Structure Aware Human Pose Estimation Using Adversarial Learning

by

Suryam Sharma

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

The University of Texas at Arlington

May 2021

Acknowledgments

I would like to thank my supervising professor Dr Vassilis Athitsos for constantly motivating and encouraging me, and also for his invaluable advice during the course of my master's studies. I wish to thank Dr William J Beksi and Dr Alex Dillhoff for their interest in my research and for taking time to serve in my dissertation committee.

I am grateful to Prof David Levine and Dr Ramez Elmasri for providing me the opportunity to work as a Graduate Teaching Assistant for three consecutive semesters at the Department of Computer Science and Engineering.

I would also like to extend my appreciation to the Computer Science and Engineering department for providing me with all the facilities and infrastructure necessary to carry out my master's studies.

Special thanks to Ph.D. student Saif Sayed for constantly supporting me and sharing his knowledge with me which helped me pass through many hurdles during my research.

Furthermore, I would like to thank my beloved parents and my sisters, who have taught me the value of hard work and persistence. I would like to thank all my friends for their continuous support that has been shown to me during my masters studies. Last but not least, I would like to thank God and all whose direct and indirect support has helped me in completing my education.

Structure Aware Human Pose Estimation Using Adversarial Learning

Suryam Sharma, M.S.
The University of Texas at Arlington, 2021

Supervisor: Dr Vassilis Athitsos

Pose estimation using Deep Neural Networks (DNNs) has shown outstanding performance in the recent years, due to the availability of powerful GPUs and larger training datasets. However, there are still many challenges due to the large variability of human body appearances, lighting conditions, complex background, occlusions and postures. Among all these peculiarities, partial occlusions, and overlapping body poses often result in deviated pose predictions. These circumstances can result in wrong and sometimes unrealistic results. The human mind can predict such poses because of the underlying structural awareness of the geometry, of a human body. In this thesis, we discuss an efficient training technique that helps us to correct structurally implausible poses caused due to partial occlusions. We introduce a pose discriminator which helps us to incorporate priors about the human-body's structure, into our model. As shown in the experiments, using this pose discriminator results in improved accuracy.

Contents

List of Tables

List of Figures

Table 1: Table of Symbols and Acronyms

G	*Generator Network*
D	*Discriminator Network*
$\mathcal{L}_G$	*Generator Loss*
$\mathcal{L}_D$	*Discriminator Loss*
M	*Number of heatmaps (or poses)*
N	*Number of stacks*
I	*Input image*
y	*Ground truth heatmaps / label*
$\hat{y}$	*Predicted heatmaps*
$\mathbb{E}$	*Expected value*
d_i	*Normalized distance between the predicted and ground-truth location of the* i_{th} *body part*
d_{fake}	16×1 *unit vector, containing 0 and 1*
δ	*Normalized distance threshold*
α	*Discriminator loss coefficient*
MSE	*Mean Squared Error*
PCK	*Percentage Correct Keypoints*
$DCNN$	*Deep Convolutional Neural Network*
GAN	*Generative Adversarial Network*
$DCGAN$	*Deep Convolutional Generative Adversarial Network*
$MPII$	*Max-Planck-Institut für Informatik*

Chapter 1

Introduction

Human pose estimation has an important impact on a wide range of applications from activity recognition, gaming, surveillance, animation to human computer interactions. For human pose estimation, joint obstructions and overlapping body poses result in deviated pose estimation.

Human vision can learn the variety and structural limits of a human body from observations. Even under extreme occlusions, human mind can deduce the possible poses. It is, however, very difficult to incorporate priors about the structural geometry of a human body into DCNNs, because DCNNs are most capable of learning features. In, this thesis we discuss a novel learning approach which uses a Discriminator to incorporate priors about the structure of a human-body into our training model, or the Generator.

Having said that, in the recent years, there have been many approaches to incorporate structure awareness while doing pose estimation. Like HRNet by sun et al. [12] does it by maintaining high resolution of the input data, PGCN by bin et al. [7] do it by modelling the structural relationships using the Graph Convolutional

Neural Networks and MSS-Net by lipeng et al. [6] fuses multiple scales of keypoint heatmaps to determine the pose output. However, [12],[7] and [6] do achieve structure awareness but at the cost of making the model bulkier, or computationally heavy. Our research focuses on an improvised learning method that helps us incorporating structure awareness into an existing pose network, without changing the network architecture.

For our research, we have taken a Stacked Hourglass network [4], as our Generator. Hourglass Network has a multi-stage conv-deconv network architecture. It focuses on contextual feature learning i.e., in matching body keypoints by combining feature heatmaps across scales. The repeated bottom-up and top-down processing within the hourglass modules can reliably extract posture features across scales and viewing variabilities, they are very good in locating local features but they do not effectively use the global relations between these features.

This thesis focuses on an adversarial learning method like Boundary aware face-alignment algorithm, by wu et al. [2] and adversarial-posenet, by chen et al. [1], which aims on improving the *existing* deep learning algorithms involved in the 2D human pose estimation problem. It does not deal with changing the architecture or parameters of the pose model in any way. Instead, it introduces a learning approach that exploits these extracted features from the pose model, by establishing a structural dependency between those features.

Using a discriminator to predict the likelihood of the pose being real or fake, we can instill the structural dependency of the human key joints into our model (or generator). We do this by establishing global relations between the locally extracted posture features by our pose model. To achieve such goals, the discriminator should be fed with sufficient information to perform classification, while the generator should have the ability to extract complicated features in pose estimation.

1.1 Thesis Outline

The thesis is organized as follows:

Chapter 1 :: Introduction - Provides a general introduction to the problem statement and proposed method.

Chapter 2 :: Related Work - Discusses a brief history of 2D Human Pose Estimation and the challenges faced by them while dealing with occlusions in the poses.

Chapter 3 :: Methodology - Talks about the proposed method.

Chapter 4 :: Datasets - Mentions the benchmark datasets used for the 2D Human Pose Estimation

Chapter 5 :: Training and Experiment Settings - Explains the training and experimental setup of our system.

Chapter 6 :: Experimental Results - Discusses the experimental results of the proposed system.

Chapter 7 :: Ablation Study - Provides the Ablation study for the proposed methodology.

Chapter 7 :: Conclusion - Concludes the work and discusses the scope for future work

Chapter 2

Related Work

2.1 Human Pose Estimation

Human pose estimation is an active research topic for decades. Human pose estimation refers to the process of inferring poses in an image. Essentially, it is predicting the positions of a human body's joints (also known as keypoints - elbows, wrists, etc) in an image or a video. This problem is also sometimes referred to as the localization of human joints. It's also important to note that pose estimation has various sub-tasks such as single pose estimation, estimating poses in an image with many people, estimating poses in crowded places, and estimating poses in videos. It can be performed in either 3D or 2D. Some common applications of Human Pose estimation are: Activity Recognition, Augmented Reality, Animation, Gaming, and many more.

Early traditional methods used to rely on hand-craft features, which formulate the problem of human keypoints estimation as a tree-structured or graphical model problem. Many recent methods on human pose estimations use Deep Convolutional Networks to predict the keypoints of the human body in an image.

