Efficient Spatial-Attention module for human pose estimation

Tien-Dat Tran, Xuan-Thuy Vo, Duy-Linh Nguyen, and Kang-Hyun Jo*

School of Electrical Engineering, University of Ulsan, Ulsan 44610, South Korea
{tdat,xthuy,}@islab.ulsan.ac.kr;ndlinh301@mail.ulsan.ac.kr;
acejo@ulsan.ac.kr

Abstract. Not only for human pose estimation but also other machine vision tasks (e.g. object recognition, semantic segmentation, image classification), convolution neural networks (CNNs) have obtained the highest performance today. Besides, their performance over other traditional networks is shown by the Attention Module (AM). Hence, this paper focuses on a valuable feed-forward AM for CNNs. First, feed the feature map into the attention module after a stage in the backbone network, divided into two different dimensions, channel and spatial. After that, by multiplication, the AM combines these two feature maps and gives them to the next stage in the backbone. In long-range dependencies (channel) and spatial data, the network can capture more information, which can gain better precision efficiency. Our experimental findings would also demonstrate the disparity between the use of the attention module and current methods. As a result, with the change to make the spatial better, the expected joint heatmap retains the accuracy while decreasing the number of parameters. In comparison, the proposed architecture benefits more than the baseline by 1.3 points in AP. In addition, the proposed network was trained on the benchmarks of COCO 2017, which is now an open dataset.

Keywords: Deep Learning · Attention module · Spatial-Attention module · Human pose estimation.

1 Introduction

In today's modern world, 2D human pose estimation performs a critical but difficult role in computer vision, which can support multiple purposes such as human pose estimation [24, 2], activity recognition [7, 11], human re-identification [27, 13] or 3D human pose estimation [1]. The key aim of human pose is to identify body parts for human body joints. Spatial and channel data play an important role in making the regression of key points more precise. As a result, this paper will focus on how to make the network learn more about the attention information.

* Corresponding author

Important developments in human pose have now been archived by deep convolution of neural networks [17, 8]. However, these networks still have a lot of issues to discuss. First of all, how to boost accuracy in different forms of networks (e.g., real-time network, accuracy network). Second, there is often a need to consider the speed of the network while changing or modifying it. Last but not least, the current network has to achieve better accuracy while maintaining speed as fast as possible. This paper describes a novel network and the reliability of the attention module for speed and accuracy. The proposed experiment shows a comparison between the focus module used and not used. The experiment also contrasts with the Simple Baseline [26] which did not use the attention mechanism and used the transpose convolution [3] for upsampling. Our experiment would concentrate on how efficient and cost-effective each case for the network.

In paricular, our approach was implement based on simple fine-tune attention module [22] which shows significant improve in mean Average Precision (mAP). Inspired by VGG16 [19], the proposed network try to improve the spatial attention module (SAM) by using two 3×3 convolution layer instead of 7×7 convolution layer. By used 3×3 kernel, the network still maintain the mAP while decrease the implementation cost. In addition, the number of parameter decrease so the speed of our network was upgraded. To make clear about modify SAM, our network increase 0.2 point in AP for accuracy and reduces around 1.6 percent of parameters compared with the Attention mechanism baseline[22] when used ResNet-50 [5] as a backbone network. This paper introduces a new SAM module for the network, which can easily respond to a range of problems in many applications, such as object recognition, image classification and human pose estimation. The suggested approach calculates joint human pose estimates based on the recovery of feature maps using the up-sampling network.

2 Related work

Human Pose Estimation The leading part of human pose estimation lies in joint detection and their relationship with spatial space, which illustrates in Fig.1. Deeppose [21], Simple baseline utilizes joints prediction thought an end to end network with higher parameter. Later, Newell with Stacked hourglass network [18] decreases the number of the setting while still keep high accuracy. All of the methods used Gaussian distribution to represent local joints. Then used a convolution neural network to estimate human pose estimation. To decrease the employment cost, they need to reduce the number of parameters, and applying suitable up-sampling methods will lower the network's parameter. So, the proposed method used interpolation as up-sampling module.

