Minor Project Report on

AI TRAINER: WORKOUT POSTURE EVALUATION USING POSENET FOLLOWED BY DNN

Yuvraj Dalia (171IT153)

Navnirman Singh (171IT126)

Akash Rao (171IT205)

Under the Guidance of,

Dr. Sowmya Kamath S

Department of Information Technology, NITK Surathkal

Date of Submission: June 22, 2020

in partial fulfillment for the award of the degree

of

Bachelor of Technology

In

Information Technology

At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

June 2020

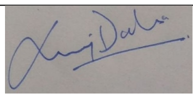
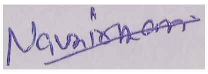
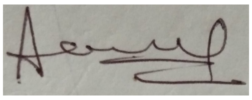
Department of Information Technology, NITK Surathkal
Minor Project
End Semester Evaluation Report (June 2020)

Course Code : IT 399

Course Title: Minor Project

Project Title: *AI Trainer: Workout Posture Evaluation using PoseNet followed by DNN*

Project Group:

Name of the Student	Register No.	Signature
Yuvraj Dalia	171IT153	
Navnirman Singh	171IT126	
Akash Rao	171IT205	

Place:

Date: *(Name and Signature of Minor Project Guide)*

Abstract

In today's day to day world, most of the people have sitting jobs. They don't do much of any movement. And being fit physically very necessary to stay fit and healthy. And to be so we need to exercise. But if we do an exercise in a wrong manner or in a bad posture it can have severe effect on our health and we might need to visit a physiotherapist.

In this project, we propose to develop a system which can observe our body movements when we do a particular exercise and notify us whenever the posture or form of us doing an exercise is not correct. We have developed a system using various computer vision and deep learning techniques to identify the pose in which person is doing a particular exercise. If posture is found to be wrong, it will notify us and we can work on our posture or form. It will also help us save money which we may have to spend on hiring a personal trainer. For the research purpose, we are doing the analysis for one exercise that is squat but the project can be extended to other exercises based on available datasets.

We will be using various Deep Learning architectures such as CNN, DNN , Pose Net etc. We will be constructing an interconnected model in which output of one model will be given as input to another model.

Keywords: *DNN, Pose Net, Open Pose, Body Key Points, posture etc.*

Contents

List of Figures

1 Introduction

These days, to stay fit, a lot of people do various kind of physical workouts such as going to gym, cross-fit, yoga etc. These kind of workouts help people stay fit and as we know a healthy mind rests in a healthy body, healthy physical conditions also lead to healthy body. But when you are doing these kind of exercises, one needs to make sure that, his/her form/posture is correct. If it is not correct, it may lead to several health disorders such as depression and stress, digestive issues, back shoulder and neck pains, tension headaches etc. An example of doing correct and incorrect form of squat is given in Figure 1.1.

Figure 1.1: Example of correct and incorrect form of doing a squat

So we propose to develop a system which will observe the movements of user and notify the user whenever user does an exercise in a wrong from/posture. To develop such a kind of system, we first need to develop a technique which helps in posture recognition of the user from the images captured from the video stream.With the more advances in Machine learning and Deep Learning techniques, such a task can be accomplished by using a CNN or other techniques used for human pose estimation. And for this particular task we use an architecture for body posture detection called PoseNet. We use outputs of PoseNet which are key body points (a list of 17 different locations) and further feed them to a neural network which learns a complex function on these body points to classify the posture of the exercise as good or bad.

So the model pipeline is as follow :

- Feed an image to the model of a person doing an exercise.
- The image is first processed by a model based on PoseNet architecture and an output of various body keypoints is processed by the model.

- These keypoints are further fed to a model which learns a complex function to associate these points with each other for correct and incorrect postures.
- The output of is given as posture is good or posture is based based on input given to second neural network.

Since there is no proper dataset available of correct and wrong poses of doing exercise, we have created our own dataset which comprises of 119 good posture images of doing a squat and 126 bad posture images of doing squat. The pictures are of different persons including male and females. There pictures in indoor environment and outdoor environment. And various other factors which add diversity to the dataset. In this project we focus on only one exercise as of now which is squat. Similarly it can be applied to other exercises after creating datasets for them too.

1.1 Motivation

- To explore the use of Neural networks for workout posture assistance.
- To help people promote health lifestyle by working out daily in a good posture to reduce bad effects on health.
- To eliminate the need of an actual trainer for workout assistance.
- Make the model highly practical in real life.

2 Literature Survey

2.1 *Related Work*

As of now, there is not much work done around recognizing the posture of a workout exercise as a good posture of bad posture, but there has been some significant work done around human activity recognition and also sitting posture classification as good or bad. Regarding the human pose estimation, there are a number of neural networks architectures which are developed in past that work with high accuracy in localizing key body points. We will talk about all those works in this section.

