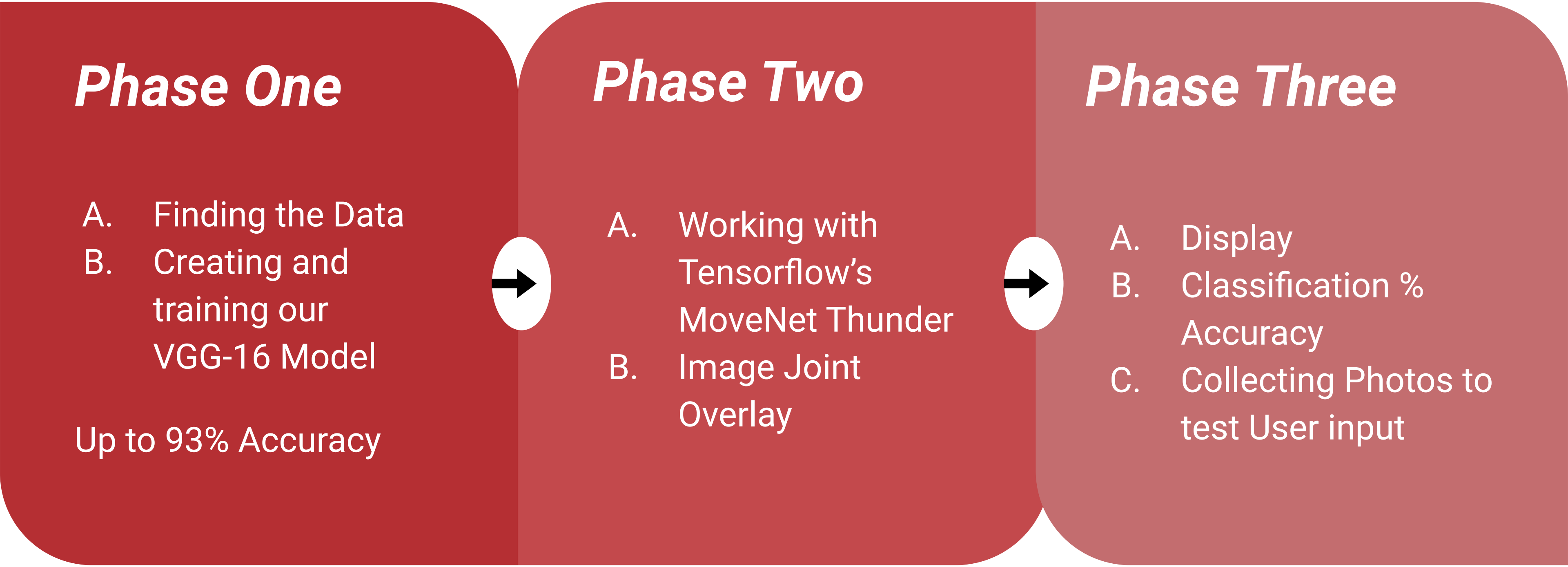
We are bad at yoga …

02

Expensive classes and memberships create a barrier for access

- Yoga classes, both online and in-person, studio memberships, and retreats are extremely expensive
- Yoga has many health benefits, if done correctly
- But without instruction, yogis risk injuring themselves