On the other hand, to enhance the speed of the network, interpolation shows a lot of benefits than the transpose convolution. However, in some complex and higher cost architecture, the transpose convolution gives better accuracy. In comparison, our up-sampling module provides an adequate view for designing the network, with a small number in parameter and high speed or higher parameter and lower speed. Then this paper shows how up-sampling will work in each

method and each result.
Attention mechanism: Human visualization plays an important role in computer vision, and there are a variety of focus processing attempts to enhance the efficiency of CNNs. Wang et al. [23] also suggested a non-local network to collect long-range dependencies. Inspired by SENet [6] and Inception [20], then SKNet [14] merged the SENet Channel Focus Module with the Inception Multi-Branch Convolution. In addition, the Module for Spatial Focus comes from the STN [10] suggested by Google, which aggregates the background details of the feature maps. In addition, the attention module shows a lot of advantages for the detection of saliency, the multi-label classification for the recognition of the individual.

In this paper, the proposed method was inspired by CBAM network [25] to make the effective between both channel and spatial module by using element-wise multiplication. After that, the feature map takes an addition to the last feature map to combine the original information and new information from the AT module.

3 Methodology

3.1 Network architecture

Backbone network There are ResNet-101 and ResNet-50[5] in the backbone network, as can be seen in Figure 1 for complete architecture. Every ResNet has four blocks, including convolution layers and shortcut connections. The input RGB picture reduces the size to 256×192 (ResNet-50, ResNet-101), the feature maps pass across each column block, and the resolution of $W \times H$ decreases twice for each block. Finally, after passing along the spine, the size of the function map is reduced to $\frac{W}{16} \times \frac{H}{16}$ with 2048 channels at the end of the spine. In addition, the size of the channels also would be doubled for each block. It's coming from 256 after the first block to 2048 in the last layer. The mission of the backbone network is to collect information and feature maps from the input image and feed it to the Up-Sampling Training System.

After extracting the information by utilized the backbone network, the up-sampling network takes the feature map from the last layer of the backbone network and upsampling to recover the information. Next, the feature map will then practice with the Ground-truth Heat Maps, as is seen in Fig.1. The default heat map size is 64×48 for 256×192 photos and 96×72 for 384×288. This heat maps need to understand the scale of the image in order to match the size of the feature maps in the training process. The network will use these heat maps and the ground truth heatmap for regression to calculate the predicting main point. For the up-sampling network, this paper uses the up-sampling module, which contains one bilinear [16] layer and one convolution layer (in Figure 2). And there's an alternative to this two-layer. Batch normalization and ReLU[9] are both within the up-sampling block.

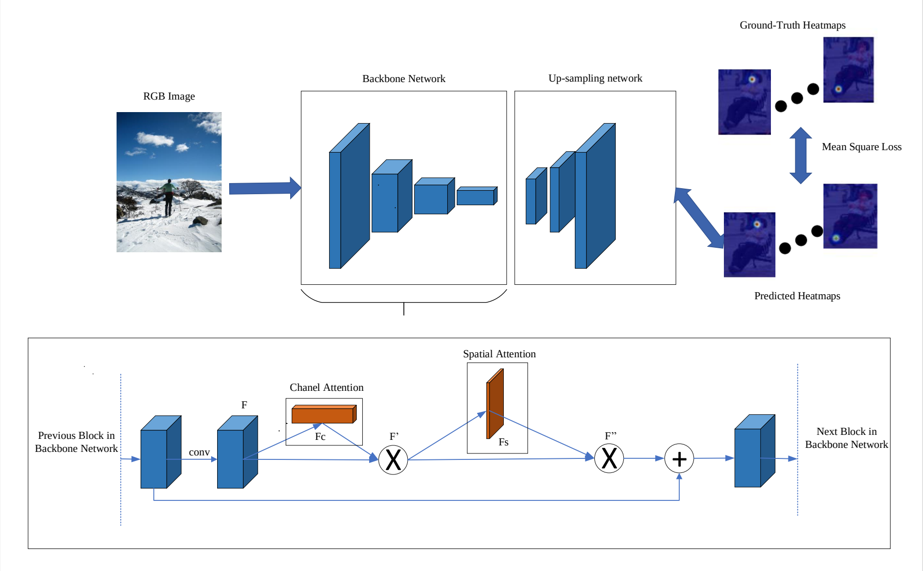

Fig. 1. Illustrating the design of the proposed human-pose estimation network. The suggested approach split the system into two sub-networks, Backbone, and Up-sampling. Backbone extracts a feature map while Up-sampling retrieves a feature map for regression. In comparison, this figure indicates the description of the attention module, which included the channel and the spatial module at the bottom of the list.