Islam at el (2017) came up with a technique of yoga posture recognition. In this project, authors used microsoft kinect to get skeletal model of human body and then extracted various body key points. Then they extracted various angles between those body points to recognize that particular yoga posture [1]. Khelalef at el (2019) did some work to to recognize human activities based on deep learning. In this project, authors took a video of some human activity. Divided it into frames and converted each frame to binary motion image using gaussian mixture models. Then they fed these these frames converted to binary motion images to an LSTM for classifying the human activity such as running,cycling etc [2].

Win at el (2020) worked on a project for human motion tracking and activity recognition. Authors categorized activities as running, walking, sitting and standing. A video is processed to extract out frames at various time stamps. Human skeletal structure is extracted out and depending upon how the movement of that structure is happening in the frame (eg: if the co-ordinates are changing very fast, slow or remain same throughout), the activity is classified as running, standing etc [3].

Boualia at el (2019) gave an overview of various approaches pose-based human activity recognition approaches. These included low level, mid level and high level approaches. Authors walked through various HAR body feature representation techniques and how these features can be processed by classical techniques and deep learning architectures.[4]

Jiang at el(2015) developed a technique for recognizing human activities such as jogging, walking,clapping etc. Authors took video frames and extracted features from them. These features included body points, body sclae, intensity histogram etc. Then using SVM classifiers they classified the input features to various activities. [5]

Sehgal at el (2018) developed an activity recognition system which uses HOG features. HOG is histogram of oriented gradients which is a local feature descriptor for activity recognition. After that, these features are fed to back propagation neural network for activity classification [6].

Chen at el (2020) developed at deep learning technique for sensor based human activity recognition. They used wearable senors such as gyroscope, accelerometer,EMG,ECG and ambient sensors such as wifi,RFID to get the data. Then this data was processed using deep learning architectures such as LSTM's and CNN's for classifying the activity [7].

Yu at el (2020) came up with a technique that uses various inter and intra joints features for recognizing human activities such as comb hair, drink water, wipe table etc. Inter joint features include joint position distance, angle between 3 joints, velocity of angles and acceleration of angles. Intra joint features include velocity in 3D space, speed, acceleration and muscle force. These features are further processed fed to deep learning models to recognize activities [8].

Attal at el (2015) developed a technique using wearable sensors for physical human activity recognition. They used data from various wearable sensors for computing features and done did principal component analysis for feature extraction. They used various machine learning models such as random forest, K nearest neighbours, SVM and markov chains for activity classification [9].

Papandreou at el(2018) came up with a state-of-the-art approach person pose estimation. This technique uses CNN's are various stages to produce heatmaps of the possibility of the presence of a body key point. Then these heatmaps are further processed using interrelated features to get final body key points and estimate the perosn pose according to them [10].

Chen at el (2017) developed a method for pose estimation based on PoseNet. It used three interlinked sub-methods to predict the final output of body key points.The three methods were Multi-Task Generative Network, Pose Discriminator and Confidence Discriminator. The ouput from the final model is position of body key points [11].

Papandreou at el (2017) developed a system for multi-person pose estimation. The technique used was very similar to [10] but it had the advantage of detecting multiple persons in the frame and then calculate body key points for those persons [12].

2.2 *Outcome of Literature Survey*

There are various techniques available for human activity recognition which uses wearable sensors and microsoft kinect. But very are also there which uses direct output from a camera. In many techniques, the data received from sensors or camera is preprocess into a structured numerical data which fed to machine learning models and deep learning models.

Most of the work according to the literature survey if focused on human activity recognition which recognize human activities such as running, drinking, yoga postures classification but negligible work has been done around classifying posture for an exercise as good or bad.

2.3 *Problem Statement*

Make an Intelligent System which takes input as an image of a person doing an exercise (in our case, a squat exercise) and output the posture of the person as good or bad for that particular exercise using Deep Learning techniques and try to achieve best possible accuracy. Also construct a dataset for consisting of good and bad posture images for training the model.

2.4 *Objectives*

1. Construct a dataset for consisting of good and bad posture images for training the model.

2. Design a model which takes input image and returns various body key points.

3. Design a model which inputs body key points and classifies the posture in image of doing a particular exercise good or bad.

4. Integrate the above two models into one final model.

5. Performance evaluation of the runtime of the model as least and accuracy as the highest.

3 Methodology

For the purpose of this project, we are considering squat as an exercise for which we will build a model to classify the posture as good or bad.

We have seen that most of the work which is done for activity recognition is done using Microsoft Kinect or using wearable sensors. The technique which we will be using will only require an image which can be taken from a smart phone camera also. This will prevent any additional costs for practical application for this project.

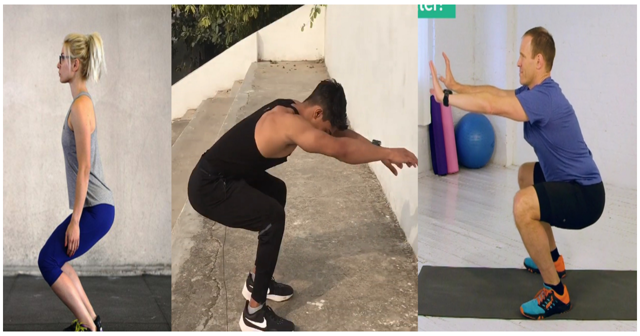


Figure 3.1: Left to right(back too high, back too low,just perfect)

If we talk about correct form of doing a squat, when doing a squat, the thighs should come parallel to the floor. Our back should slightly bent forwards. It is like sitting on an invisible chair. Our back should not be too straight and it should not be too bent forwards. In Fig 3.1, it can be seen that (from left to right), the back is too straight, back is too much bent and just perfect.

