WORKOUT POSE DETECTION

Ankit Kalra, Senior Data Scientist, Fractal Anlaytics
Aakash Goel, Data Scientist, Fractal Analytics

OUTLINE

- What, why, Application
- Dataset Discussion
- Model Training
- Model Comparison
- Result
- Challenges and Future Scope

WHAT, WHY, APPLICATION

- Workout Pose Detection – Tracking User Movement, classify it corresponding to specific trained workout and determine whether performed pose correct or not.

- **Why**
  - Flexibility - Indoor or Outdoor Gym
  - Act as Personal fitness trainer
  - Wrong Postures during workouts are dangerous – can lead to permanent joint damages
  - One tenth the cost of what a fitness seeker pays a real-life fitness trainer
  - Members prefer workout regimens that are fun, customizable and social

- **Application**

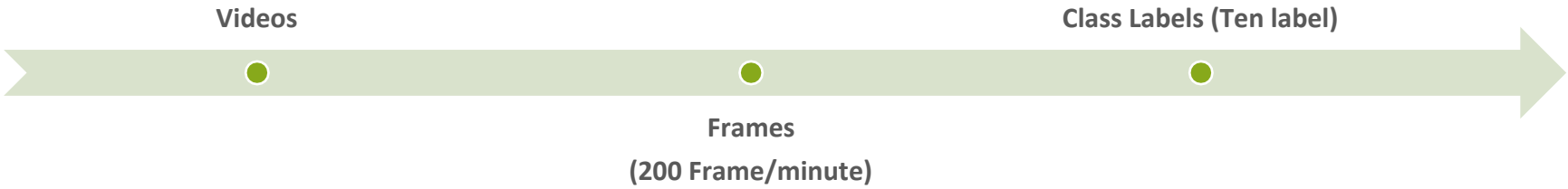
Intelligent conversational agent, requires speech recognition, voice recognition and NLP to interact with the exerciser in real time

Computer vision to do workout pose recognition (correct and wrong poses) by deep learning networks

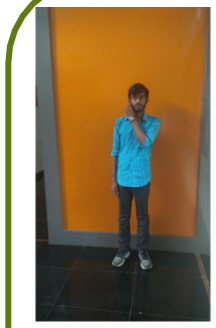
Intelligent conversational agent for pose correction in real time

ABOUT THE DATASET

- Total 40 videos of brain yoga workout captured containing workout poses performed by different individuals, recorded by mobile phone.
- Out of 40 videos, 20 videos contained only accurate poses and were recorded by changing the camera orientations – Front, Right and Left. The length of these initial set of videos were approximately 1 minute each.
- Data cycle



CORRECT POSES


Left hand on right ear – **Correct Pose I**


Cross folds hands with opposite ears; left hand below the right hand – **Correct Pose 2**


Cross folds hands with opposite ears; left hand below the right hand while sitting – **Correct Pose 3**

WRONG POSES


Right hand on right ear – **Wrong Pose I**


Right hand on left ear – **Wrong Pose 2**


Left hand on left ear– **Wrong Pose 3**


Both hands on respective ears– **Wrong Pose 4**


Cross folds hands with opposite ears; left hand above the right hand – **Wrong Pose 5**


Cross folds hands with opposite ears; left hand above the right hand while sitting – **Wrong Pose 6**


Both hands on respective ears while sitting– **Wrong Pose 7**

ABOUT THE DATASET - OPERATIONS

- **Image Augmentation**
  - Need because of variation in background
  - To increase dataset size
  - Techniques Used – Gaussian blur, Hue and saturation, color, Elastic Transformation

Before Augmentation– Correct Pose 2

Post Augmentation– Correct Pose 2

- **Person extraction from Frame**
  - To minimize background interferences while modelling

Real-time object detection system, prepares a bounding box around the image and predict the object

