Tilting at windmills

Data augmentation for deep pose estimation does not help with occlusions

Rafal Pytel

Tilting at windmills

Data augmentation for deep pose estimation does not help with occlusions

by

Rafal Pytel

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 31, 2020 at 12:30 AM.

Student number:	4933478	
Project duration:	November 1, 2019 – August 31, 2020	
Thesis committee:	Prof.Dr.Ir. M.J.T. Reinders,	TU Delft, Chair of the thesis committee
	dr. J. C. Gemert,	TU Delft, supervisor
	Dr. ir. C.C.S. Liem	TU Delft, External thesis committee member

TU Delft

Preface

The current report "Tilting at Windmills: Data augmentation for deep pose estimation does not help with occlusions" presents work done for my master's graduation project. The research was conducted within Computer Vision Lab at TU Delft, under the supervision of Dr. J.C. van Gemert and Osman Semih Kayhan as my daily supervisor.

As I was introduced during studies to a variety of challenges in Computer Vision and also have the passion for the different sport I decided to follow the direction which could join both of them - human pose estimation. I would like to thank Jan for very helpful support with a variety of research decisions, which resulted in an enjoyable 10 months of research. I would also like to thank for giving me access to TU Delft cluster, which I have used at its full capacity for the past few months, resulting in top places in "usage per User" table.

Also, I would like to thank Osman for the tremendous support in our weekly meetings, brainstorming solutions on a daily basis and successful motivation on each step. It was very helpful to have someone fully understanding all the ins and outs of your subject with whom I could always have a valuable discussion whenever I needed.

I would like to thank Prof.Dr.Ir. Reinders and Dr. ir. Liem, for their interest in my thesis and evaluation of my work.

I would like to also thank my parents for their emotional, moral and financial support in the last two years, often motivating me to follow my dreams and try to learn as much as I can instead of choosing easier less demanding paths. My sister, for both emotional support and teaching me the importance of not overworking myself when it is not needed and its benefit in the long run. I would also like to thank my friends to always motivating me for more concentrated work, exchanging valuable input on each other problems and celebrating together small successes along the way.

Rafal Pytel
Delft, August 2020

Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions

Rafal Pytel
Computer Vision Lab
Delft University of Technology

Osman Semih Kayhan
Computer Vision Lab
Delft University of Technology

Jan C. van Gemert
Computer Vision Lab
Delft University of Technology

***Abstract*—Occlusion degrades the performance of human pose estimation. In this paper, we introduce targeted keypoint and body part occlusion attacks. The effects of the attacks are systematically analyzed on the best performing methods. In addition, we propose occlusion specific data augmentation techniques against keypoint and part attacks. Our extensive experiments show that human pose estimation methods are not robust to occlusion and data augmentation does not solve the occlusion problems.** [1]

I. Introduction

Human Pose Estimation is the task of localizing anatomical keypoints such as eyes, hips, knees and localizing body-parts like head, limbs, corpus. It has many applications on segmentation [24], [25], [52], action recognition [28], [30], [40], pose tracking [14], [54], gait recognition [39], [44], autonomous driving [12], [32], [50], elderly monitoring [10], [31] and social behaviour analysis [22], [48]. All these applications rely on correct and robust pose estimation. In this paper we investigate the robustness of human pose estimation methods to a natural and common effect: Occlusions.

Occlusions are common and occur frequently in the wild as for example by a random object, another person [15], and self-occlusion [18]. Prior works address the occlusion problem in a general way and exploits segmentation [32] or depth information [33]. [36] checks the robustness of the estimators with image-agnostic and domain-agnostic universal perturbations. In contrast, we systematically analyze targeted occlusion attacks not only for keypoints, but also for and body parts and investigate the sensitivity of pose estimation to the proposed occlusion attacks.

Data augmentation is now such a common practice that it has become a default setting for deep learning applications [37] to improve subtle difference between training and testing data and leads to better generalization. Flipping, rotation, scaling are often employed the in computer vision tasks [6], [37], [45]. Recent works show that the usage of regional dropout and mixup methods improve the generalization performance of image classification [9], [16], [41], [46], [49], [55], [59], [60], object localization and detection [7], [11], [38] and segmentation [13]. In pose estimation, [19] applies region based augmentation by exchanging a single keypoint patch with a random background patch. More recent approaches [42], [53] use half-body augmentation wherewith the presence of more than 8 keypoints, by choosing upper or lower body keypoints. We implement systematic data augmentation methods for occlusion for keypoint and body parts to investigate how data augmentation can remedy occlusion attacks.

