Human Pose Estimation Using Body Part Tracking.

Submitted By

Sanchit Jain

18MCEC09

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Human Pose Estimation Using Body Part Tracking.

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (CSE)

Submitted By

Sanchit Jain

(18MCEC09)

Guided By

Dr. K.P. Agrawal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

Certificate

This is to certify that the Major Project entitled **"Human Pose Estimation Using Body part Tracking."** submitted by **SANCHIT JAIN (18MCEC09)**, towards the fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Dr. K.P. Agrawal
Guide and Professor,
Department of CSE,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech CSE,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
Department of CSE,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. R. N. Patel
I/C Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Sanchit Jain**, Roll. No. **18MCEC09**, give undertaking that the Major Project entitled "**Human Pose Estimation using Body Part Tracking.**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (CSE)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. K.P. Agrawal

_____

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. K.P. Agrawal**, Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Computer Engineering/ Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. R. N. Patel**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Sanchit Jain

18MCEC09

Abstract

Human pose estimation is a major computer vision problem that means to detect the spatial area (for example coordinates) of human body joints in unconstrained pictures and images.In other ways we can say it is to predict the location of various human key-points(joints and landmarks) such as elbows, knees, neck, shoulder, hips, chest etc.Estimating human pose is quite challenging problem due to the fact that the human body parts are small and hardly visible parts, occlusions and huge variability in articulations.As strong image processing models, convolutional neural networks (CNNs) comes to the rescue.

In this report I included different approaches for human pose estimation, literature survey of major approaches for pose estimation, detailed analysis, showed how wec can use activity recognition as a use case using estimating of pose for instance here i used drowning detection as an example, also comparison study of Performance of various deep learning models like mobilenet on various Datasets like COCO Dataset, MPII dataset etc .

Contents

Chapter 1

Introduction

1.1 Objective

Here I aimed at targeting the Posture Estimation of Human or Pose Estimation using body Part Tracking and using deep learning approach on how we can efficiently map the body part keypoints and estimate the pose and use ot for various Use Cases as per our requirement for example human activity recognition like walking, sitting etc. In this report I included how can use activity recognition as a use case using estimating of pose for instance here i used drowning detection as an example like person is drowning using his posture, also comparison study of Performance of various deep learning models like mobilenet on various Datasets like COCO Dataset, MPII dataset etc .

1.2 what do we mean by pose estimation?

When we think of pose estimation one thing instinctively strikes our mind that it might be related to something posture of humans and you guessed it correctly, its to estimate the posture of a person or multiple persons by successfully associating each body part keypoints such as knee, arms, shoulders etc in any image or video through deep learning

algorithms.We might think its an easy task but it is way harder then it looks! From Number of its applications including Gaming and activity recognition,improving Pose Estimation still a challenge and a curious topic of research for researchers across the globe.

1.3 Categories and Applications of Pose Estimation

We human beings are quite flexible and we can create a posture by ourselves through bending knees,arms or legs and hence we will be having different body part keypoint relative to others.But in contrast to the living beings, most of the lifeless entities are rigid in nature, for e.g. we have a tile or a brick the distance would be same from one corner to another regardless of their orientation, hence for these entities rigid posture or pose estimation came into picture. Broadly we can classify pose estimation in two main categories i.e. two dimension and three dimension pose estimation . Two dimension posture estimation refers to associating each body part keypoints in two dimension space with reference to any image or video.Whereas three dimension just adds a third dimension.

Applications of human pose estimation ranges from different activity recognition such as falling down of a person to estimating the basic activities like sitting, standing, hands up, hands on hips and many more such activities.

Its application also includes autonomously teaching of work out activities in gym,various sports techniques and different dance activities. we can have uses cases such as understanding complete body sign language including Airport runway signals, traffic police men signals, etc.

