Chapter 3

THE TWO DEEP NETWORKS FOR 2D HUMAN POSE ESTIMATION

This chapter presents two deep learning models for 2D HPE. Section 3.1 discusses the theoretical background of the models. Section 3.2 gives literature for both methods. Section 3.3 presents the proposed methodologies. Section 3.4 shows the results and discussion for both models. At last, section 3.5 conclude this chapter.

3.1 Introduction

HPE is the technique of deriving an exact pixel position from human body key points. It is an essential tool to solve other high level tasks, such as human tracking, action recognition, human-computer interaction, motion capture, content retrieval, social signal processing, and animation [158].

HPE from the image is a tough job as the human being is completely articulated, few portions may not be apparent because of the low-quality image and the occlusion, and the visible features of the body parts can vary considerably from one to another pose.

The classical approaches utilize the joint detectors to obtain the local information, that was fused to develop the pictorial structures [159]. This technique suffers from the problem of occlusion. To handle difficult instances of partial visualization or occlusion, contextual information is usually needed to provide visual pieces of information that can be derived from the wide region throughout the part section [160] or by cooperation amongst the detected parts [161]. Chu et al. [9] proposed a deep network where CNN was utilized with the multi-context attention to estimate the human pose. So, the conclusion is that contextual information is highly useful to handle the occlusion problem.

Usually, HPE has been observed from two distinct perspectives, i.e detection, and regression problem. The detection based techniques produce a probability heatmap for every joint and recognize the joint with the highest value point in the heatmap [111] [158] [112]. The uncertainty of the heatmaps has been diminished by utilizing the dependence of the joints with various manners, such as, by exploiting the multi-stage procedure [111]. Such

techniques are preeminent for 2D HPE but do not comfortably generalize to the 3D HPE scenario [11], because of the high demand of computational and memory for the 3D heat maps.

Regression-based techniques directly regress output joints from the input image. The working is more direct and general. They do not work very well though. They are often used to estimate 3D poses [10] [162]; the result is not very satisfactory. The main difficulty is that they independently minimize the positioning errors per joint connection, but overlook the internal body structures of the pose.

Detection-based procedures commonly attempt to detect keypoints separately, which are aggregated in the post-processing steps to produce the prediction result. In opposition, approaches based on regression control a function that maps the input images directly to the joint coordinates.

There is a progression of constructing networks with branches, such as, ResNets [163], Inception models [164] and ASPP-nets [165] utilized for classification and semantic segmentation, Stacked hourglass networks [111] and convolution pose machines [166] for HPE, where one layer is input from several distinct layers or the product of one layer is utilized by other layers.

The above discussed observations motivate us to propose the first model of this chapter having a multi-stage deep architecture for HPE. The method utilizes the heatmap information while regressing the body joint locations. The contextual information has been also used to make system efficient towards occlusion.

The proposed method consists of three consecutive parts: first, as the traditional DCNN networks for HPE utilizes the single stream input branch [111], in contrast to these techniques, we propose a two-branch input DCNN network for feature extraction, which exhibits better result over single-branch techniques. The method is inspired by Inception-v4 [167] and VGG-19 [168]. We observe that the existing single stream feature extraction techniques using less width of CNN scheme are not that much efficient as with more layers with branches. Therefore, we perform a feature fusion concept based over Inception and VGG deep networks for feature extraction and shows that the resulted initialization of network with more branches into consideration gives better result over other state of the art techniques. Second, to facilitate the refinement, we proposed an cascaded feature integration technique on the stacked hourglass method as a basis. This work aims to study the issue of the existing stacked hourglass technique. We find out that the unsatisfactory performance of the method is mainly because of the imperfection in the design ideas. Because of the repetitive down and up sampling steps, there are high chances of data loss, and the optimization becomes more challenging. We propose a cascaded features integration across different stages to reinforce the data flow and alleviate the obstruction in training. At last, fusion is performed to the detected part heatmap with the context heatmap to make the system more accurate. Finally, the result attained by our approach outdoes that of regression and detection based state of the art.

The CNN recently highly utilized to improve the performance of computer vision tasks. HPE is a case, in which the joint location is predicted from the image, CNN is largely utilized for the same as [140][149]. Irrespective of the huge improvements occur due to

these fascinating model design and procedures, the problem seems to be that there is no perfect representation of features that easily differentiate the visual information and the different manifestations of human account. The whole condition has been changed after the evolution of deep learning in the field of computer vision. The CNN architectures have the ability to gather more and efficient image cues. Like in [111], CNN has utilized to process features among all scales to largely acquire the spatial information of the body for joint prediction. But accurate joint location prediction is difficult because the human body is complex due to its articulation by occlusion, foreshortening, change in viewpoint and body limbs. Due to the evaluation of deep learning, most of the recent techniques makes the model complex only to improve the performance or result. In spite of impactful result they have, they are compute-intensive and have a highly complex architecture which requires a very high configuration graphics card.

The second model of this chapter presents an efficient technique for HPE that utilize less GPU memory and give good comparable result. The main focus of the architecture is to easily learn the feature from multiple layers like [169], as this make system efficient towards difficult body context joints like ankles and wrists. So we do multiple layers feature integration for making system robust, we all know that the last layer features are efficient for classification purpose not for localization due to pooling and middle layers are for localization. There are many techniques in the literature which follows the same flow as ours by taking a detection system used for extracting human ROI and then apply pose estimator to predict the joint coordinates. The advantage of using this procedure is their proficiency in arranging a task into multiple subtask for making it easier and more

accurate. If the human detector is excellent in detecting the hard candidates, the HPE will usually get accurate with a focused regression space. Our technique give state of the art results on most of the popular datasets of human detection and pose estimation like INIRIA person dataset and MPII dataset.