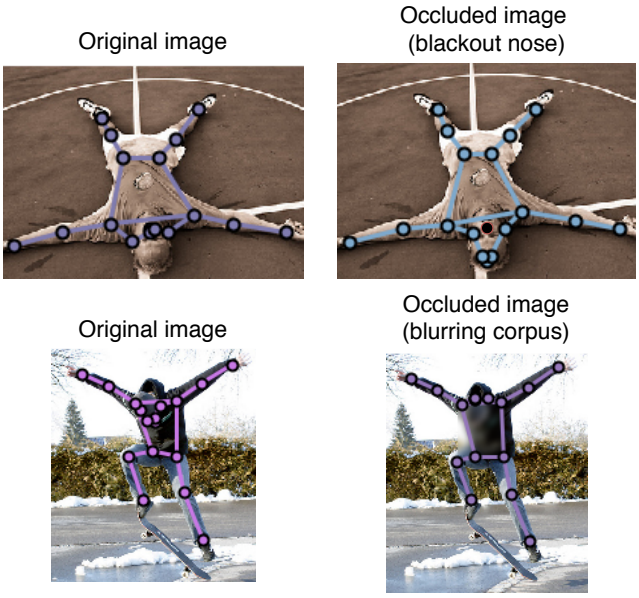


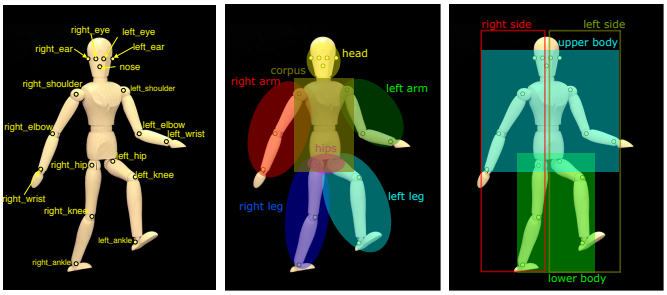
Fig. 1: Qualitative example of changes for HRNet-32 in position prediction of joints after performing keypoint blackout on nose (first row) and part blurring on the corpus (second row). With both of the examples we observe change in head keypoints, nose, eyes and ears.

We have the following contributions: First, we conduct a structured investigation on the occlusion problem of pose estimation and introduce occlusion attacks. Second, we investigate occlusion-based data augmentation methods. Third, we show that data augmentation does not provide robustness to occlusion attacks.

II. Related work

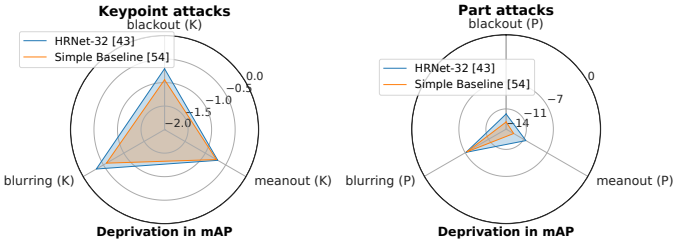
Human Pose Estimation. Deep learning methods in human pose estimation can be divided into 2 categories: bottom-up and top-down. Bottom-up approaches [3], [6], [21], firstly localize identity-free keypoints and then group them into person instances. Top-down approaches [5], [29], [42], [53] firstly detect a person in the image and then perform a single person

[1] submitted to ICPR 2021 conference.



(a) COCO keypoints annotations. (b) Part mapping for smaller parts. (c) Part mapping for larger parts.

Fig. 2: Visualization of keypoint annotations in COCO dataset and proposed part mapping.



