

**JSS MAHAVIDYAPEETHA**

**JSS Science and Technology University**



**“Human Pose Estimation”**

A technical project report submitted in partial fulfillment of the award of the degree  
of

**MASTER OF COMPUTER APPLICATIONS**

**IN**

**Department of Computer Applications**

**BY**

**ASHOK KUMAR H G**

**01JST20PMC006**

Under the guidance of

**Prof. Nandini S**

Assistant Professor

Department of Computer Applications

JSS STU, Mysuru-06

**2021-2022**

**Department of Computer Applications**

**JSS MAHAVIDYAPEETHA**  
**JSS Science and Technology University**



**Certificate**

This is to certify that the work entitled

“**Human Pose estimation**” is a Bonafide work carried out by **Ashok kumar H G**, in partial fulfilment of the award of the degree of **Master of Computer Applications** in **Department of Computer Applications** for the award of **Master of Computer Applications** by JSS Science and Technology University, Mysuru, during the year 2021-2022. The project report has been approved as it satisfies the academic requirements in respect to project work prescribed for the Master of Computer Applications degree in Department of Computer Applications.

**Under the guidance of**

**Prof. Nandini S**  
Assistance Professor  
Dept. of Computer Applications,  
JSS STU, Mysuru-06

**Head of the department**

**Dr. V N. Manjunath Aradhya**  
Professor and Head,  
Dept. of Computer Applications  
JSS STU, Mysuru-06

Examiners : 1.....

2.....

# DECLARATION

I, **Ashok kumar H G** bearing USN **01JST20PMC006** student of IV Semester MCA, **JSS Science and Technology University, Mysuru-570006**, hereby declare that the project entitled “**Human pose estimation**” has been independently carried out by me under the guidance of **Prof. Nandini S**, Assistant Professor, Department of Computer Applications, JSS Science and Technology University, Mysuru, and submitted in the partial fulfillment for the award of degree of Master of Computer Applications, during the academic year 2021-2022. Further the matter embodied in the report is an original and bonafide work done by me.

To my knowledge this project has not been submitted to any other college or university or published at any time prior to this.

**Place : Mysuru**

**Date :**

**ASHOK KUMAR H G**

**01JST20PMC006**

## Plagiarism Scan Report

Report Generated on: Jul 26,2022



### Content Checked for Plagiarism

#### 1.0 Introduction

Being one of the most difficult PC vision issues with a large number of applications, human posture assessment has been one of the essential examination regions that the PC vision local area attempted to address with Deep Learning and Convolutional Neural Networks (CNNs). Considering that the outcomes delivered by existing cutting edge techniques look basically noteworthy both subjectively and quantitatively, addressing how much is normal progress can be anticipated on this issue throughout the following years and whether there is space for further improvement.

[IEEE 2017 9th IEEE-GCC Conference and Exhibition (GCCCE ... [↗](#)]

segmentation is accomplished, the human pose can be recognized accurately using  
<https://ur.booksc.me/book/72246992/778b3a>

3

[IEEE 2017 9th IEEE-GCC Conference and Exhibition (GCCCE ... [↗](#)]

include head, neck, shoulders, chest, arms, elbow, thighs, legs, ankle and foot.  
<https://ur.booksc.me/book/72246992/778b3a>

2

## ACKNOWLEDGMENT

The success of any endeavor depends a lot on the goals set at the onset as well as the constant guidance and motivation received throughout. It's my duty to acknowledge and thank the individuals who has contributed in the successful completion of the project.

I truly express my deep sense of gratitude and sincere thanks to my respected guide **Prof. Nandini S**, Assistant Professor, Department of Computer Applications, sustaining interest and dynamic guidance shown in aiding me to complete this project immaculately and impeccably and being the source of my strength and confidence.

I feel immense pleasure to thank **Dr. V N Manjunath Aradhya**, Associate Professor and Head, Department of Computer Applications, for his encouragement and support throughout the project.

I express my heartfelt thanks to our principal **Dr. S B Kivade** at the esteemed institution **JSS Science and Technology University, Mysuru** for providing me an opportunity to reach my goal.

I sincerely express our thanks and gratitude to our institution **JSS Science and Technology University, Mysuru - 570006** for providing me an opportunity to fulfill our most cherished desire of reaching my goal and thus helping me to make a bright career.

I would like to thank all the **Teaching and Non-Teaching Staff** of Department of MCA for their kind Co-operation during the course of the work. The support provided by the **Departmental library** is gratefully acknowledged.

This successful completion of my project would not have been possible without **my parents Sacrifice, guidance and prayers**. I take this opportunity to thank very much for their continuous encouragement. I convey my thankfulness to all **my friends who were with me to share my happiness and agony**. They gave valuable suggestion which was the solution that helped me to a great extent to complete the project successfully.

ASHOK KUMAR H G

## **ABSTRACT**

Human pose recognition is considered a well-known process of estimating the human body pose from a single image or a series of video frames. Human pose estimation has always been a challenging problem that holds great attention, it has the widespread and extensive variety of uses from the classification of images to activity acknowledgment, main challenge is the detection and localization of the key points in the variation of several body poses. There exist many applications that can benefit from human pose technology e.g. activity recognition, human tracking, 3D gaming, character animation, clinical analysis of human gait and other HCI applications. Due to its many challenges, such as illumination, occlusion, outdoor environment and clothing, it is considered one of the active areas in computer vision. For the last 15 years, Human pose recognition problem significantly gained interest of many researchers and therefore, many techniques were proposed in order to address the challenges of human pose recognition. In this study, we review the recently progressed work in human pose recognition using computer vision feature extraction and machine learning classification techniques.

## Table of Contents

<b>Chapter 1</b> .....	1
Introduction .....	1
1.0 Introduction .....	1
1.1 Description .....	2
1.2 Problem Statement .....	3
1.2.1 Objectives .....	3
1.2.2 Aim .....	3
1.2.3 Scope .....	3
 <b>Chapter 2</b> .....	4
Literature Survey .....	4
2.1 Survey papers .....	4
 <b>Chapter 3</b> .....	13
Dataset .....	13
3.0 Dataset Description .....	13
3.1 MPII Dataset .....	13
3.2 COCO Pose dataset .....	15
 <b>Chapter 4</b> .....	17
System Design .....	17
4.0 System Design .....	17
4.1 Dataflow Diagram .....	17
Fig 4.1.1 : General Dataflow Diagram.....	18
Fig 4.1.2 : Level-0 Dataflow Diagram .....	19
Fig 4.1.3 : Level-1 Dataflow Diagram .....	19
4.2 Usecase Diagram .....	20
Fig 4.2.1 : Usecase Diagram .....	20
4.3 Sequence Diagram .....	21
4.3.1 Purpose .....	21
Fig 4.3.2 : Sequence Diagram .....	22

4.4 Activity Diagram .....	23
Fig 4.4.1 : Activity Diagram .....	23
<b>Chapter 5 .....</b>	<b>24</b>
System Implementation .....	24
5.0 Introduction .....	24
Fig 5.0.1: Software Architecture .....	25
5.1 Python .....	25
5.2 Machine Learning v/s Deep Learning .....	26
5.3 Deep Learning Overview .....	26
5.4 Libraries and modules used .....	27
5.4.1 Pandas .....	27
5.4.2 OS .....	27
5.4.3 PIL .....	27
5.4.4 TensorFlow .....	28
5.4.5 Keras .....	29
5.4.6 OpenCV .....	29
5.5 Algorithm used .....	29
5.5.1 CNN .....	29
Fig 5.5.2 : CNN Architecture .....	29
Fig 5.5.3 : System Architecture .....	33
<b>Chapter 6 .....</b>	<b>34</b>
Discussion and Results .....	34
6.1 Discussion .....	34
6.2 Results .....	34
<b>Chapter 7 .....</b>	<b>38</b>
Conclusion and Future Scope .....	38
7.1 Conclusion .....	38
7.2 Future Scope .....	38
<b>Chapter 8 .....</b>	<b>39</b>
References .....	39



## CHAPTER - 1

### INTRODUCTION

#### 1.0 Introduction

Being one of the most challenging computer vision problems with a multitude of applications, human pose estimation has been one of the primary research areas that the computer vision community tried to solve with Deep Learning and Convolutional Neural Networks (CNNs). Given that the results produced by existing state-of-the-art methods look at least impressive both qualitatively and quantitatively, it is natural to question how much progress can be expected on this problem over the next years and whether there is room for further improvement.

Human pose estimation holds extraordinary potential from single, 2D pictures to aid an extensive variety of uses from the classification of images and recordings, activity acknowledgment, active investigation, and grabbed great attention in computer vision and human PC interaction. However, human posture estimation has always been a challenging problem that acquires great attention. It involves huge difficulty for the identification and localization of key points of the body that mainly includes various joints and body movement forecast and also shares difficulties in detection, for example in clustering, lighting, perspective, and scale, are the significant troubles interesting to human postures.

Human pose plays an important role in the human communication process. The human posture is used to represent different emotions. A recent study shows that human body poses express emotions better than facial expression. According to a study, spoken words represent only 7% of human communication while non-verbal actions, such as posture and facial expressions, represent 55% of the overall communication process. Human pose is a non-verbal communication method, which is realized by recognizing the pose of a human. A pose can be extracted from any action such as eating, walking, sitting, waiting, and discussion to mention a few. Human pose can be recognized by localizing joints on human body and dividing the body around these joints into body parts. One such division may include head, neck, shoulders, chest, arms, elbow, thighs, legs, ankle and foot. Once such segmentation is accomplished, the human pose can be recognized accurately using segmented body parts.

The detection of body key points has been a great problem due to little joints, impediments, and the need to catch content. Hence convolutional neural networks have a remarkable approach to image classification and object identification issue.

## 1.1 Description

Human pose estimation is the task of estimating the joint locations of one or multiple people within an image. It is a core challenge in computer vision because it forms the foundation of more complex tasks such as activity recognition and motion planning. For example, joint locations have been used to supplement other visual features to determine the trajectory of a person through a sequence of video frames

Human pose estimation is one of the challenging fields of study in computer vision which aims in determining the position or spatial location of body key-points (parts/joints) of a person from a given image or video, Thus, pose estimation obtains the pose of an articulated human body, which consists of joints and rigid parts using image based observations.

Human pose estimation refers to the process of inferring poses in an image and these estimations are performed in either 3D or 2D. To solve this problem, several approaches in the literature have been proposed. Early works introduced the classical approaches to articulated human pose estimation called the pictorial structures . In these models, the spatial correlations of the body parts are demonstrated as a tree structured graphical model and they are very successful when the limbs are visible however faced problems when the tree structured fails capturing the correlation between variables.

Solving the problems and challenges related to human pose estimation has been advanced and progressed remarkably with the help of deep learning and publicly available datasets.

Robust interactive human body tracking has applications including gaming, human-computer interaction, security, tele-presence, and even health-care.

Being one of the most challenging computer vision problems with a multitude of applications, human pose estimation has been one of the primary research areas that the computer vision community tried to solve with Deep Learning and Convolutional Neural Networks (CNNs). Given that the results produced by existing state-of-the-art methods look at least impressive both qualitatively and quantitatively, it is natural to question how much progress can be expected on this problem over the next years and whether there is room for further improvement.

