Human Pose Estimation using Deep Learning Models

Deepankar Kansal (MT20007)
Palak Tiwari (MT20103)
Mohd. Naki (2018052)

Abstract

Human Pose Estimation is an approach to detect key points from Human Bodies or detecting various Human Poses. The dataset used for this problem is MPII Human Pose Dataset which contains nearly 25k images, covering multiple categories of Human Activities. The problem consists of two parts classification of human poses and localization of joints, which is solved using Classification and Regression. We tried implementing various powerful Deep learning models to classify Human Poses, including CNNs, VGG16, VGG19, ResNet50, and Xception. For joint localization, we use Regression models like CNN and ResNet50, and the metrics used are PCP and PCK. We summarize and compare the performance of various models.

Keywords: Xception, ResNet, VGG, Convolutional Neural Networks as CNN, etc.

1 Introduction

In the past few years, drastic growth in digital data has given rise to Deep Learning applications. Such applications include image classification or object detection. One of the similar problems is Human pose estimation that involves the classification of activity performed by the human in the given image and identifying the major body parts and joints.

Some major application areas of Human pose estimation are action or activity classification and body movement prediction. Even though researchers are digging more profound into this area, various problems are still to be resolved, such as multi-person overlaps, complexity in the calculation, dark background, rotation and orientation of the figure, overlap of multiple subjects, etc. Our problem consists of identifying the coordinates of joints of the body by regression models and activity classification, for this we are using MPII Human Pose Dataset that covers 410 human activities annotated body joints.

This report presents baselines and variants of CNN models and the Xception model that we build, and we try to solve this problem from both the point of views, one as the *regression* problem and one as the *classification* problem. We can think of regression problem as detecting various human body joints that are present in the image. These body joints include the head, shoulder, wrist, elbow, pelvis, thorax, hip, etc. The evaluation of the correct body joint can be done using PCP (percentage of correct parts) and PCK (percentage of correct key points). In comparison, the classification problem includes detecting poses according to the labels provided and print these labels. Evaluation in this approach can be done using the accuracy, f1-score, or by printing the confusion matrix.

Further, section 2 briefly describes the past work done in the field. Sections 3 defines methodology of the project, 4 defines the dataset used and preprocessing done, 5 presents baselines used for the human pose estimation and section 6 present the results and analysis. Finally, we conclude in section 8.

2 Related Work

[1] This research paper focues on the introduction, classification and performance comparison of the methods for 2-d Human Pose Estimation. Single CNN methods combine CNN which extracted features, with traditional PS model which models the distribution of joints which further leads to low complexity of the network while on the other hand, Multi-stage CNN methods faciliates the process of feature extraction using cascading CNNs and completely models the relationships of neighboring joints with the use of receptive fields.

[2] The research paper uses 'Pose Machines'. A Pose Machine comprises an image feature computation module which is followed by a prediction module. They follow a sequential prediction framework for learning rich spatial models and hence are very useful for human pose prediction. Convolutional Pose Machines have a multi-stage architecture that can be trained end to end and are also completely differentiable. PCKh-0.5 score achieves state of the art at 87.95%. On the ankle which is the most challenging part for the learning, PCKh-0.5 score is 78.28%.

[3] Authors constructed CNN for regression on Human joint estimation and classification of human activity on images. For classification of 20 activities, achieved the accuracy of 80.5% and for the regression problem, used Percentage of Detected Joint metric for evaluation and achieved the accuracy of around 60%. The reason for using PDJ metric was that, can vary the threshold for the distance between ground truth and predicted joints.

[4] In this paper, the dataset used has images extracted from the videos recorded by daily used surveillance cameras. The dataset contains diverse outdoor images that belong to ten different categories. The evaluation metric used is the Percentage of Correct Keypointson this dataset. The models used were VGG-f, VGG-19, ResNet-50, ResNet-101 among these the best mean pose accuracy id given by ResNet-50 of 88.56%.