Chapter 2

Literature Survey

2.1 Deep Learning Approaches for Pose Estimation:

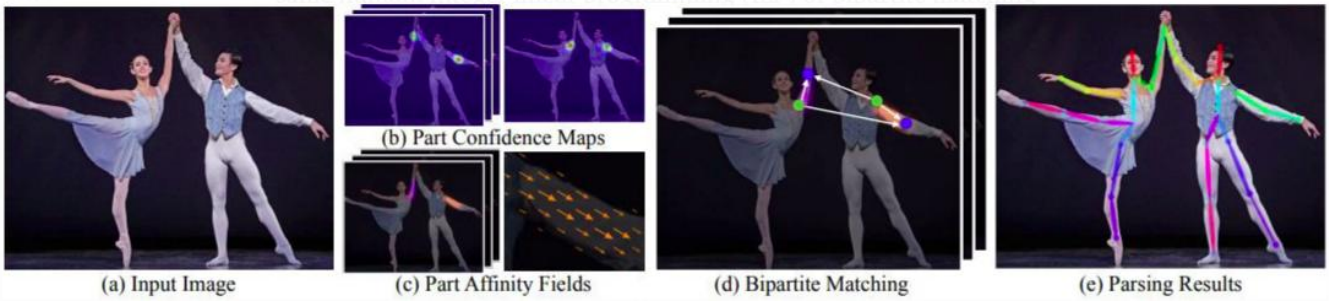
In this section we shall see how through various approaches we can estimate posture of a person and how deep learning plays a vital role to efficiently map the body points through various techniques and give good results using suitable deep learning model as per the requirements.

2.1.1 OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields(Dec, 2018)

The authors worked on the technique which make use of some non-parametric representation that is named as Part Affinity fields which is generally referred as PAF in order to identify human body parts that attains peak precision and excellent concurrent responses irrespective of number of people present in that particular image[1].

- Part affinity fields:

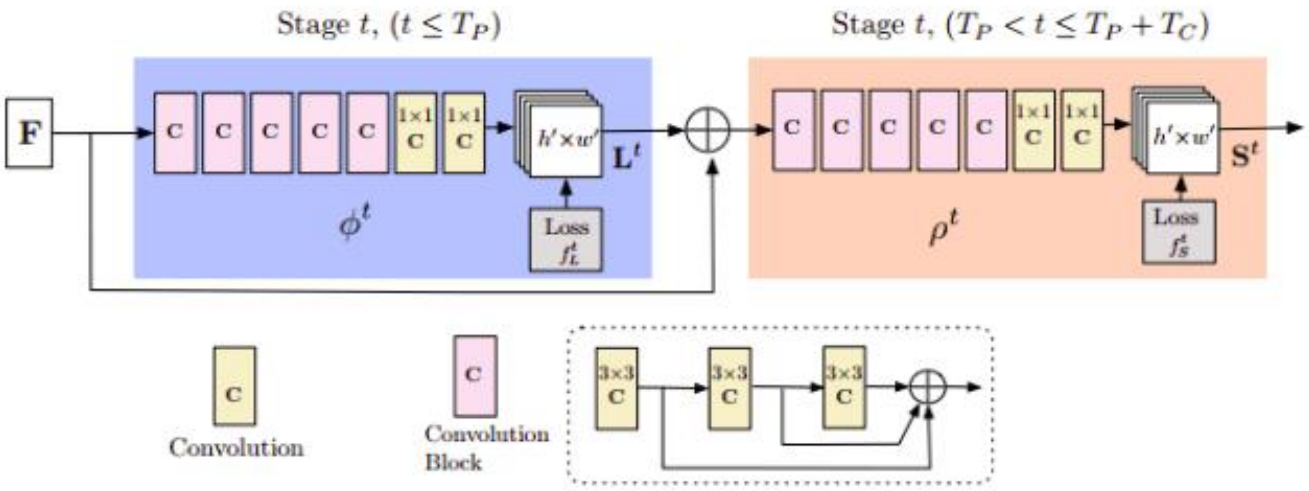
  A pair of 2D Vector Field which is represented or annoted as L which shows or encodes the level of association between body parts is refered as Part affinity field or PAF



- Part Confidence Maps:

  In order to Identify of bodypart localization a pair or set of 2D confidence map S Known as Part Confidence maps.Each joint location has a map.

- Multi Stage CNN:



Multistage CNN can be seen as the following architecture . Following are the steps which are been followed:

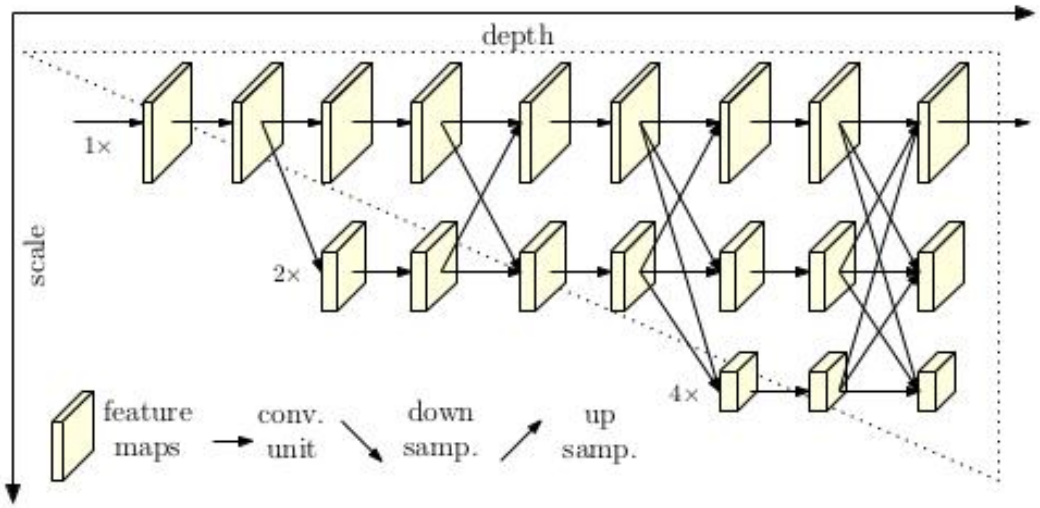
- Stage 1: PAF is been computed,from the base network the feature maps and denoted as L1, F. Let φ be the CNN at the stage 1.
- Stage t to Stage Tp: In this stage it refines the predictions of PAFs from previous stage using the feature maps F and the previous PAF L^{t-1}. Let φ t be the CNN at the stage t.
- After Tp iterations,the confidence maps are found by repeating the process initialising with frequently updated part affinity field(PAF) Let ρt be the CNN at the stage t. The process is repeated for Tc iteration
- The final S and L are the confidence maps and the part affinity fields (PAFs) that will be further processed by the greedy algorithm.

2.1.2 Deep High-Resolution Representation Learning for Human Pose Estimation(Feb, 2019)

In this paper, Authors majorly worked on evaluating human pose estimation.As per them traditional approaches uses recovery techniques for high resolution from comparatively low resolution representations created from high to low resolution networks, rather their approach steadily maintains high resolution throughout complete process. [2].

The initialisation phase happens from a high-resolution subnet and subsequently adding high to low resolution subnets consecutively to process further stages and combine multiple resolution subnets in parallel.Repeated multiple scale fusions were held such that every high to low resolution representations gets the information from different parallel representations repeatedly, resulting to very good high resolution representations.Hence due to above method used the output is more accurate and precise.

HRNET architecture can be seen above contains mainly parallel high to low resolu-



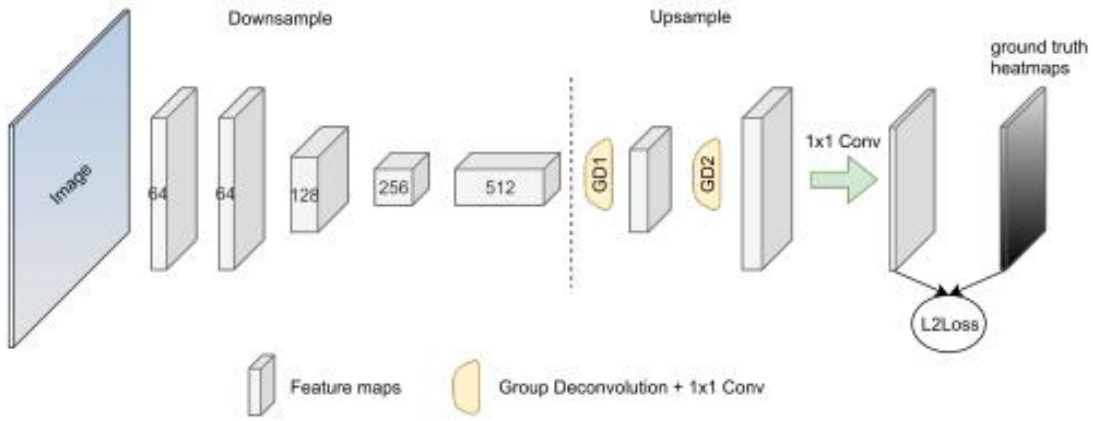
tion subnets with redundant data to be processed is exchange across multiple resolution subnets which can also be called as multi scale fusion.The depth of network and scale of the feature maps are related with each other as horizontal and vertical directions respectively.[2]

Authors with the help of this paper aptly solved the human pose estimation problem by presenting a high resolution network that results in accurate body part keypoints heatmap. The major success is due to these two aspects: (i)Good high resolution is been kept throughout the complete process and not needing high resolution recovery and (ii) merging multiple-resolution representations redundantly,providing efficient high resolution representations. There is huge opportunity and scope in this specially with dense prediction tasks, for example, semantic segmentation, object detection, face alignment, image translation, as well as the investigation on aggregating multi- resolution representations in efficient way [2].