Yet, from a practical perspective, many applications cannot fully enjoy the high accuracy demonstrated by recent advances. The reason is for this is twofold: (a) the bulk of current work assumes the abundance of computational resources (e.g. GPUs, memory, power) to run these models which for many applications are not available. (b) In many application domains (e.g. autonomous driving) accuracy is absolutely essential, and there is very little room for accuracy drop when, for example, more lightweight, compact, and memory efficient methods are used

### 1.2 Problem Statement

The problem consists of human pose estimation. For estimation of human postures, the network takes a raw image as input and a vector of coordinates of the body key points as outputs. The purpose is to identify x-y pixel coordinates for 18 body joints, connecting them to form a skeleton structure of the human and finally classifying them based on the each joint(points) value which will differ for each and every different positions. By training the regression CNN that compensate and classify the model.

#### 1.2.1 Objectives :

The main objectives of the project are :

- To detect the body joints (key points) from the human body.
- Drawing a skeleton structure by connecting the detected key points.
- Building a model to estimate the position of a human at an instance.
- We are building a model which detect the activity or posture of a person in given image or video.
- The model classifies the skeleton structure (which is obtained by connecting the key points) based on the values of the each joints.
- Will estimate the posture from an image or video which will be useful in many areas.

#### 1.2.2 Aim :

The main Aim of this project is to efficiently identify the position of the human in an image or a video

#### 1.2.3 Scope :

The scope of future research in Pose Estimation is immense and creating a learning slope can get more people interested..

# CHAPTER – 2

## LITERATURE SURVEY

### 2.1 Survey Papers

#### 1. 2D Human Pose Estimation from Monocular Images: A Survey

Author : Sun Jingtian, ChenXue, Lu Yanan, Cao Jianwen

Year : 2020

Findings :

This survey collected a wide range of papers of human pose estimation from classic models using handcrafted features to the newest state-of-art deep learning methods. This paper divided them into two main categories: bottom-up and top-down, in each category, differences between single person pose estimation and multi- person pose estimation is clarified, within each sub-category, some representative works are introduced in detail.

The formation of this paper is composed as follow: bottom-up methods and top-down methods will be introduced in Section 2 and Section 3 respectively, where the comparison will be made as well. In each section single person method and multi-person methods will be divided into sub-sections and introduced correspondingly.

One general way to categorize the human pose estimation algorithms is to divide them into generative methods which can also be called top-down methods and discriminative methods which can also be called bottom-up methods. Though having a little bit of difference of meaning in single person and multi- person situations, generative methods often refer to methods that require a human body model in prior and treat the pose estimation problem as a geometric projection, while discriminative methods focus on local evidence first.

#### 2. A Study on the Learning Based Human Pose Recognition

Author : Faisal Sajjad, Adel F. Ahmed, Moataz A. Ahmed

Year : 2017

Findings :

Younesset al, in 2016, recognized human pose in realtime using Microsoft Kinect sensor. The features were extracted from skeleton data provided by Kinect. These features were then fed into different machine learning classification techniques in order to identify which classifier performs best in real time. Figure 2 shows the average accuracies of different classifier. The authors tested the performance of the classifier by initially using 22% of the data as training data and then increase the training data up to 88% of the entire dataset. As can be noticed

from Figure 2, all the classifiers achieved almost 99% accuracy except naive Bayesian classifier. The naive Bayesian classifier reached the accuracy of 98.21% only. Similarly, Ishan et al. [5] applied different machine learning classification techniques to detect the emotional states of each individual subject in a team using Microsoft Kinect. Table IV shows the different classifiers' accuracies. Each classifier performance was tested with up to 4 subjects, i.e. team members. As can be seen from the Table IV, when emotions are recognized with one team member in the frame, the Random Forest classifier outperformed others followed by the IBK classifier. But when the number of team members increased in the frame, the performance of all classifier decreased slightly except for the naive Bayesian classifier who's accuracy dropped significantly from 98.44% to 53.44%.

### **3. Combining detection and tracking for human pose estimation in videos**

**Manchen Wang, Joseph**

Author : Tighe, Davide Modolo

Year : 2020

Findings :

We propose a novel top-down approach that tackles the problem of multi-person human pose estimation and tracking in videos. In contrast to existing top-down approaches, our method is not limited by the performance of its person detector and can predict the poses of person instances not localized. It achieves this capability by propagating known person locations forward and backward in time and searching for poses in those regions. Our approach consists of three components: (i) a Clip Tracking Network that performs body joint detection and tracking simultaneously on small video clips; (ii) a Video Tracking Pipeline that merges the fixed-length tracklets produced by the Clip Tracking Network to arbitrary length tracks; and (iii) a SpatialTemporal Merging procedure that refines the joint locations based on spatial and temporal smoothing terms. Thanks to the precision of our Clip Tracking Network and our merging procedure, our approach produces very accurate joint predictions and can fix common mistakes on hard scenarios like heavily entangled people. Our approach achieves state-of-the-art results on both joint detection and tracking, on both the PoseTrack 2017 and 2018 datasets, and against all top-down and bottom-down approaches.

At a high level, our method works by first detecting all candidate persons in the center frame of each video clip (i.e. the keyframe) and then estimating their poses forward and backward in time. Then, it merges poses from different clips in time and space, producing any arbitrary length tracks. More in details, our approach consist of three major components: Cut, Sew and Polish. Given a video, we first cut it into overlapping clips and then run a person detector on their keyframes. For each person bounding box detected in a keyframe, a spatial-temporal tube is cut out at the bounding box location over the corresponding clip. Given this tube as input, our Clip Tracking Network both estimates the pose of the central person in the

keyframe, and tracks his pose across the whole video clip (sec. 3.1, fig. 2). We call these tracklets. Next, our Video Tracking Pipeline works as a tailor to sew these tracklets together based on poses in overlapping frames (sec. 3.2, fig. 3). We call these multiple poses for the same person in same frame hypotheses. Finally, Spatial-Temporal merging polishes these predictions using these hypotheses in an optimization algorithm that selects the more spatially and temporally consistent location for each joint (sec. 3.3, fig. 4). In the next three sections we present these three components in details

### **4. Human Pose Estimation Combined with Depth Information**

Author : Yuan Sha, Ping Shi, Da Pan, Shaojing Zhou

Year : 2016

Findings :

In this paper, the resolution of color image and depth image is both 640\*480. Each pixel in depth map is composed of two bytes, 16 bits in total. Higher 13 bits represent the distance from the device to infrared camera to the headmost object. Because of the effective depth range of the camera is limited, the default depth offield ranges from 800 mm to 4000 mm. We will set the pixel value to 0 who is beyond the scope. Scaling the original depth data into the grayscale of 0-255 output darker pixels represent the farther object; the brighter represent the closer to the camera.

However such a figure directly used as segmentation mask of following position estimation process is not suitable, because the distribution of small pieces of wrong foreground area may also surround the human body. They will significantly cause large search space and complex computation. Especially for the lower arms part, they will have a great influence on estimating its appearance model. So the next step is to implement morphological processing on the mask image. From the results after opening and closing operation we can easily find the edge is smoothed, error points are removed, and the internal tiny holes are filled; with the whole body region has no obvious increase and decrease. But the large-area noise still remains in a small size. So, we drop them by setting a threshold.

In this paper, we have presented an improved pose estimation framework based on the state-of-art pose estimation method, access to a certain data and the current development of this problem. At first, this paper has introduced the basic principle of pose estimation, and understands the general process model based on the segmentation prior. To solve the foreground background segmentation problem, we used the depth data to reduce the interference of environment and manual work. On several subjects with different appearance and a variety of postures, we compared the results between the improved method and the original method, and analyzed the reasons of which may cause unsatisfactory problems. Through the experimental results, we confirmed that the improved method has reduced

uncertainties brought by the artificial intervention, so that improved foreground background segmentation effect which is crucial to the subsequent process.

### **5. Human Pose Estimation Using Convolutional Neural Networks**

Authors : Anubhav Singh, Shruti Agarwal, Preeti Nagrath, Anmol Saxena, Narina Thakur

Year : 2019

Findings :

The researcher uses Convolution Neural Network for human posture estimation where primary responsibility is a CNN cell structure especially planned for adjusting part associations and spatial contexts. The initial segment performs part identification heat maps and second part performs regression on these heat maps managing the framework where to focus in the image and enough encodes part requirements and context. Likewise, author shows that the planned course is sufficiently adaptable to expeditiously allow the blend of various CNN structures for both acknowledgment and relapse.

The authors in their paper proposed named Location based or relapse based verbalized human posture estimation. The Location constructed techniques are depending with respect to intense CNN-based part identifiers which are then solidified using a graphical model. Relapse-based strategies endeavor to take in a mapping from the picture and CNN features to part areas

The authors in their paper uses CNN which is trained for working on scene marking, by characterizing a multiclassification characterization for every pixel. Rather, they describe location over sliding windows in the image. Since, they allow their revelation over each window to contain various body parts, where every identification task is basically a paired characterization job in a window. Network optimization is been done with L2 loss without seeing the effect on outliers on the training process which would be affected by outliers

The author has used a regression network with ConvNets that attains robustness against these outliers by reducing Tukey's bi-weight function an m-estimator robust to outliers. The author illustrates quicker combination with improved speculation of the robust loss function for the estimating the poses from images of faces.



### **6. Multi-source Deep Learning for Human Pose Estimation**

Authors: Wanli Ouyang Xiao, Chu Xiaogang Wang

Year : 2014

Findings :

This paper has proposed a multi-source deep model for pose estimation. It non-linearly integrates three information sources: appearance score, deformation and appearance mixture type. These information sources are used for describing different aspects of the single modality data, which is the image data in our pose estimation approach. Extensive experimental comparisons on three public benchmark datasets show that the proposed model obviously improves the pose estimation accuracy and outperforms the state of the art. Since this model is a post-processing of information sources, it is very flexible in terms of integrating with existing approaches that use different information sources, features, or articulation models. Learning deep model from pixels for pose estimation and analyzing the influence of training data number will be the future work.

Our model extracts high-order representations of appearance, deformation and mixture types and better models their dependence at the top layer. For example, if the mixture types are upright upper- and lower-arms, the weighted combination of the locations of wrist and shoulder is a good estimation on the location of elbow. If the mixture types change, such estimation should change correspondingly. Such complex dependence cannot be modeled linearly and deep model is a better solution. When different information sources are extracted separately with the first several layers, the connections across sources are removed and the number of parameters is reduced.

### **7. Toward fast and accurate human pose estimation via soft-gated skip connections**

Authors : Adrian Bulat, Jean Kossaifi, Georgios Tzimiropoulos, Maja Pantic

Year : 2020

Findings :

This paper is on highly accurate and highly efficient human pose estimation. Recent works based on Fully Convolutional Networks (FCNs) have demonstrated excellent results for this difficult problem. While residual connections within FCNs have proved to be quintessential for achieving high accuracy, we re-analyze this design choice in the context of improving both the accuracy and the efficiency over the state-of-the-art. In particular, we make the following contributions: (a) We propose gated skip connections with per-channel learnable parameters to control the data flow for each channel within the module within the macro-



module. (b) We introduce a hybrid network that combines the HourGlass and U-Net architectures which minimizes the number of identity connections within the network and increases the performance for the same parameter budget. Our model achieves state-of-the-art results on the MPII and LSP datasets. In addition, with a reduction of  $3\times$  in model size and complexity, we show no decrease in performance when compared to the original HourGlass network.