2.2 Literature Review

DeepPose by toshev et al. [9] began the shift from classic approaches to the use of deep neural networks. Most of the recent pose estimation systems have universally adopted Convolutional Neural Networks as their main building block, largely replacing hand-crafted features and graphical models; this strategy has yielded drastic improvements on standard benchmarks.

The work by tompson et al. [10] adopted the heatmap representation of human body keypoints to improve their localization during training. They used a multi-resolution CNN architecture (coarse heatmap model) to implement a sliding window detector to produce a coarse heatmap output.

Followed by the work of Tompson, came Convolutional Pose Machines by shihen et al. [14]. This was an interesting work, Convolutional Pose Machines used a sequential prediction framework to learn long range spatial relationships by using larger receptive fields and they proved to work very well for Human Poses.

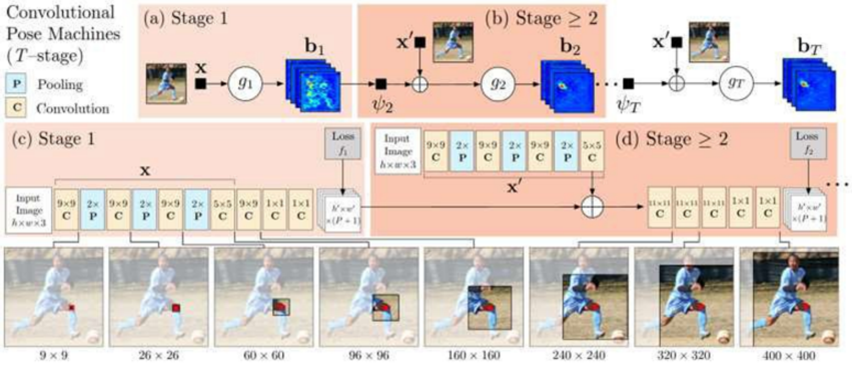


Figure 2.1: Convolutional Pose Machines [14]

Down the line, came the Stacked Hourglass Network, proposed by Newell et al. [4]. This was a landmark paper that introduced a novel and intuitive architecture and beat all previous methods. It's called a Stacked hourglass network since the network consists of series of downsampling and upsampling layers which looks like an hourglass, and these are stacked together.

To understand Stacked Hourglass network, first we need to understand convolutional autoencoders.

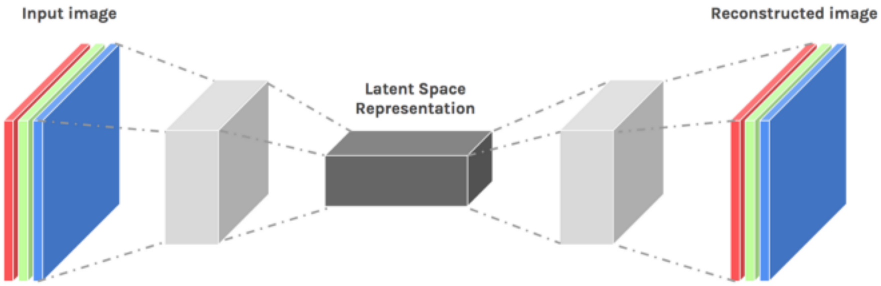


Figure 2.2: Convolutional Auto Encoders [15]

Convolutional autoencoders Fig. 2.2, are used to reduce the high-dimensional image input into a lower dimensional state-space using downsampling layers, and then try to reconstruct the input from this representation using upsampling layers. Downsampling is called as encoding and upsampling is the decoding phase. By reducing the number of dimensions, we force the model to learn how to keep only meaningful information, from which the input is reconstructable.

The design of the hourglass is motivated by the need to capture information at every scale. While local evidence is essential to identify features like neck or hands, a final pose estimate requires a global context. The person's orientation, the arrangement of their limbs, and the relationships of adjacent joints are among the many indications that are best recognized at different scales in the image.

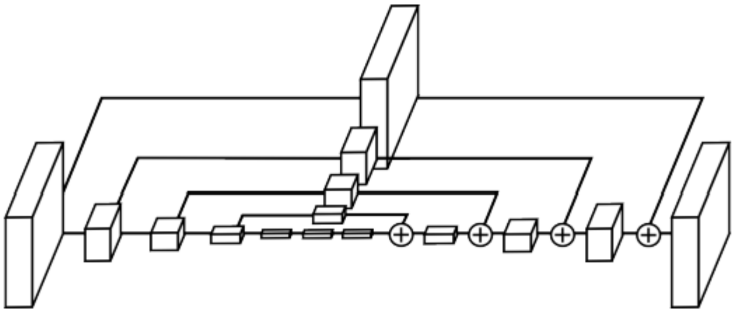


Figure 2.3: Single Hourglass Module [4]

Fig 2.3 Illustrates a single ”hourglass” module. Each box in the figure corresponds to a residual module. The number of features is consistent across the whole hourglass.

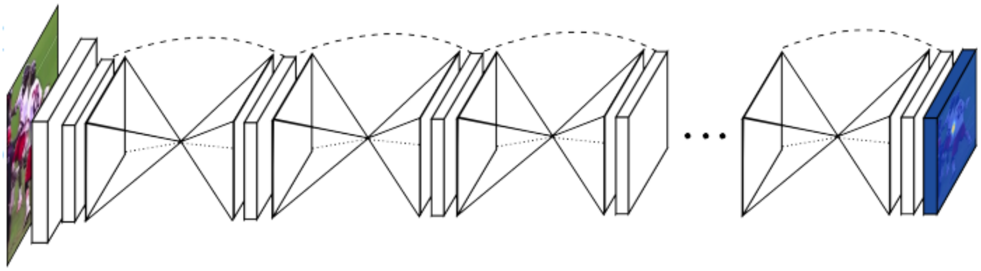
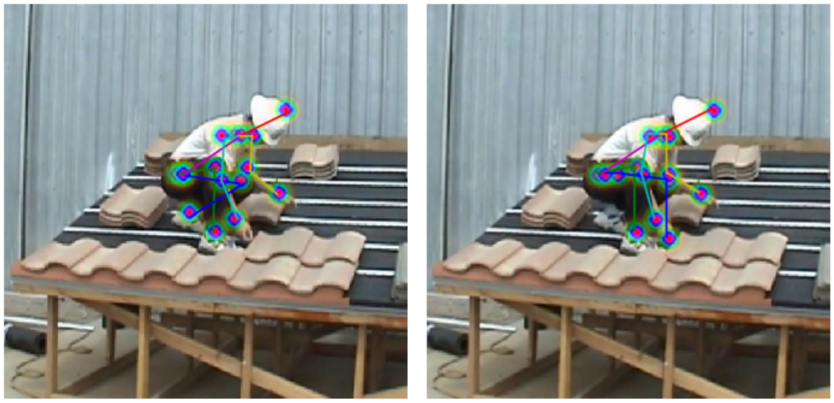


Figure 2.4: Stacked Hourglass Network [4]

Intermediate supervision is applied to the predictions of each hourglass stage. The hourglass captures information at every scale. [4] This way, global and local information is captured efficiently and are used by the network to learn the predictions.

However, DCNNs are still limited in the capability of modeling human body’s structural integrity. Existing methods rely on a brute-force approach, of increasing the network depth to implicitly enrich the keypoint relationship modelling capability, which makes them very good in locating local features but tends to ignore the global relation between these features. This leads to implausible pose predictions in cases involving partial occlusions and overlapping body poses.

