Multi-task learning for human pose estimation and part segmentation with hard parameter sharing

Abhinav Agrawal (14011) Ayushya Agarwal (14168) Shubh Gupta (14670)

1 Introduction

Human pose estimation relates to recognizing the configuration of the different body parts in a picture or video. It is a problem that has been studied for more than two decades in the computer vision community. Recent advances like PAF [2] and CPM [4] have been able to solve the task in near real time. These methods build upon the existing graph based modeling and have integrated deep learning(read convolutional neural networks) to achieve state of the art results while being completely supervised. There have been other approaches[3] which have combined the task of human pose estimation with part segmentation.

Motivated by the work in [3] we aim to develop a multi task pipeline which learns the segmentation and pose estimation tasks end to end. This builds upon the ideas that human part segmentation and pose point recognition are complementary task and can learn from each other. Also the difficulty in acquiring segmentation annotations leads to paucity of labeled data for human parts. On the other hand, the pose point annotations are easy and hence enjoy large datasets. Thus an empirical experiment that joins both these tasks into one single network was warranted. We make an attempt at this task, which to our knowledge is the first joint pose and segmentation multi-task network.

We have been able to implement the PAF and the FCN pipelines separately and calculated results on the jointly annotated subset of the PASCAL VOC part dataset. Further we have implemented the joint PAF and FCN pipelines for multi-task learning of segmentations and pose points. The results of the individual pipelines, with respect to change in different parameters, have been compared with the results obtained by the joint pipeline.

2 Implementation

2.1 Dataset

The two task of part segmentation although seem very complementary, the exploitation of this idea has been constrained due to lack of a common dataset that has annotations for both the task. [3] came up with their own annotation set to train their two different networks for the tasks. We in our formulation aim to liberate joint learning from this constraint by using a multi-task objective. This essentially gives us the ability to use datasets such as MPII, which has $\tilde{2}$5k of images annotated with human pose points and combine it with PASCAL VOC part datasets which only has 3k human body part annotated images.

We currently use the dataset provided by [3] for our preliminary tests. We further processed the dataset to create the two sets of ground truths for the pose estimation pipeline(expanded in section 2.3).

2.2 Segmentation Pipeline

This has been implemented as stated in [1], along with the adaptation of the number of channels to match the number of limbs and background as annotated in the dataset used - The subset of PASCAL VOC part annotations with human annotations. The network uses the pretrained weights of the VGG16 network[]. The implementation of FCN according to the paper uses skip connections to get finer segmentations - the outputs of the inner layers are added to the outputs of the outer