In this they proposed gated skip connections with per-channel learnable parameters to control the data flow for each channel within the module. This has the simple effect to learn how much information from the previous stage is propagated into the next one per channel and encourages each module learn more complicated functions.

### **8. Real-Time Human Pose Recognition in Parts from Single Depth Images**

Authors : Jamie Shotton Andrew Fitzgibbon Mat Cook

Year : 2020

Findings :

We propose a new method to quickly and accurately predict 3D positions of body joints from a single depth image, using no temporal information. We take an object recognition approach, designing an intermediate body parts representation that maps the difficult pose estimation problem into a simpler per-pixel classification problem. Our large and highly varied training dataset allows the classifier to estimate body parts invariant to pose, body shape, clothing, etc. Finally we generate confidence-scored 3D proposals of several body joints by re-projecting the classification result and finding local modes.

The system runs at 200 frames per second on consumer hardware. Our evaluation shows high accuracy on both synthetic and real test sets, and investigates the effect of several training parameters. We achieve state of the art accuracy in our comparison with related work and demonstrate improved generalization over exact whole-skeleton nearest neighbour matching.

### 9. Recurrent Human Pose Estimation

Authors : Matthew Chen, Melvin Low

Year : 2018

Findings :

We hypothesize that estimating joint locations sequentially will be able to better capture the dependency between joint locations since we explicitly condition on previous joint decisions. We choose to use a recurrent neural network model because it is able to express dependencies across sequences. Such models are common in natural language tasks which are naturally modelled as sequences. To test this hypothesis we choose our comparison baseline model to be a generic CNN which regresses all joint locations in one pass. This is similar to the method proposed in DeepPose, however we do not implement their cascade of network classifiers so we can isolate the effects of the sequence modelling without the added effect of refinement through higher resolution. For our experimental model we use a RNN whose inputs are feature which were extracted from the last convolutional layers of the CNN used for the base model

In this project, we investigated the possibility of modelling pose estimation as a sequence task. We tested this hypothesis via use of a convolutional network linked to a recurrent one. For comparison, we also tested just the convolutional network on the same task. Our preliminary results show that the CNNRNN performed worse than the CNN. We also found that adding attention did not improve the result of the CNN-RNN. Further work can will expand upon specific reasons why the the model performed worse than the baseline with additional quantitative results to test our hypothesis.

### 10.The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation

Authors : Tewodros Legesse Munea, Yalew Zelalem Jembre

Year : 2018

Findings :

This paper presented a review of the most outstanding and influential models in human pose estimation progress. As introduced early a 2D human pose estimation has been a fundamental yet challenging problem in computer vision. The main objective of human pose estimation is to localize human anatomical key points (e.g., head, shoulder, elbow, wrist, etc.) or joints. This article started by introducing human pose estimation, then classified pose estimation based on tracing the number of people as a single or multi-person. Furthermore, approaches used in pose estimation are explored before discussing its applications and flaws.

## Human Pose Estimation

---

Thus, this article provides a guideline for new readers about human pose estimation. Furthermore, this paper can be a base for research to innovate new models by combining the techniques used in different papers mentioned above. This can be done by changing the backbone architecture or combining the two or three models to create new, or adding new architecture on one of the mentioned papers.

There are very large datasets publicly available on the net. Using these datasets, we have seen substantial progress in 2D human pose estimation with deep learning. However, in addition to the issues discussed in the summary and discussion section, some challenges remain to be addressed in the near future works. Such as i) occlusion of body parts by clothes and other people, ii) interactions between people, iii) human body structure constraints, and iv) barely visible joints are some of the prominent issues that need immense attention to be resolved in the coming works.

SI No	Title	Year	Author	Methods
1	2D Human Pose Estimation from Monocular Images: A Survey	2020	Sun Jingtian, ChenXue, Lu Yanan, Cao Jianwen	By estimating human body joints location.
2	A Study on the Learning Based Human Pose Recognition	2017	Faisal Sajjad; Adel F. Ahmed	SVM
3	Combining detection and tracking for human pose estimation in videos Manchen Wang, Joseph	2020	Manchen Wang; Joseph Tighe	Body joint detection and tracking simultaneously on video clips
4	Human Pose Estimation Combined with Depth Information	2016	Yuan Sha; Ping Shi	Image segmentation

## Human Pose Estimation

---

5	Human Pose Estimation using Convolution Neural Networks	2019	Anubhav Singh, Shruti Agarwal	CNN
6	Multi-source Deep Learning for Human Pose Estimation	2014	Wanli Ouyang; Xiao Chu	Multi-source deep model , SVM
7	Toward fast and accurate human pose estimation via soft-gated skip connections	2020	Adrian Bulat, Jean Kossaifi, Georgios Tzimiropoulos, Maja Pantic	FCN
8	Real-Time Human Pose Recognition in Parts from Single Depth Images	2020	Jamie Shotton Andrew , Fitzgibbon Mat Cook	SVM
9	Recurrent Human Pose Estimation	2018	Matthew Chen, Melvin Low	ConvNet model
10	The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation	2018	Tewodros Legesse Munea , Yalew Zelalem Jembre	Skeleton based, SVM

## CHAPTER – 3

### DATASET

#### 3.0 Dataset Description

Principally every state-of-the-art (SOTA) pose estimation model includes a component that detects body joints or estimates their position and making pairwise terms between body part hypotheses which assist categorizing the pairwise terms into valid human pose configurations. In doing so, some challenges are faced. Such as position and scale of each person in the image; barely visible joints; interactions between people, which brings complex spatial interference due to clothing, lighting changes, contact, occlusion of individual parts by clothes, backgrounds, and limb articulations which makes the association of parts difficult.

Dataset links : <http://human-pose.mpi-inf.mpg.de/#overview>

<https://cocodataset.org/#keypoints-2018>

#### 3.1 MPII Human Pose Dataset

##### Introduction

MPII (Max Planck Institute for Informatics) Human Pose dataset is a state of the art benchmark for evaluation of articulated human pose estimation. The dataset includes around 25K images containing over 40K people with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities. Overall the dataset covers 410 human activities and each image is provided with an activity label. Each image was extracted from a YouTube video and provided with preceding and following un-annotated frames. In addition, for the test set we obtained richer annotations including body part occlusions and 3D torso and head orientations.

MPII Human Pose dataset contains around 25,000 images from which composed of more than 40,000 individuals with annotated body joints. These images are collected on the purpose to show human activities every day. In MPII human pose dataset, each individual's body is labeled with 18 body joints as mentioned in the introduction section. As FLIC dataset, MPII Human pose dataset has also been used for a single person and multiperson pose estimation models.

Following the best practices for the performance evaluation benchmarks in the literature we withhold the test annotations to prevent overfitting and tuning on the test set. We are working on an automatic evaluation server and performance analysis tools based on rich test set annotations.

Therefore for the advancement and solution for this challenging problem MPII Human Pose dataset has been used for analyzing human postures. It involves around 25k images of 40k people having various illustrated body joints .Every picture was collected from YouTube videos and has former and un-illustrated frames involved 3D torso and head movements.

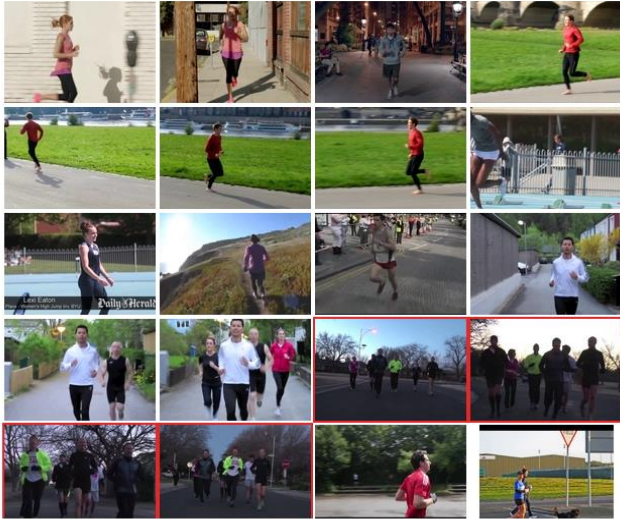
## Human Pose Estimation

Therefore the proposed benchmark “MPII Human Posture” fundamentally progresses in terms of appearance variability and complexity, and incorporates in excess of 40,000 pictures of individuals or full-body human posture estimation.

While the methodology depends on convolutional systems (convnets), testing dataset, and furthermore an investigation of what is required to make convnets work in estimation of human pose. The model inputs a color image having size  $w \times h$  and outputs 2D positions of key points for each person present in the picture itself. The model has layers used for creation of input image has two branch multistage CNN where first branch forecast about the confidence maps of various body parts localization like elbow, knee etc. and second branch tells about 2D vector fields(L) and tells about the degree of correlation between parts and therefore produce the 2D key points for all members present in the picture. The Multi-Person Dataset(MPII) is learned specialized model and has 15 outputs points

The MPII human pose estimation dataset, which is composed of 25,000 RGB images annotated with over 40,000 human poses. The images were collected by sampling 3,913 videos from YouTube in various frames, which are at least 5 seconds apart, and filtering images which contained people.

[Overview](#) [Browse](#) [Download](#) [Evaluation](#) [Results](#) [Related Benchmarks](#) [References](#) [Contact](#)

Activity Categories	Activities	Images
bicycling	jogging (74) - 913	
conditioning exercise	jogging, on a mini-tramp (26) - 983	
dancing	running (148) - 998	
fishing and hunting	running, cross country (41) - 280	
home activities	running, marathon (35) - 307	
home repair	running, on a track, team practice (29) - 653	
inactivity quiet/light	running, stairs, up (40) - 335	
lawn and garden	running, training, pushing a wheelchair or baby c... (1) - 7	
miscellaneous		
music playing		
occupation		
religious activities		
running		
self care		
sports		
transportation		
volunteer activities		
walking		
water activities		
winter activities		

### MPII Dataset

### 3.2 COCO Pose Dataset

#### Introduction

The COCO Key-point Detection Task requires localization of person key-points in challenging, uncontrolled conditions. The key-point task involves simultaneously detecting people *and* localizing their key-points (person locations are *not* given at test time). For full details of this task please see the key-point evaluation page.

This task is part of the Joint COCO and Mapillary Recognition Challenge Workshop at ECCV 2018. For further details about the joint workshop please visit the workshop page. Please also see the related COCO detection, stuff, and panoptic tasks.