(a) (b)

Fig. 3: Robustness comparison of HRNet and Simple Baseline methods against (a) keypoint and (b) part occlusion attacks. HRNet is more robust against both attacks, yet both attacks drop performance, where part attacks deteriorate more.

estimation within the bounding box. The top-down approaches achieve the state of the art results on various multi-person benchmarks such as COCO [26], MPII [1]. Within top-down approaches 2 categories can be distinguished: regressing direct location of each keypoint [4], [47] and keypoint heatmaps estimation [8], [29], [42], [51], [53] followed by choosing the locations with the highest heat values as the keypoints. The best performing methods on COCO keypoint challenge use a cascade network [5], [23] to improve keypoint prediction. The 'Simple Baseline' [53] proposes simple but effective improvement by adding few deconvolutional layers to enlarge the resolution of output features. HRNet [42] which is built from multiple branches can produce high-resolution feature maps with rich semantics and performs well on COCO. Some works advance performance of HRNet via improvement over standard encoding and decoding of heatmaps [58] and basing data processing on the unit length instead of pixels [17] with an additional off-set strategy for encoding and decoding. Because of their good accuracy and wide adaptation, we focus on top-down methods: HRNet and Simple Baseline.

Occlusion in pose estimation. Occlusion in pose estimation is an under-studied problem. In [36] analyses of occlusions are done for deep pose estimators by domain-agnostic universal perturbations. More recently, attempts to solve the occlusion problem in pose estimation are suggested via the usage of segmentation of occluded parts [32] and depth of in an image [33]. OcclusionNet [34] predicts occluded keypoints via graph-neural networks yet it is applied only on vehicles. Different from these methods, in our paper we introduce keypoint occlusion attacks and body part occlusion attacks and give a structured analysis of occlusion on human pose estimation.

Data augmentation. Data augmentation is a strong, simple and popular approach to increase model robustness. Removing part of the image improves generalization of image classification [9], [55], [60] and object localization-detection [7], [11], [38]. Mixup [16], [46], [59] approaches which create a combination of two images are often used in image classification. [13][57] combine regional dropout and MixUp methods for image segmentation [13] and image classification [57] task. [19] proposes a cutmix-like approach where a small patch from the background is pasted on the single keypoint or vice versa. For the human pose estimation methods [4], [51], [56], scaling, rotation and flipping is commonly used as data augmentation. Random cropping is also used in bottom-up approaches [3], [6], [21]. More recent top-down approaches [5], [42], [53] employ the usage of half body transform by a probability of 0.3 choosing either upper or lower body keypoints. We introduce and evaluate new data augmentation methods for keypoint and for body parts specifically designed against occlusion attacks for human pose estimation.

III. Sensitivity to Occlusion Attacks

We investigate the effect of occlusion attacks on MS COCO dataset [26]. COCO contains challenging images with the unconstrained environment, different body scales, variety of human poses and occlusion patterns. The dataset contains over 200k images with 250k person instances labelled with 17 keypoints. Models are trained on COCO train2017 datasets which includes 57k images and 150k person instances. The evaluation is done on val2017 set which contains 5k images.

The occlusion attack experiments are conducted with HRNet [42] and Simple Baseline [53] for two aspects: (i) keypoint attacks, where the occlusion area is a centred circle on the chosen keypoint, (ii) body part attacks, where the occlusion area is the minimum rectangle covering all keypoints of a chosen part. The COCO keypoints and the proposed groups of body parts can be seen in Figure 2. For the analyses, COCO pretrained HRNet and Simple Baseline are evaluated by the performance of the network against keypoint and part occlusion attacks on COCO validation set.

HRNet and SimpleBaseline produce heatmap instead of predicting direct single location for each keypoint. The ground truth heatmaps are generated by using 2D Gaussian of size 13x13. Thus, as a default, we choose the size of the occlusion circle with a radius of 6 pixels for keypoint attacks to cover the keypoint heatmap. We have 3 different keypoint attacks: (i) Gaussian Blur (blurring) attack, (ii) attack by filling with black pixels (blackout), (iii) attack by filling with a mean intensity value of a given image (meanout).

Body parts occlusion attacks are designed to draw a minimum rectangle which covers all the keypoints of a chosen

part. Similar to the keypoint attacks, we have 3 different part attacks which are applied to the occlusion area: blurring with the kernel size 31 and sigma 5, blackout and meanout. These attacks can be applied on both small parts such as head, arms, hips and larger parts like upper body, lower body, left and right side (Figure 2 b and c).