The novelty of this work lies in the architecture of the proposed methods. The contributions of this chapter are:

- Introduced a multi-stage deep learning architecture for HPE, which gives high accuracy as by reducing the PCK error value compared to other recent state-of-the-art techniques.
- Proposed a two branch input DCNN network for Feature Extraction. This method is inspired by Inception-v4 and VGG-19 deep network.
- Proposed a cascaded feature integration technique on the stacked hourglass method as a basis for feature refinement.
- Utilized the detected part heatmap with the context heatmap to make the system efficient towards occlusion.
- Proposed a detection followed by estimation 2D HPE model that does not use very high configuration memory and also give better result in terms of PCK score compared to other state-of-the-art techniques.

Through the experimental demonstrations, we analyze that the calculated results of both the techniques outperform the other state-of-the-art techniques in terms of the PCK metric.

3.2 Literature

3.2.1 Multi-stage deep model for 2D human pose estimation

In literature, there have been many approaches for 2D and 3D HPE. The early classical based techniques handle pose estimation as poslet based [170] and pictorial structure-based prediction [159]. These classical approaches are largely replaced by the DCNN, the main reason for largely increased pose estimation performance over many other vision-based techniques. Mostly pose estimation techniques adapt to use DCNN as their core architecture part. Therefore, we are going to discuss only CNN based approaches for HPE.

Single-stage Network: Many of the single-stage approaches [171] [172] has based on the ResNet [173] or VGG [168] backbone systems, that are very well tuned for the task of classification. The authors [172] gives an architecture that creates a heatmaps with their absolute offsets to predict the body joints. Xiao et al. [171] proposed to use the deconvolutional layers over the ResNet to estimate the pose. The modified optical flow is used here to track the poses. Sun et al. [141] focuses to learn the high resolution representation using multi-scale feature fusion concept for HPE. Despite their great job, these technique have faced a very common bottleneck. Solely developing the model capability does not improve the performance too much.

Multi-stage Network: This type of approaches try to provide a more accurate prediction. Newell et al. [111], proposed an architecture having repeated top-down and bottom-up processing for collection of multiple features at different scale named "stacked hourglass"

for HPE from images. Su et al. [174] propose an feature aggregation method over the stackhourglass to make an more efficient and robust prediction. Similarly, [10] proposes an structure based regression technique for HPE. While these techniques give good results on MPII and LSP datasets in terms of PCK scores. But still require to improve the accuracy in terms of PCK score. So, in this chapter, we introduce an multi-stage network that shows better performance compared to them.

All the deep learning architecture have multiple layers. DCNN is mostly the main building block of deep learning. It contains alternative convolution and pooling layers. The basic components of the DCNN are shown below:

Convolution layer This component is a type of linear function. The layer utilization is for property extraction, noise reduction, and characteristic improvement. The interconnections of input neurons can take the local information to the receptive field of previous layer neurons. The image I is a two-dimensional matrix with size of n $\times$ n . After applying convolution with k filters of f$\times$f size, the output is of size $((n-f) \div sl + 1) \times ((n-f) \div sl + 1)$:

$$y_i = b_i + \sum K_{ij} \times x_i \tag{3.1}$$

where $\times$ is the convolutional operation, x_i represents the input of the convolutional layer, k_{ij} denotes the parameter for the convolutional kernel, b_i denotes bias, and sl is the step length for the convolution; each filter is associated with the particular feature.

Feature mapping layer A non linear activation operator is utilized to map results from the filter layer, thereby producing the feature graph F.

$$f_s = \sigma(b_i + \sum k_{ij} \times x_i) \tag{3.2}$$

Here σ is a non linear activation function. We have many activation functions like sigmoid, softplus, and tanh. Mostly ReLU (Restricted Linear Units) is utilized for the activation of neural networks.

Pooling Layer To remove redundant information pooling layer is utilized. The feature graph is divided into m $\times$ m non overlapping sections, the pooled feature is calculated by taking the statistical mean value of each section and having the size of $(\frac{((n-f)\div sl+1)}{m}) \times (\frac{((n-f)\div sl+1)}{m})$. The layer makes the model more accurate and robust by largely reducing it, and by avoiding overfitting. The DCNN architecture considers a single layer as a whole with the combination of all convolution, mapping, pooling layers to extract image features. After applying these operations over an input image, the output is the set of learned features. The next section is the classification part. Features to be learned may be placed in the logistic regression for classification. These classifier utilizes the softmax activation function for output-layer. All the DCNN parameters are usually trained using back propagation [175] along with Stochastic Gradient Descent algorithm. The dropout strategy [176] is used to avoid overfitting and improve the unreliability of the networks. These strategy is usually utilized for fully connected layers.

3.2.2 2D human pose estimation using detection followed by estimation approach

In computer vision, this topic is highly motivated and improve the performance by utilizing current techniques [7]. The classical approach for human detection uses handcrafted features like HOG, Gradient, wavelet, etc. Few handcrafted features are ACF(Active contour feature) [8]. After the evolution of CNN, the task of detection, localization and classification becomes easy. The few popular techniques which utilizes the deep learning procedure are [5][9]. Currently, [9] introduced an CNN based deep network along with early and late fusion using multi-spectral image data to improve the performance.

The HPE has been intensively studied in recent decades. In early days, the graphical models based on tree structures like pictorial structure models [22] was highly utilized for HPE. CNN based techniques are highly utilized for the estimation in last few years. First, Toshev et al.[10] proposed an CNN based architecture to regress the body joint coordinates for belief map. Similarly, [11] utilize the CNN for HPE.

3.3 Methods

This section explains the procedures of both of the introduced models in detail. Section 3.3.1 defines the first suggested model named "Multi-stage Deep learning network for 2D HPE". Section 3.3.2 discusses the second proposed model: "Detection followed by estimation Deep learning model for 2D HPE".