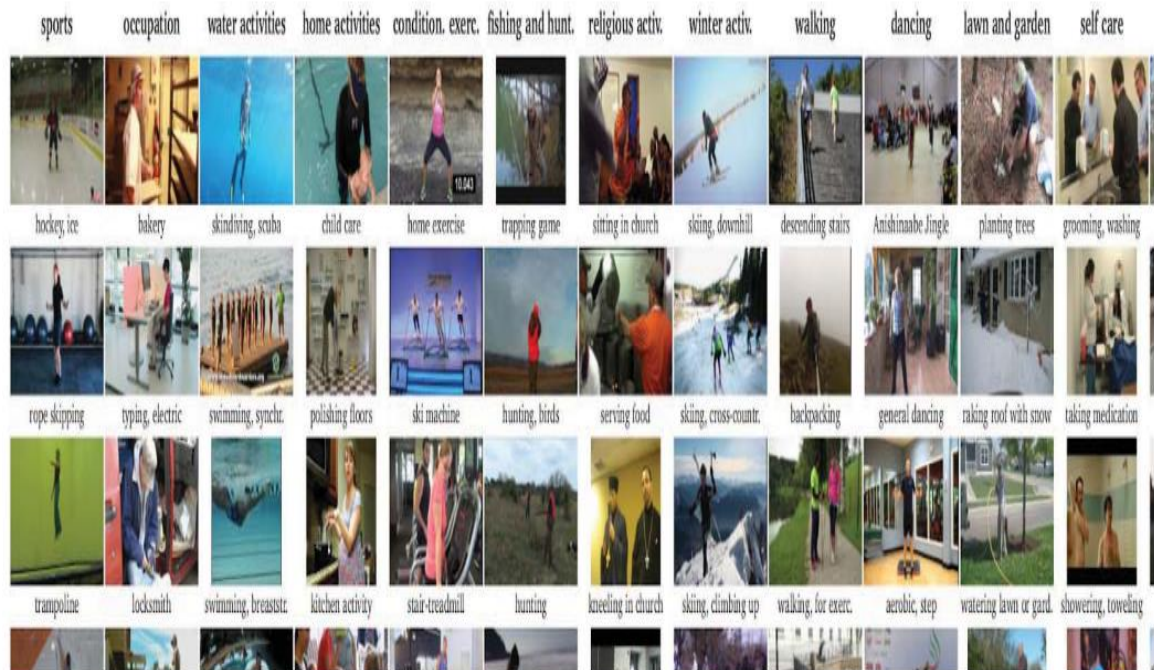
The COCO train, validation, and test sets, containing more than 200,000 images and 250,000 person instances labeled with key-points (the majority of people in COCO at medium and large scales) are available for download. Annotations on train and val (with over 150,000 people and 1.7 million labeled key-points) are publicly available.

This is the third iteration of key-point task and it exactly follows the COCO 2017 Key-point Detection Task. In particular, the same data, metrics, and guidelines are being used for this year's task.

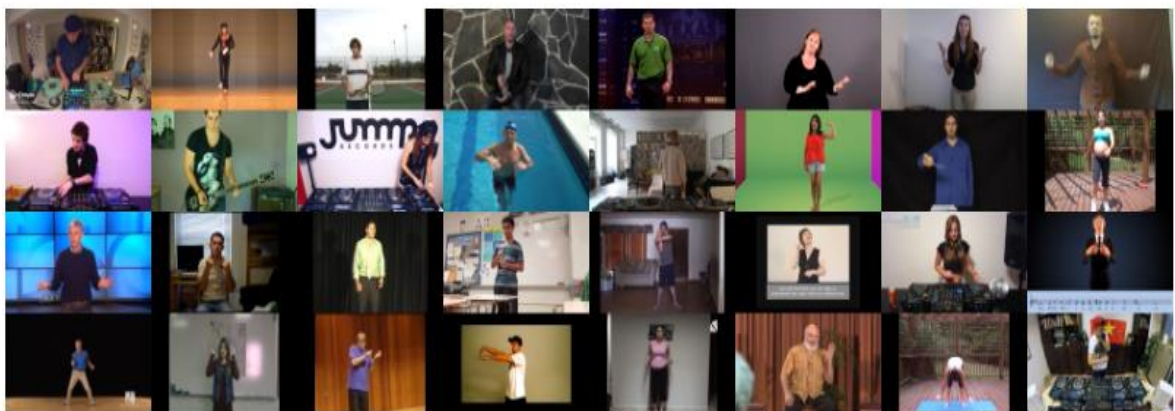
Finally, the MS-COCO dataset has got huge attention for multi-person pose estimation models. MS-COCO or usually called COCO (Common Objects in Context) is a product of Microsoft (MS). COCO dataset is a collection of a very large dataset with annotation types of object detection, key-point detection, stuff segmentation, panoptic segmentation, and image captioning. A JSON file is used to store annotations. COCO dataset brought to the table a very interesting mix of data, with various human poses used in different body scales, also containing occlusion patterns, with unconstrained environments. COCO dataset contains a total of 200,000 images and these contain 250,000 people with key-points from which each individual's instance is labeled with 18 joints. COCO dataset has been producing dataset starting from 2014 with a large amount.



# Human Pose Estimation



## YouTube Pose [1]



## COCO Pose Dataset



# CHAPTER – 4

## SYSTEM DESIGN

### 4.0 System Design

Detailed design starts after the system design phase is completed and the system design has been certified through the review. The goal of this phase is to develop the internal logic of each of the modules identified during system design.

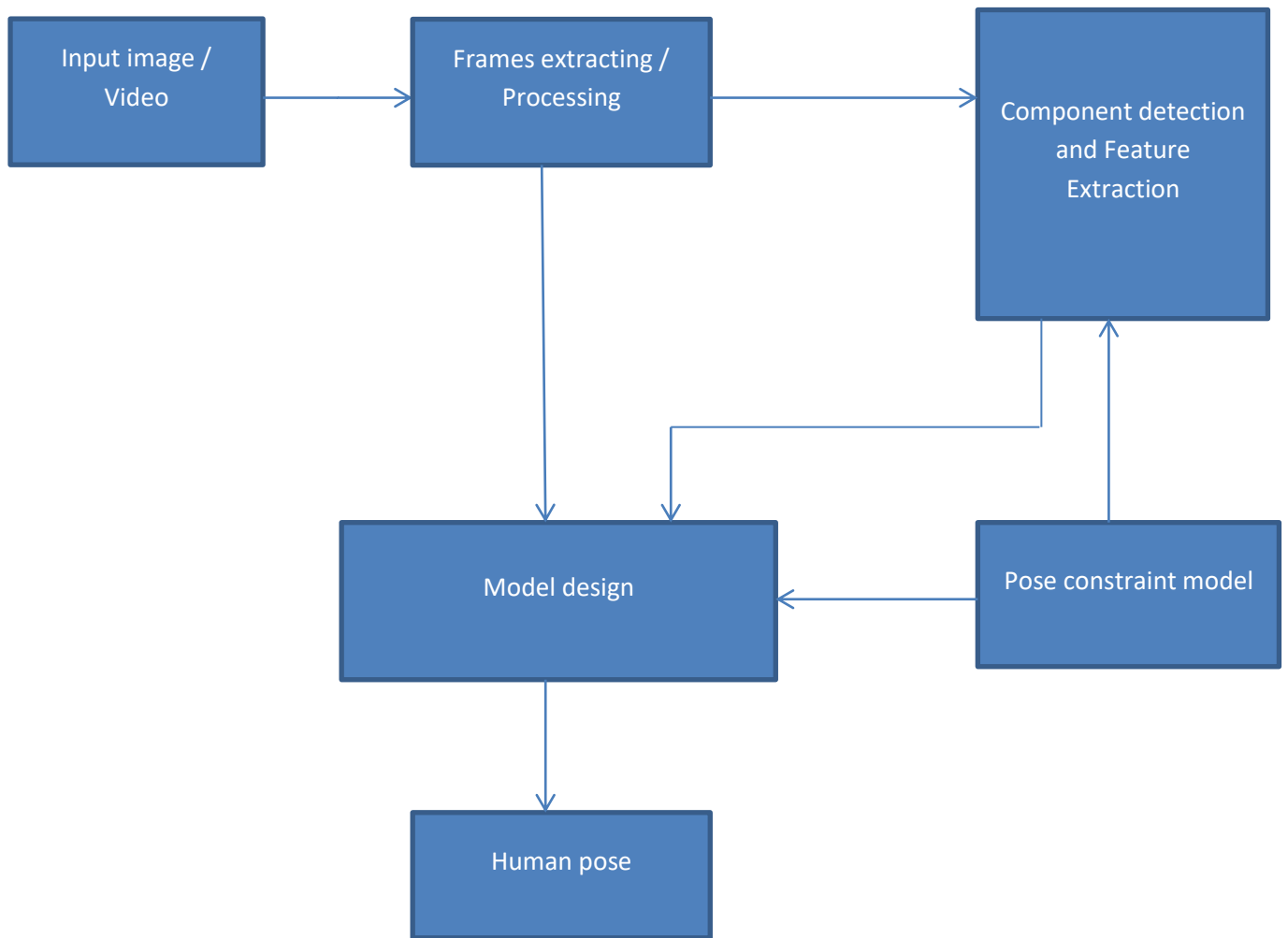
In the system design, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for the modules. In other words in system design attention is on what components are needed, while in detailed design how the components can be implemented in the software is the issue.

The design activity is often divided into two separate phase system design and detailed design. System design is also called top-level design. At the first level focus is on deciding which modules are needed for the system, the specifications of these modules and how the modules should be interconnected. This is called system design or top level design. In the second level the internal design of the modules or how the specifications of the module can be satisfied is decided. This design level is often called detailed design or logic design.