Since there is no dataset available for good and bad posture of doing a squat, we have created our own dataset for 238 images from various youtube videos and internet sources. We have tried to make this dataset as much diverse as possible about which we have talked in introduction section.

The deep learning model which we purpose is divided into two parts.

3.1 Part 1

The model in part one takes image as input and gives out 17 body key points namely nose, left eye, right eye, left eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle and right ankle in the same order. This model is based on the PoseNet architecture also called as Open Pose [13].. The architecture of this model is given in Fig 3.2.

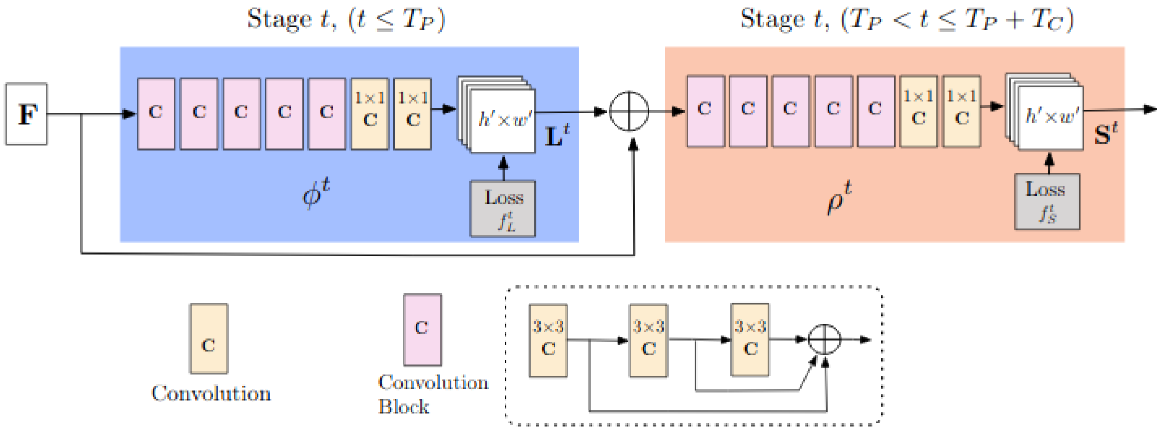


Figure 3.2: Openpose architecture taken from [2]

In this at the stage 1, part affinity fields are given as outputs. Part affinity fields (PAF) represent the relationship of various body parts with each other. It gives information about how two body key points like elbow and shoulder form a limb i.e Upper arm. So all the pixels which are on limbs have high confidence value in PAF maps.

In stage 2 which takes input as PAF image maps generated in previous stage as well as the original image, it generates confidence maps as output which is confidence as presence of a particular body part at a given pixel. From this, for a particular body part, the pixel which has highest confidence value in an image is labeled as that particular body key point.

Both stage 1 and stage 2 are deep convolutional neural networks which include several layers of convolutional filters for extracting the features out from input layers.

17 confidence maps are created for all 17 body key points and the pixel which has highest value in each of the 17 confidence maps is labelled as body key-point for that particular confidence map.

The generated outputs for two of the input images can be seen is Fig 3.3.

Also the output of the x and y coordinates of body key points for the two images is given in Fig 3.4.

These points are in the order (nose, left eye, right eye, left eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle and right ankle) from start to end.

For the input to model in part 2, we only take the body key points left shoulder, right