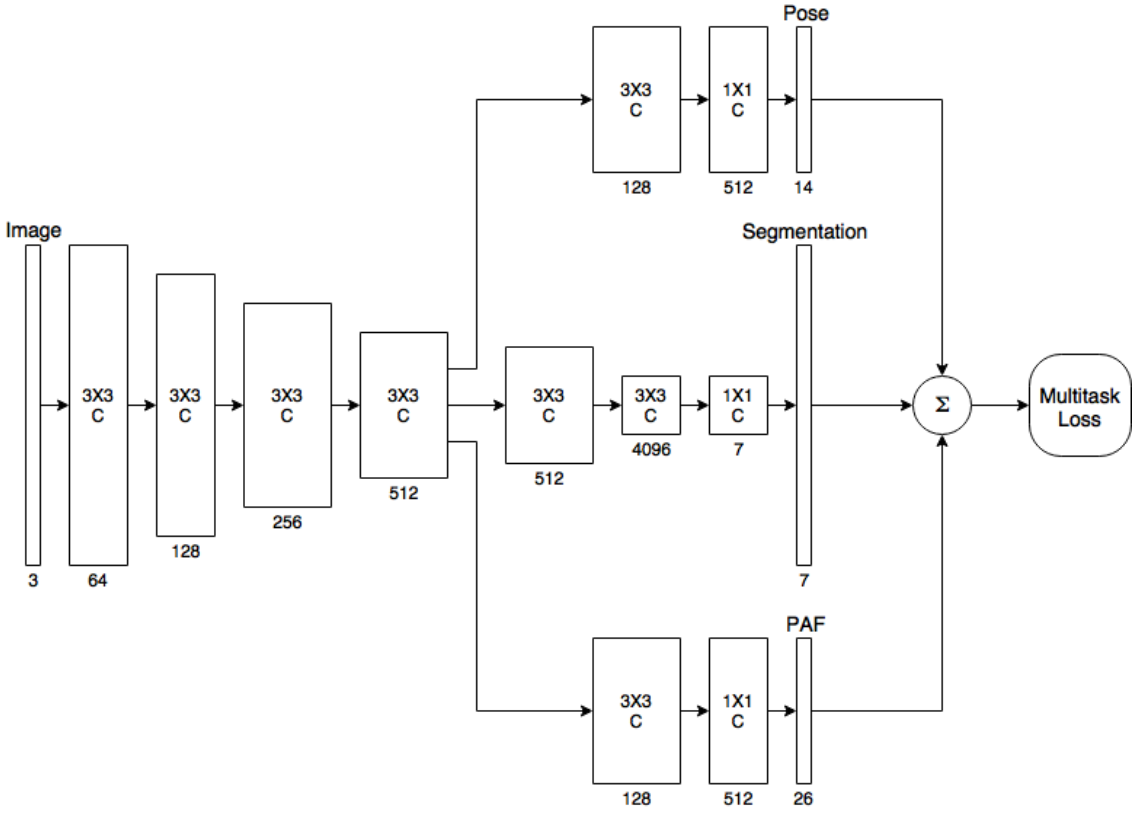


Figure 1: Model architecture

layers (after up-sampling). This is found to increase the information available as both coarse and fine level structural information gets available and hence results in better performance[1].

Without the skip connections, the basic pipeline of (popularly known as FCN32) fully convolutional layers outputs features of size $\frac{H}{32} \times \frac{W}{32}$ which are then resized to image dimensions $(H \times W)$. In our implementation we work only with this basic pipeline of FCN32. This is done to easily merge into a joint PAF-FCN pipeline and a quick comparison of results. The standard pixel wise Cross Entropy Loss is used for segmentation training with a Stochastic Gradient Optimizer. We also played with Adam, but it was found empirically that SGD outperforms it. The Cross Entropy Loss Function is defined as:

$$Loss_{seg} = -\sum_{i=0}^{i=H*W-1} \sum_{c=1}^{C} t_{ic} log(p_{ic}) \tag{1}$$

where, the $t_i c = 1$ when i^{th} pixel belongs to the c^{th} class, p_{ic} is the probability of i^{th} pixel belonging to the c^{th} class. The following metrics are calculated for performance analysis:

- pixel accuracy : $\sum_i n_{ii} / \sum_i t_i$
- mean accuracy : $(1/n_{cl}) \sum_i n_{ii}/t_i$
- mean IU : $(1/n_{cl}) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$
- frequency weighted IU : $(\sum_k t_k)^{-1} \sum_i n_{ii} t_i/(t_i + \sum_j n_{ji} - n_{ii})$

where n_{ij} is the number of pixels of class i predicted to belong to class j, n_{cl} is the total number of classes and $t_i = \sum_i n_{ii}$ .The results can be seen in Table 1

2.3 Pose Estimation Pipeline

The Pose Estimation pipeline has been implemented through a PAF[2] like architecture. We do some modifications to their architectures so as to be able to combine this with segmentation and

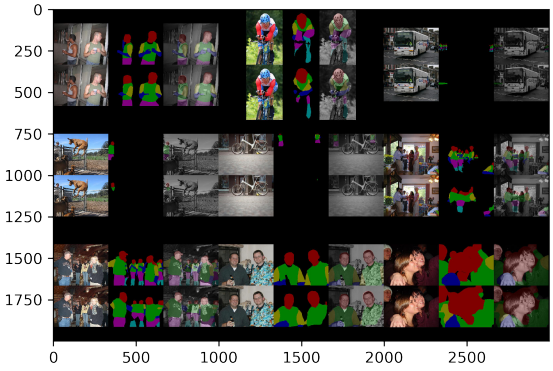


Figure 2: Segmentation using FCN32

run a proof of concept experiment. The PAF architecture learns pose points and their associations with another through part affinity fields. It predicts the 2D heat maps of the body part locations($S = [S_1, S_2, ...S_J]$),one per pose point,and a set of 2D vector fields ($L = [L_1, L_2, ...L_C]$),one per limb(a limb is defined by two human points and not necessarily the usual four limbs of human beings), which encode the degree of association between pose points. The image is first pushed through the first ten layers of VGG16 generating a layer of feature maps which is then used as input to the two branches of the first stage - one predicting the heat maps and another the vector fields. The architecture according to the paper, has multiple stages of this. After each stage, the predicted heat maps, vector fields and the original features are concatenated and pushed through to the initial layers of the 2 branches of the next stage.The paper implements a network with 6 such stages. We currently implement a stand alone PAF pipeline with 1 stage model and 2 stage model. The loss is calculated in both the pipelines at the end of each stage of the network. The loss used is the masked MSE loss, In the case of heat maps the expression is as follows:

$$Loss_{pose} = \sum_{j=1}^{J} \sum_{p \epsilon I} W(p) \left\| S_j(p)^{pred} - S_j(p)^{gt} \right\|^2 \tag{2}$$

The $S_j(p)^{pred}$ are the predicted heat maps and $S_j(p)^{gt}$ are the ground truth heat maps for the j^{th} body part. $W(p) = 1$ if p has ground truth heat map value above a threshold of 0.02, otherwise $W(p) = 0$ In the case of 2D vector fields, the loss is defined as

$$Loss_{paf} = \sum_{c=1}^{C} \sum_{p \epsilon I} W(p) \left\| L_c(p)^{pred} - L_c(p)^{gt} \right\|^2 \tag{3}$$