Figure 2.5: Relevant example showing failure of DCNN dealing with heavy occlusions



In the recent years, there have been some significant approaches to incorporate structure awareness while doing pose estimation. The use of graph convolutional network by bin et al. [7] focuses on exploiting correlations between the local areas of adjacent key points to refine the location of predicted keypoints.

Some works tend to increase the receptive field large enough for learning the long-range spatial relationship [12] and [6], refining the process by doing intermediate supervision. Taking HRNet (High-Resolution Network) by sun et al. [12] for example. Most of the previous papers went from a high to low to high resolution representation. HRNet maintains a high-resolution representation throughout the whole process and out performs all existing methods on keypoint detection.

Although approaches taken by [6],[12] and [7] focuses on structure awareness of the human body, they do so either by increasing the resolution or by increasing

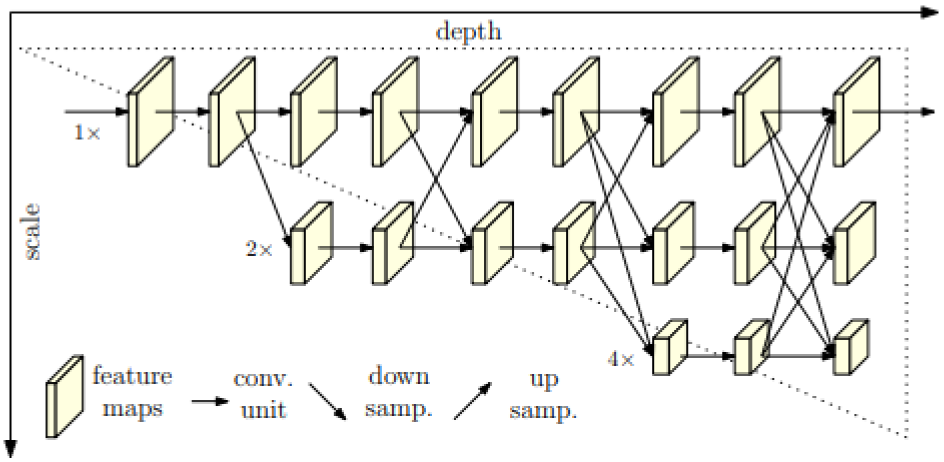


Figure 2.6: HRNet consists of parallel high-to-low resolution subnetworks with repeated information exchange across multi-resolution subnetworks (multi-scale fusion) [12]

the receptive field large enough for learning the long-range spatial relationship, in both the cases the computational or memory-usage demands of the model increases significantly which again is undesirable when you have processing constraints.

But then there are other works which introduced a novel approach to use conditional GANs for instilling structural awareness into the DCNN pose model.

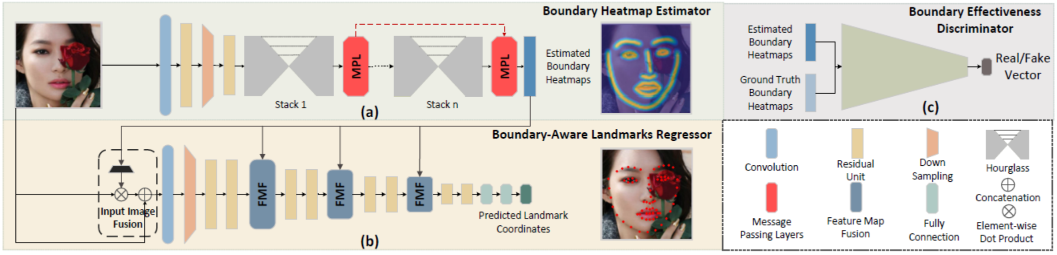


Figure 2.7: Look at Boundary: A Boundary-Aware Face Alignment Algorithm [2]

Fig 2.7 is an overview of the Boundary-Aware Face Alignment framework which is a facial landmark detection algorithm by wu et al. it uses a discriminator to further improve the quality of boundary heatmaps generated by the generator (here a stacked hourglass network) and lead to better landmark coordinates prediction.

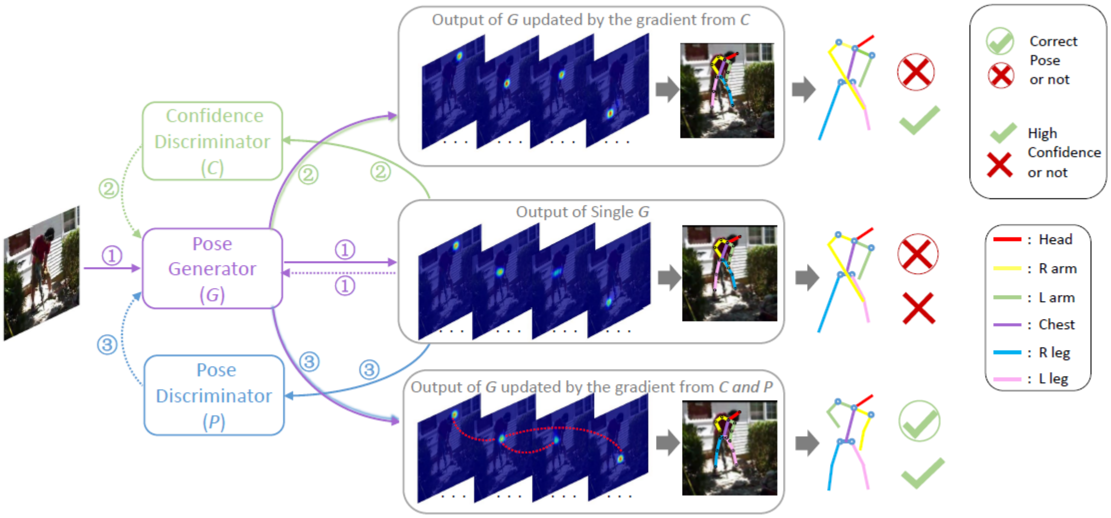


Figure 2.8: Adversarial PoseNet [1]

Adversarial Posenet [1] by chen et al. works on a similar approach. But they use two discriminators to train their generator adversarially as shown in Fig. 2.8.

Generative adversarial networks (GANs) excel in generating natural images such as human faces and indoor scenes. With the introduction by Goodfellow et al. [16], the two-player minimax game allows unsupervised training of generative models and avoids the blur effect of using variational autoencoders.

Radford et al. [17] introduce DCGAN, an all convolutional architecture which is easier to train. They propose some elements to increase the model stability such as eliminating the fully connected layer and employing batch normalization to prevent from mode collapsing. DCGAN uses an effective network configuration to make the training of GANs more feasible.

Due to the success of GANs on generating images, it also drew attention to the field of supervised learning. The concept of conditional GAN [18] is introduced for incorporating class information. Several methods combine the conditional GAN loss and the L1 or L2 distance between generated data and ground-truth data. The methods of [19], [20], and [21] use this solution to perform tasks of super-resolution, image in-painting, and image translation. Also, the methods of [2] and [1] also discusses the use of conditional GANs for keypoint detection in facial alignment, and pose estimation.