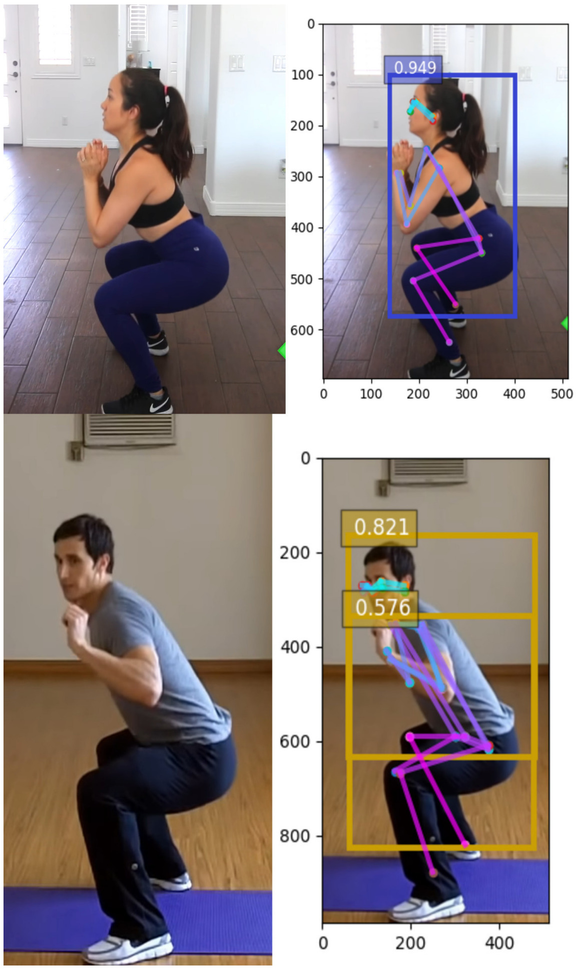


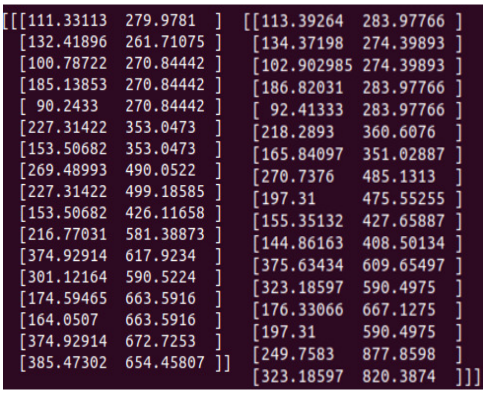
Figure 3.3: Input and output images for Model 1 which is based on PoseNet architecture from OpenPose paper

shoulder, left hip, right hip, left knee, right knee, left ankle and right ankle as only these are involved for if checking the posture of the squat is good or bad.

3.2 Part 2

The output given by model in part 1 are 8 body key points and each body point includes x and y co ordinates, hence total 16 values. So, the input to this model will be those 16 values. The architecture of this model can be seen in Fig 3.5.

So after giving the 16 points (left shoulder (x), left shoulder (y), right shoulder (x), right shoulder (y), left hip(x), left hip(y), right hip (x), right hip (y), left knee (x), left knee (y), right knee (x), right knee (y), left ankle (x), left ankle (y), right ankle (x) and right ankle (y)) input to model 2, it passes through a deep neural network (DNN) which gives output Y. If Y is equal to 1, it means the posture was good and if output is 0, it



```
[[[111.33113  279.9781  ]  [[113.39264  283.97766 ]
  [132.41896  261.71075 ]   [134.37198  274.39893 ]
  [100.78722  270.84442 ]   [102.902985 274.39893 ]
  [185.13853  270.84442 ]   [186.82031  283.97766 ]
  [ 90.2433   270.84442 ]   [ 92.41333  283.97766 ]
  [227.31422  353.0473  ]   [218.2893   360.6076  ]
  [153.50682  353.0473  ]   [165.84097  351.02887 ]
  [269.48993  490.0522  ]   [270.7376   485.1313  ]
  [227.31422  499.18585 ]   [197.31     475.55255 ]
  [153.50682  426.11658 ]   [155.35132  427.65887 ]
  [216.77031  581.38873 ]   [144.86163  408.50134 ]
  [374.92914  617.9234  ]   [375.63434  609.65497 ]
  [301.12164  590.5224  ]   [323.18597  590.4975  ]
  [174.59465  663.5916  ]   [176.33066  667.1275  ]
  [164.0507   663.5916  ]   [197.31     590.4975  ]
  [374.92914  672.7253  ]   [249.7583   877.8598  ]
  [385.47302  654.45807 ]]  [323.18597  820.3874  ]]]
```

Figure 3.4: Output Co-ordinates from Model 1 which is based on PoseNet architecture from OpenPose paper [13]

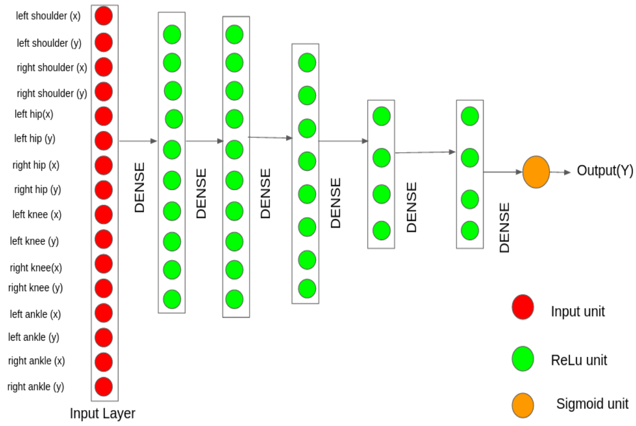


Figure 3.5: Architecture of DNN in part 2 model.

means the posture was bad.

This architecture has one input layer, five hidden layers and one output layer. In the hidden layers, first two hidden layers have 10 hidden units each, next one layers has 8 hidden units and the next two layers have 4 hidden units each. This architecture is developed after trying out various architectures and this one gave best results.

The model tries to learn the association between these three points and learns what should be the association between these 16 points if the posture is good. And if the association deviates from the perfect association is classifies the posture as bad.