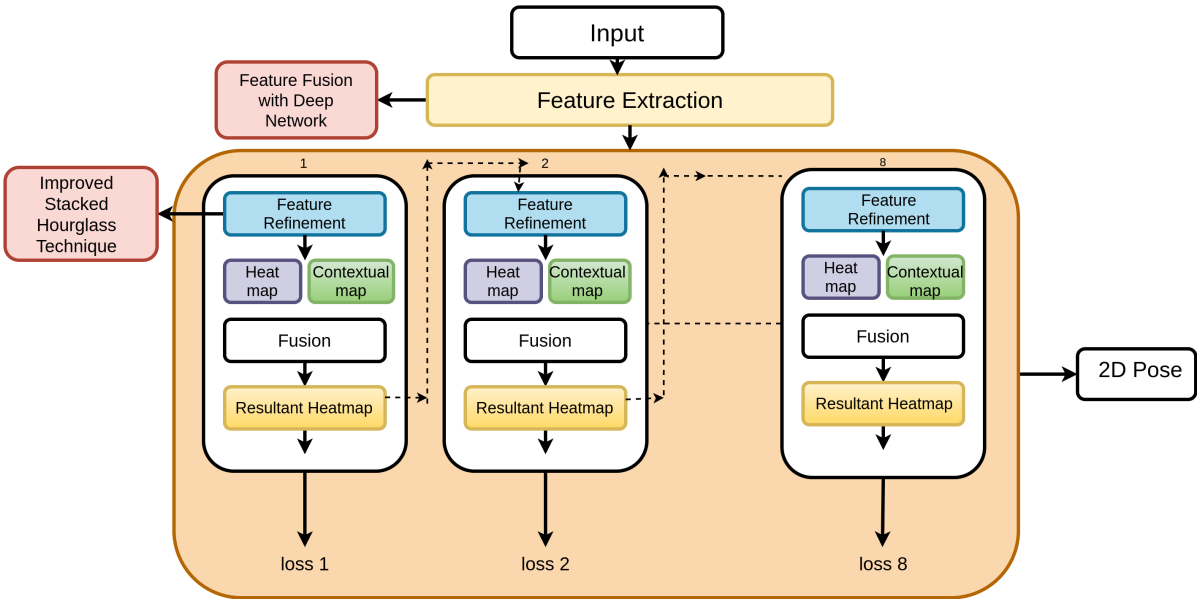

FIGURE 3.1: The proposed architecture of first model

3.3.1 Multi-stage Deep learning network for 2D HPE

This section discusses the first proposed work for 2D HPE. An overview of the proposed framework have been illustrated in the Figure 5.1. The proposed architecture have eight stages, which contains the DCNN based pose prediction modules that takes RGB image as input and gives output as joint coordinate and its probability. We give a summarize discussion of the network modules and then briefly review the structure of hourglass network and at last presents the detailed discussion of each part of the proposed architecture.

The proposed DCNN architecture have main three parts: Feature Extraction, Feature Refinement and Fusion. The method first perform the feature extraction over the image that extracts the basic features.For this part we propose a two-stream DCNN based feature fusion concept that is highly motivated by the Inception-v4 and VGG-19 deep modules. These extracted features are fed to the first stage of the network. For all stages mainly

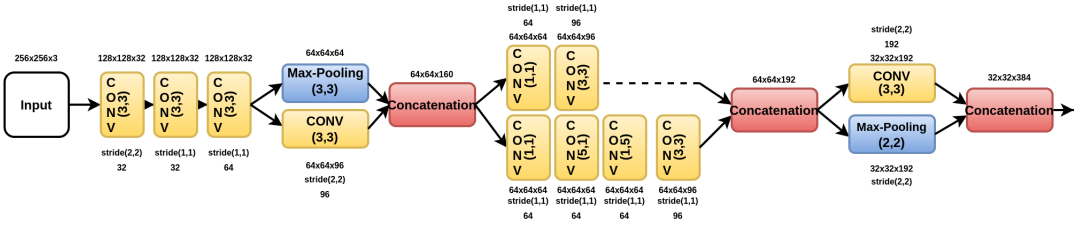

FIGURE 3.2: First stream of feature extraction.

two steps have been performed, i.e. feature refinement and heatmap fusion. The last stage gives the final pose prediction output. We employ the highly modularized Stacked Hourglass Network [111] as the basic network structure and proposed an cascaded feature integration concept of the hourglass to investigate the feature refinement for HPE. The proposed cascaded feature integration stacked hourglass technique work as a 'part' detector, which regresses the keypoint heat maps and context heat maps. At last the part-based and context map have been fused to make prediction at each stage, which have also been used as intermediate supervision by passing it to the next stage with 1×1 convolution.

The local information is important for recognizing features such as hands and faces, a final HPE requires a systematic understanding of the entire body. There are many cues which are perfectly identified at distinct scales of the image. These cues are the orientation of the human body, limb arrangement, and the adjacent joints relations. In literature, the hourglass procedure is very efficient and have a minimal design, which is sufficiently capable to acquire all these features and carry them collectively to make the pixel-wise prediction as output.

The description of the hourglass method is given as: the convolutional and pooling layers have been utilized to extract feature at the low resolution. For every max pooling branch, the system branches and add more convolutions at the original resolution. After the lowest

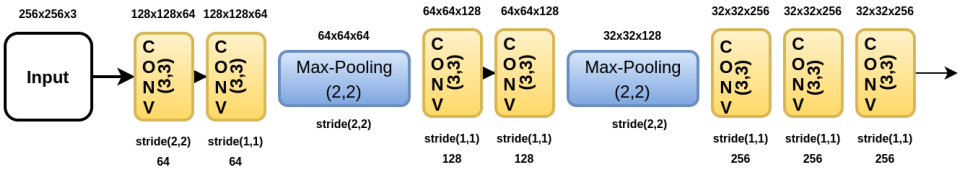

FIGURE 3.3: Second stream of feature extraction.

resolution is achieved, the network starts with the top-down sequence of upsampling and combination of features across scales. To combine information on two adjacent resolution, they utilize the method given by Tompson et al. [169] and perform the upsampling of the closest neighbor to the lowest resolution, followed by an elementary addition of the two groups of characteristics. The framework of the hourglass method is symmetric, so for each layer that is present in the downward path, a corresponding layer rises.