2.1.3 Simple and Lightweight Human Pose Estimation(Jan, 2020)

In this paper the authors have redesigned the pre existing light weight Block from a normal baseline structure using 2 main mechanisims that are depthwise convolution and attention mechanism.By significantly reducing the model size(no. of Params) to their own network architecture known as Lightweight Pose Network (LPN) to only 9% of the original simplebaseline which is Resnet50, and FLOPs to 11% .
The Dataset the authors used for evaluating performance was COCO keypoint detection dataset. The Proposed authors Model Architecture LPN-50 can achieve 68.7 in AP score on the COCO test-dev set[3].

The model architecture which the authors proposed can be seen below:



Moreover to increase the efficiency and enhanced results for the predected results β-Soft-Argmax funtion was introduced which handled the postprocessing part of the network which help in improving LPN network performance.

Few of the upgradation which authors did to the normal Baseline Backbone models are:

- Setting the Hyperparameter to 1
- Replacement of Convolution(3x3) to Depthwise convolution(3x3)

Performance evaluation on various model backbones:

Method	Backbone	Pretrain	Input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass [6]	Hourglass	N	256 × 192	25.6M	26.2G	66.9	–	–	–	–	–
CPN [27]	ResNet-50	Y	256 × 192	27.0M	6.2G	68.6	–	–	–	–	–
SimpleBaseline [1]	ResNet-50	Y	256 × 192	34.0M	8.9G	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline [1]	ResNet-101	Y	256 × 192	53.0M	12.4G	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline [1]	ResNet-152	Y	256 × 192	68.6M	15.7G	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32 [11]	HRNet-W32	N	256 × 192	28.5M	7.1G	73.4	89.5	80.7	70.2	80.1	78.9
HRNet-W32 [11]	HRNet-W32	Y	256 × 192	28.5M	7.1G	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48 [11]	HRNet-W48	Y	256 × 192	63.6M	14.6G	75.1	90.6	82.2	71.5	81.8	80.4
SimpleBaseline [1]	ResNet-152	Y	384 × 288	68.6M	35.6G	74.3	89.6	81.1	70.5	79.7	79.7
HRNet-W48 [11]	HRNet-W48	Y	384 × 288	63.6M	32.9G	**76.3**	**90.8**	**82.9**	**72.3**	**83.4**	**81.2**
LPN (Ours)	ResNet-50	N	256 × 192	**2.9**M	**1.0**G	69.1	88.1	76.6	65.9	75.7	74.9
LPN (Ours)	ResNet-101	N	256 × 192	5.3M	1.4G	70.4	88.6	78.1	67.2	77.2	76.2
LPN (Ours)	ResNet-152	N	256 × 192	7.4M	1.8G	71.0	89.2	78.6	67.8	77.7	76.8

2.1.4 DeepPose: Human Pose Estimation via Deep Neural Networks(June, 2014)

Human Pose estimation using Deep neural network is been proposed by authors.Using DNN bodypart keypoints are been calculated and used for pose estimation. They proposed a cascade containing these specific DNN regressors that has consequence in high precision pose estimates [4].There are many befitting benefits of such an efficient method for pose estimation which is easy to understand as well as implement to extract various use cases out of it using advanced deep learning approaches.The authors presented an in-depth factual examination with better results on four academic benchmarks of vivid images.

In this authors majorly put their focus on how to combinely formulate something for human posture estimation, the regression problem as well as displaying successful DNN setting casting.An input of complete image is passed through a seven layered generic DNN and regressed to find exact location of each body part keypoint in that individual. The two major benefits of this approach is as follows. First of all,our Deep neural network has certain potential of gaining complete information regarding each body part keypoint.Every joint regressor utilizes complete image as a signal. Secondly, the applied methodology is quite effectively easier to manipulate than other methods for graphical models, in this there is no needfor a direct approach for designing feature representations and detectors for body parts,also regarding directly require to have a outline of a model bule print or we can say topology and interactions between human body parts.As an al-

ternate the author showed that learning is highly possible for the generic DNN in context to solve such problem.[4]

In the left image above,we can see the dimensions of the layers of network in which blue indicates convolutional layers whereas fully connected are green in color.Moreover the layers which are free from parameters are not been shown.Towards right on stage S,there is a refining regressor is been put in over a sub image in order to improve prediction from preceding stage.