So the architecture of overall final model which is combination of models in Part 1 and and Part 2 can be seen in Fig 3.6.

Hence in our final model, the whole process goes like following:

1. We feed an image of a person doing a squat to the the model part 1 (Open Pose).

2. It extracts the 8 body key points which are involved in maintaining a good posture while doing the squat. So we are given total 16 values (8 x coordinates and 8 y coordinates).

3. These 16 points are fed to the model part 2 (DNN).

4. The DNN gives output as 0 or 1. Zero means the form is bad, 1 means the form is good.

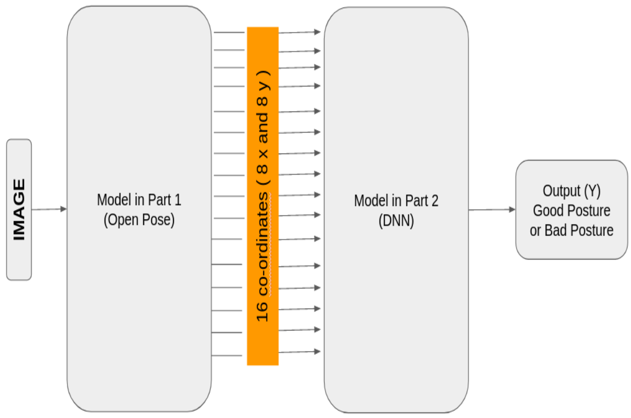


Figure 3.6: Architecture of the overall model

Hence, we can see how much healthcare benefits can be taken from this model if are doing work out as bad posture while doing work outs can lead to some serious health problems. Also this saves us from hiring a personal trainer to work on our form as this model guides us for maintaining a good posture while doing an exercise.

4 Work Done

The image taken as input is in the RGB format. After that the image is given to the Pose Net model which is based on the Open Pose paper [13] and available in the tensorflow library [14].

So the part 1 of our model is developed using Pose Net model which is available in tensorflow. This model has been pre-trained on datasets such as COCO,LSP,FLIC etc and hence is very accurate with it's result. We give the image of the perosn in squat posture to this model (Pose Net) and it gives out output which is 17 location co-ordinates of various body points mentioned earlier. From these body points, we filter out the body points which are involved with the good or bad posture of squat namely left shoulder, right shoulder, left hip, right hip, left knee, right knee, left ankle and right ankle. So from these 8 body points we get total 16 values (x and y coordinates).

We feed 238 images to the Pose Net and store the corresponding outputs in a CSV file saving 16 values per image. Hence our CSV file has 16 columns for these 16 values and each row represents one image and a column for the class of the image as good posture (represented as 1) or bad posture (represented as 0). Then the data is split between train and validation sets with 90 % data for training and 10 % for validation. There is seperate test set containing 22 images and a set for real time test consisting two images of the author, one in good posture and one in bad posture of the squat. Some of the images also have a person doing weighted squats carrying a bar with weights attached on his/her shoulders.

This data is fed to part 2 of our overall model this as a deep neural network. This network is implemented using tensorflow library. It has the layers as following:

1. Input layers : This is the layers to which input data of values is fed to. It is connected to next layer in a dense manner which means it is fully connected.

2. Hidden layers: There are 5 hidden layers. First two hidden layers have 10 hidden units each, next one layers has 8 hidden units and the next two layers have 4 hidden units each. These hidden layers are fully connected with each other. All the hidden units of all hidden layers have Relu as activation function. Mathematically this function is defined as :

$$Y = Relu(x) = max(0, x) \tag{1}$$

3. Output Layer: The last layer as one unit as an output unit. It is fully connected to the previous layer. This single neuron as sigmoid activation function, output of which lies between zero and one. Sigmoid function is as follows:

$$Y = \frac{1}{1 + e^{-}x}$$

We have used binary crossentropy as loss function which is mathematically represented as :

$$J(Loss) = -(y\log(\hat{y}) + (1 - y)\log(1 - \hat{y})$$

Here, $\hat{y}$ is value generated by the model and y is ground truth. Hence the loss be minimum if ground truth value and generated value are very close.

The optimizer used is adam optimizer as it was giving the best results in this case. The optimizer tries to make the weights of the model such that it gives out 1 as output for good posture and 0 for bad posture by using gradient descent for updating weights of the model. And the models is trained for 2000 epochs.

5 Results and Analysis

The inputs (left) outputs (right) of the model part 1 (Open Pose) which gives out key body points (16 points) as output can be seen in Fig 5.1 for a good posture (above) and a bad posture (below).