The method takes input RGB images and gives two output: the joint coordinate $Y_x = (m_x, n_x)$, and the probability P_x of that joint. Where x is the x = 1, 2, 3,....., NO_J. NO_J is the number of joints. The refined feature map is transformed into the body part (H_p) and context map (H_c). At each stage, the soft-argmax [177] [178] have been utilized over generated part and context maps, that are fused to generate the joint coordinate. The resulted fused heat maps are represented as H = $H_p + H_c$, where H_c is ((NO_c) (NO J)).

In the coming section, we give the detail description of the each modules of the architecture and the loss function that have been utilized for the training. The whole system perform the end-to-end training.

Algorithm 1: Summary of the proposed method:

Result: ϕ : 2D pose

1. Initialize
2. No of context-per-joint =2
3. α = 8
4. No of heatmap = (No of context-per-joint +1) $\times$ No of joints
5. 2: Input $\leftarrow$ image: x,
6. 3: x $\leftarrow$ **FeatureExtraction** (x)
7. 4: **for** stage =1 to 8
8. Input-shape $\leftarrow$ x
9. i. x $\leftarrow$ **Refinement**(x)
10. ii. ident-map = x
11. iii. x = conv(x)
12. iv. h = conv(x)
13. v. (h_p, h_c) = split-heatmap-for-part-and-context-information(h)
14. vi. Apply differentiable soft-argmax as per equation (5) and build the joint probability for each heatmap Estimated joint location : y = $(\psi_x(h), \psi_y(h))^T$)
15. Joint-prabability = activation-sigmoid (joint location)
16. vvi. Joint coordinate estimation and fusion of part and context information heatmap as per equation (2)
17. vvvi: if stage ¡ 7
18. h = conv(h)
19. x = add (ident-map, x, h)
20. **end**
21. **end**

3.3.1.1 Feature Extraction

This section structures the proposed framework for feature extraction using the fusion of two DCNN based features, which we called here as a two-stream input DCNN network. The proposed two-stream deep fusion architecture have been motivated from the Inception-v4 [167] and VGG-19 [168]. The fused feature is fed to feature refinement module as described in the Figure 5.1. We are going to describe the structure and source of feature for each specific stream.

The approache [179] motivates us to utilize the stem part of the inception and a very small

portion of very deep VGG architecture for feature extraction. The inception and VGG network is extremely tunable, where we can easily make modifications at the level of filter count on various layers, which does not disturb the fully trained system performance and extract relevant features that make the prediction accurate with the low computational cost. We observe that this CNN module for feature extraction have abundant quality knowledge to make an accurate 2D prediction. The detailed description of the part is shown in Figure 6.6 and 3.3.

We have utilized the serial fusion strategy to combine the two stream sets of features as shown in equation 1, where $F_1(I)$ is the first stream feature and $F_2(I)$ is the second stream feature. The resulted dimension of the combined features is identical as the summation of the dimension of the two sets of features.

$$F(I) = F_1(I) + F_2(I) \tag{3.3}$$

3.3.1.2 Feature Refinement

Pose estimation is more challenging than image classification. After the introduction of a stacked hourglass, the multi-task architecture is increasingly utilized for the technique. But we found that the stacked hourglass is insufficient in their design choices because the repeated up and down sampling, having the information loss. To overcome this, we propose a cascaded feature integration technique over each stage of the hourglass network. It reinforces the data flow and alleviates the obstruction of training by circulating the multiple scale features from one stage to another.

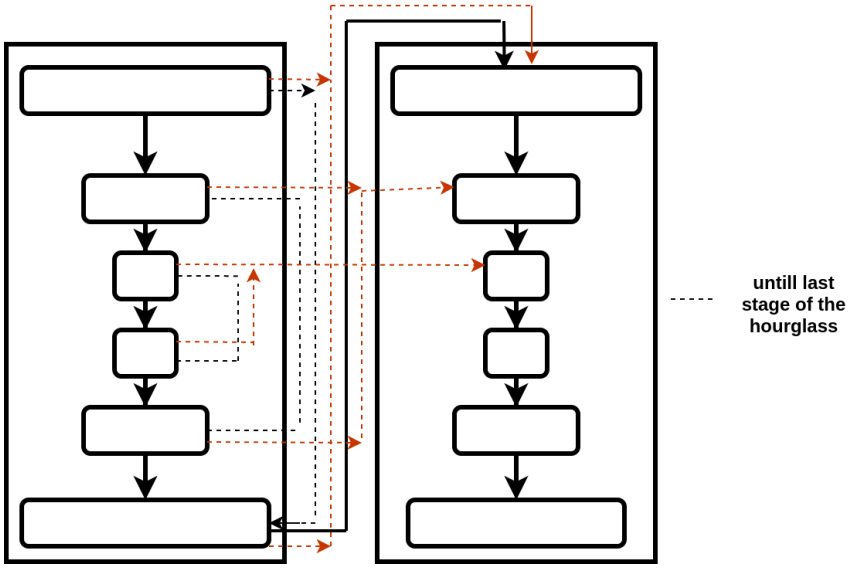

FIGURE 3.4: Cascaded feature integration architecture.

The red color arrow in Figure 3.4, represents the information flow of the proposed cascaded feature integration method. For every scale, two separate data flows are proposed from downsampling and upsampling units of the preceding stage to the downsampling operation of the present stage. The 1×1 convolution operation have been added on every flow.

3.3.1.3 Fusion

After getting motivated by multi-context attention [9], we find that the learning of joint coordinate's correlation takes place in hidden layers, while regressing joint coordinate by many techniques, identify the body part independently, make it hard to learn the context. For example, the techniques give a positive result on the head, also positively reacting to sunglass and cap. So there is a need to add some more information to the system. We add context information along with part-based to make the system more accurate. We call it

part heat map (H_p) and context heat map (H_c).

$$H_p = [h_1^p, ..., h_{NO_j}^p] \tag{3.4}$$

$$H_c = [h_{1,1}^c, ..., h_{NO_c,NO_j}^c] \tag{3.5}$$

The joint probability for context map is $P_{r,s}^c$, r = 1, ..., NO_c, s = 1,....., No_j.