Attention Module The Attention Mechanism contains two major modules seen in Fig.2. First, after block one in the backbone network, the feature map was fed to the channel attention module (CAM). In CAM, the feature map takes global average pooling to squeeze the feature map from $H \times W \times C$ to $1 \times 1 \times C$. First, it goes into the convolution layer, which transforms the feature map to $1 \times 1 \times \frac{C}{r}$, which r is the reduction ratio, and r is set to 16. The CAM then used the ReLU to trigger the weight. The final step in CAM is to use 1x1 convolution layer again to restore the channel to $1 \times 1 \times C$ and use the sigmoid to normalize the feature map. After that, the element-wise multiplication was used to merge the details for CAM.

When the feature map goes through the CAM, it will be fed into the Spatial Attention Module (SAM). In SAM, the feature map takes the average channel pooling from $H \times W \times C$ to $H \times W \times 1$. After pooling, two 3×3 convolution layers were used to extract the spatial information attribute diagram, and the final stage in SAM is identical to the CAM that can be seen in Fig.2. Finally, the planned approach used the element-wise extension to the initial feature map and the feature map after AT to be merged and a new feature map for the next block in the backbone network.

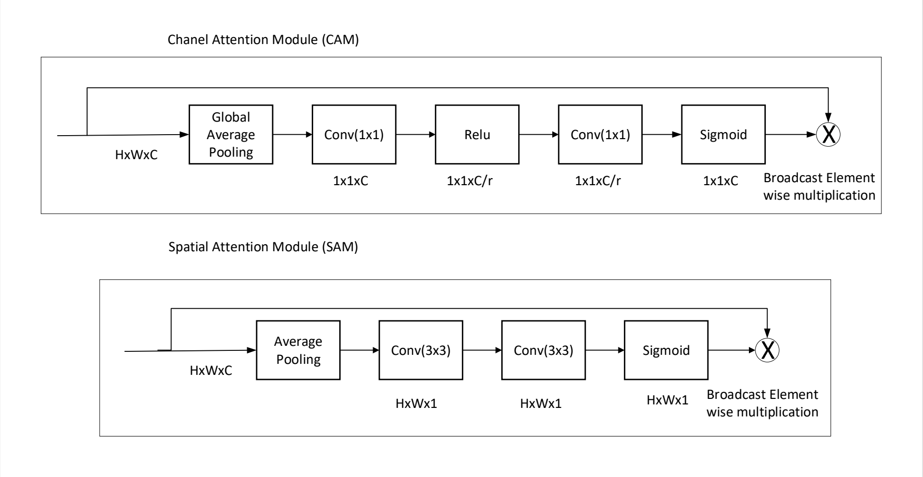

Fig. 2. Architecture of Channel Attention Module (CAM) and Spatial attention Module (SAM)

3.2 Loss Function

This paper uses heat maps to represent body joint positions for the loss function. As the position of the ground-truth in Fig. 1 by $a = \{ak\}\, k = 1^K$, where $xk = (xk, yk)$ is the spatial coordinate of the kth body joint in the picture. Then the ground-truth heat map value Hk is generated from the Gaussian distribution with the mean ak and the variance $\sum$ as follows.

$$H_k(p) \sim N(a_k, \sum) \tag{1}$$

where $\mathbf{p} \in R^2$ denotes the coordinate, and $\sum$ is empirically set as an identity matrix $\mathbf{I}$. Final layer of neural network predicts K heat maps, *i.e.*, $\hat{S} = \left\{\hat{S}_k\right\}_{k=1}^{K}$ for K body joints. A loss function is defined by the mean square error, measured as::

$$L = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \left\| S_k - \hat{S}_k \right\|^2 \tag{2}$$

Where N is the number of samples in the training session. The network developed predictive heat maps from ground-truth heat maps using information from the last layer of the backbone network.

4 Experiments

4.1 Experiment Setup

Dataset. During the tests, the suggested approach used the Microsoft COCO 2017 dataset [15]. This dataset contains about 200K images and 250K human samples, which have 17 keypoint labels for one person. The data collection of the study included three folder train set, validation set, test-dev set, respectively, with training, validation and testing images. In addition, the validation and training annotations are public and accompanied by the original. **Evaluation metrics.** This paper used Object Keypoint Similarity (OKS) for COCO[15] with $OKS = \frac{\sum_i exp(-{d_i}^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$. Here d_i is the Euclidean distance between the predicted keypoint and the groundtruth while v_i is the visibility flag of the target, s is the object scale and k_i is a keypoint for each joints. Then calculate the standart average precision and recall score. In table I, AP and AR is the average from OKS=0.5 to OKS=0.95, while AP^M for medium object and AP^L for large object.