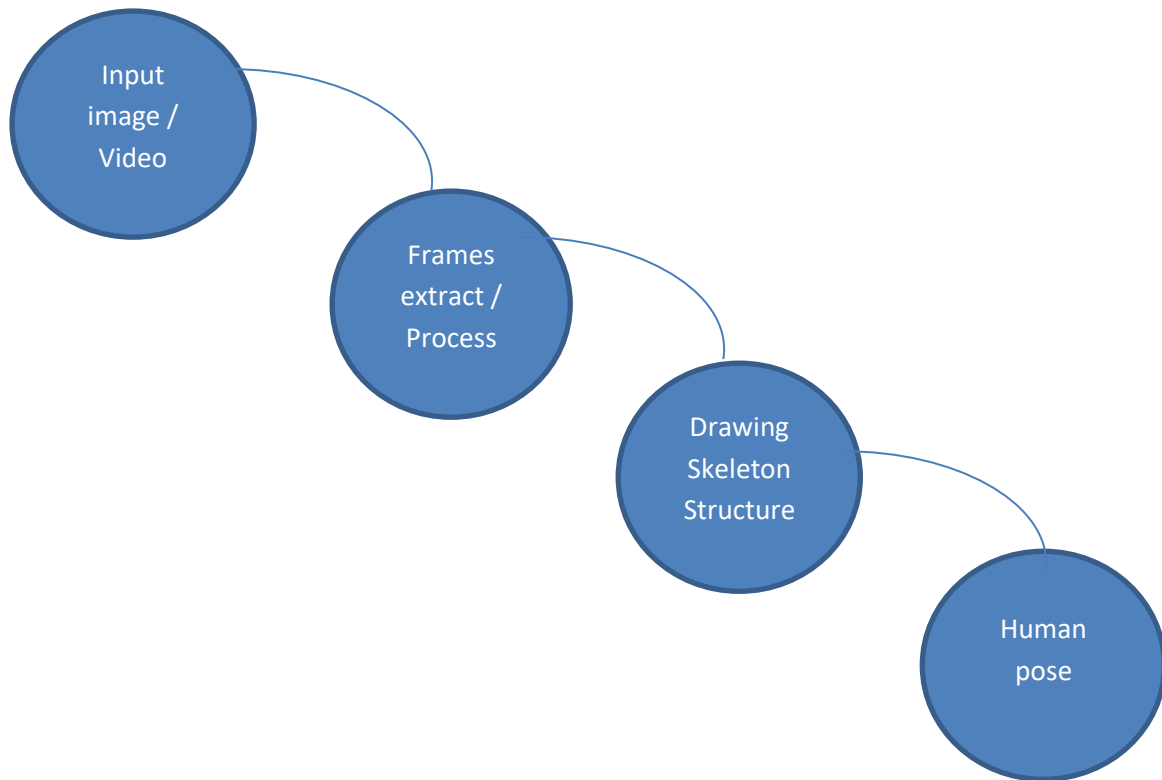
### 4.1 Dataflow Diagram

DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

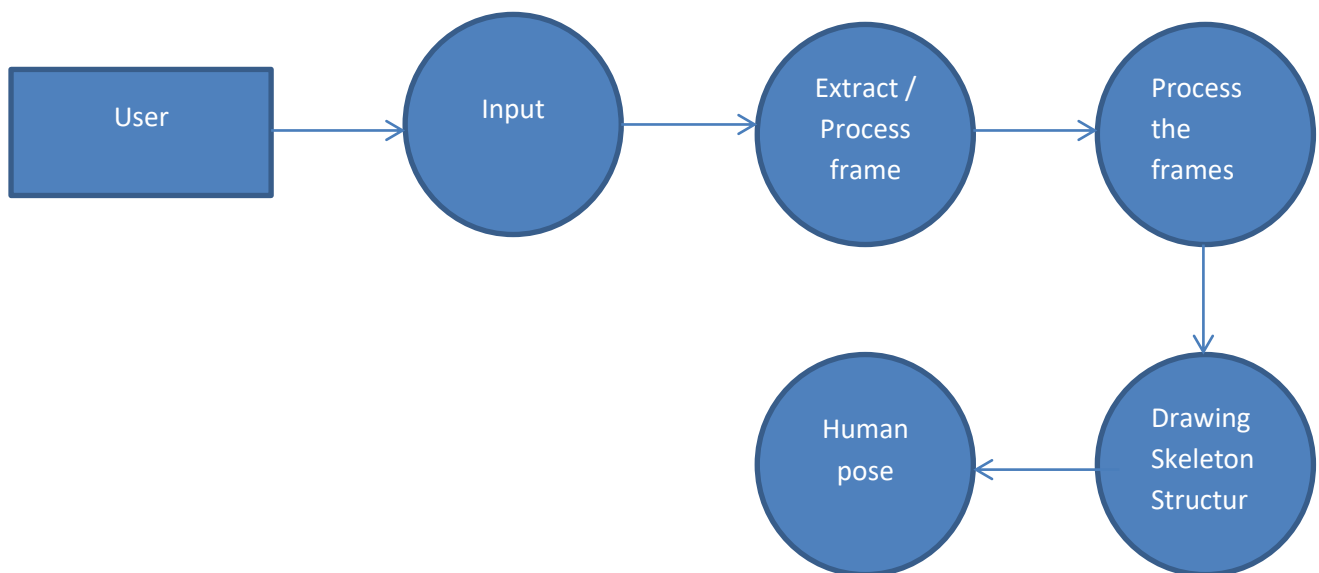
- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements



**Fig 4.4.1 : General Dataflow Diagram**



**Fig 4.1.2 : Level 0 - Dataflow Diagram**

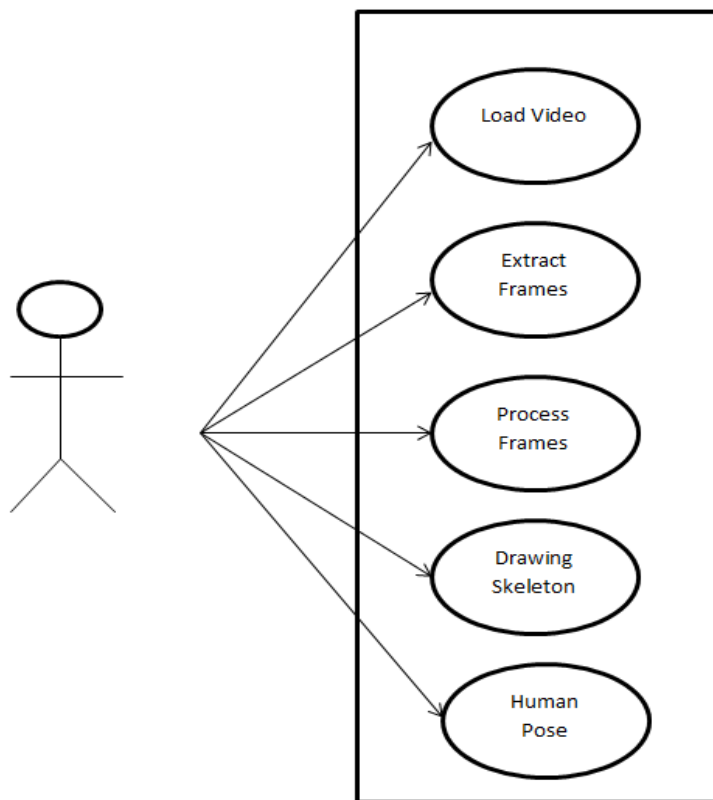


**Fig 4.1.3: Level 1 – Dataflow Diagram**

### 4.2 Usecase Diagram

Use case diagram is a graph of actors, a hard and fast of use instances enclosed by means of a device boundary, conversation associations among the actor and the use case. The use case diagram describes how a gadget interacts with out of doors actors; each use case represents a bit of functionality that a machine provides to its users. A use case is called an ellipse containing the call of the use case and an actor is shown as a stick figure with the call of the actor beneath the parent.

The use instances are used at some point of the evaluation phase of a task to pick out and partition system capability. They separate the device into actors and use case. Actors represent roles which might be played by using person of the system. Those users may be people, different computer systems, portions of hardware, or maybe other software structures.



**Fig 4.2.1 : Usecase Diagram**

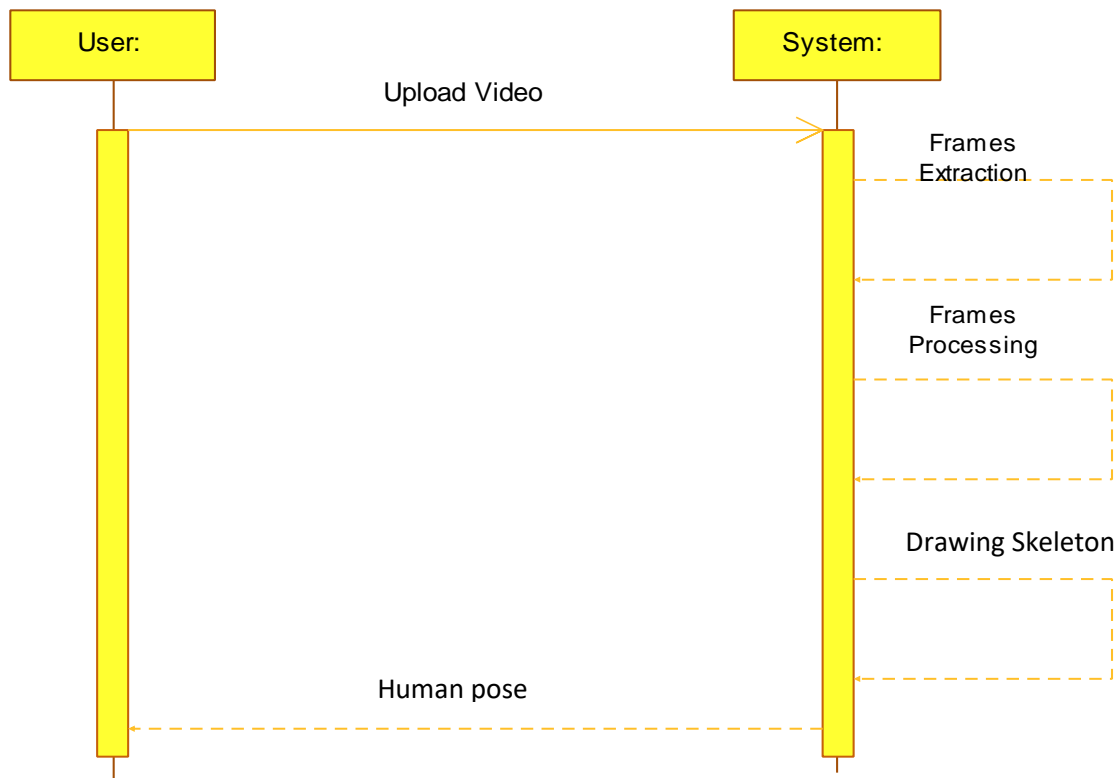
### 4.3 Sequence Diagram :

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called event diagrams, event scenarios.

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram. Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behaviour.

#### 4.3.1 Purpose

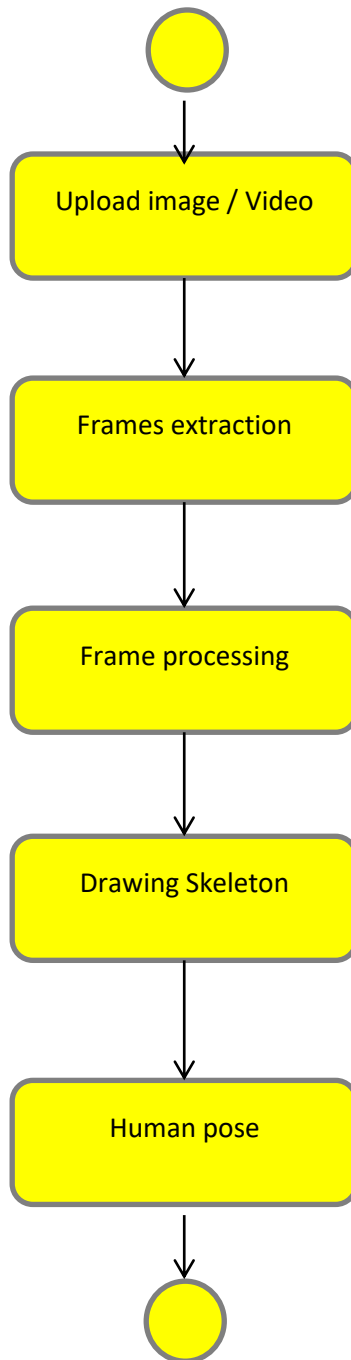
The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement.



**Fig 4.3.2 : Sequence Diagram**

### 4.4 Activity Diagram

In this developed project the Activity diagrams illustrate the overall flow of control. This diagram symbolizes the goings-on taking place in the project. There are different accomplishments for member.