We compare HRNet and Simple Baseline according to their robustness to keypoint and part occlusion attacks. Figure 3 shows that both attacks are quite successful as occlusion causes the performance to drop. HRNet is more robust against keypoint and part occlusion attacks. For further analyses, we only use HRNet as a baseline for our investigations.

A. How sensitive to key point occlusion attacks?

First, we analyze the effect of the occlusion size on the average performance of the pose estimator on all keypoints. Figure 4 indicates that pose estimator performance is inversely proportional to the occlusion size and blurring, blackout, and meanout attacks on average perform similarly. The size of the occlusion decreases the average performance of the estimator by approximately 3% when the radius of the occlusion circle is 18 pixels.

Second, we show the class-specific performance drops for each individual keypoints for each attack. In Figure 5, attacking nose causes serious loss in mAP, almost 5% for blackout, 4.4% for meanout and 1.2% for blurring. The empirical results indicate that **the nose** is the most important keypoint since the occlusion of the nose causes notable performance drop. After the nose, each eye influences the performance of other keypoints mostly by approximately 1% with each occlusion attack. Keypoints from less densely annotated places like ankles or wrists are the least influential.

If we check the analysis of the reduced accuracy per keypoint for the case of attacking nose (Figure 6a), the most affected keypoints are the ones within close distance, which are eyes and ears due to being a part of the head. Interestingly, occluding nose affects the performance of the left eye estimation more than occluding the left eye itself, respectively by approximately 10% and 5% (Figure 6a, 6b). If we investigate per keypoint performance for occluding left ankle, it can be seen that the deprivation is by several magnitudes smaller than in case of the nose or left eye occlusions. From the observation of the analyses, it can be drawn that HRNet [42] is not robust to keypoint occlusion attacks.

B. How sensitive to part occlusion attacks?

We analyze the effect of the part occlusion attacks on each body parts given in Figure 2. Attacking the upper body, left and right sides influence the overall performance the most, by more than 44%, 24% and 24% with blackout attack respectively since these three parts include the majority of the keypoints (Figure 7). When we examine keypoint-specific accuracy drops for the remaining keypoints of the upper body, it is clear that blackout is the most influential attack, with a drop of almost 3% for left and right ankle (Figure 8a). If we investigate per-keypoint behaviour for the corpus (Figure 8b), we observe

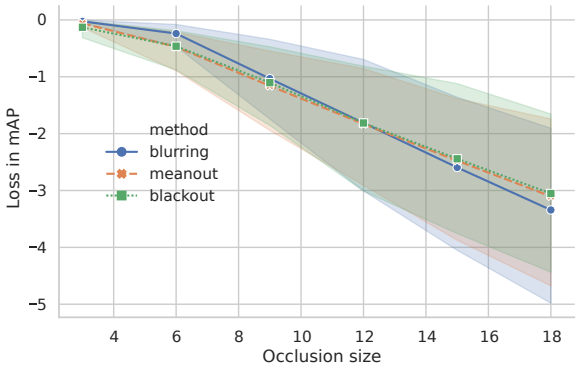


Fig. 4: The relation between occlusion size and average loss in performance for keypoint level methods. Occlusion size greatly affects the performance.

significant degradation of the performance on all the keypoints, with left and right ankle affected the most. Interestingly, attacking on one side improves performance of the the other side (Figure 8c). Attacking on left side increases the mAP score of right side such as shoulder, ear, elbow keypoints. The analysis demonstrates that the pose estimator is sensitive to part occlusion attacks.

IV. Occlusion augmentation against attacks

The experiments are performed on two of the main human pose estimation datasets: COCO and MPII. The MPII dataset consists of 40k person instances and each instance is labelled with 16 joints. The train set and validation sets includes 22k and 3k person instances respectively. For the evaluation, the validation set is used.

Training on COCO is done on 3 NVIDIA 1080TI GPUs for roughly 90 hours. The setup of training procedure was adopted from the baseline - HRNet[42]. Human detection boxes are extended to fit 4:3 aspect ratio, and then the boxes are cropped from the image, which is resized to a fixed size, $256x192$. The pose estimator is trained with the keypoint location of the joints. The data augmentations that are used in HRNet training include random rotation $\in [-45^o, 45^o]$, random scale $\in [0.65, 1.35]$, random flipping and half-body augmentations. Adam optimizer [20] is used to train the network with the learning rate schedule following [53], starting with $1e-3$ and reduced to $1e-4$ and $1e-5$ at 170th and 200th epochs respectively and the training is completed at the 210th epoch. For MPII dataset, the training procedure of baseline is as followed: 256x256 input size is used and half-body augmentations are discarded. For the evaluation of the models, Object Keypoint Similarity (OKS) for COCO and Percentage of Correct Keypoints (PCK) for MPII are used.