The equation represents the s^{th} joint coordinate position by fusing the part-based and context information as:

$$K_s = \alpha K_s^P + (1-\alpha)\frac{\sum_{r=1}^{NO_c} P_{r,s}^c K_{r,s}^c}{\sum_{r=1}^{NO_c} P_{r,s}^c} \tag{3.6}$$

where K_s^P is the soft-argmax operation over h_s^P, represents the predicted location of the s^{th} part-based heatmap. $K_{r,s}^c$ is the soft-argmax over $h_{r,s}^c$ represents the predicted location of the r^{th} context map for joint s. $P_{r,s}^c$ is the probability of the the r^{th} context map for the joint s. Here α is the hyper parameter.

The equation represents the pose prediction, the fusing of part-based prediction and the context prediction maps.

The softmax over a heatmap h ϵ $R^{W\times H}$ is represented as:

$$\Phi(h_{m,n}) = \frac{e^{h_{m,n}}}{\sum_{i=1}^{W}\sum_{j=1}^{H} e^{h_{i,j}}} \tag{3.7}$$

W $\times$ H : heatmap size $h_{m,n}$: heatmap value at location (m,n) Rather than using cross-channel softmax, we utilize spatial softmax to make the heatmap normalized. The definition of soft-argmax is given as:

$$\Psi(h) = \sum_{m=1}^{W} \sum_{n=1}^{H} W_{m,n,d} \Phi(h_{m,n}) \tag{3.8}$$

d: components x or y

W: weight matrix for coordinate (x,y)

The definition of the Weight matrix components are as follows:

$$W_{m,n,x} = m/W, W_{m,n,y} = n/H \tag{3.9}$$

At last, the regressed joint location is expressed as:

$$z = (\Psi_x(h), \Psi_y(h))^T \tag{3.10}$$

The soft-argmax [177] [178] is defined as the weighted mean of the points allocated on a uniform grid, including the weights remain similar to the respective heat map.

The differentaible version of soft-argmax is given as:

$$\frac{\delta \Psi(h_{m,n})}{\delta h_{m,n}} = W_{m,n,d} \frac{e^{h_{m,n}} \sum_{i=1}^{W} \sum_{j=1}^{H} e^{h_{i,j}} - e^{h_{m,n}}}{(\sum_{i=1}^{W} \sum_{j=1}^{H} e^{h_{i,j}})^2} \tag{3.11}$$

The advantage of using soft-argmax is to make system directly regress the joint coordinate without using any post-processing step.

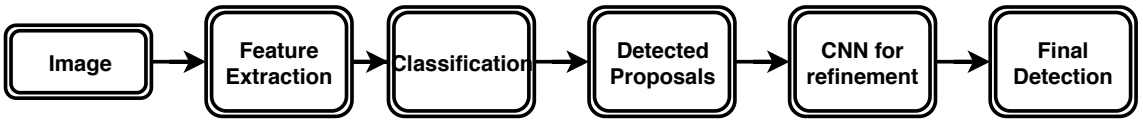

FIGURE 3.5: The architecture of Detection module.

3.3.2 Detection followed by estimation Deep learning model for 2D HPE

We go along with detection followed by estimation like detecting the region of interest having human as subject and then estimating the location of the respective body joints. The detail architecture for both individual steps are given below, respectively.

3.3.2.1 Detection module

For human detection, we follow the architecture design mentioned in Figure 5.1. First, we apply preprocessing in which we do data augmentation as discussed below in section 4. Second, we apply basic machine learning strategy of feature extraction and classification, which identifies the region that might have human. Then finally the detected outputs have been utlized by the CNN based deep learning to make more accurate classification.

3.3.2.2 Feature Extraction and Classification

We extract HOG(Histogram of oriented gradient) and LBP(Local Binary Pattern) features from the images. The main purpose to use HOG is that it is view invariant and global feature. The reason for using LBP is that it is texture based feature, uniform and rotation

invariant. The mathematical explanation of the steps involved in HOG computation is as follows: Both the horizontal and vertical gradient computation are:

$$g_h(m,n) = I(m+1,n) - I(m-1,n) \tag{3.12}$$

$$g_v(m,n) = I(m,n+1) - I(m,n-1) \tag{3.13}$$

Gradient pixel(m,n) have amplitute and direction as:

$$g(m,n) = \sqrt{g_m(m,n)^2 + g_n(m,n)^2} \tag{3.14}$$

$$\Theta(m,n) = \tan^{-1}(g_m(m,n)/g_n(m,n)) \tag{3.15}$$

The steps require to compute LBP with NP number of pixels over ra radius are given below:

np: np^{th} pixel neighbor

g_{cp}: central pixel gray value

g_{np}:np^{th} neighbor gray value

$$LBP_{NP,ra}(m,n) = \sum_{np=0}^{NP} X(g_{np} - g_{cp})2^{np} \tag{3.16}$$

Here X(q) was utilized to threshold:

$$X(q) = \begin{cases} 1 & q \geq 0 \\ 0 & q < 0 \end{cases} \tag{3.17}$$

The circular pixel does ot contain the np^{th} neighbor. Thats why the computation of gray value is as follows:

$$g_{np} = I(m_{np}, n_{np}) \tag{3.18}$$

np = 0,1,...NP-1 and

$$m_{np} = m + ra\sin(2\pi np/NP) \tag{3.19}$$

$$n_{np} = n - ra\cos(2\pi np/NP) \tag{3.20}$$

The modified rotational invariant LBP is as follows:

$$LBP^{rin}_{NP,ra}(m,n) = \begin{cases} \sum_{np=0}^{NP-1} X(g_{np} - g_{cp}) & \text{if } V(m,n) \leq 2 \\ z+1 & \text{if } otherwise \end{cases} \tag{3.21}$$

$$V(m,n) = \sum_{np=1}^{NP} |X(g_{np} - g_{cp}) - X(g_{np-1} - g_{cp})| \tag{3.22}$$

The resulted feature descriptor have been utilized with lib-SVM for detection. After that the detection proposals are again utilized by the CNN based classification model as described below.