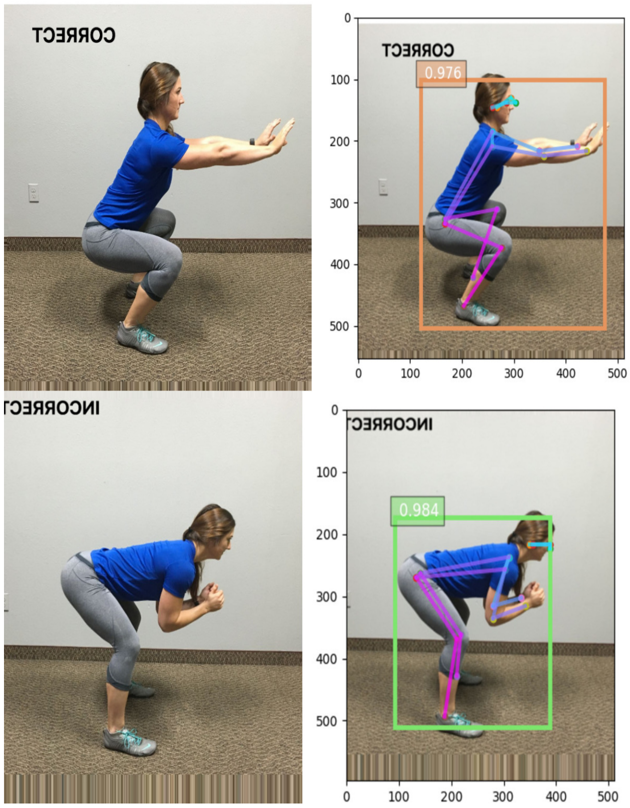


Figure 5.1: An example of inputs and outputs for good and bad posture images from model part 1.

After training the model for 2000 epochs, the accuracy on training set reaches around 96% and accuracy for validation set reaches at around 82%. The accuracy reaches at around 95% around the 500th epoch and remains constant around 95% mark with a few fluctuations. The case is same for validation accuracy which moves to 80% mark at around 500th epoch and remains constant around that value with few fluctuations. Training accuracy and validation accuracy reach there peak at 96.3% and 90% during the 2000 epochs. All these observations can se seen in Fig 5.2.

Th value of the loss function for the training set starts at around 2.5 for the first epoch and goes down to 0.14 till the end of training process with some fluctuations. And during the same process, for the validation set, it goes down till the value 0.4. The loss

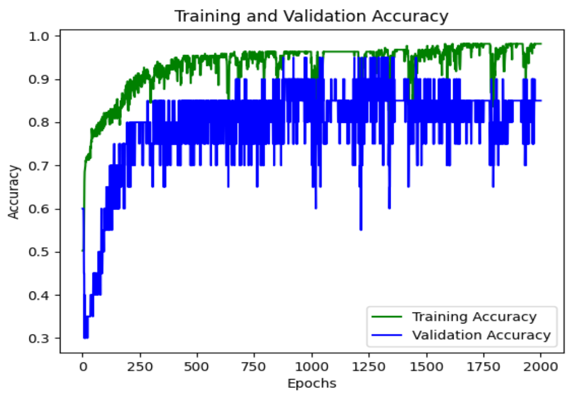


Figure 5.2: Accuracy for training and validation set

reduces very quickly in the first few epochs for both test and validation set and then decreases slowly for the training set to the value of 0.14 till the 2000th epoch. And for validation set remains constant at around 0.4 after 1000th epoch. All these observations can be seen in Fig 5.3.

The model achieves 81% accuracy for the test set which contains 22 images which the model has never seen before. And a loss 0.44 for the same.

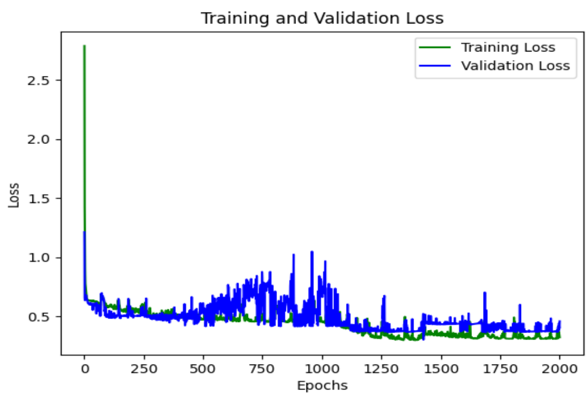


Figure 5.3: Loss for training and validation set

The metrics for Train, Validation and Test set can be seen in Fig 5.4.

	Accuracy	Loss
Training Set	96% (highest during training)	0.1469 (for 2000th epoch)
Validation Set	90% (highest during training)	0.4106 (for 2000th epoch)
Test Set	81%	0.4467

Figure 5.4: Accuracy and loss metrics table

For the real time test using author's pictures, it classifies the picture with bad posture(right) as bad and picture with good posture(left) as good. Hence giving 100% accuracy for this case. The sample of those images can be seen in Fig 5.5.