Implementation details In model training, the proposed approach used data raise, such as flip, rotation at 40 degrees by design, and scale, which set the factor at 0.3. Set the batch size to 4 and use the shuffle for training photos. The total of the epoch is 270, while the based learning-rate at 0.001 and multiple by 0.1 (learning rate factor) at the 170-th and 200-th epoch in our experiment. The momentum is 0.9, and the Adam optimizer[12] was used.

All experiments are implemented with Pytorch framework and testing in two datasets. The input resolution of images resized to 256x192. The model was trained on one NVIDIA GTX 1080Ti GPUs with CUDA 10.2 and CuDNN 7.3.

4.2 Experiment Result

Table 1. The result of using the different kernel of convolution layer in the SAM module. 3×3 + 3×3 means using continuously two 3×3 kernel and 3×3 // 3×3 means using two parallel 3×3 kernel

Backbone	Convolution layer	#Param	mAP
ResNet-50	7×7	31.2M	71.4
ResNet-50	3×3 + 3×3	30.7M	71.6
ResNet-50	3×3 // 3×3	30.7M	71.5
ResNet-101	7×7	51.3M	72.3

To show clearly about performance of SAM in AT, The proposed method compares each situation when used different convolution kernel, which show in Table 1. The Average Precision (AP) shows that used two 3×3 convolution kernel gain 0.2 in mAP than used 7×7 while the number of parameter decrease 1.6 percents in ResNet50.

COCO datasets result The AP in the suggested approach is greater than the Basic benchmark in all situations of 1.3 AP, 1.1 AP in ResNet-50, ResNet-101, respectively. The number of parameters also reduces relative to the standard due to the variation in the up-sampling network. Although the benchmark used transposes convolution costs a number of parameters, the proposed approach

Table 2. Comparison on COCO TEST-DEV Dataset. AM is mean attention module, new AM is new attention module proposed in this paper

Method	Backbone	Input size	#Params	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-Stage Hourglass[18]	8-Stage Hourglass	256×192	25.1M	66.9	-	-	-	-	-
Mask-RCNN[4]	ResNet-50-FPN	256×192	-	63.1	87.3	68.7	57.8	71.4	-
SimpleBaseline[26]	ResNet-50	256×192	34.0M	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline[26]	ResNet-101	256×192	53.0M	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline[26]	ResNet-152	256×192	68.6M	73.7	91.9	81.1	70.3	80.0	79.0
Fine-tuning AM[22]	ResNet-50	256×192	31.2M	71.4	91.6	78.6	68.2	75.7	76.3
Fine-tuning AM[22]	ResNet-101	256×192	50.2M	72.3	92.0	79.4	68.3	77.1	77.1
Our + new AM	ResNet-50	256×192	30.7M	71.7	91.8	80.3	69.0	78.2	76.9
Our + new AM	ResNet101	256×192	49.7M	72.5	92.2	80.9	69.9	79.5	78.1

uses bi-linear interpolation at no cost for settings. Our solution adds a new module (AT) but changes the up-sampling module such that the number of parameters are different. In this experiment the number of parameter was smaller 9.7 percent and 6.2 percent in case or ResNet-50, ResNet-101, respectively. Besides, the average recall (AR) gain better result in case of ResNet-101 with 1.0 points higher. In all cases of Fine-tuning AM, our network gain 0.3 in mAP for both ResNet-50 and Resnet-101.

However, as with many architectures today, human pose estimation also has many problems that need to be tackled. The first problem being that the pictures included unseen joints that were impossible to train and predict. Second, low-resolution photographs of humans need to be properly extracted for human body joints. Next, there are photos of crowd scenes, which are often hard to establish all the locations of the joints for all participants. Finally, there is a lack of details on photographs containing partial sections for estimating human poses.