**Fig 4.4.1 : Activity Diagram**

# CHAPTER – 5

## SYSTEM IMPLEMENTATION

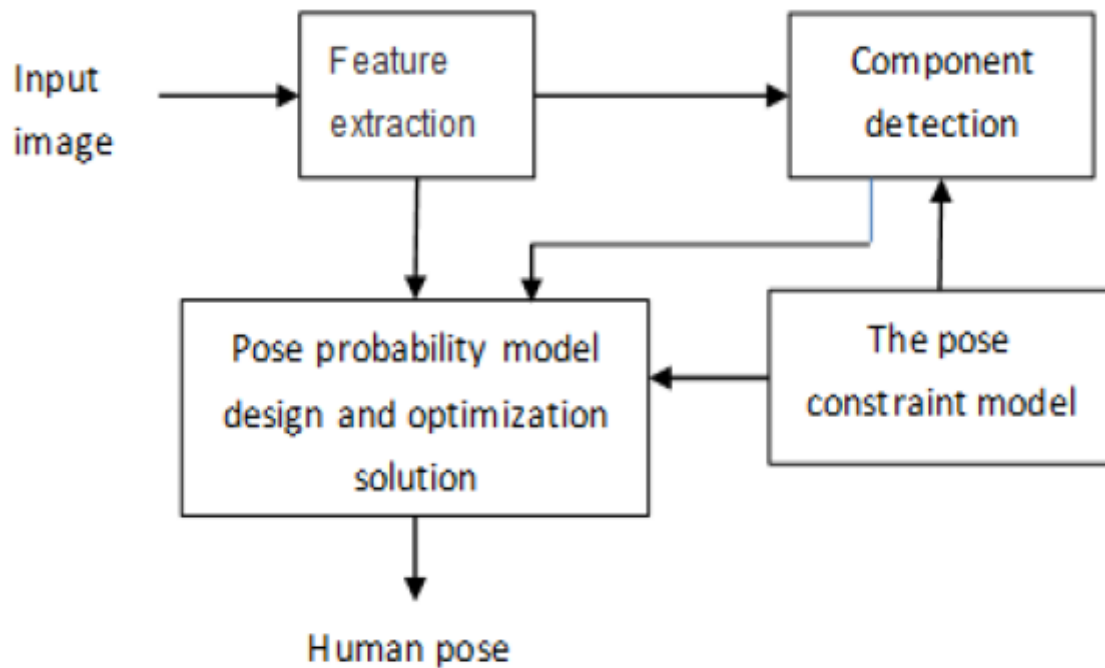
### 5.0 Introduction

The most critical step of any project development is the implementation phase, which produces the final solution that answers the problem at hand. In order to produce the required final report, the implementation step involves the actual materialisation of ideas that are described in the analysis document in an appropriate programming language. The coding phase should be directly related to the design .

In this project I have implemented using python which is an object-oriented programming language and procedure-oriented. programming language. Object-oriented programming is an approach that provides a way of the modularizing program by creating partitioned memory area of both data and function that can be used as a template for creating copies of such module on demand. And procedure oriented programming is an approach to break down the task into collection of variables through the sequence of instructions.

This project is implemented using python programming language. Python is very useful for type dynamically and it also have garbage-collection feature. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is oftendescribed as a "batteries included" language due to its comprehensive standard library. The machine Learning techniques are used in this project.





**Fig 5.0.1 : Software Architecture**

### 5.1 Python :

Python is a backend programming language. Python is similar in many ways to Ruby, but is less verbose than other programming languages - a little less wordy. It can be used to create a variety of different programs and isn't specialized for any specific problems. Python is progressively composed and trash gathered. It underpins numerous programming standards, including procedural, object-arranged, and practical programming. Python is frequently portrayed as a "batteries included" language because of its thorough standard library. Python is a multi-worldview programming language. Article arranged programming and organized writing computer programs are completely upheld, and a significant number of its highlights uphold useful programming and angle situated programming counting by meta programming and meta objects (enchantment methods). Many different standards are upheld by means of expansions, including plan by agreement and rationale programming.

### 5.2 Machine Learning V/S Deep Learning

Experts in machine learning and deep learning have not yet reached consensus on these concepts. In this context, almost every day new ideas are being discussed. Machine Learning is an older concept than Deep Learning. Deep learning can also be called a technique that performs machine learning. The differences are listed below;

- In deep learning, too much data is needed to bring the algorithm structure to the ideal. In machine learning, the problem can be solved with much less data because the person gives specific features to the algorithm.
- Deep learning algorithms try to extract features from data. In machine learning, the features are determined by the expert.
- While Deep Learning algorithms work on high performance machines, Machine Learning algorithms can work on ordinary CPUs.
- In machine learning, the problem is usually divided into pieces, these parts are solved one by one and then the solutions are formed as a result of the solutions. In deep learning, the problem is solved end-to-end. It takes a long time to train deep learning algorithms.

### 5.3 Deep Learning Overview

The term Deep Learning or Deep Neural Network refers to Artificial Neural Networks (ANN) with multiple layers. Over the last few decades, it has been considered to be one of the most powerful tools, and has become very popular in the literature as it is able to handle a huge amount of data. The interest in having deeper hidden layers has recently begun to surpass classical methods performance in different fields; especially in pattern recognition. One of the most popular deep neural networks is the Convolutional Neural Network (CNN). It takes this name from mathematical linear operation between matrixes called convolution. CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully connected layer. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters. The CNN has an excellent performance in machine learning problems. Specially the applications that deal with image

data, such as largest image classification data set (Image Net), computer vision, and in natural language processing (NLP) and the results achieved were very amazing

In this paper we will explain and define all the elements and important issues related to CNN, and how these elements work. In addition, we will also state the parameters that effect CNN efficiency. This paper assumes that the readers have adequate knowledge about both machine learning and artificial neural network.

Convolutional Neural Network has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN . This achievement has prompted both researchers and developers to approach larger models in order to solve complex tasks, which was not possible with classic ANNs; The most important assumption about problems that are solved by CNN should not have features which are spatially dependent. In other words, for example, in a chest X ray detection application, we do not need to pay attention to where the chest X rays are located in the images. The only concern is to detect them regardless of their position in the given images . Another important aspect of CNN, is to obtain abstract features when input propagates toward the deeper layers.

## 5.4 Libraries and Modules used

### 5.4.1 Pandas

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data.

### 5.4.2 OS

OS provides functions for the interaction with the Operating system.

### 5.4.3 PIL

Python Imaging Library is the open-source library to the python programming language which helps for opening, manipulating, and saving the many different images file formats. It

incorporates the lightweight image processing tool for creating, editing and saving the images.

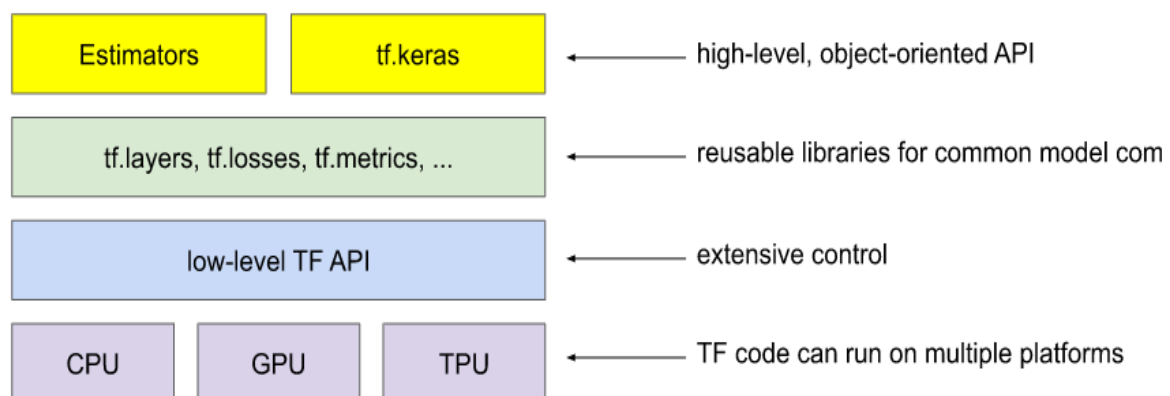
### 5.4.4 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Tensorflow library is used to fast numerical computing/calculations. And it is used to create Deep learning models directly or by using the wrapper library which simplifies the process.

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.

TensorFlow APIs are arranged hierarchically, with the high-level APIs built on the low-level APIs. Machine learning researchers use the low-level APIs to create and explore new machine learning algorithms. In this class, you will use a high-level API named `tf.keras` to define and train machine learning models and to make predictions. `tf.keras` is the TensorFlow variant of the open-source Keras API.



### 5.4.5 Keras

Keras is open-source Software Library which provide the interface for Neural network. Keras also acts as the interface for the tensorflow Library. It is also uses for Developing and Evaluating Deep learning algorithms

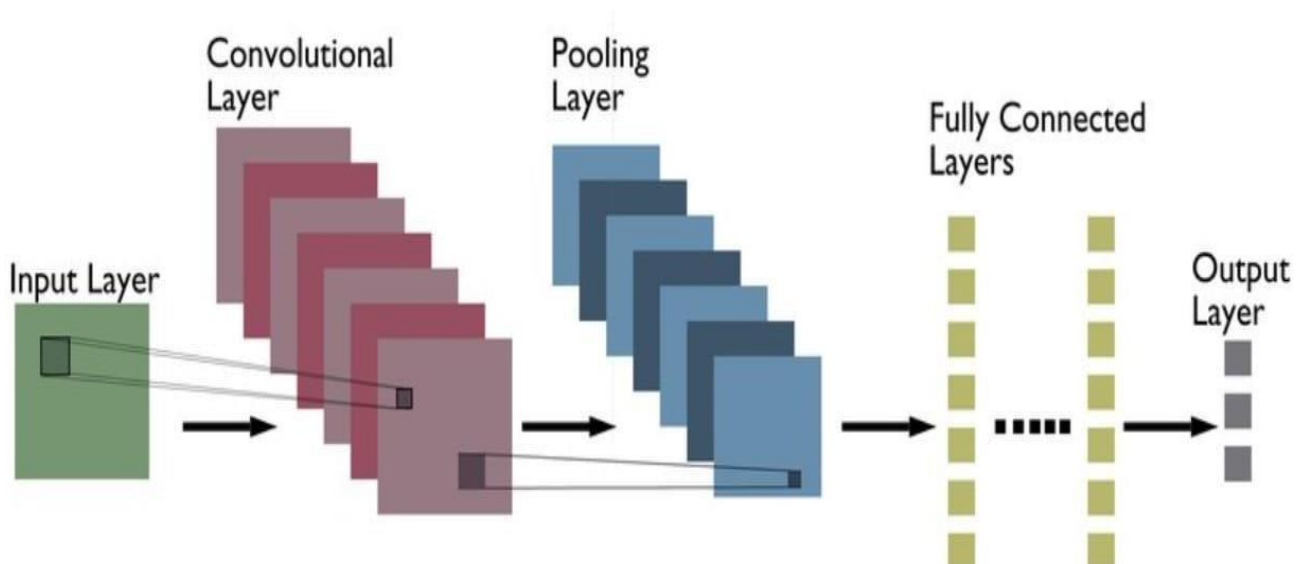
### 5.4.6 OpenCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers. It promotes OpenVisionCapsules, which is a portable format, compatible with all other formats.

## 5.5 Algorithm used :

### 5.5.1 CNN (Convolutional Neural Networks):



**Fig 5.5.2 : CNN (Convolutional Neural Networks) Architecture**

Convolutional Neural Networks have the following layers:

- Convolutional Layer
- ReLU Layer
- Pooling
- Fully Connected Layer
- Softmax / Logistic Layer
- Output Layer

### Convolution Layer :

The real power of deep learning, especially for image recognition, comes from convolutional layers. It is the first and the most important layer. In this layer, a CNN uses different filters to convolve the whole image as well as the intermediate feature maps, generating various feature maps. Feature map consists of a mapping from input layers to hidden layers. Convolutional neural networks apply a filter to an input to create a feature map that summarizes the presence of detected features in the input.