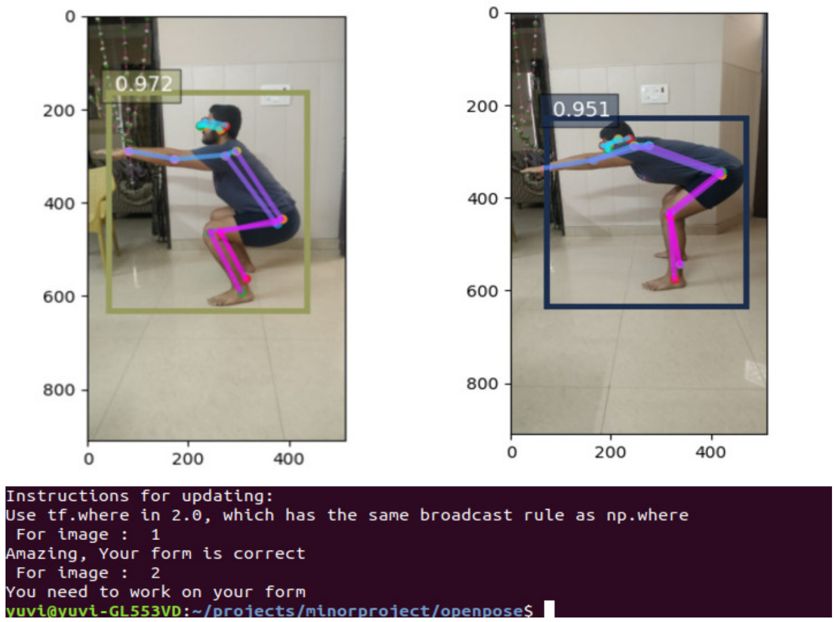


Figure 5.5: Accuracy and loss metrics table

6 Conclusion & Future Work

We can see that the model architecture proposed in this project gives good results and can be implemented practically. It can be similarly trained for other exercises and yoga poses too for improving the posture of the person trying to do a particular exercise. To stay fit, we definitely need to move our body and do some work out but if we do it in a wrong posture it can have very adverse effects on our health, but if we have a system such as this one, it will help us stay fit along with good health by correcting our posture. It also help us save money which we might have to pay to an actual trainer to help us with our posture of doing an exercise. It does not uses any wearable sensors or Microsoft Kinect to process data but can simply work with an image taken from a smart phone too. Hence it is very practical and can be used in our everyday life.

And if we talk about future work, new datasets for other exercises and yoga postures with images having common good and bad postures can be created and the model can be trained on those images. We chose squat because it is one a exercise which has a complex structure and if our method works good on this, it should work good for other exercises also. And then we can make a mobile app for helping users maintain good posture while doing various exercises.

7 Individual Contribution

- Yuvraj Dalia :
  1. Literature Survey
  2. Data Collection
  3. Data Pre processing
  4. Implementation of model architecture
  5. Result and Analysis
  6. Report
- Navnirman Singh :
  1. Data Collection

2. Preprocessing

3. simple app that takes photo , saves and shares it

- Akash Rao :

1. Testing

2. simple app that takes photo , saves and shares it

References

[1] M. U. Islam, H. Mahmud, F. B. Ashraf, I. Hossain and M. K. Hasan, "Yoga posture recognition by detecting human joint points in real time using microsoft kinect," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 668-673, doi: 10.1109/R10-HTC.2017.8289047.

[2] Khelalef, A., Ababsa, F. Benoudjit, N. An Efficient Human Activity Recognition Technique Based on Deep Learning. Pattern Recognit. Image Anal. 29, 702–715 (2019). https://doi.org/10.1134/S1054661819040084

[3] S. Win and T. L. L. Thein, "Real-Time Human Motion Detection, Tracking and Activity Recognition with Skeletal Model," 2020 IEEE Conference on Computer Applications(ICCA), Yangon, Myanmar, 2020, pp. 1-5, doi: 10.1109/ICCA49400.2020.9022822.

[4] S. N. Boualia and N. Essoukri Ben Amara, "Pose-based Human Activity Recognition: a review," 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), Tangier, Morocco, 2019, pp. 1468-1475, doi: 10.1109/IWCMC.2019.8766694.

[5] K. Xu, X. Jiang and T. Sun, "Human activity recognition based on pose points selection," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 2930-2834, doi: 10.1109/ICIP.2015.7351339.

[6] S. Sehgal, "Human Activity Recognition Using BPNN Classifier on HOG Features," 2018 International Conference on Intelligent Circuits and Systems (ICICS), Phagwara, 2018, pp. 286-289, doi: 10.1109/ICICS.2018.00065.

[7] Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., Liu, Y. (2020). Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges and Opportunities. ArXiv, abs/2001.07416.

[8] Bruce X. B. Yu and Yan Liu and Keith C. C. Chan (2020). Effective Human Activity Recognition Based on Small Datasets. ArXiv, abs/2004.13977

[9] Attal, Ferhat Mohammed, Samer Dedabrishvili, Mariam Chamroukhi, Faicel Oukhellou, Latifa Amirat, Yacine. (2015). Physical Human Activity Recognition Using Wearable Sensors. Sensors. 15. 31314-31338. 10.3390/s151229858.