We have proposed a method which also focuses on improving structure awareness by using conditional GANs. Our approach is very similar to that of [2] and [1]. The detailed explanation is provided into the next chapter i.e. Chapter 3. The benefit of using this method over other methods like [12], [6] or [8] is that we achieve structural awareness into our model without making it computationally heavy.

Chapter 3

Methodology

3.1 Overview

Our model splits into two networks, the pose generator and the discriminator. The first network, pose generator, is a fully convolutional network with residual blocks and a conv-deconv architecture, also known as stacked hourglass network. We have used 4 stacks for our purpose unlike 8 stacks used by newell et al. [4].

The inputs to the generator G are RGB images, after feeding forward through the generator network, we get a set of heatmaps that indicate the confidence score at every location for each keypoint, corresponding to the 16 keypoints of the human body specified in the MPII dataset [13]. The second network, discriminator D, has the same architecture as the generator but it encodes the collective heatmap predictions, generated by the Generator G, combined with the original RGB images and decodes them into new set of heatmaps.

The set of new heatmaps is then used to discriminate the real heatmaps from fake ones. The framework of our model is illustrated in Fig. 3.1.

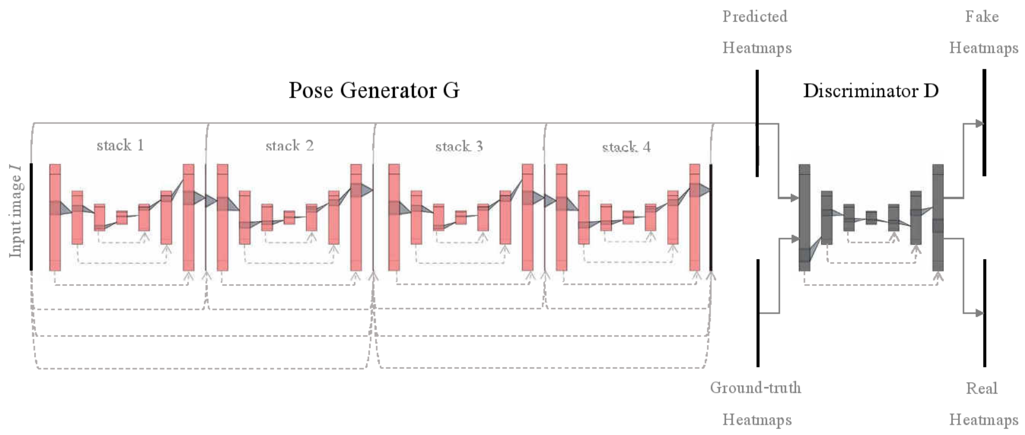


Figure 3.1: The framework of our structure aware convolutional pose network. We incorporate a Autoencoder Convolutional Network based pose estimator as the generator (on the left) with a discriminator (on the right) that aims to discriminate whether the generated pose is reasonable or not by reconstructing the input heatmaps. The generator and the discriminator have the same architecture.

3.2 Generator

The task of the generator G is to learn map the relations from an RGB image to keypoint heatmaps. The DCNN architecture allows itself to learn contextual feature representation from the input images. Furthermore, the adversarial loss from the discriminator is also combined with the mean-squared error between the generated heatmaps and the ground-truth heatmaps. This process helps the generator to learn not only the local features and spatial dependencies, but also the priors of the human body configurations.

3.2.1 Pose Gnerator Network Architecture

We have used the stacked hourglass architecture [4] as our pose generator network. It is a fully convolutional network with residual modules as its building blocks as shown in Fig 3.2. The network starts with an initial process of a 7 × 7 convolution with stride 2, followed by several residual modules and max-pooling layers. The initial process reduces the resolution of the feature maps from 256 x 256 to 64 x 64 [4][22]. Then, a sequence of hourglass modules are stacked to predict the keypoint heatmaps.

A single hourglass module is an encoder and decoder design, Fig. 2.2, to extract the features at every scale. For human pose estimation, we need to explore both the local evidence, such as a small region around the neck, and the relative relationships between the joints. To maintain this information and to integrate global and local context simultaneously, skip connections are required, and features at each resolution can be better preserved [22]. A single hourglass module is shown in Fig 3.2 ref [4], each box corresponds to a residual module.

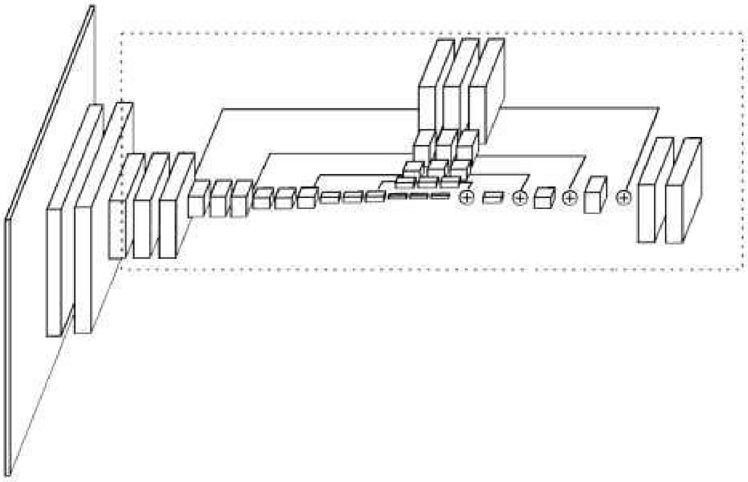


Figure 3.2: Single Hourglass module, each box represents a residual module

3.2.2 Training the Pose Generator

Training the generator is done by back-propagating the loss $\mathcal{L}_G$, which is the loss $\mathcal{L}_{MSE}$ from generator itself combined with the discriminator loss $\mathcal{L}_{adv}$ from the discriminator. The generator consists of 4 stacks of hourglass modules as shown in Fig 3.1. The expected output for each hourglass module contains M heatmaps, each of which is a 64 x 64 heatmap with a Gaussian centered at the ground-truth location of the j_{th} pose-joint. The supervision is conducted at the end of each hourglass. The $\mathcal{L}_{MSE}$ can be expressed as.

$$\mathcal{L}_{MSE} = \frac{1}{2MN} \sum_{n=1}^{N} \sum_{i=1}^{M} ||y_{ij} - \hat{y}_{ij}||^2 \tag{1}$$

where y_{ij} is the ground-truth heatmap of the j_{th} pose-joint at the i_{th} stack, and $\hat{y}_{ij}$ is the predicted heatmap. We calculate the MSE loss between them to impose the generator to learn the features of the pose, as well as learn to localize the pose keypoints.

In addition to the MSE loss described in Eq. (1), we add a discriminator loss, which helps G to produce plausible poses. The discriminator loss $\mathcal{L}_D$ is explained in later sections, ref Eq. (3) and Eq. (3.1).

Therefore, the training loss for the generator $\mathcal{L}_G$ is defined as follows.

$$\mathcal{L}_G = \mathcal{L}_{MSE} + \alpha \times \mathcal{L}_D \tag{2}$$

where α is the hyperparameter to control the influence of the discriminator loss.