2.1.5 Deep Learning Based 2D Human Pose Estimation: A Survey(Dec, 2019)

A Rigorous Survey was conducted by the authors in order to compute human pose estimation approaches.the authors encapsulated and discussed most trending and latest works with a approach based taxonomy. The survey mainly focused on single and multi person pose or psoture estimation consecutively.Due to Recent advancement in this field and variety of uses in almost all the domains human pose estimation has received significant heed and there is on going improvements through state of the art deep learning techniques [**?**].

In case of single person pose estimation, explained with the help of image below various different human bodyparts including neck center, head,elbows and shoulders of eighter side, left or right arms etc a single person algorithm for identifying the respective body parts should be applied if person as a sole is been given as an input.

In case of multipose estimation in which exact count of total people is unrevealed in

any image or video as an input then the algorithm for pose estimation is applied which is more challenging then single person estimation, it is required by the deep learning algorithm then to precisely associate each body part keypoint of that exact person even in a populated place.

While differentiating with single and multiple pose estimation approaches,we find multipose estimation as more challenging problem due to the fact that both number and spot of that particular person is unknown to us.In order to make our way out of this problem we mainly need to deal with bodypart keypoint detection or locating it and position of human in that image, there are two ways to solve:

- top down approach or pipeline
- bottom up approach or pipeline

With the help of this survey the main intentions of the authors were to present those efficient methods for human posture estimation thar applies deep learning approaches.With the recent development in research field in this domain of deep learning there is a significant improvements in deep learning based human posture estimation approaches, there is still room for improvement and refining efficiency.As per authors present deep learning algorithms lacks speed and unable to meet real time concurrent predictions, hence this part

needs to be worked upon.Furthermore there has been work done for compression in networks and network acceleration, but not with reference to human pose estimation,because its requirement is high resolution featuremaps in contrast with object detection and classification problem.Hence for pose estimation the accelerating approaches have to be more digged into.

At Present dataset is quite huge,and for unbalanced datasets the human psoe estimation results are not up to the mark hence there is need of refinement of the techniques.Some of the approaches may include data augmentation technique, and efficient training approach.

Moreover blockages which are also known as occlusion creates problem for efficient human pose estimation results hence decreases the overall accuracy and performance of model.Still there is room for improvement in human pose estimation techniques for better accuracy and efficient results.

2.2 Models related Literature Survey

In this section there will be a discussion regarding the deep learning models which are used in pose estimation and see the architectures of each of them.

2.2.1 MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications(April, 2017)

Mobilenets are one of the most efficient Neuralnetwork which has wide range of applications in applications which are related to mobile vision and embedded applications.

In this Model architecture first and foremost thing is that it uses Depthwise seperable Convolution to create the neural network. We will see the architecture of mobilenet as we proceed.

The Mobilenet model architecture is created basically from the depthwise sep. conv. which are introduced in few networks before like the Inception neural network model, to decrease the compitation of top layers and then comes the factorized networks which are used to create the flatten layers or networks.Then to increase or scale up the depthwise sep filters it is been demonstrated in the Xception network[5].

Mobilenet Network Architecture:

We will be discussing the core architecture starting from the base of the network to the final layers of the mobile net model.Then we will see the depthwise seperable conv. how it is used with its proper description and finally concluding the model by discussing its hyperparameter as well.

- Depthwise Seperable Convolution:

  In this mobilenet architecture major role is been played by this depthwise separable convolution which can be described as factorized convolution into 1x1 conv. refered as pointwise convolution, the authors proposed in the model that this conv. is splitted in 2 layers namely for filtering as seperate layer and seperate for combining.

  Hence using this factorizing there is noteable reduction in computation and model size.Also, MobileNet model uses 3x3 depthwise separable convolutions which uses between 8 to 9 times less computation than standard convolution[5].

- Network Structure and Training :

  The network structure mainly consist of the depthwise seperable convolution layers as been seen previously,it also has ReLU and softmax operation for classification.After taking consideration into depthwise and pointwsie convolution as seperate layer the model of mobilenet has 28 layers, the table below shows the architecture[5].

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

That was all about majorly the architecture part of the famous deep learning model Mobilenet and now its used in wide range of computer vision applications and use cases

2.2.2 MobileNetV2: Inverted Residuals and Linear Bottlenecks(2019)

The Authors Explained or rather proposed a new and more efficient model which is certainly an enhancement over the first mobilenet model proposed, the accuracy, latency and effeciency is beeter then mobilenet and various other network models for deep learning applications.

The architecture is explained or proposed as such in which inverted residual structures are been utilized for thin layered bottleneck.The intermediate or middle layers are consists of depthwise seperable convolution and in the concluding stages of the architecture decoupling on the input and output domain is been done from the transformation and furthur analysis is done by selecting appropriate framework for the same.