5 Conclusion

This paper demonstrates the influence of the attention module on CNNs, and reveals that the attention module used has a stronger effect by not changing the number of parameters. In comparison, the Attention Module highlighted the essential function maps instead of the other component. The network will therefore boost efficiency, particularly for several tasks in the field of computer vision. Future analysis is to define certain applications or environments to be added to our study, such as the surveillance system. Another task is due to the difficulties of human exposure assessment, which limits the precision of the network.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government. (MSIT)(2020R1A2C2008972)

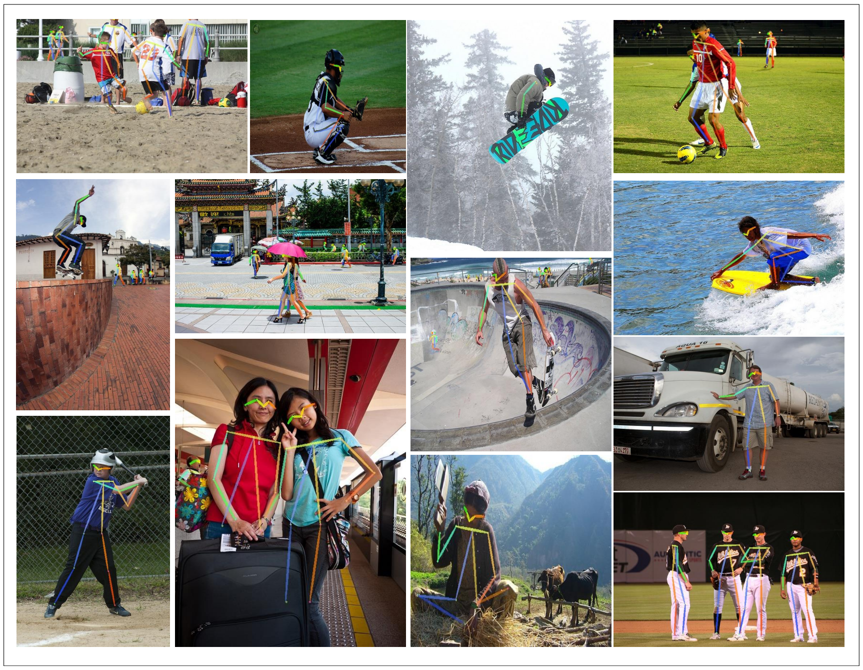

Fig. 3. Qualitative result for human pose estimation in COCO2017 test-dev set

References

1. Chen, C., Ramanan, D.: 3d human pose estimation = 2d pose estimation + matching. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5759–5767 (July 2017). https://doi.org/10.1109/CVPR.2017.610
2. Chou, C.J., Chien, J.T., Chen, H.T.: Self adversarial training for human pose estimation (2017)
3. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning (2016)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn (2017)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
6. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks (2017)
7. Hussain, Z., Sheng, M., Zhang, W.E.: Different approaches for human activity recognition: A survey (2019)
8. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model (2016)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
10. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks (2015)
11. Kim, E., Helal, S., Cook, D.: Human activity recognition and pattern discovery. IEEE Pervasive Computing **9**(1), 48–53 (Jan 2010). https://doi.org/10.1109/MPRV.2010.7

12. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
13. Li, W., Zhao, R., Wang, X.: Human reidentification with transferred metric learning. In: Asian Conference on Computer Vision (ACCV). pp. 31–44 (11 2012)
14. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks (2019)
15. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014), http://arxiv.org/abs/1405.0312
16. Mastyło, M.: Bilinear interpolation theorems and applications. Journal of Functional Analysis **265**, 185–207 (07 2013). https://doi.org/10.1016/j.jfa.2013.05.001
17. Moon, G., Chang, J.Y., Lee, K.M.: Posefix: Model-agnostic general human pose refinement network (2018)
18. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR **abs/1603.06937** (2016), http://arxiv.org/abs/1603.06937
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
20. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning (2016)
21. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. CoRR **abs/1312.4659** (2013), http://arxiv.org/abs/1312.4659
22. Tran, T.D., Vo, X.T., Russo, M.A., Jo, K.H.: Simple fine-tuning attention modules for human pose estimation. In: International Conference on Computational Collective Intelligence. pp. 175–185. Springer (2020)
23. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. CoRR **abs/1711.07971** (2017), http://arxiv.org/abs/1711.07971
24. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines (2016)
25. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module (2018)
26. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. CoRR **abs/1804.06208** (2018), http://arxiv.org/abs/1804.06208
27. Yang, X., Wang, M., Tao, D.: Person re-identification with metric learning using privileged information. CoRR **abs/1904.05005** (2019), http://arxiv.org/abs/1904.05005