[5] In this paper, pose estimation is formulated as a DNN-based regression problem towards body joints. Also present a cascade of such DNN regressors which results in high precision pose estimates. Worked on two datasets *FLIC*(Frames Labeled In Cinema) and *LSP*(Leeds Sports Pose). Percentage of Correct Parts(PCP) and Percent of Detected Joints(PDJ) are used as evaluation metrics for regression problem. Achieved *0.69%* PCP using DeepPose network proposed in the paper.

In [6], the authors focus on the "MPII Human Pose" dataset, which contains near about 20 general human activities. These 20 activities are further divided into various sub-categories, making a total of 410 activities. PCP, PCK and their respective variations like PCPm and PCKh are used as evaluation metrics. All these evaluations are done on the different body joints of the human body. Achieved *69.3%* of PCPm after retraining the previously trained models.

[7] This paper uses an approach called "PoseRefiner" which introduces an effective post-processing technique for body joint refinement in human pose estimation tasks. This technique can works on top of any existing human body pose estimation approach. The pose refinement network proposed in the paper is efficient due to its feed-forward architecture, simple and end-to-end trainability. The paper proposes a training data augmentation scheme which is used for the error correction, which further enables the network to identify incorrect and erroneous body joint predictions and to learn a way to refine them. The refinement

network was able to improve the best reported results on MPII Human Pose and PoseTrack datasets for multi-person pose estimation and pose tracking tasks.

[8] The paper focuses on building a novel recurrent architecture with Long Short-Term Memory (LSTM) to capture the time-related geometric consistency and the dependency among video frames for pose estimation. The methods introduces a new architecture which decouples the relationship among network stages and results in a faster inference speed for videos. This method also probes into the LSTM memory cells and visualizes the method for improving the prediction of joints in videos. The method used in the paper surpassed all the existing approaches for pose estimation on two large-scale benchmarks.

3 Methodology

This problem can be solved from the classification point of view to detect human pose activity, as well as from the regression point of view to detect body joints.

3.1 Activity Classification

For classification, we have to convert the image activities/classes into labels. After converting classes into labels, we can perform one-hot encoding of the labels because the proposed model will predict probabilities of various classes using *softmax* activation function and select the maximum predicted probability class.

After converting classes to their one-hot encodings, we can now design our model to classify the images. As the model that we are going to create requires standard custom input of the shape of images. We are using $(224 \times 224 \times 3)$ as an input shape.

After creating inputs to the model, we require to design the model that best classifies the data. As deep neural networks work like a black box, one has to try various methods and hyper-parameters to improve the classification accuracy. There are three variants of convolutional neural network (CNN) that we are proposing, and all of them are working well for the classification tasks as results shown in 6. The sequential CNN model variant that gives the best classification accuracy is described in figure 1.

First, we provide an input of shape $(224 \times 224 \times 3)$ to the network. After this, 3-conv2D blocks are there, which are defined as: $block1 = (64 \times 3 \times 3)$, $block2 = (128 \times 3 \times 3)$, $block3 = (256 \times 3 \times 3)$. Each block contains 2 conv2D layers, each convolutional layer is followed by a batch-normalisation layer, each block is followed by max-pool2D layer of (3×3) pool size and a dropout layer.

After 3 blocks of convolutions, the final output gets flattened and further processed on dense layer of shape $(256 \times 1 \times 1)$ followed by a batch-normalisation and dropout layer. After this, the final layer contains number of classes our data has and softmax activation is used to predict class probabilities. All the other convolutional layer has ReLU activation.

Other models that can be used for the classification problem includes VGGnets, ResNets and XceptionNets. These are pre-trained on the imagenet data and thus have weights obtained from such large data (imagenet). We also used these pre-trained models for classifications. When using these models, some initial layers left un-trainable, and these layers will not update their weights while training.