3.3 Discriminator

The purpose of the discriminator is to differentiate between the fake poses (poses which do not satisfy the constraints of the human body joints) and real poses. The inputs to the Discriminator are the generated heatmaps or the ground truth heatmaps, combined or added together with the corresponding RGB image, as shown in Fig. 3.3

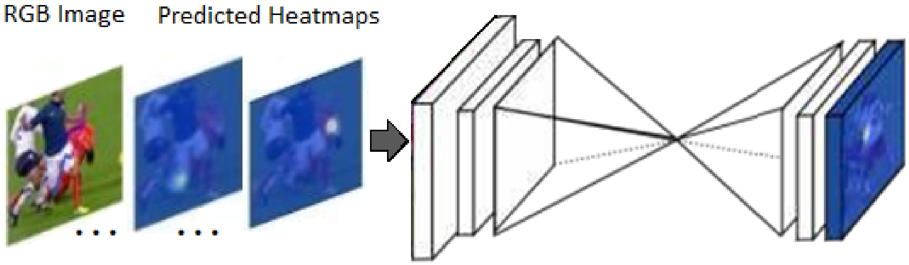


Figure 3.3: Discriminator input and output explained

Discriminator network architecture is same as of the generator network, we have used a single stack hourglass architecture [4], described in the generator section, as our discriminator network. The discriminator attempts to reconstruct a new set of heatmaps.

3.3.1 Training the Discriminator

For each training image I, the discriminator will be forwarded with the generated and ground-truth heatmaps separately. The pose GAN is set in the conditional manner. Unlike GANs which focuses on generative modelling, conditional GANs (cGANs) learn a conditional generative model [19]. The objective function for the conditional discriminator network D is expressed as follows:

$$\mathcal{L}_D = \mathbb{E}[\log(D(y))] + \mathbb{E}[\log(1 - |D(G(I)) - d_{fake}|)] \tag{3}$$

Where, y is the ground truth pose heatmap combined with the original image I and $D(G(I))$ gives the predicted heatmaps also combined with the corresponding

RGB image I as shown in Fig. 3.

In traditional GAN, the term d_{fake} is usually kept as 0, but to achieve convergence using the conditional GAN loss as prescribed by [2], we have taken d_{fake} as a 16 x 1 unit vector containing 0 and 1. d_{fake} is calculated as follows.

$$d_{fake} = \begin{cases} 1, & \text{if } d_j < \delta \\ 0, & \text{if } d_j \geqslant \delta \end{cases} \tag{3.1}$$

where δ, is the normalized distance threshold parameter and d_j is the normalized distance between the predicted and ground-truth location of the j_{th} body part.

As prescribed by [21], [2] and [1], conditional GANs perform better when GAN objective is combined with a traditional loss, such as ℓ_2 distance. For our task, like [2] and [1], the generator will try to fool the discriminator but, at the same time it will also learn to approximate ground-truth in an ℓ_2 manner as shown in Eq. (3). Therefore, final mini-max objective function is presented as follows:

$$\arg\min_{G} \underset{D}{max}\, \mathcal{L}_G(I) + \alpha\mathcal{L}_D(G, D) \tag{4}$$

3.4 Adversarial Training

Based on generative adversarial networks (GANs) [16] and conditional GANs (cGANs) [19], our training scheme is supervised learning along with a two-player mini-max game. As evident from Eq. (4), the generator aims to minimize $\mathcal{L}_{MSE}$ from Eq. (1), and the discriminator focusses on maximizing the $\mathcal{L}_D$ given in Eq. (3). **Algorithm 1** demonstrates the whole training process as the pseudo codes.

3.5 Algorithm

Algorithm 1: The training process of our method

```
input : Training images: $I$, the corresponding ground-truth heatmaps $y$;
while $\hat{y}$ improves do
    Forward $G$ by $\hat{y} = G(I)$;
    Compute gradient $\nabla G$ w.r.t. Eq. (1);
    Forward $D$ by $\hat{y}_{real} = D(\hat{y})$, and optimize $D$ by maximizing the first
      term in Eq. (3);
    Forward $D$ by $\hat{y}_{fake} = D(G(I))$, and optimize $D$ by maximizing the
      second term in Eq. (3);
    Optimize $G$ according to Eq. (2)
end
```

3.6 Hyperparameters α and δ mentioned in this chapter

Hyperparameters		
Hyper-parameters	**Value**	**Description**
α	$\frac{1}{200}$	α is the hyperparameter to control the influence of the discriminator loss.
δ	0.003	δ is the normalized distance threshold parameter and d_i is the normalized distance between the predicted and ground-truth location of the i_{th} body part

Table 3.1: Training hyperparameters α and δ

Training hyperparameters is discussed in detail, in Chapter 5, training and experimental settings.

Chapter 4

Datasets

The two widely used benchmarks for 2D Human Pose Estimation are MPII Human Pose [13], and extended Leeds Sports Poses (LSP) [23].

4.1 MPII Dataset

MPII Human Pose dataset is a benchmark dataset for 2D human pose estimation evaluation. The dataset includes 25K images containing 40K+ people with annotated body joints. The images were methodically collected using an established taxonomy of general human activities. The dataset covers 410 human activities, where each image is provided with its activity label.

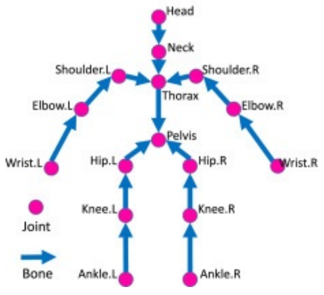


Figure 4.1: Shows 16 key-joints present in the MPII Dataset [13]

4.2 Leeds Sports Pose (LSP) Dataset

The Leeds Sports Pose (LSP) dataset is widely used as the benchmark for human pose estimation. The extended LSP dataset consists of 11,000 poses for training and 1,000 for testing. Each image is annotated with 14 keypoint locations. The images are gathered from Flickr and contain people who are doing sports such as baseball, parkour, tennis, and so on.

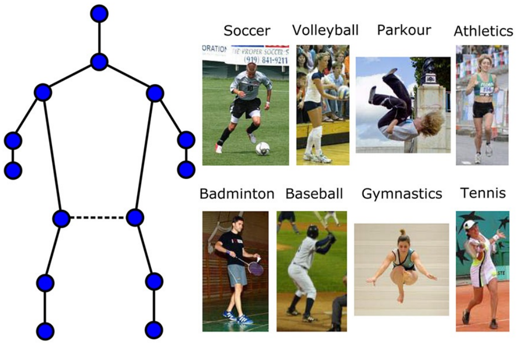


Figure 4.2: Shows 14 key-joints present in the Leeds Sports Pose (LSP) Dataset [13]

For our experiments we have trained and tested our model on MPII dataset. Further elaborated in Chapter 5, training and experimental settings.

Chapter 5

Training and Experimental Settings

5.1 Dataset

As discussed in the previous chapter, we evaluate our method on MPII Human Pose Dataset [13]. In the following experiments, we train our model on a subset of training images and evaluate on a validation set of 2958 images, along with a test set of 300 images.

5.2 Data Augmentation

We have followed the same data augmentation methods as used by newell et al. [4]. We randomly flip an input image horizontally, rotate it by an angle in $[-30, +30]$ degrees, and scale it in $[0.75, 1.25]$ to make the network more robust to different scales and directions.