3.3.2.3 Pre trained CNN based model for making detection accurate

VGG deep architecture [168] have been utilized for the pre-trained network. The module have total thirteen convolution, three FC and one logistic regression layer. They have 5 max-pooling of 2×2 after the 2^{nd}, 5^{th}, 7^{th}, 11^{th} and 13^{th} layers. ReLU was used for non-linearity of 3×3.

The organisation of filters is as follows:

1 and 2 layers have 64 filters

3 and 4 layers have 128 filters

5,6, and 7 layers have 256 filters

8 to 13 layers have 512 filters

14 to 15 layers have 4096 filters

16 layers have 1000 filters

Then at last there is a soft-max function. In the VGG deep network, we apply some changes. First, the network is retrain on the INIRIA dataset. Second, the two class classification is required (c=2) for identifying human or nonhuman rather than thousand classes of Imagenet. The sixteenth layer and softmax was modified respectively. The network was fine-tuned with INIRIA dataset. The fine-tune hyperparameter is as follows, 10 number of epochs, 0.001 learning rate, 100 batch size and 0.9 momentum. For testing, the SVM detector was applied over 288 test images and the resulted output was run by the pretrained CNN module for final classification.

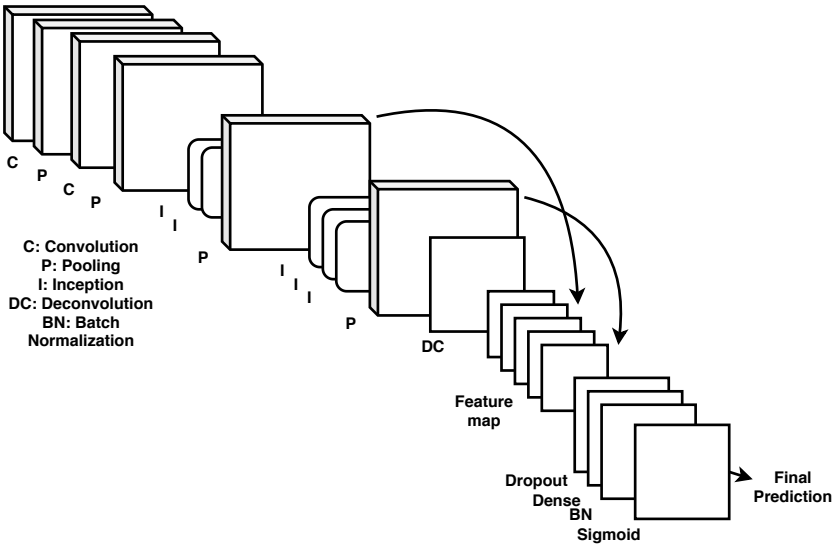

FIGURE 3.6: The multi-layer architecture with Fully Convolutional GoogleNet.

3.3.2.4 Deep architecture for HPE

In this case, we have to predict the body joint location coordinate. The proposed network is based on the Inception deep network. We freeze the last stages of the network like pooling, drop-out, linear and soft-max layers and starting all layers have been remain the same. To collect multi-layer information, we combine the feature map of multiple inception output layers shown in Figure 6.6 with the help of deconvolution filter with stride 2 and of size 2×2.

The resulted feature map have low resolution because of pooling as compared to input image. So, the produced feature map immediately upsampled with a deconvolution filter of stride 16 and of size 32×32 to get all joints belief maps. At last dropout, dense, batch normalization and sigmoid function utilized for normalizing and regulazing the network.

3.4 Result and Discussion

This section of the chapter discusses the results produced by the methods and their evaluation study.

3.4.1 Multi-stage Deep learning network for 2D HPE

The first proposed method is validated on the following datasets:

3.4.1.1 MPII Dataset

MPII [127] is the benchmark dataset for the single person HPE. The dataset has collection of images from the You tube videos which contains the daily human activities. It contains total 25k images along with 40k annotations. The 30k images have been utilized for training and 10k for testing. The annotated data have total 16 joints. In respect to other datasets, MPII give more knowledge like fully annotated frames and activity label with good image resolution. The keypoint location have been utilized for training.

3.4.1.2 LSP Dataset

LSP dataset [180] provides 1K training and 1K testing images. The extended LSP provide extra 10k images for training. So, to train the network we utilize 11k images from both one. The images in this dataset was mainly collected from the Flicker and have sports persons doing different sports like tennis, baseball, and so on.

3.4.1.3 Training Details

In the introduced architecture, the joint regression and probability have been trained at the same time. We utilize both L1 and L2 distances for joint coordinate loss, which is the distance between the estimated and ground truth joints and act as a loss function. The (l1 + l2) loss function is utilized to train the system for joint regression is given below:

$$l1 + l2 = \frac{1}{NO} \sum_{s=1}^{NO_J} \left\| M_s(m,n) - M_s(\hat{m},n) \right\|_1 + \left\| M_s(m,n) - M_s(\hat{m},n) \right\|_2^2 \tag{3.23}$$

M_s : ground truth of the s^{th} joint coordinate

$\hat{M_S}$: predicted joint coordinate

Binary cross entropy loss is used to train the system for joint probability P is given as:

$$L_p = \frac{1}{NO_J} \sum_{s=1}^{NO_J} (P_s - 1) log(1 - \hat{P}_s) - P_s log(\hat{P}_s) \tag{3.24}$$

P_s and $\hat{P}_s$: ground truth and predicted joint probability.

RMSProp optimizer and back propagation have been utilized along with 12 batch size.