The $L_c(p)^{pred}$ are the predicted 2D vector fields and $L_c(p)^{gt}$ are the ground truth vector fields for the c^{th} limb. $W(p)$ is defined as above.
The metric used for comparing performances has its basis in the PCKh metric. The metric calculates the average distance between the predicted pose points and the corresponding ground truth pose points. A particular ground truth-prediction pose point pair is included in the calculation for the metric if the distance between the two is within a threshold. The threshold we have chosen is 5 times the length of the shortest limb. The results can be seen in Table 1

(a) PAF predictions and ground truth

(b) Pose predictions and ground truth

Figure 3: Results of 1 stage deep PAF pipeline

2.4 Joint Pose and Segmentation pipeline

2.4.1 Single-Stage Pipeline

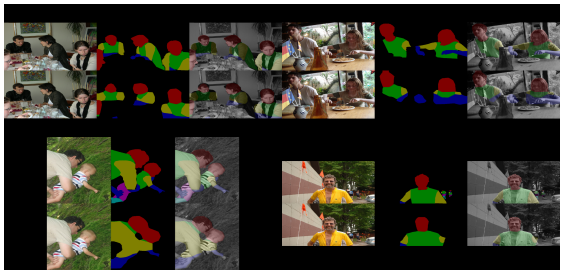
The joint PAF and FCN pipeline aims at combining the individual pipelines of the complementary tasks to form an end to end network that learns the 3 tasks together. The individual pipelines are split from the main pipeline after the initial 10 layers of VGG16. The loss generated in each pipeline is back-propagated through the pipeline to ultimately form a weighted sum at the splitting layer. This combined loss is then back-propagated through the common pipeline.

$$Loss_{total} = Loss_{seg} + Loss_{pose} + Loss_{paf} \tag{4}$$

In our limited results we find that SGD performs best with this setting. This experiment serves as a proof of concept for our original formulation and future work.

(a) PAF predictions and ground truth

(b) Pose predictions and ground truth



(c) Segmentation predictions and ground truth

Figure 4: Results of 1 stage deep combined pipeline

2.4.2 Multi-Stage Pipeline

The PAF pipeline individually has been implemented with 6 stages. We have defined a network with 1 stage that has all 3 pipelines corresponding to the multiple tasks. Based on these results1, we aim to expand this pipeline in which the three heads of the multi-task objective, take the learned saliency maps from the the previous stage. This has been implemented to one more stage but the results could not be collected in the limited time. We wish to proceed in this direction as a future work.

Table 1: Results on Different Metrics

	PCK_dist	Pixel Acc.	Mean Acc.	Mean IU	Frequency IU
Joint Pipeline	38.01	89.93	60.12	48.41	82.77
FCN32		89.90	61.82	50.05	83.52
PAF (Stage 1)	76.34				

3 Future Work

We aimed to conduct an empirical experiment to verify whether the pose estimation and human body part segmentation were compatible or not. The results of the limited experiments conducted by us seem to suggest that with some engineering the proposed networks would be promising. Also the pipeline with one more stage could be extended for segmentation too. This shall again be an interesting experiment to conduct where the whole network is fully convolutional and takes

supervision at each stage for the three complementary tasks. Also we feel that once the MPII data set is also used for pose estimation we may further see improvements for both the tasks.

This pipeline further be extended to a self supervised setting where we take images from wild and refine predictions form the network. Pose point predictions can be filtered using the Cyclic consistency methods and then segmentation can be refined using the CRF procedures. These can then be used for further training with a hope of leveraging the vast unlabelled data.

References

[1] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[2] Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." arXiv preprint arXiv:1611.08050 (2016).

[3] Xia, Fangting, et al. "Joint Multi-Person Pose Estimation and Semantic Part Segmentation." arXiv preprint arXiv:1708.03383 (2017).

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

[5] D. Comanicu, P. Meer: "Mean shift: A robust approach toward feature space analysis". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, May 2002. Jean, Frédéric, "Python Module for Mean Shift Image Segmentation" , https://github.com/fjean/pymeanshift

[6] Zhou, Tinghui, et al. "Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences.",(2017), GitHub repository, https://github.com/tinghuiz/flowweb

[7] Pathak, Deepak, et al. "Learning features by watching objects move." arXiv preprint arXiv:1612.06370 (2016).

[8] Zhou, Tinghui, et al. "Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[9] Chakraborty, Prabuddha, and Vinay P. Namboodiri. "Learning to Estimate Pose by Watching Videos." arXiv preprint arXiv:1704.04081 (2017).

[10] Poirson, Patrick, et al. "Fast single shot detection and pose estimation." 3D Vision (3DV), 2016 Fourth International Conference on. IEEE, 2016.