The Process



Phase One

A. Finding the Data

B. Creating and training our VGG-16 Model

Up to 93% Accuracy

Phase Two

A. Working with Tensorflow's MoveNet Thunder

B. Image Joint Overlay

Phase Three

A. Display

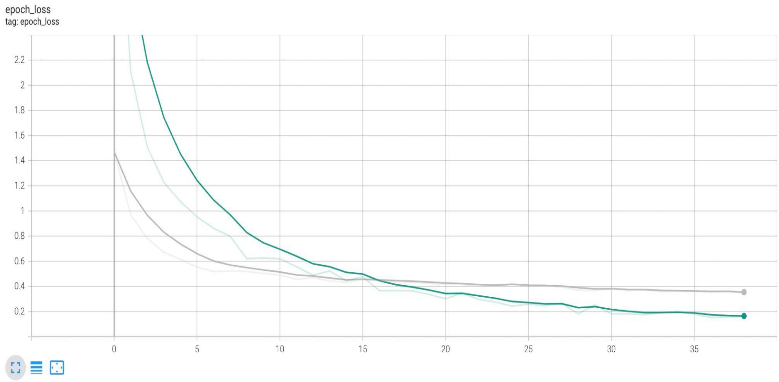
B. Classification % Accuracy

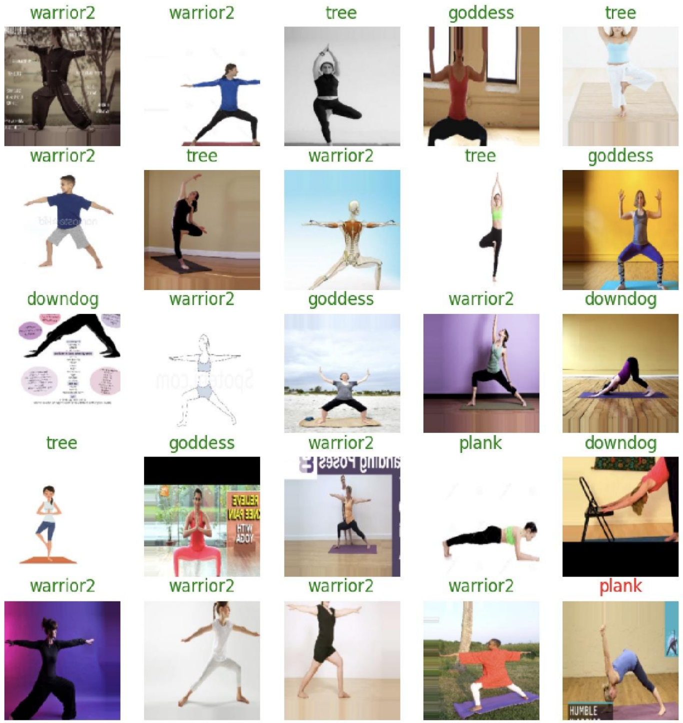
C. Collecting Photos to test User input

Phase One

Trained the VGG-16 CNN classifier (up to 90% accuracy)

Examined loss function and tensorboard output





Phase Two

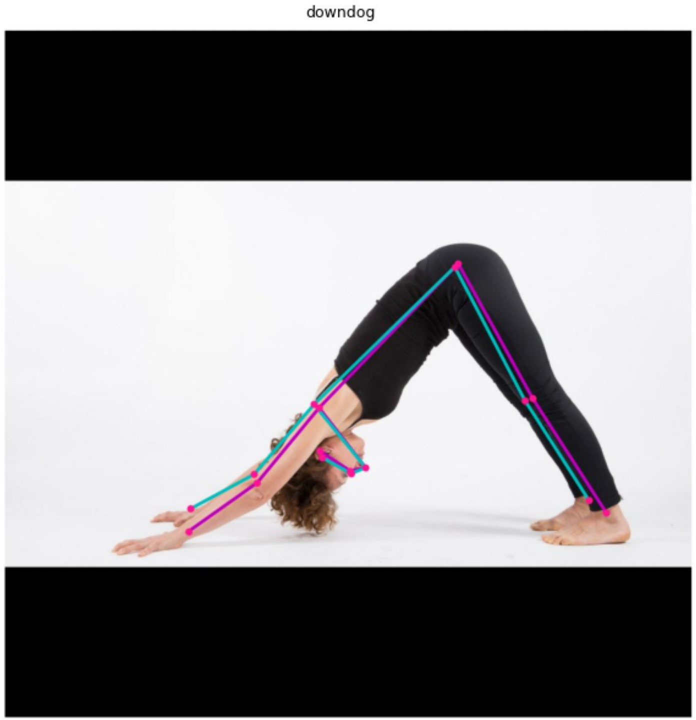


Image display in pop-up

Used MoveNet to find 17 “keypoints” of joints in a human body in an image

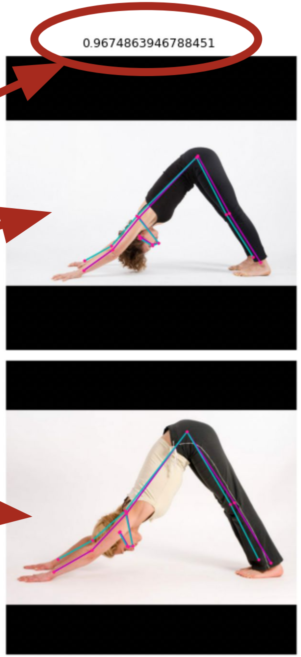
Developed a visual to overlay on the image that shows the connections between the 17 keypoints