3.4.1.4 Evaluation Metric Used

Percentage of correct keypoints (PCK) metric is utilized for the result evaluation purpose. The correct estimation is evaluated by computing the distance between the predicted and ground-truth keupoint. A predicted keypoint is taken as correct if the distance between

TABLE 3.1: **The result of the first introduced technique and its comparison with other state of the art on MPII dataset using modified PCK called PCKh with @0.5.**

Technique	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Pishchulin et al. [181]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Newell et al. [111]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Lifshitz et al. [182]	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Carreina et al. [183]	95.7	91.6	81.7	72.4	82.7	37.1	66.4	81.3
Insafutdinov et al. [184]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Bulat et al. [149]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
wei et al. [166]	97.8	95.0	88.7	84.0	88.4	83.4	78.0	88.5
Rafi et al. [185]	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3
Chen et al. [186]	98.5	96.5	92.5	88.5	90.2	89.6	86.0	91.9
Chu et al. [9]	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Belagiannis et al. [158]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
Chou et al. [112]	98.2	96.8	92.2	88.0	91.3	89.1	84.9	91.8
Sun et al. [141]	98.6	96.9	92.8	89.0	91.5	89.0	85.7	92.3
Sun et al. [10]	97.5	94.3	87.0	81.2	86.5	78.5	75.4	86.4
Ours	99.1	98.6	94.0	89.2	92.6	90.0	85.7	92.7

TABLE 3.2: **The result of first introduced technique and comparison with other state of the art on LSP dataset using PCK with @0.2.**

Technique	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Pishchulin et al. [181]	87.2	56.7	46.7	38.0	61.0	57.5	52.7	57.1
Tompson et al. [169]	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.3
Carreina et al. [183]	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Yang et al. [148]	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6
Rafi et al. [185]	95.8	86.2	79.3	75.0	86.6	83.8	79.8	83.8
Yu et al. [176]	87.2	88.2	82.4	76.3	91.4	85.8	78.7	84.3
Belagiannis et al. [158]	95.2	89.0	81.5	77.0	83.7	87.0	82.8	85.2
Lifshitz et al. [182]	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
Pishchulin et al. [181]	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Insafutdinov et al. [184]	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1
wei et al. [166]	97.8	92.5	87.5	83.9	91.5	90.8	89.9	90.5
Bulat et al. [149]	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Chu et al. [9]	98.1	93.7	89.3	86.9	93.4	94.0	92.5	92.6
Ours	99.4	94.8	90.1	85.3	94.7	93.0	91.7	92.7

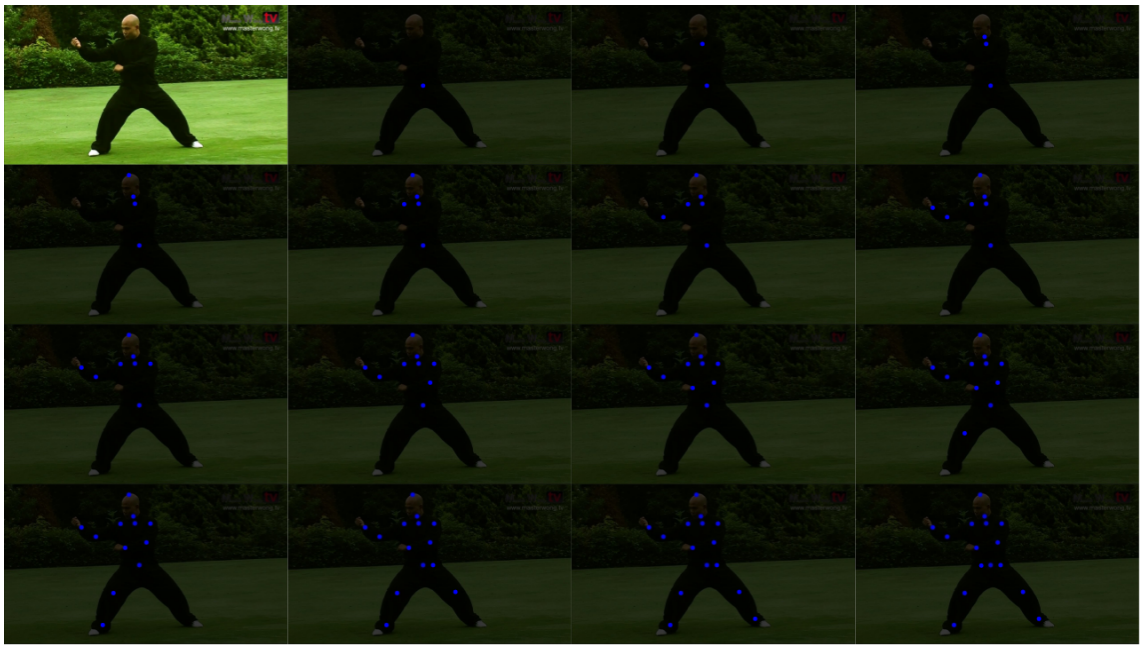

FIGURE 3.7: The visualization of heatmap for all 16 joint locations on the MPII dataset. The images from left to right and top to bottom have been added the heatmaps for pelvis, thorax, upper neck, head top, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right knee, right ankle, left hip, left knee, left ankle, right hip.

predicted and ground truth keypoint is lesser than α factor of head segment length (PCKh). The representation of the metric is as PCKhα.

3.4.1.5 Preprocessing

The input image was cropped and resized to 256×256 pixels taking the main subject at the center. The data augmentation comprises, random data translation (with +-2% of the given image size), random rotation (+- 40 degree) and scaling from 0.7 to 1.3 on MPII and from 0.85 to 1.25 on LSP. In all the experiments, the base learning rate is 1e-3. It drops by factor of 0.4 when the loss on the validation set saturates. The used validation split is same as [187]. The training time 3 days using Nvidia GPU with 8 GB of memory.

FIGURE 3.8: The visualization of the Qualitative evaluations on both MPII dataset.

The proposed architecture have 8 stages, which predicts the 16 joints. The 2 context map is used for each joint. The value of parameter α is 0.8.