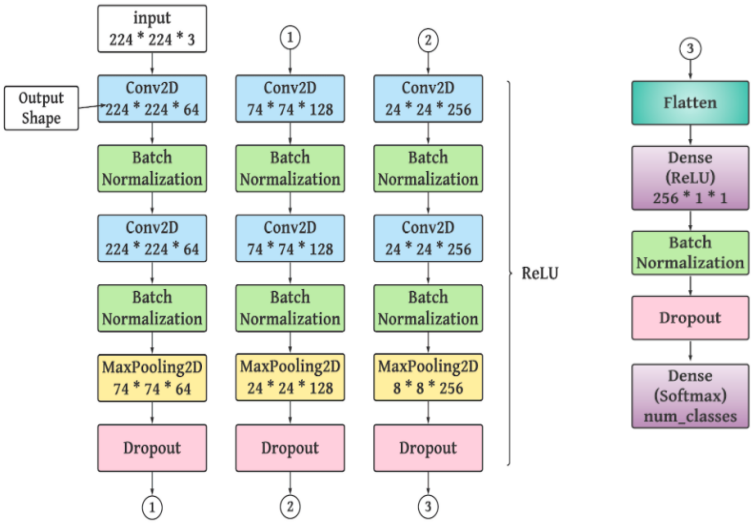


Figure 1: Best CNN variant for classification task

3.2 Joints Detection

Regression problem is of a different kind in a way that it only focuses on locating body joints efficiently. There are 32 points present in the data containing 16 body joints. These are; right-ankle, right-knee, right-hip, left-ankle, left-knee, left-hip, pelvis, thorax, upper-neck, head, right-wrist, left-wrist, right-elbow, left-elbow, right-shoulder and left-shoulder. We can change the final layer to a linear activation layer of $(32 \times 1 \times 1)$ output shape for the regression problem. This output layer helps the model generate outputs according to the training data outputs. By calculating loss on epochs, the model tries to minimize the loss, which can be seen as a gap between predicted values and actual values as described in the figure 3.

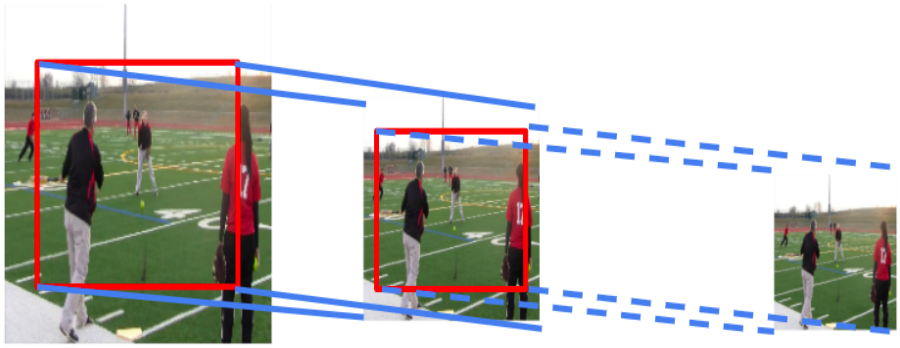
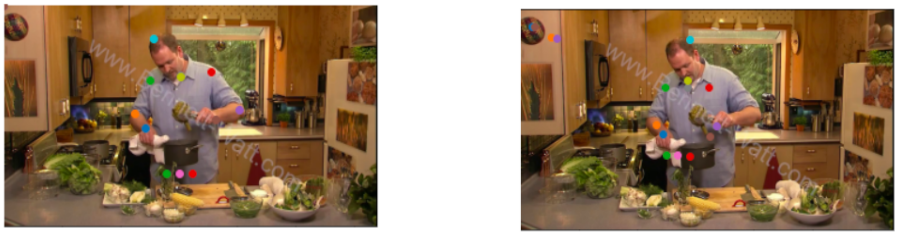


Figure 2: Visualising CNN model working

Since the output layer contains linear activation with an output shape of $(32 \times 1 \times 1)$, these output results will then be calculated for loss. Various loss metrics that can be used for

this problem contains, *mean absolute error (MAE), mean squared error (MSE), etc.*

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$



Original Joints on image

Predicted Joints on image

Figure 3: Visualising predicted joints with original joints, given in the data

$$MSE = \frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}$$

Where y_i is the predicted value, x_i is the original output, n is the total number of data points. Thus by minimising these errors, models are able to recognise body joints efficiently.