During testing, HRNet firstly employs an object detection algorithm to obtain boxes with a single person. Afterwards the pose estimator produces the keypoint location of the joints.

A. Occlusion augmentation

To mitigate the occlusion problems in human pose estimation, we investigate the following three methods: (i) Targeted

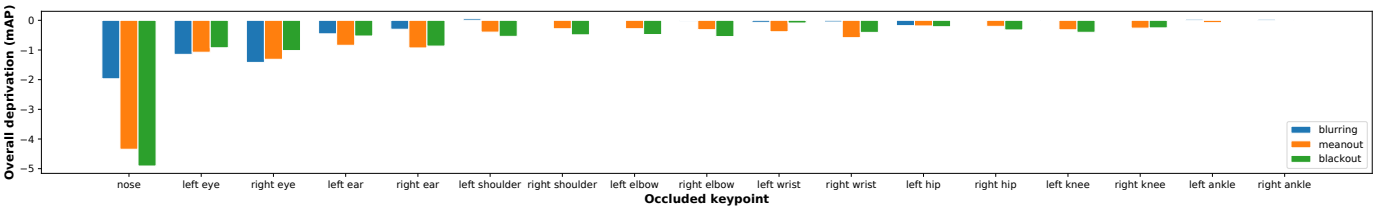
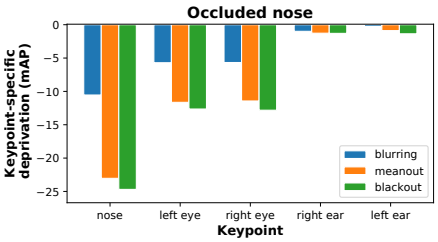
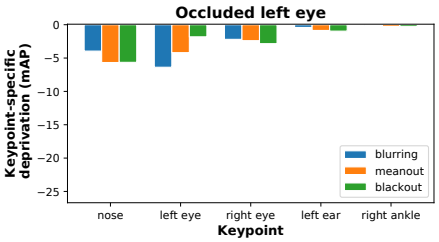


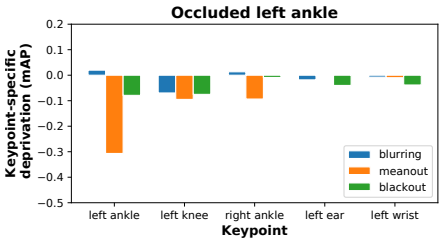
Fig. 5: Overall loss in mAP after performing keypoint level occlusion. To note that, the occluded keypoint is included in the evaluation. Occluding nose causes the highest loss in performance.



(a) The nose is the most influential keypoint causes a significant drop in the performance for the closest keypoints - left eye and right eye by around 10%.



(b) When we occlude the left eye, there is a smaller loss in keypoint-specific performance for the left eye than while occluding nose.



(c) Left ankle is one of the least influential keypoints with loss only visible for meanout for occluded keypoint.

Fig. 6: Loss in AP for top 5 keypoints with largest deprivation, when an individual key point is occluded.

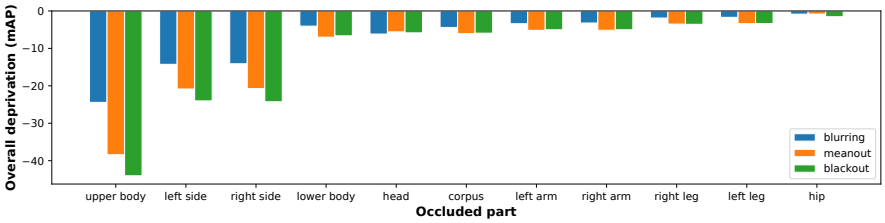
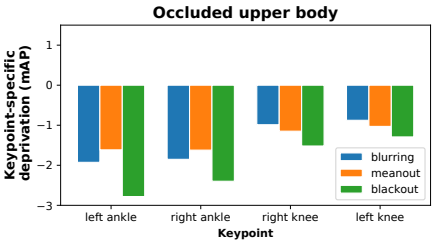
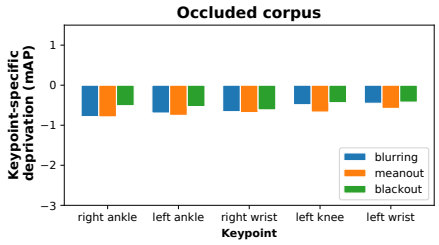