We have three hyper parameters to control the size of the output volume of the convolutional layer: the depth, stride, and zero-padding.

- The Depth  
Depth of the output volume controls the number of neurons in the layer that connect to the same region of the input volume. All of these neurons will learn to activate for different features in the input. For instance, if the first Convolutional Layer takes the raw image as input, then different neurons along the depth dimension may activate in the presence of various oriented edges, or blobs of color.
- Stride  
Stride controls how depth columns around the spatial dimensions (width and height) are allocated. When the stride is 1, then we move the filters one pixel at a time. This leads to heavily overlapping receptive fields between the columns, and also to large output volumes. When the stride is 2, then the filters jump 2 pixels at a time as we slide them around. The receptive fields will overlap less and the resulting output volume will have smaller dimensions spatially.
- Zero-padding  
Zero-padding and it is suitable to pad the input with zeros on the border of the input volume. Zero padding deals with the control of the output volume spatial size. In particular, sometimes it is needed to exactly preserve the spatial size of the input volume. For example (Fig. 8), the input volume is  $32 \times 32 \times 3$ . If we pad two borders of zeros around the volume, we obtain a  $36 \times 36 \times 3$  volume. Then, when we apply the

convolution layer with our 5x5x3 filters and a stride of 1, then we will also get a 32x32x3 output volume.

### ReLU Layer :

It is the Rectified Linear Units Layer. This is a layer of neurons that applies the non-saturating nonlinearity function or loss function:

$$f(x) = \max(0, x)$$

It yields the nonlinear properties of the decision function and the overall network without affecting the receptive fields of the convolution layer. We have saturated nonlinear functions that are much slower. Also, we have the tan(h) function: In this layer, we remove every negative value from the filtered images and replaces them with zeros .It is happening to avoid the values from adding up to zero.

Also, we have the tan(h) function:

$$f(x) = \tanh(x)$$

or the logistic sigmoid function

$$f(x) = 1/(1+e^{-x})$$

ReLU results in the neural network that is training several times rapidly, without making a significant difference to generalization accuracy. Rectified Linear unit (ReLU) transform functions only activates a node if the input is above a certain quantity. While the data is below zero, the output is zero, but when the information rises above a threshold. It has a linear relationship with the dependent variable.

### Pooling Layer

In the layer, we shrink the image stack into a smaller size. Pooling is done after passing by the activation layer. We do by implementing the following 4 steps:

- Pick a window size (often 2 or 3).
- Pick a stride (usually 2).
- Walk your Window across your filtered images.
- From each Window, take the maximum values.

### **Fully Connected Layer**

The last layer in the network is fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers

This mimics high-level reasoning where all possible pathways from the input to output are considered.

Then, take the shrunk image and put into the single list, so we have got after passing through two layers of convolution relu and pooling and then converting it into a single file or a vector.

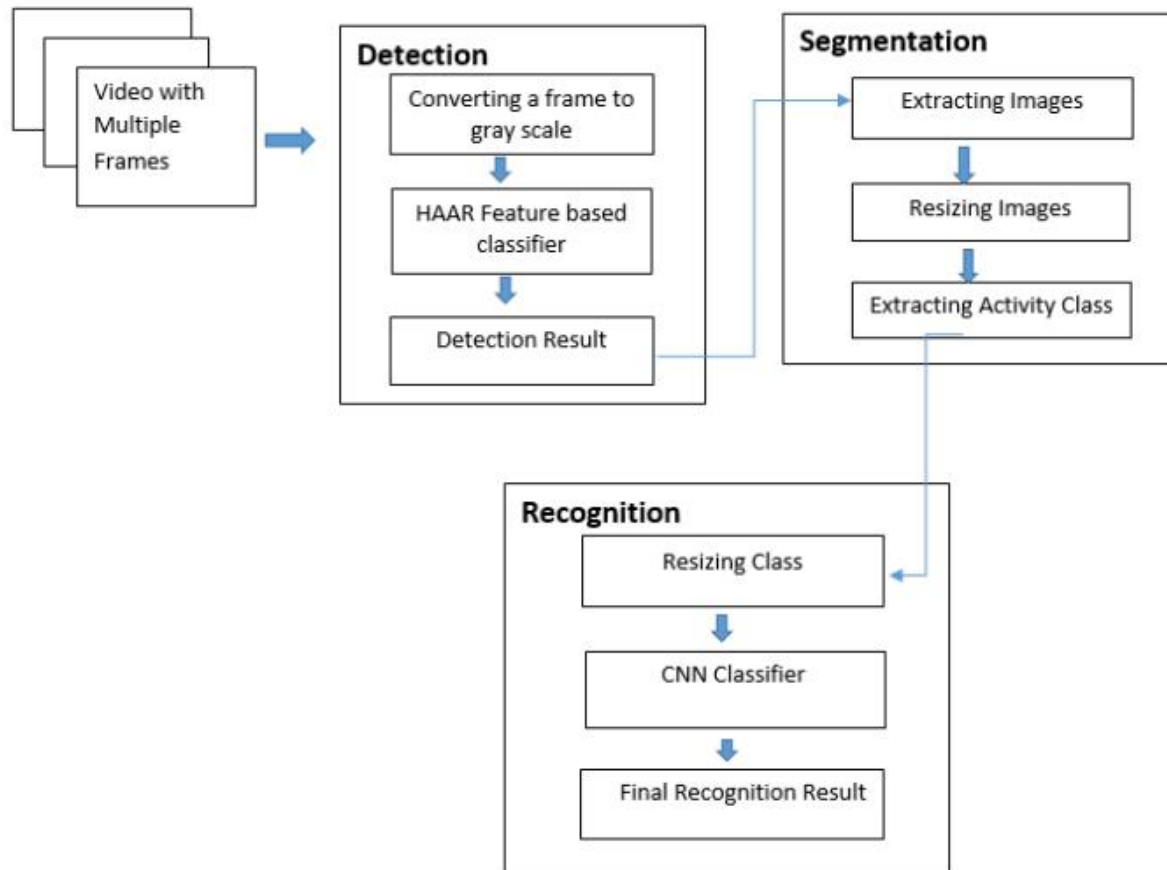
### **SoftMax / Logistic Layer :**

The SoftMax or Logistic layer is the last layer of CNN. Logistic is used for binary classification problem statement and SoftMax is for multi-classification problem statement.

### **Output Layer :**

This layer contains the label and classifies the Input Dataset.