4 Dataset

4.1 Dataset Description

To apply Deep Learning algorithms, it is essential to have a massive dataset in quantity and diversity. For Human Pose Estimation, we choose MPII Human Pose Dataset, one of the popular choice among various available datasets for Human Pose Estimation. As this dataset is huge as well as diverse and also publicly available. This dataset consists of 25k images with annotations file, which contains the information of the joints.The annotation file consists of head position information and various joints location with their ids. There 16 joints (0-15), including ankle, knee, hip, upper neck, pelvis, thorax, elbow, shoulder, head top, wrist. The images present in this dataset are collected from YouTube videos. Overall there are 20 categories and 397 activities present in the dataset. However, due to computational challenges, we reduce the dataset to 17 categories and three sets containing 14, 20 and 50 activities. Figure 4 shows the original distribution of classes where as Figure 5 shows the class distribution after dataset reduction.

4.2 Preprocessing

MPII Human Pose Dataset contains images from YouTube videos thus various images have uneven dimensions. We resize the images to the shape of (244x244x3). Using data augmentation techniques too, we resize and re-scale the images to feed into the Deep Learning models. The batch size used for data augmentation was 30.

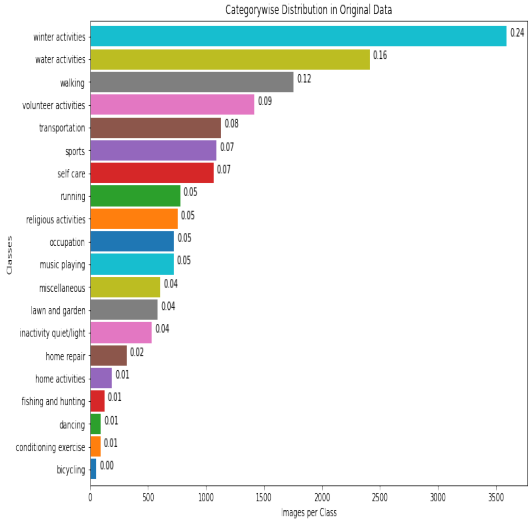


Figure 4: Original data Distribution

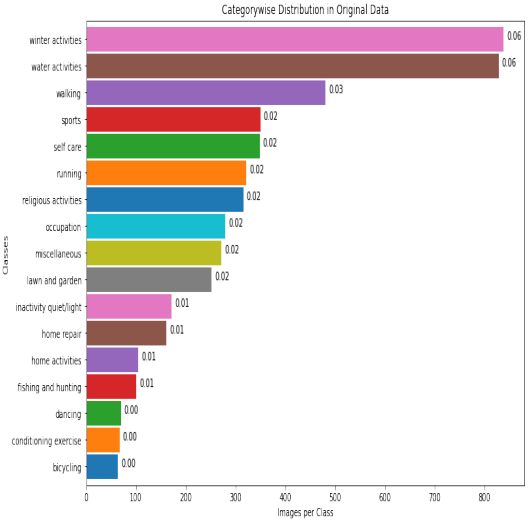


Figure 5: Reduced data Distribution

5 Baselines

5.1 Midsem Project Baselines

We performed the activity classification on twenty categories as multiclass classification. We took the images from these categories and performed data augmentation first on them by re-scaling them and resized them to (244 x 244 x 3) size. Adam optimizer and a learning rate of 0.0001 has been used, categorical cross-entropy is used as loss function and accuracy as a metrics. The baseline architecture, followed from [2], contains five convolution layers with the ReLU activation function, followed by normalization and max-pooling. The final dense layer uses Softmax as an activation function to make the model for multiclass classification.