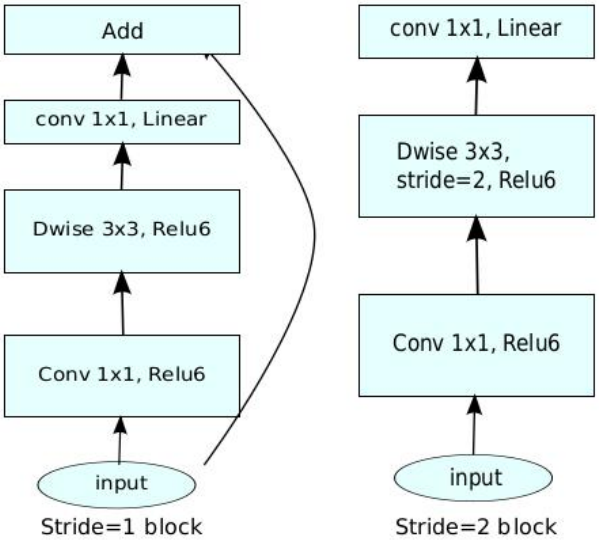
Major components of architecture includes:

- Depthwise Separable Convolution:
  As Discussed in the mobilenet model as well, this architecture of mobilenetV2 is also highly dependent or majorly constucted using this layers. just for the sake of the idea it is diving into two forms filtering on single convolutional layer and second pointwise convolution.
- Linear Bottlenecks:
  Linear bottlenecks majorly discuss Relu transformation and its coressponding linear transformation and how complete info is stored or manifold if input is in lowdimentional sub space.
- Inverted Residue:
  The concept of inverted residue is that the important information in the middle layers are the implementation details for no linear transformation of tensors.

Below is the architecture of MobilenetV2 model: [6]

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

In the above architecture described, the stride is s for 1st layer and others as stride 1. Kernal Size is 3x3 also the total number of convolutional layers with 32 filters.Due to low precision Relu works efficiently with non linearity.The diagrametic representation is below [6]



Chapter 3

Implementation

Here we will discuss the tensorflow based implementation of OpenPose which we saw earlier in literature survey.As we saw number of use cases of the human pose estimation, hence here i have depicted one of the possible use case, i.e. in flooded areas or any other situation where people are in urgent need or help the drone or any video capture device would detect send the message for help.The outputs are shown in the section.

3.1 Dependencies

These are dependencies of the libraries and framework for implementation of OpenPose for human pose estimation:

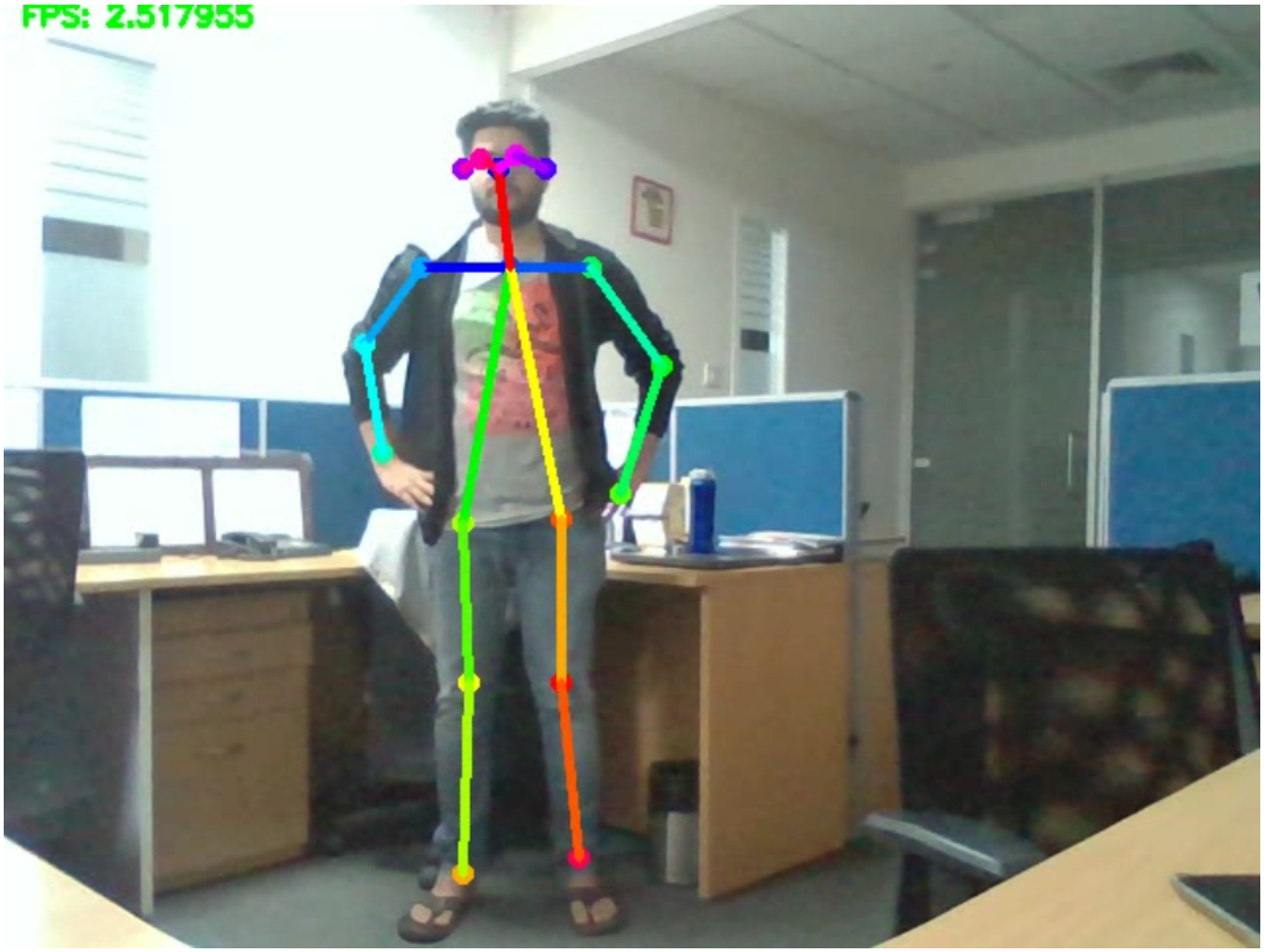
- Python3
- tensorflow 1.4.1 or higher
- opencv3, protobuf, python3-tk