3.4.1.6 Experiments

The proposed model is evaluated and compared with all the recent methods of 2D HPE from a single image. The MPII Human Pose and LSP dataset is used for evaluation. The

MPII and LSP dataset have been used to evaluate for single HPE. The keypoint annotation is given by both MPII and LSP dataset with 16 and 14 keypoints. The Figure 3.7 and 3.8 shows the predicted joint locations in the form of heatmaps and qualitative results obtained using proposed method.

Experiment 1: The score of the MPII have been available by the dataset providers, since the MPII test label is not publically available. The test evaluation is taken on the 3k validation data set which is taken out from the available training set, the following Table 3.1 shown below, is the result of the proposed method on the MPII dataset and the comparison with other state of the art techniques.

Experiment 2: Table 3.2 gives the PCK score value of proposed method with threshold of 0.2 and comparison with other state of the art techniques with same threshold. The training procedure is as same as many other state of the art [158] [184], which add MPII and LSP training set. We trained on person-centric (PC) annotations and have been evaluated on proposed method using the Percentage Correct Key-points (PCK) metric.

The visualization of heatmap using proposed architecture for all 16 joints on MPII dataset are shown in Figure 3.7. it can be seen that the predictions given by our method is very accurate for all joints. The visualization of some qualitative poses produced by the proposed framework is shown in Figure 3.8.

TABLE 3.3: **Some Evaluated Metrics on INIRIA Person Dataset**

	Metric	HOG+LBP	HOG+LBP+CNN
INIRIA Dataset	TP	530	545
	FN	37	42
	FP	1282	248
	FPS	14.12	3.20
	MR%	15.80	14.13

3.4.2 Detection followed by estimation Deep learning model for 2D HPE

3.4.2.1 Experiments

INIRIA Person dataset: INIRIA person dataset have 1832 training images from which 1218 are negative images and 614 are the positive images. The dataset have 288 test images. Here we build both training and validation set. To make the positive class set, the positive train ground truth bounding box has been utilized resulting in 1237. After that horizontal flipping was applied resulting in a set of 2474.Then, translation and scaling was utilized with a range of [0,5], results in a final set of size 4948. To make negative train set, we randomly extract the windows from the images using [188]. The final set have 12552 negative samples. The whole procedure give 175000 samples from which 15754 is utilized for training and 1749 for validation.The fine-tune hyperparameter is as follows for detection module, 10 number of epochs, 0.001 learning rate, 100 batch size and 0.9 momentum.

TABLE 3.4: PCKh metric at 0.5 for MPII Dataset.

Method	Shoulder	Wrist	Head	Elbow	Ankle	Knee	Hip	PCKh0.5
Ours	93.7	81.0	97.1	86.2	73.8	80.8	86.4	85.5
Tompson et al. [15]	91.9	77.8	96.1	83.9	64.8	72.3	80.9	82.0
Pishchulin et al. [17]	90.2	77.3	94.1	83.4	68.6	75.4	82.6	82.4
Pishchulin et al.[12]	49.0	34.1	74.3	40.8	35.2	34.4	36.5	44.1
Lifshitz et al. [18]	93.3	80.4	97.8	85.7	70.2	76.6	85.3	85.0
Tompson et al. [13]	90.3	72.4	95.8	80.5	62.8	69.7	77.6	79.6
Hu et al. [16]	91.6	76.6	95.0	83.0	69.5	74.5	81.9	82.4
Carreira et al. [14]	91.7	72.4	95.7	81.7	66.4	73.2	82.8	81.3

TABLE 3.5: PCKh metric at 0.2 for LSP Dataset.

Method	Shoulder	Wrist	Head	Elbow	Ankle	Knee	Hip	PCKh0.5
Ours	82.8	71.8	95.6	74.8	74.4	78.4	84.6	80.34
Wang et al. [19]	57.1	36.7	84.7	43.7	50.8	52.4	56.7	54.6
Fan et al. [20]	75.2	64.0	92.4	65.3	70.4	68.3	76.7	73.0
Chen et al. [21]	78.2	65.5	91.8	71.8	63.4	70.2	73.3	73.4
Tompson et al. [13]	79.2	63.4	90.6	67.9	64.2	71.0	69.5	72.3
Pishchulin et al. [12]	56.7	38.0	87.2	46.7	52.7	57.5	61.0	57.1

MPII and LSP Dataset: The dataset have total 40,000 images, from which 25,925 utilized for training and 2,958 for validation. The dataset gives an approximate location of the human for both train and test images. The LSP HPE dataset have 10000 train and 1000 test images.

For HPE, we do the training of the network using the merged dataset of MPII and LSP for 73 epochs, 8mbatch size and learning rate of 0.00092. The quantitative outputs of the proposed module for MPII datset are given in Table 3.4 and for LSP given in Table 3.5. The proposed method outperforms the other state-of-the-art methods.

3.5 Conclusion

This chapter introduced the two-deep learning-based 2D HPE frameworks that give good results on challenging joints and occluded subjects. The first model was composed of three consecutive modules: DCNN based feature extraction, cascaded feature integration over stacked hourglass for feature refinement, and integration of context and part heatmaps information to make the system accurate towards occlusion. Evaluation was performed on two standard datasets MPII and LSP and found that the model gave improved pose prediction results on PCK metrics than other state-of-the-art techniques. We observed that the limitation of the first model is the utilization of high memory space.

The second model had an advantage in using the detection followed by estimation procedure: their proficiency in arranging a task into multiple subtasks for making it easier and utilizing less memory space. If the human detector is excellent in detecting the hard candidates, the HPE will usually get accurate with a focused regression space. The pose estimator architecture is not as complex as other state-of-the-art methods based on deep learning. So the overall system act as simple and efficient for HPE. We have examined the architecture modules on the popular publicly available dataset like the INRIA person dataset, LSP, and MPII pose estimation dataset. The method gives an impressive performance on these datasets as compared to other states of the art methods. The first one is more accurate among both models, whereas the second model takes less memory space than the first. Both the models give good results as compared to other state-of-the-art techniques.