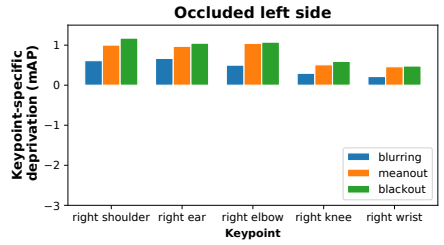
Fig. 7: Change in mAP for various parts occluded. Upper body and sides are the parts that cause the highest loss in the performance.



(a) Significant loss in performance for all of the remaining keypoints. Blackout affects the method most.



(b) Similar loss across remaining keypoints, indicating that corpus is one of the most influential parts.



(c) Occluding the left side of the body improves the performance of right shoulder, ear and elbow.

Fig. 8: Change in AP for top 5 keypoints with the largest difference, when chosen part is occluded.

Blurring, (ii) Targeted Cutout, (iii) Targeted PartMix. The augmentation techniques are called as *targeted*, because we apply them on directly the locations of keypoints or parts, instead of random location in the image. It is important to state that the proposed augmentation techniques are introduced during the second step, for a single person instance and it does not affect the detection method. Unless it is specified, we remove the word *targeted* from the name of the techniques.

Targeted Blurring. The method is originated from the analysis performed in the occlusion attacks. We envision two types of targeted blurring: keypoint blurring (Figure 9a) and part blurring (Figure 9d). To blur a keypoint or a part, Gaussian blur is applied with a kernel size of 9 pixels and 31 pixels respectively.

Targeted Cutout. The size of the keypoint cutout (Figure 9b-9c) and part cutout (Figure 9e) are similar to the

(a) Blurring on left hip (almost not visible).

(b) Cutout on left hip.

(c) Multi keypoint Cutout.

(d) Part Blurring.

(e) Part Cutout.

(f) PartMix on the right leg with a pasted random arm.

Fig. 9: Targeted keypoint augmentations: a, b, c and targeted part augmentations: d, e, f.

blurring equivalents. Instead of blurring, the area is colored with mean value of the image.

Targeted PartMix. The method is designed to mitigate the occlusions caused by another person (Figure 9f). In this approach, a different part from a random image is pasted in the place of a body part area. In this process, the keypoint labels of newly pasted part are not introduced to heatmap labels. This augmentation is only performed on body parts. Similar to the part level blurring and cutout augmentation methods, the occluded keypoints under the pasted area are still predicted.

B. Analyses of occlusion augmentation

All the following augmentation methods, except baselines, already include flipping, rotation, scaling and half-body augmentations. Each network obtains the boxes from Cascade RCNN [2] detector which has ResNet50 backbone. The results of each method can be seen in Table I.

Baselines. Table I indicates 3 baseline variants. Firstly, HRNet without any augmentations obtains only 65.3% mAP score. Secondly, adding flipping, rotation and scaling augmentations improve non-augmented baseline by 8.6%. Last variant is half body augmentation which adds only 0.4% improvements on rotation and scaling augmentations.

Single keypoint augmentations. We check the performance of 3 different augmentations: blurring, cutout and a combination of two of them which are applied on a single keypoint with the varying probability of 0.2 and 0.5 (Figure 9a-9b). We observe the highest improvement for blurring and cutout by 0.2% when the probability is chosen as 0.5 (Table I). Other single keypoint variants do not improve the performance.

Multi-keypoint augmentations. We applied random multi-keypoint variant blurring and cutout with a maximum of 5 keypoints with a probability of 0.2 (Figure 9c). The augmentation decreases the model performance by 0.4%.