Regression has been performed as the second baseline architecture, input size of the image has been resized to (244 x 244 x 3). Adam optimizer and a batch size of 32 has been used, keeping the default learning rate, mean absolute error is used as loss function. The architecture consists of three convolution layers with ReLU activation function, followed by pooling layer and the final dense layer using linear activation. PCP (Percentage of Correct Parts) is used as evaluation metric.

5.2 VGG16

The name VGG16 signifies that all 16 layers contain weight. The architecture of VGG16 includes five-stage blocks, starting with a convolution layer with a filters of 64, then followed by 128, 256, 512, 512. Each convolutional layer consists of ReLU as an activation function, and each block consists of max-pooling layers with stride(2x2) and pooling size(2x2). Ending with three fully connected layers with ReLU as an activation function. The input dimension required is (244x244x3).

5.3 VGG19

Similar to VGG16, VGG19 consists of 19 weighted layers as it contains three more Conv-3 layers. Starting with a convolution layer with a filter size of 64, followed by 128, 256, 512, 512 with ReLU as an activation function. Like VGG16, each block is followed by a max-pooling layer with stride (2x2) and pooling size (2x2). The architecture ends with three fully connected layers with ReLU as an activation function.

5.4 ResNet50

The only difference between usual network and residual network is that it contains identity connection between the layers. The idea behind residual block is to increase the accuracy by learning the deviation between input and output. The ResNet architecture consists of five-stage containing a convolutional layer followed by an identity block. Identity blocks are where input and output dimensions are the same, whereas, in convolutional blocks, input and output dimensions are different. Also, in the convolutional block, the shortcut layer consists of a convolutional layer too.

5.5 Xception

Xception is one of the robust deep convolutional neural network architectures developed by Google Researchers. It consists of entry flow and middle flow followed by batch normalization then exit flow after repeating middle flow eight times. Xception architecture consists of 14 modules having linear residual around them except the first and last module and overall 36 convolutional layers with ReLU as an activation function forming base for the feature extraction. Xception architecture consists of pointwise convolution then Depthwise convolution. Depthwise convolution means applying convoultion channel by channel. Similar to ResNet architecture, Xception also has shortcuts between convolutional blocks.

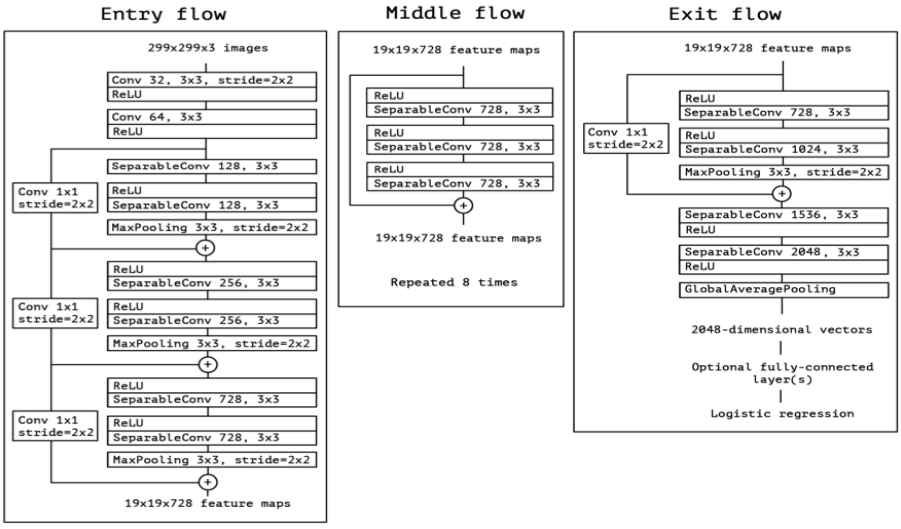


Figure 6: Xception Architecture

6 Results and Analysis

For the Activity Classification we measure the results using Accuracy as a mertic, for various subsets of data. Xception model performed relatively accurate in all the three sets. However for 14 classes set ResNet50 performed the best among all models.