- sliding window

This is Tensorflow implementation of the paper, the original implementation is in caffe.

3.2 ScreenShots of the output

First lets see normal pose of a person.



Now lets see how our output looks when a person needs help!!

Chapter 4

Experimental Evaluation

In the performance measurement section we will be discussing the below model performance on various datasets.The performance metrics we are using is average precision(AP) and average recall(AR) for different Intersection over unions(IOU's).

Models used for comparison are:

- CMU(VGG16 as backbone)
- Mobilenet-thin
- Mobilenetv2-small
- Mobilenetv2-large

4.0.1 Brief about some Datasets Available:

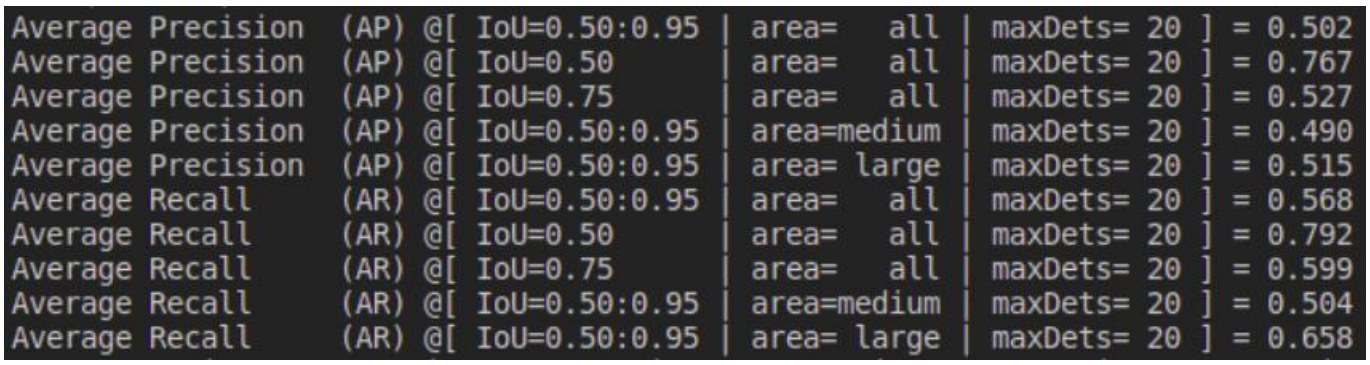
These are the datasets which are very famous and freely available for performance measurement for pose estimation, in my case i will be testing the performance on COCO keypoint datasets for year(2014,2017) and discussing few others :