Part augmentations. 4 different part augmentation methods are used: part blurring, part cutout, a combination of both them and PartMix (Figure 9d, 9e and 9f respectively). To demonstrate the effect of each augmentation, we apply them with a probability of 0.2 and 0.5. In addition, the effect of removing the labels of the occluded keypoint is also investigated as *removal* column in Table I.

In the bottom part of Table I, cutout and PartMix show 0.2% and 0.1% improvements respectively. In all the variants of blurring, small degradation or no improvement is observed. The combination of part level variants of cutout and blurring indicate some decreases of the performance for the removal configuration with probability of 0.2 and 0.5 and do not improve in non-removal configuration.

To conclude to findings from the Table I, flipping, rotation and scaling augmentations add a huge performance gain to the HRNet. However, including half-body, the occlusion based augmentation methods do not improve the performance of the pose estimator significantly.

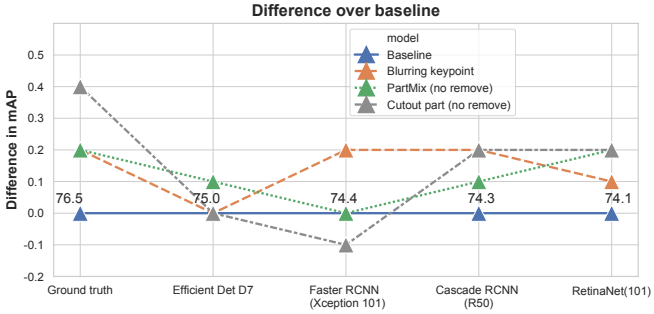


Fig. 10: Performance of chosen augmentations for HRNet-32 on various detection backbones and ground truth boxes. The ground truth bounding box performs best. Yet, none of the data augmentation methods help to improve performance over 0.2% for any object detector.

The effect of the object detection algorithms. HRNet, a Top-Down approach, uses an object detection algorithm to obtain human instances. Thus, the performance of the pose estimation considerably depends on the detection performance.

By the evidence of the Table I, we choose keypoint blurring, part cutout and PartMix methods for further analysis as they are most promising augmentations.

In this section, we evaluate the pose estimation performances of HRNet and HRNet with the chosen augmentation methods with 2 2-stage detectors, Faster RCNN [35] with XCeption 101 backbone and Cascade RCNN [2]; 2 single-

Augmentation	*level*	*removal*	*p*	**Evaluation results** AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Baseline (no augments)	-	-	-	65.3	86.4	72.6	62.6	70.7	70.2
Baseline (flip, rot, scale)	-	-	-	73.9	90.0	80.9	70.4	80.3	78.3
Baseline (flip, rot, scale, half-body)	-	-	-	74.3	90.6	81.7	70.7	80.7	78.8
Blurring	k	✗	0.2	74.3	90.4	81.6	70.8	80.6	78.7
	k	✗	0.5	74.5	90.4	81.8	70.8	80.8	78.7
Cutout	k	✗	0.2	74.3	90.4	81.7	71.0	80.3	78.7
	k	✗	0.5	74.5	90.5	81.7	70.9	80.7	78.8
Cutout + Blurring	k	✗	0.2	74.0	90.4	81.1	70.4	80.3	78.4
	k	✗	0.5	74.3	90.5	81.1	70.8	80.6	78.6
Blurring	p	✓	0.2	74.3	90.5	81.7	70.6	80.8	78.6
	p	✓	0.5	74.0	90.5	81.1	70.5	80.4	78.4
	p	✗	0.5	74.1	90.3	81.1	70.6	80.2	78.5
Cutout	p	✓	0.2	74.2	90.5	81.2	70.8	80.4	78.6
	p	✓	0.5	74.2	90.3	81.1	70.6	80.4	78.6
	p	✗	0.5	74.5	90.5	81.6	70.9	80.7	78.8
Cutout + Blurring	p	✓	0.2	73.4	90.3	80.8	69.9	79.5	77.8
	p	✓	0.5	73.9	90.4	81.0	70.5	80.0	78.3
	p	✗	0.5	74.3	90.4	81.2	70.6	80.5	78.6
Multikeypoint (max. 5)	-	-	0.2	73.9	90.1	80.9	70.5	80.2	78.3
PartMix	-	✓	0.5	74.3	90.5	81.1	70.7	80.6	78.7
	-	✗	0.5	74.4	90.7	81.5	71.1	80.5	78.8