5.3 Training configurations

We followed the standard routine to crop image patches using the given position and scale [4]. The input to the generator network is of dimensions 64 x 64 pixels. When input with an RGB image of resolution 256 x 256, the initial pre-processing involves 7 x 7 convolutional layer with a stride of 2, followed by a residual module, and a max pooling layer to drop the resolution to 64 x 64.

We train our model using PyTorch [25]. Our network is trained using Adam optimizer with the initial learning rate of 1×10^{-3} and decay learning rate of 2×10^{-4}, where the decay iterations is taken as 100k.

The model was trained on the MPII dataset for 200 epochs, where 1 epoch is of 1000 iterations. I takes about 2 days to train the model on Nvidia GeForce GTX 1080 Ti. Table 5.1 shows the training configuration and hyperparameters used in the training.

5.4 Evaluation Metrics

For MPII dataset we use PCKh error as a common metric used by the state of the art methods to measure accuracy of the predictive model.

5.4.1 PCKh (Percentage of Correct Keypoints with respect to head)

To understand PCKh error [13], we need to understand PCK (Percentage of Correct Keypoints) error [24] first. PCK gives the percentage of correct keypoint detection that happens to be within certain tolerance range. The tolerance range is the fraction of torso size. The equation can be expressed as:

$$\frac{||y_i - \hat{y}_i||_2}{||y_{lhip} - y_{rsho}||_2} \leqslant r, \tag{5}$$

where y_i is the ground-truth location of the i_{th} keypoint and $\hat{y}_i$ is the predicted location of the i_{th} keypoint. The fraction r is bounded between 0 and 1.

PCKh is almost the same as PCK except for the tolerance range r, it is a fraction of the head size.

Model Training Configuration		
Config	**Value**	**Description**
N	4	Number of stacks in the hourglass network, or the generator.
learning rate	1×10^{-3}	Initial learning rate.
decay learning rate	2×10^{-4}	Decay Learning rate.
decay iterations	100K	Decay iterations; number of iterations after which the initial learning rate changes to decay learning rate.
batch size	8	Number of images trained per iteration.
train iterations	1000	Training iterations per epoch.
validation set	2958	Number of images used for validation.
epochs	200	Number of epochs for which the model is trained.
α	$\frac{1}{200}$	α is the hyperparameter to control the influence of the discriminator loss.
δ	3×10^{-3}	δ is the normalized distance threshold parameter and d_i is the normalized distance between the predicted and ground-truth location of the i_{th} body part.

Table 5.1: Training configuration and hyperparameters

Chapter 6

Experimental Results

Evaluation is done using the standard Percentage of Correct Keypoints w.r.t. head (PCKh) metric, discussed in the previous chapter, on the MPII dataset. We experiment on several network configurations. The settings differ in the number of stacks of the generator. The size of the discriminator is fixed (1-stack). The discriminator seems to perform well even when the image of the person is not provided. A possible reason is that the implausible pose could be recognized by merely the pose information, chen et al. [1] uses two discriminators, one is fed with the image and the other one is not. The image of the person is an extra information, but the discriminator does not always need it. Instead of increasing the number of discriminators, like chen et al., we can just increase the value of α which is the hyperparameter to control the influence of the discriminator loss.

We compare our result with the original stacked hourglass network by newell et al. [4] and investigate the benefit of using adversarial learning. The goal of this research is not to draw comparisons with other state-of-the-art methods but rather emphasise on a training methodology which can be used to achieve better results.

The following tables give a quantitative comparison between the original stacked hourglass network [4] and adversarially trained stacked hourglass networks. Table 6.1 compares the validation PCKh error of 4 stack and 2 stack hourglass network with their counterparts (proposed model), and similarly Table 6.2 compares the training accuracy.

Methods	**Head**	**Sho.**	**Elb.**	**Wri.**	**Hip.**	**Knee.**	**Ank.**	**Total**
4 Stack Hourglass Network	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Our Method (4 Stack HG)	**98.2**	**96.7**	**92.1**	**87.8**	**91.1**	**88.6**	**84.3**	**91.3**
2 Stack Hourglass Network	95.5	94.6	88.0	83.3	87.0	81.3	77.7	88.3
Our Method (2 Stack HG)	**95.2**	**94.5**	**87.4**	**81.8**	**87.0**	**80.9**	**76.1**	**87.8**

Table 6.1: Validation: Stack Hourglass comparisons on the MPII dataset. (PCKh)

Looking at the validation results we realise that the original 2-stack hourglass performs better than our adversarially trained 2-stack hourglass network. However, when we look at the validation accuracy of the original 4-stack hourglass network in comparison to the adversarially trained 4-stack hourglass network, we realise that our method outperforms the original 4-stack hourglass.

Reason being, 2-stack hourglass due to lack of its depth collects relatively less features when compared with 4-stack hourglass network. As a result the discriminator does not have enough features to effectively differentiate between the set of plausible poses. This tends to shift the predicted keypoint location farther from the ground truth and hence the accuracy is lesser as compared to its counterpart.

Methods	**Head**	**Sho.**	**Elb.**	**Wri.**	**Hip.**	**Knee.**	**Ank.**	**Total**
4 Stack Hourglass Network	99.0	99.2	91.2	95.2	94.0	87.4	89.8	93.7
Our Method (4 Stack HG)	**99.1**	**99.4**	**91.3**	**95.6**	**93.9**	**87.6**	**89.8**	**93.8**
2 Stack Hourglass Network	98.0	98.8	95.4	88.3	93.4	91.8	87.2	94.2
Our Method (2 Stack HG)	**97.0**	**98.3**	**82.7**	**87.4**	**89.3**	**89.8**	**83.4**	**92.5**

Table 6.2: Training: Stack Hourglass comparisons on the MPII dataset. (PCKh)

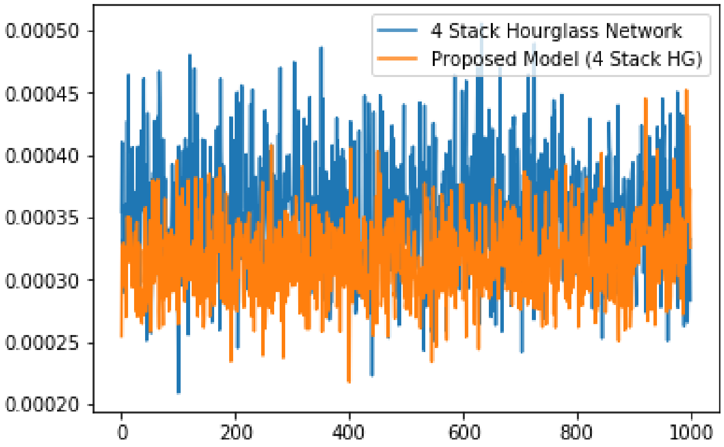


Figure 6.1: Training loss for original 4-stack HG vs adversarially trained 4-stack HG (last epoch)