Classification Results

	CNN 1	CNN 2	CNN 3	VGG16	VGG19	Xception	ResNet50
14 Classes	69.69%	83.37%	69.10%	86.79%	90.33%	71.23%	91.04%
20 Classes	73.47%	79.48%	72.88%	81.18%	83.14%	91.18%	62.56%
50 Classes	57.75	73.57%	65.74%	--	--	82.18%	46.36%

Figure 7: Classification Results with Validation Accuracy

Regression Results

	CNN	ResNet50
PCP Metric	82.71%	86.36%
PCKh Metric	88.22%	88.84%

Figure 8: Regression Results with PCP and PCKh

From the proposed CNN variants, the 2nd variant works quite well for both regression and classification tasks. When run for 14, 20 and 50 classes, this CNN gives 83.37%, 79.5% and 73.57% accuracies, respectively. The only limitation of this model is, it does not beat the baselines. Anyhow it is very accurate when compared to proposed baselines. Because other pre-trained models are taking weights from the imagenet dataset and thus their weights are already saturated for the number of images and give good accuracy.

For joint detection, CNN working quite well when compared to ResNet50. Both these models used mean absolute error as loss function and thus try to minimize loss using the difference between original joints and predicted joints as discussed in section 3.2. Proposed CNN gives 82.71% PCP, 88.22% PCKh, and ResNet50 model give 86.36% PCP and 88.84% PCKh.

7 Contribution

Deepankar Kansal (MT20007): Literature surveys, Data preprocessing (reducing data for compatibility with google colab), CNN variants (2), Regression models (CNN), VGG19 for classification task, visualisation of final predited output values for reduced data models, report writing, adding images to the report for better understanding, methodology.
Palak Tiwari (MT20103): Literature surveys, Data augmentation, Data analysis, Xception model, Resnet50 for both classification and regression, visualisation of final output values for data augmentation models, report writing, baselines, results and analysis plotting and explaining.
Mohd. Naki (2018052): Literature surveys, CNN variants (1), VGG16 for classification, PCP and PCKh metrics and other eveluation results, report writing, adding visualising results in code files, literature review writing.

8 Conclusion

For addressing the Human Activity Classification problem, seven different models were applied and compared. For the Joint localization problem, two models were used and compared. However, the results achieved in the case of regression could be improved by minimizing the loss. We also present the Xception model in this paper and found that it gives outstanding classification accuracy and out-performing other baselines. Due to lack of resources, we are not able to check the proposed model's results for the whole data because the whole data consists of 25k images and near about 13 GB, but colab's RAM is around 12 GB, So it is not feasible to run large data on this environment.

Future aspect is to run proposed models for the whole data and change their architecture accordingly to get maximum accuracy. One can also use other emerging deep learning models like GAN's to check for classification and joints detection using regression.

References

[1] Y. Liu, Y. Xu, and S. Li, "2-d human pose estimation from images based on deep learning: A review," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 462–465.

[2] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," *CoRR*, vol. abs/1602.00134, 2016. [Online]. Available: http://arxiv.org/abs/1602.00134

[3] S. Park, J. Hwang, and N. Kwak, "3d human pose estimation using convolutional neural networks with 2d pose information," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 156–169.

[4] Q. Chen, C. Zhang, W. Liu, and D. Wang, "Shpd: Surveillance human pose dataset and performance evaluation for coarse-grained pose estimation," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 4088–4092.

[5] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *CoRR*, vol. abs/1312.4659, 2013. [Online]. Available: http://arxiv.org/abs/1312.4659

[6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[7] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, "Learning to refine human pose estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 318–31 809.

[8] Y. Luo, J. S. J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin, "LSTM pose machines," *CoRR*, vol. abs/1712.06316, 2017. [Online]. Available: http://arxiv.org/abs/1712.06316

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.