**Fig 5.5.3 : System Architecture**

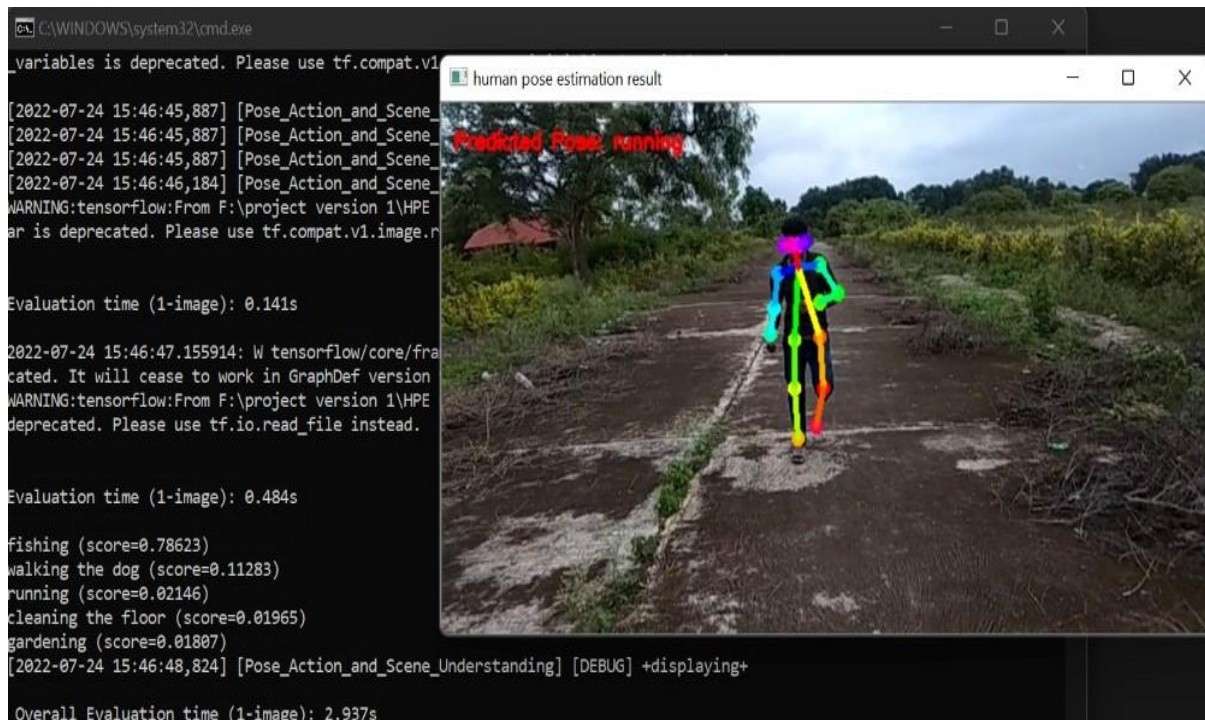
## CHAPTER-6

### DISCUSSION AND RESULTS

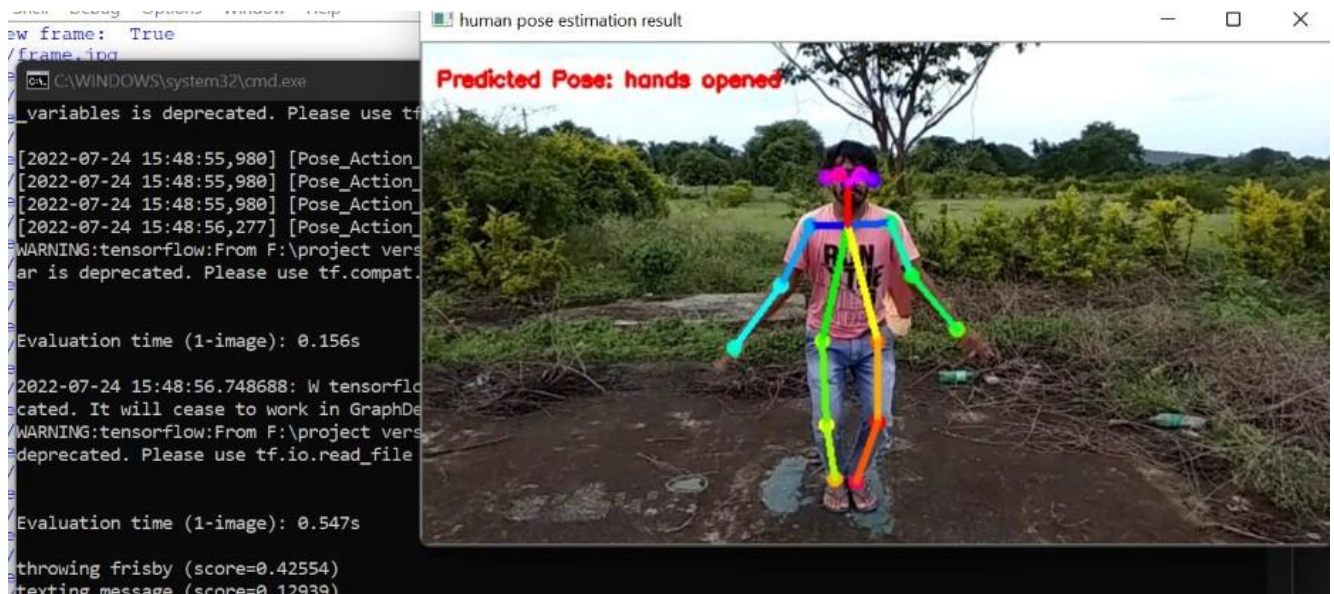
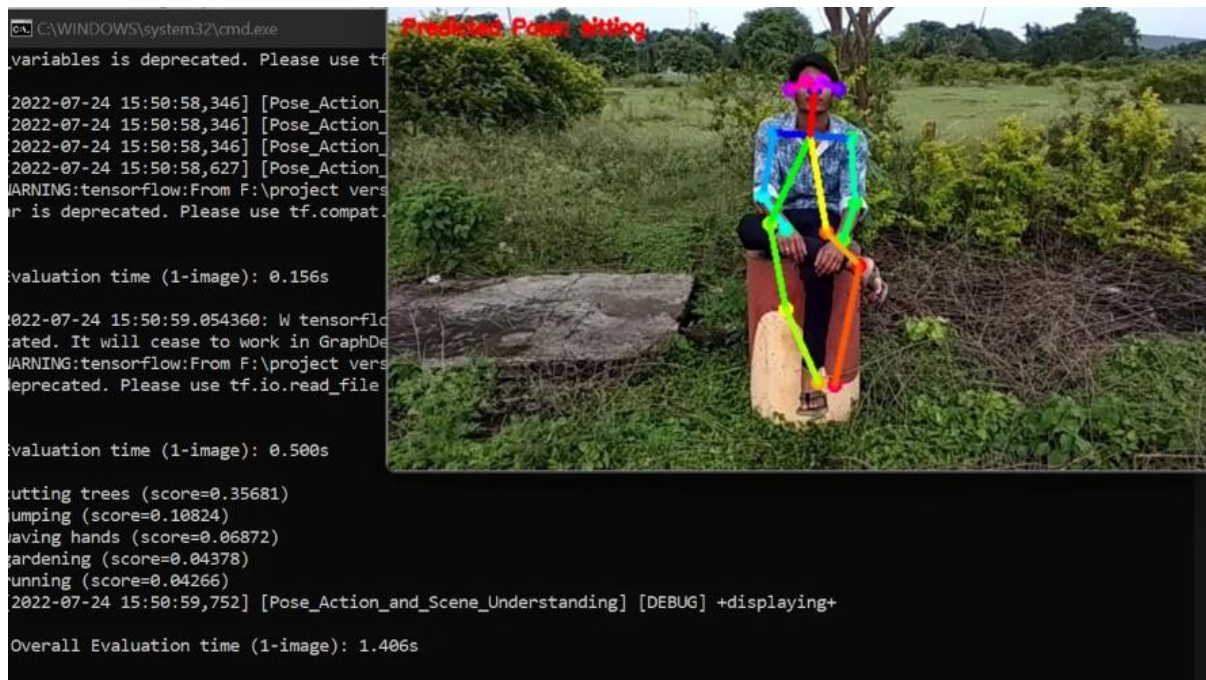
#### 6.1 Discussion

The Human pose estimation system is successfully implemented using python language Code and tested on some Activity videos. It takes frames from the given video and draw the skeleton structure by detecting the 18 key points to classify the pose.

#### 6.2 Results

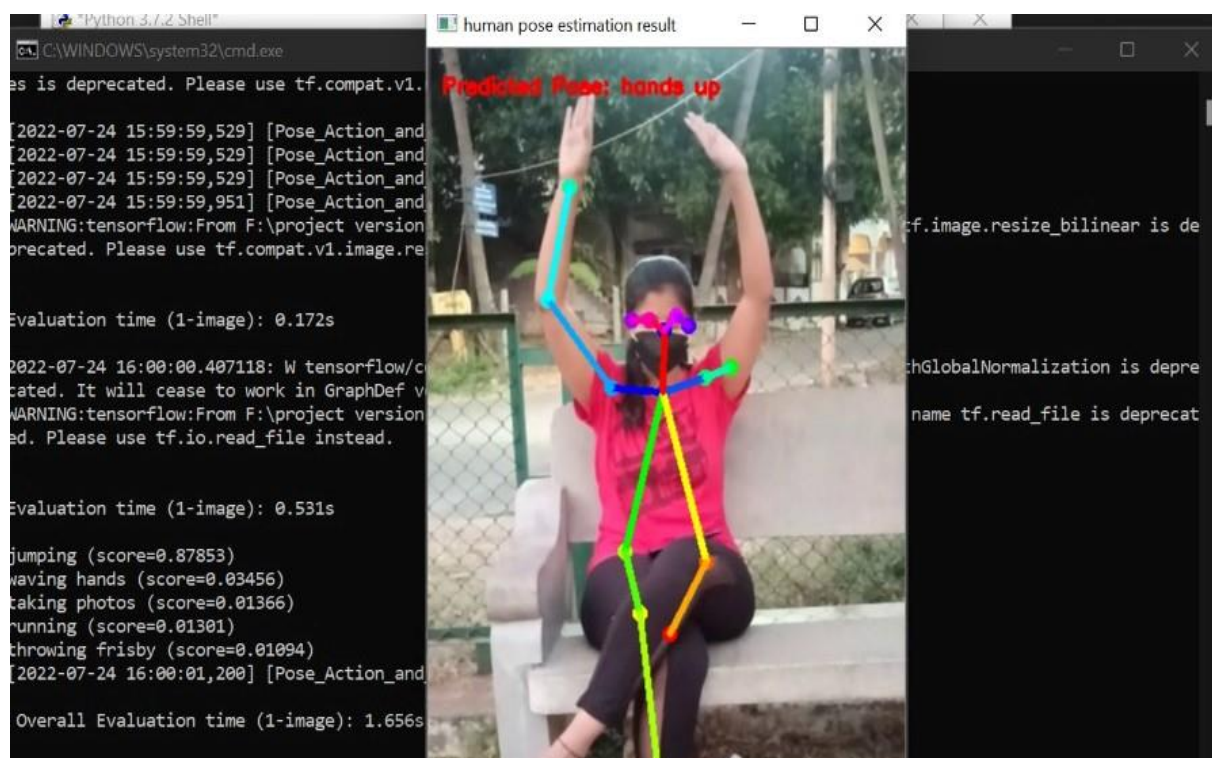


## Human Pose Estimation

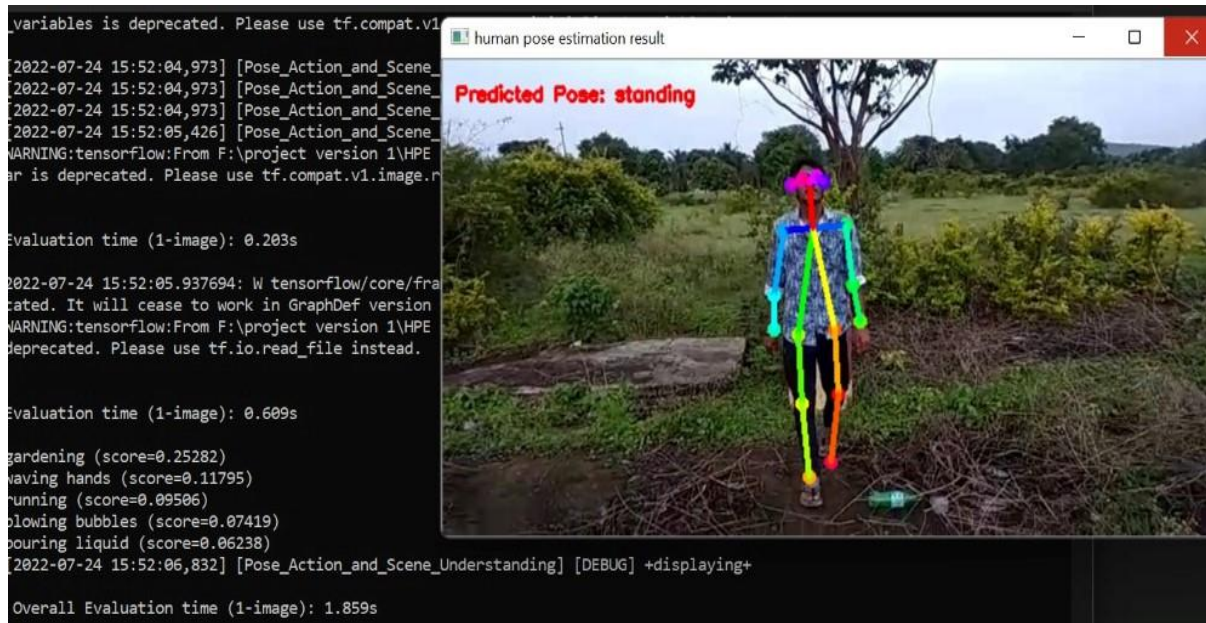




# Human Pose Estimation



# Human Pose Estimation



### CHAPTER – 7

## CONCLUSION AND FUTURE SCOPE

### 7.1 Conclusion

Summarizing all the introduction above, we can see a general line of how human pose estimation problem has been evolved through all these years: methods are kind of, in light of present perspective, primitive at the beginning to only try to capture the outline of human pose using features like silhouette and edges. Later, as people start to realize the importance to take correlation of body joints into consideration, different body models are established to reproduce the relationship between body parts. By that time, most body models are empirical and based on some impossible assumption. The great development of Convolutional Neural Network and Deep Learning recent years gives human pose estimation a huge boost just like other Computer Vision fields.

### 7.2 Future Scope

This project has various applications in the future like surveillance systems, human monitoring, action detection, gaming and many more, and this can be enhanced in future by Adopting some more algorithms and for another set of activities then by getting 100% of accuracy. We can build the models using Some more efficient Algorithms.

### CHAPTER - 8

### REFERENCES

- M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In CVPR, June 2014.
- V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In FG 2017. IEEE, 2017.
- A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In ECCV, 2016
- A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In ICCV, 2017.
- Y. Chen, C. Shen, H. Chen, X.-S. Wei, L. Liu, and J. Yang. Adversarial learning of structure-aware fully convolutional networks for landmark localization. IEEE TPAMI, 2019.
- ] C.-J. Chou, J.-T. Chien, and H.-T. Chen. Self adversarial training for human pose estimation. arXiv preprint arXiv:1707.02439, 2017.
- X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In CVPR, 2017.
- K. Greff, R. K. Srivastava, and J. Schmidhuber. Highway and residual networks learn unrolled iterative estimation. arXiv preprint arXiv:1612.07771, 2016.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016
- Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. CoRR, abs/1705.00389, 2017.