Hard-coded the title and image path

Phase Three

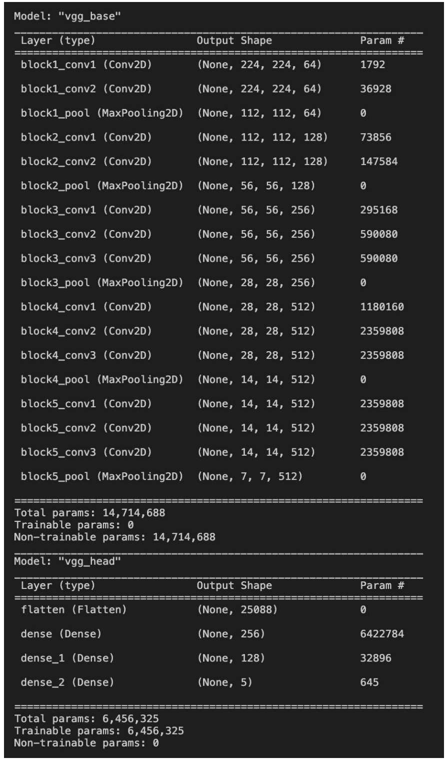
Calculated percent accuracy of user pose in input image from MoveNet score (displayed above image)

Displayed user image side by side with "perfect pose" both with MoveNet keypoints overlay

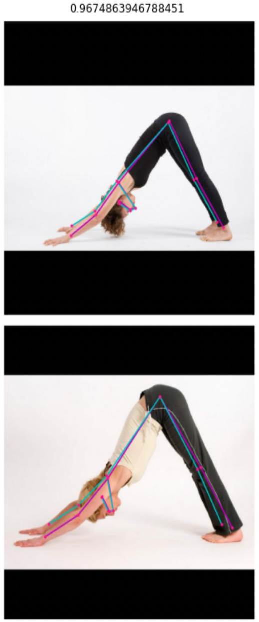


```
0.9674863946788451
```

Phase Three (cont.)



```
Model: "vgg_base"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
Model: "vgg_head"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 256)               6422784

 dense_1 (Dense)             (None, 128)               32896

 dense_2 (Dense)             (None, 5)                 645

=================================================================
Total params: 6,456,325
Trainable params: 6,456,325
Non-trainable params: 0
```



Turned run.py (which trains the CNN) into a class

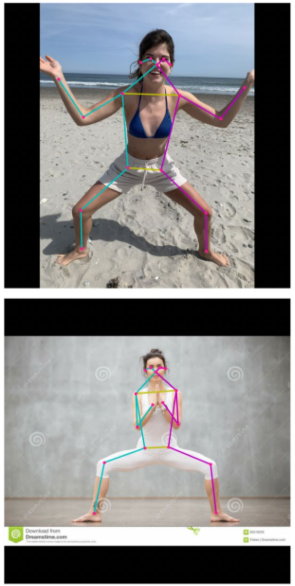
Running movenet.py now both:

A. trains the CNN

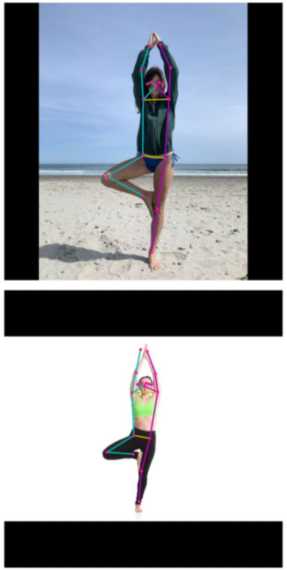
B. gives our popup window with overlay

The Final Result & What’s Next

What the user will see



Goddess Pose, Percent Accuracy:72.73775916914241



Tree Pose, Percent Accuracy:90.60936577298784

Remaining problem spots

1. Hardcoding the path to the user input image → create a GUI where user can input image

2. Accessing title from run.py

3. Play around with the % accuracy calculation