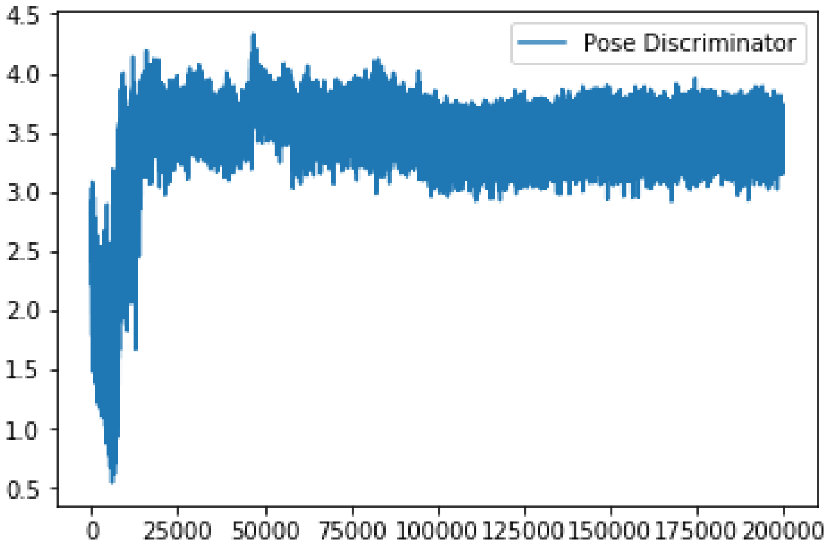


Figure 6.2: Training loss curve loss curve for pose discriminator

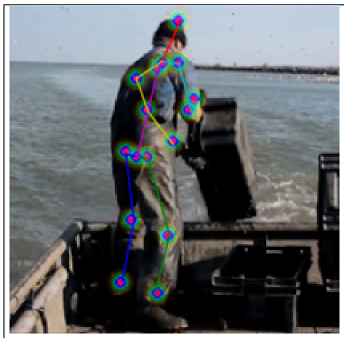
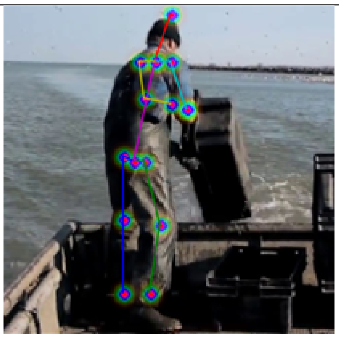
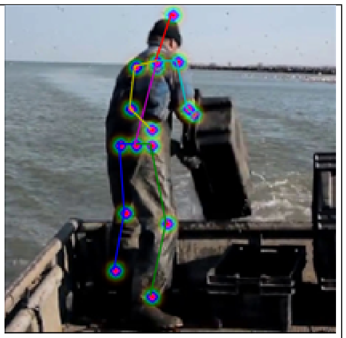
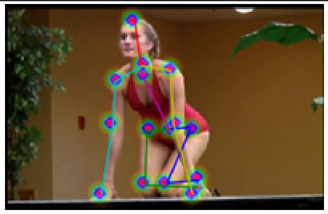
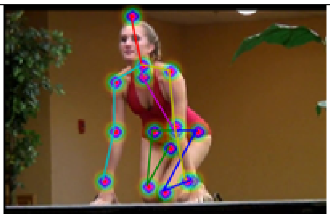
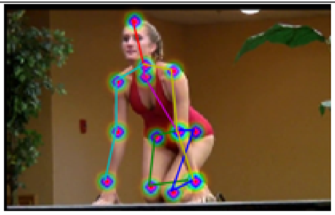
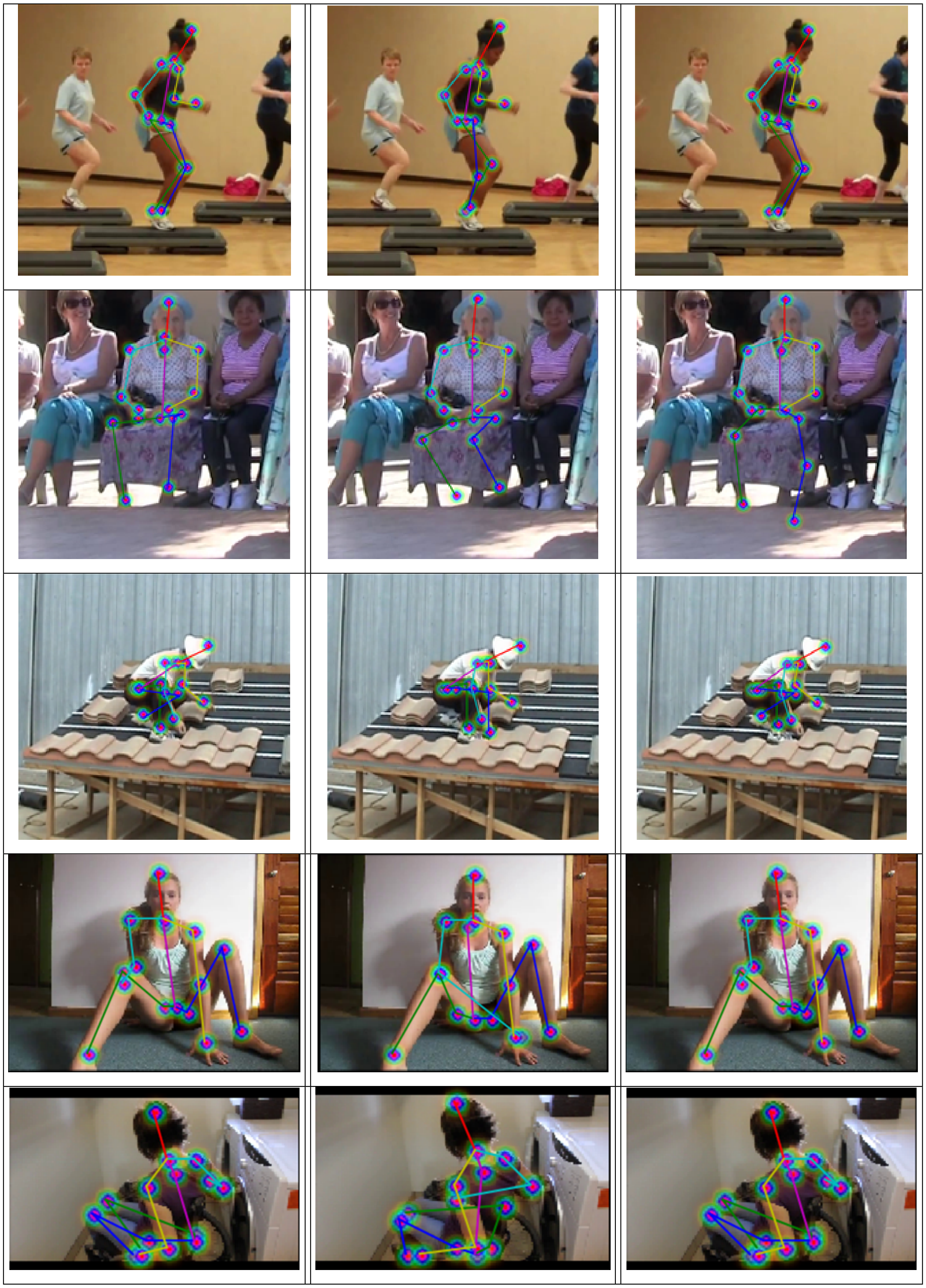
Prediction Samples		
Ground Truth	**Stacked HG Network**	**Proposed Method**
		
		

Table 6.3: Prediction samples on the MPII dataset. The first row: ground truth. The second row: results by stacked hourglass network [4]. The third row: results by our method. *Continued...*



Chapter 7

Ablation Studies

Since we have performed our experiments on Stacked Hourglass Networks which is a well accepted and confirmed work by the computer vision research community. Therefore, we need not go into the ablation study of the training configurations of the Stacked Hourglass Network. In this chapter we will mainly focus on the detailed analysis of the effect of only two hyperparameters, α and δ.