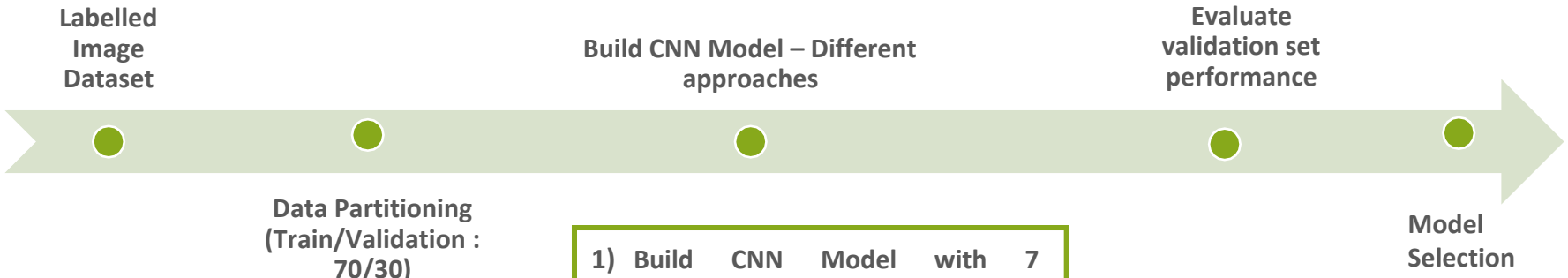
Open Source Computer Vision Library - an open source computer vision and machine learning software library

Open source framework built on top of Tensor Flow that makes it easy to construct, train and deploy object detection models

All the three techniques, helped us in extracting persons from frames, however ***Tensor flow API*** was found to be more robust and scalable for all ten different classes

MODEL TRAINING

We started exploring several approaches for building an accurate and robust model for ten class classification problem



1) Build CNN Model with 7 convoluted layer & 13 activation layers
2) Build CNN Model with 10 convoluted layer & 16 activation layers
3) Build CNN model using VGG-Net pre-trained weights
4) 1st model with different learning rates & batch sizes

CNN Model Architecture (7-layers)

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 200, 200)	896
activation_1 (Activation)	(None, 32, 200, 200)	0
conv2d_2 (Conv2D)	(None, 32, 198, 198)	9248
activation_2 (Activation)	(None, 32, 198, 198)	0
max_pooling2d_1 (MaxPooling2	(None, 32, 99, 99)	0
dropout_1 (Dropout)	(None, 32, 99, 99)	0
conv2d_3 (Conv2D)	(None, 64, 99, 99)	18496
activation_3 (Activation)	(None, 64, 99, 99)	0
conv2d_4 (Conv2D)	(None, 64, 97, 97)	36928
activation_4 (Activation)	(None, 64, 97, 97)	0
conv2d_5 (Conv2D)	(None, 64, 95, 95)	36928
activation_5 (Activation)	(None, 64, 95, 95)	0
conv2d_6 (Conv2D)	(None, 64, 93, 93)	36928
activation_6 (Activation)	(None, 64, 93, 93)	0
conv2d_7 (Conv2D)	(None, 64, 91, 91)	36928
activation_7 (Activation)	(None, 64, 91, 91)	0
max_pooling2d_2 (MaxPooling2	(None, 64, 45, 45)	0
activation_8 (Activation)	(None, 64, 45, 45)	0
max_pooling2d_3 (MaxPooling2	(None, 64, 22, 22)	0
activation_9 (Activation)	(None, 64, 22, 22)	0
dropout_2 (Dropout)	(None, 64, 22, 22)	0
max_pooling2d_4 (MaxPooling2	(None, 64, 11, 11)	0
activation_10 (Activation)	(None, 64, 11, 11)	0
max_pooling2d_5 (MaxPooling2	(None, 64, 5, 5)	0
activation_11 (Activation)	(None, 64, 5, 5)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 512)	819712
activation_12 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_13 (Activation)	(None, 10)	0

Total params: 1,001,194
Trainable params: 1,001,194
Non-trainable params: 0

MODEL COMPARISON

- Evaluating performances of above discussed Architectures that were optimized for different parameters