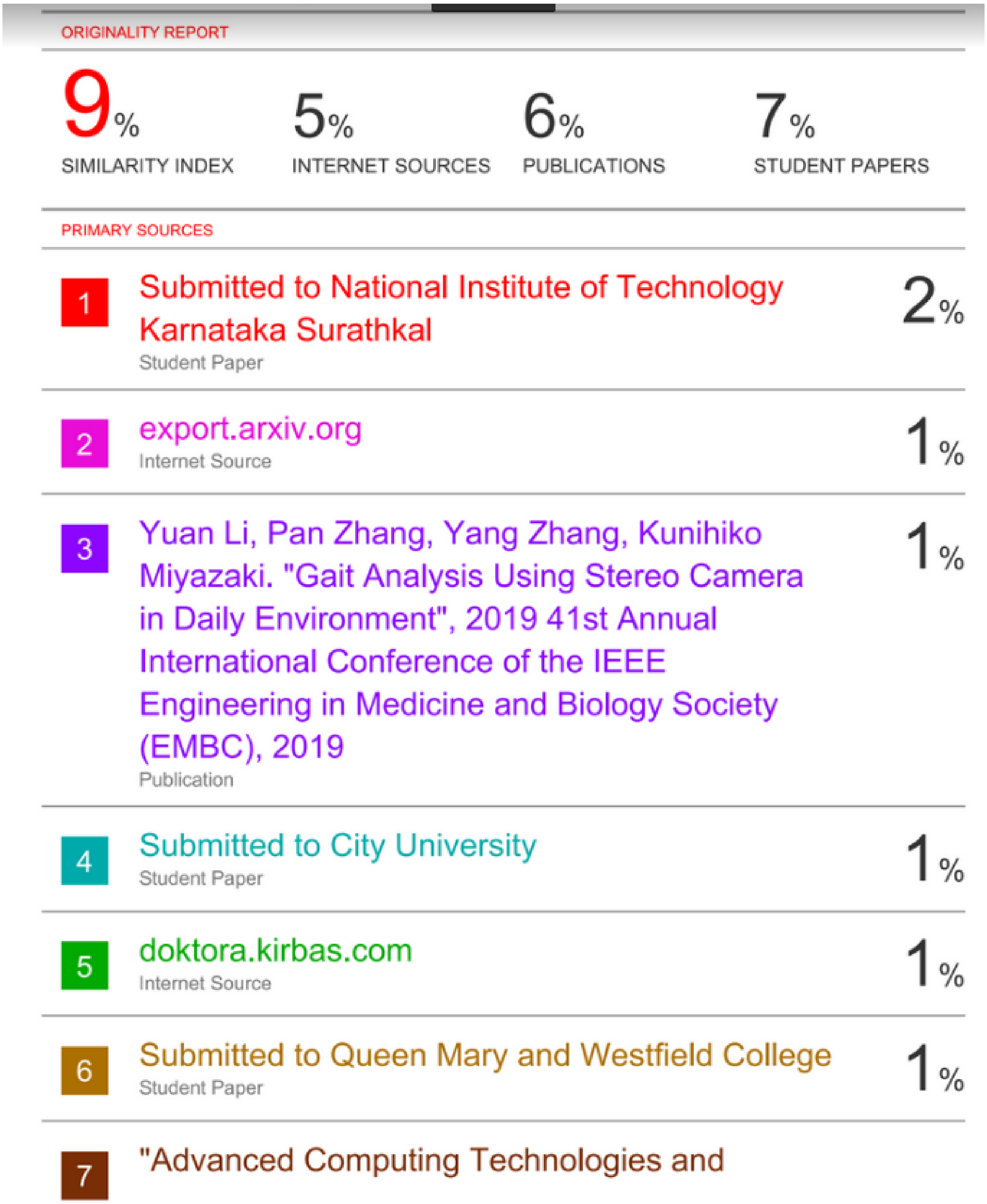
[10] Papandreou G., Zhu T., Chen LC., Gidaris S., Tompson J., Murphy K. (2018) PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11218. Springer, Cham

[11] Y. Chen, C. Shen, X. Wei, L. Liu and J. Yang, "Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 1221-1230, doi: 10.1109/ICCV.2017.137.

[12] Papandreou, George Zhu, Tyler Kanazawa, Nori Toshev, Alexander Tompson, Jonathan Bregler, Christoph Murphy, Kevin. (2017). Towards Accurate Multi-person Pose Estimation in the Wild.

[13] Cao, Zhe Martinez, Gines Simon, Tomas Wei, Shih-En Sheikh, Yaser. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-1. 10.1109/TPAMI.2019.2929257.

[14] https://www.tensorflow.org/lite/models/poseestimation/overview



ORIGINALITY REPORT

9%	5%	6%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to National Institute of Technology Karnataka Surathkal Student Paper	2%
2	export.arxiv.org Internet Source	1%
3	Yuan Li, Pan Zhang, Yang Zhang, Kunihiko Miyazaki. "Gait Analysis Using Stereo Camera in Daily Environment", 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019 Publication	1%
4	Submitted to City University Student Paper	1%
5	doktora.kirbas.com Internet Source	1%
6	Submitted to Queen Mary and Westfield College Student Paper	1%
7	"Advanced Computing Technologies and	

Figure 7.1: Similarity Index Report