Where, α is the hyperparameter to control the influence of the discriminator's loss. Refer Eq. 4.

As we decrease the α by a significant amount, the influence of the discriminator decreases and hence the discriminator becomes ineffective. The Fig. (7.1) shows the training loss curve during the last epoch of the training, here α has been reduced to $\frac{1}{270}$.

On the other hand, If we increase the value of α by a sufficient amount it will lead us to a less accurate model (or generator), as shown in the Fig. (7.2)

δ is the normalized distance threshold parameter and d_i is the normalized distance between the predicted and ground-truth location of the i_{th} body part. To

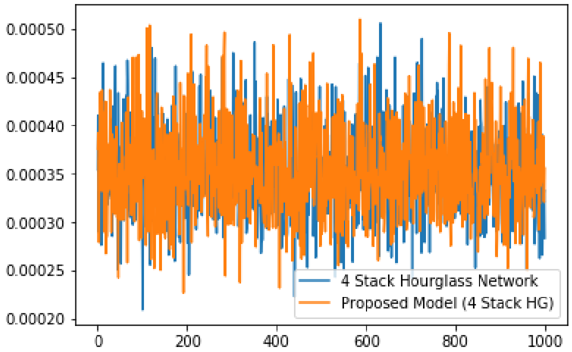


Figure 7.1: Training loss curve loss curve for pose generator w.r.t. original HG network ($\alpha = \frac{1}{270}$)

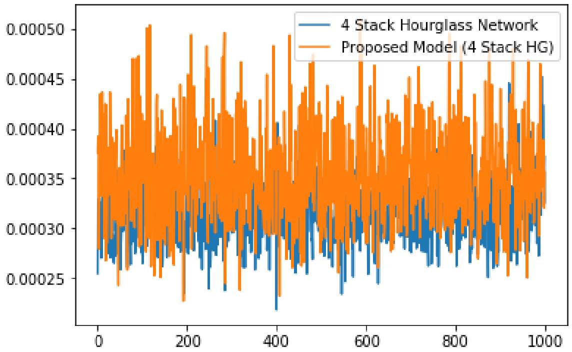


Figure 7.2: Training loss curve loss curve for pose generator w.r.t. original HG network ($\alpha = \frac{1}{110}$)

understand better, refer Eq. 3 and Eq. 3.1

On increasing the value of δ, discriminator's loss sharply plummets, which again diminishes the influence of the discriminator on our pose generator. The Fig. (7.3) shows the training loss curve of the pose discriminator, here β equals 0.8.

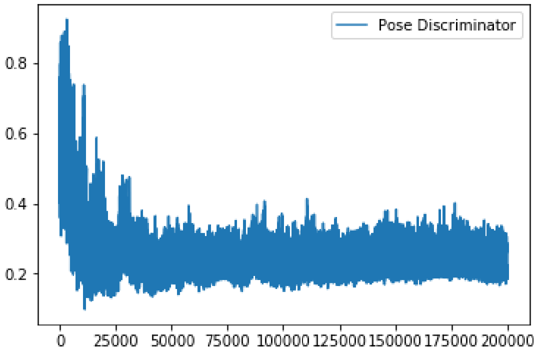


Figure 7.3: Training loss curve for pose discriminator ($\beta = 0.8$)

And let's suppose, if we take the value of β significantly low, say 2×10^{-6}, it results in gradient explosion.

Chapter 8

Conclusion

We can conclude from the experimental results that adversarial learning can help us to achieve structure awareness. However, there are couple of ways this work can be extended. Firstly, we can start with improving the model's receptivity by increasing the resolution of the subnet layers, as achieved by sun et al. in HRNet [12]. We can also refine it by performing intermediate supervision, as demonstrated by lipeng et al. [6]. This can help us with locating the local features more accurately and will give us more information for the discriminator to locate plausible keypoints.

For the discriminator module, we can try to choose a different network architecture, or we can also assign more than on discriminators in parallel, supervising the intermediate loss between the subnets of the generator model.

We can also use this same methodology to get better results in image segmentation problems, or 3D human or hand pose estimation problems. Hence the scope of this methodology is vast as well as exciting.

Bibliography

[1] Chen, Yu, et al. ”Adversarial posenet: A structure-aware convolutional network for human pose estimation.” Proceedings of the IEEE International Conference on Computer Vision. 2017.

[2] Wu, Wayne, et al. ”Look at boundary: A boundary-aware face alignment algorithm.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[3] Toshev, Alexander, and Christian Szegedy. ”Deeppose: Human pose estimation via deep neural networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[4] Newell, Alejandro, Kaiyu Yang, and Jia Deng. ”Stacked hourglass networks for human pose estimation.” European Conference on Computer Vision. Springer, Cham, 2016.

[5] Mirza, Mehdi, and Simon Osindero. ”Conditional generative adversarial nets.” arXiv preprint arXiv:1411.1784 (2014).

[6] Ke, Lipeng, et al. ”Multi-scale structure-aware network for human pose estimation.” Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[7] Bin, Yanrui, et al. ”Structure-aware human pose estimation with graph convolutional networks.” Pattern Recognition 106 (2020): 107410.

[8] Shamsolmoali, Pourya, et al. ”AMIL: Adversarial Multi-instance Learning for Human Pose Estimation.” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16.1s (2020): 1-23.

[9] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, 2014.

[10] Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network +and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems. (2014)

[11] M. Mirza and S. Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014.

[12] Sun, Ke, et al. ”Deep high-resolution representation learning for human pose estimation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[13] M. Andriluka, L. Pishchulin, P. V. Gehler, and B. Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 3686–3693, 2014.

[14] Wei, Shih-En, et al. ”Convolutional pose machines.” Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2016.

[15] Convolutional Auto Encoders: https://towardsdatascience.com/aligning-hand-written-digits-with-convolutional-autoencoders-99128b83af8b. Accessed: 04-28-2021.

[16] Goodfellow, Ian J., et al. ”Generative adversarial networks.” arXiv preprint arXiv:1406.2661 (2014).

[17] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015. 2

[18] M. Mirza and S. Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014. 2

[19] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Imageto-image translation with conditional adversarial networks. CoRR, abs/1611.07004, 2016. 2

[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photorealistic single image super-resolution using a generative adversarial network. CoRR, abs/1609.04802, 2016. 2

[21] D. Pathak, P. Krahenb ¨ uhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016. 2

[22] Chou, Chia-Jung, Jui-Ting Chien, and Hwann-Tzong Chen. ”Self adversarial training for human pose estimation.” 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018.

[23] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In Proceedings of the British Machine Vision Conference, 2010.

[24] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In CVPR, 2011.

[25] Paszke, Adam, et al. ”Automatic differentiation in pytorch.” (2017).