- COCO Keypoint Dataset(2014,2017): COCO is a Large image dataset for various use cases such as segmentation, object detection, keypoint detection, We will be using the year 2014 and 2017 dataset they released for keypoint detection[7].
- MPII dataset: MPII is another prominent dataset for human pose estimation which is a state-of-art benchmark evaluation which has 25 thousand images with annoted images of 40 thousand people for body joints with 800 human activites[8]
- Pose track Dataset: PoseTrack is another great dataset for pose estimation, in the paper iteself the authors explained how dataset in the form of video is efficiently useful and handle many challenges[9]

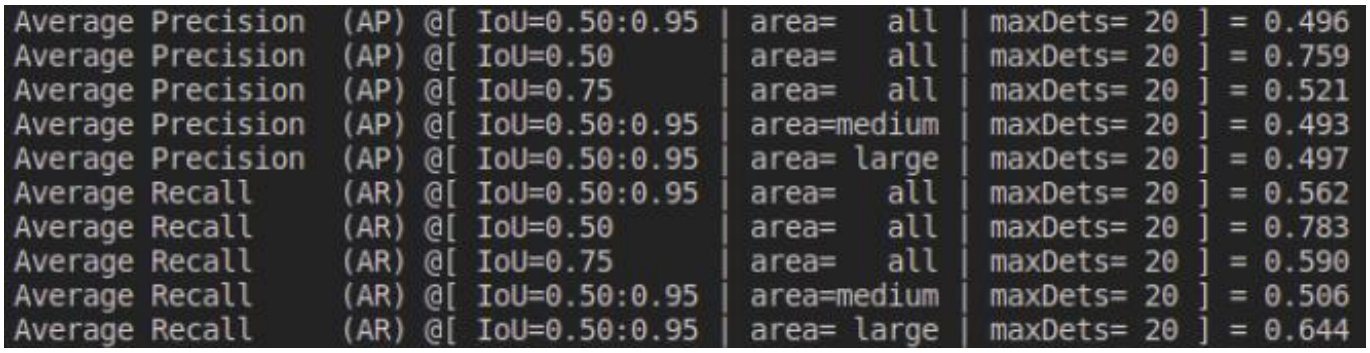
4.0.2 Results:

In this section we will discuss what is the performance of the models on COCO dataset for year 2014,2017.

- CMU model on COCO 2014 Dataset:



- CMU model on COCO 2017 Dataset:



- Mobilenet-thin model on COCO 2014 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.261
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.555
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.213
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.238
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.295
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.321
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.599
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.294
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.257
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.407
```

- Mobilenet-thin model on COCO 2017 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.255
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.556
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.201
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.243
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.281
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.313
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.594
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.282
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.259
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.390
```

- MobilenetV2-Small model on COCO 2014 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.172
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.400
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.124
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.149
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.209
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.219
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.448
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.186
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.159
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.300
```

- MobilenetV2-small model on COCO 2017 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.164
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.390
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.111
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.151
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.191
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.210
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.432
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.172
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.160
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.279
```

- MobilenetV2-Large model on COCO 2014 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.312
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.577
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.294
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.261
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.383
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.367
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.609
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.363
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.276
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.490
```

- MobilenetV2- Large model on COCO 2017 Dataset:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.292
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.556
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.263
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.251
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.355
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 20 ] = 0.345
Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets= 20 ] = 0.582
Average Recall     (AR) @[ IoU=0.75      | area=   all | maxDets= 20 ] = 0.335
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets= 20 ] = 0.264
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets= 20 ] = 0.458
```

Chapter 5

Conclusion and Future Scope

On the Concluding note i would like to wrap this report by just giving the overview of what i have done, We saw what is human pose estimation in detail, Different approaches to deal with human pose estimation with the help of literature survey, then implementation of one of the papers which is openpose which is beautifully written and well read in easy language followed by the checking the performance of various models on COCO Dataset some output for the same Finally concluding i cansy that for few cases the CMU model performed better for COCO dataset of 2014 while for other mobielnet V2 gave good result, few bodypart keypoints AP was good for mobilenet model as well, so will be concluding Thats a Wrap!

This particular problem of human pose estimation is one of hot topics among the researchers around the world seeing the vast variety of uses for real world problems! Some of the future scopes include:

- Improvements in estimation of pose in case of multiple person
- improvements in frames per seconds(fps)
- Implementation on Edge Devices for mobile applications!

Bibliography

[1] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[2] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *arXiv preprint arXiv:1902.09212*, 2019.

[3] Z. Zhang, J. Tang, and G. Wu, "Simple and lightweight human pose estimation," *arXiv preprint arXiv:1911.10346*, 2019.

[4] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.

[5] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[8] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[9] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.