Attributes	MODEL I	MODEL II	MODEL III	MODEL IV
CNN Layers	7	10	19	7
Activation layers	13	16	-	13
Learning rate	0.0005	0.001	0.001	0.001
Batch Size	32	32	32	64
Pretrained weights	NA	NA	VGG-Net	NA
Steps/epoch	1000	1000	1000	1000
No. of epochs	12	21	21	6
Validation accuracy	80%	60%	60%	91%

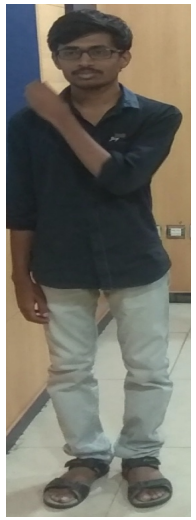
- Model 4 seems to work best.

RESULT

- Identifying correct pose with Accuracy 90 %

INPUT - TEST VIDEO 1 MINUTE

OUTPUT - FRAMES WITH PROBABILITY SCORES



5012_0_pose 1_0.5779795

5079_2_pose 2_0.83053553

5101_0_pose 3_0.9778807

5101_1_pose 3_0.99151784

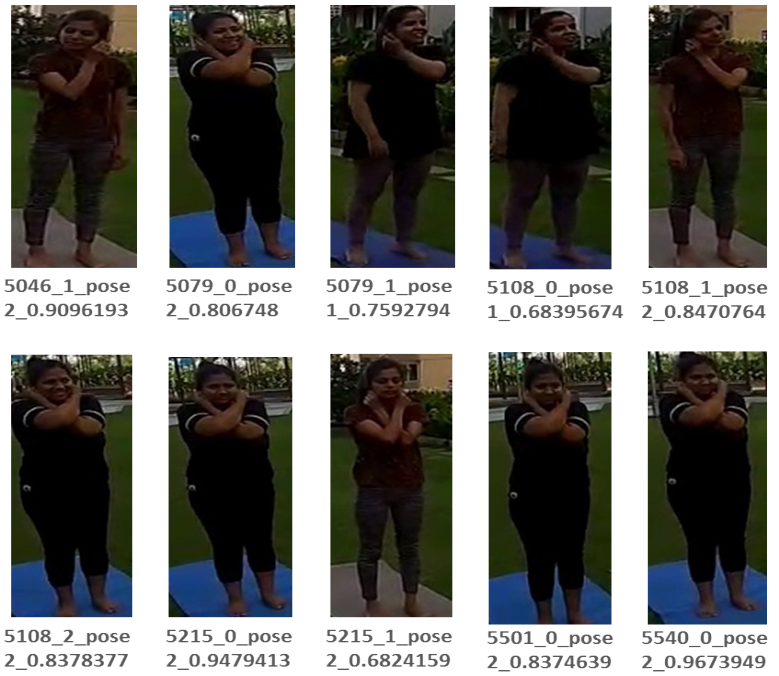
Accuracy of the test 90%

RESULT CONTD..

- Identifying correct pose with Accuracy 80 %

INPUT - TEST VIDEO 1 MINUTE 17 SEC.

OUTPUT - FRAMES WITH PROBABILITY SCORES



Accuracy of the test 80%

The low test accuracy was the result of different camera orientations (model not trained on different orientations based on limited dataset).

RESULT CONTD..

- Identifying incorrect pose

5570_0_pose 4_0.7579795

CHALLENGES & FUTURE SCOPE

- **Challenge**
  - Model will not perform good if videos or frames are captured at different orientations and if person is too close or far in frames because model is not trained on certain type of distribution of images. So, need more images with different person, different angles and person at different depths.
  - Data Preparation – Labelling the brain yoga dataset finally into ten class labels was the biggest challenge as the number of wrong poses can be done in different combinations in real-time scenarios.

- **Future Scope**
  - Leverage Mask R-CNN to avoid person extraction from frame.
  - For proper cropping of person from images, using TensorFlow object detection API and giving accuracy of more than 95%. This can be further improved by using TensorFlow object detection API for custom object detection with label marked down as person.