TABLE I: Comparison of augmentation variants on COCO validation set for HRNet using CascadeRCNN bounding boxes. Upper-part indicates single-keypoint augmentation and bottom-part shows multiple-keypoint augmentation. k and p in the level column represent keypoint and part augmentations respectively. Removal column indicates if the occluded keypoints are removed from prediction. Column p is the probability of augmentation. Keypoint cutout and blurring, and part cutout and PartMix improve the performance. Other variants obtain results either on a par with baseline or worse than baseline.

Augmentation	*level*	*remove*	*p*	**Evaluation results** *Head*	*Shoulder*	*Elbow*	*Wrist*	*Hip*	*Knee*	*Ankle*	***Total***
Baseline	-	-	-	97.1	95.9	90.4	86.4	89.1	87.2	83.3	90.3
Blurring	k	✗	0.5	97.3	95.9	90.5	86.2	89.2	86.4	83.1	90.3
Cutout	p	✗	0.5	97.2	96.3	90.7	86.7	89.4	86.7	83.3	**90.5**
PartMix	-	✗	0.5	97.4	96.2	91.0	86.8	89.2	86.7	83.0	**90.5**

TABLE II: Results on MPII dataset. Keypoint blurring obtains on a par with the baseline, yet part cutout and PartMix increase the performance.

stage detectors, RetinaNet [27] and EfficientDet D7 [43]; and by using ground truth boxes of human instances (Figure 10).

All the augmentations indicate improvements using ground truth bounding boxes by 0.2% for keypoint blurring and PartMix, and 0.4% for part cutout. All the chosen augmentation methods obtain better result with Cascade RCNN and RetinaNet $0.1 - 0.2\%$ depending on the augmentation. With EfficientDet D7 detector, keypoint blurring and part cutout result in similar to baseline except 0.1% improvement by PartMix. For Faster-RCNN, keypoint blurring shows 0.2% increase, yet part cutout degrades the performance by 0.1%.

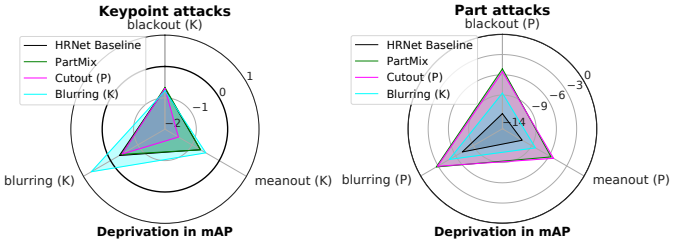
The performances of baseline and the augmentations vary depending on the object detector. The augmentation methods improves the results slightly, yet the gain is insignificant.

Performance on MPII. We also test the data augmentation methods on MPII dataset (Table II). If we check the total contribution of the proposed augmentations, keypoint blurring result in on a par with baseline, yet part cutout and PartMix increase the performance by 0.2% for the metric PCK@0.5. The largest improvement per keypoint is observed for elbows by 0.6% and wrists by 0.3%, with the degradation on knees and ankles by 0.4% and 0.2% respectively.

Similar to analyses on the COCO dataset, the proposed augmentations can only improve the performance slightly.

How much robustness does data augmentation bring to the occlusion problem? The analysis of the robustness of the baseline and the proposed augmentations to the occlusion attacks can be seen in Figure 11. The analysis is done on COCO dataset and the results are shown as mAP score of all keypoints. We can clearly see that training with the keypoint blurring augmentation makes the network more robust against blurring attack, but there is no significant improvement for the other keypoint attacks. In case of part attacks, we observe an improvement across all augmentation methods over the baseline. For the part augmentations, there is a significant

improvement against all part level attacks in comparison to baseline. Specifically, PartMix has almost no advantages against keypoint attacks, however, it improves part level methods about more than 5% in comparison to baseline. Part cutout obtains similar performance with PartMix against part attacks. Proposed augmentations reduce the performance deprivations when we apply occlusion attacks, yet data augmentation still does not solve the occlusion problem.



(a) (b)

Fig. 11: Robustness comparison of proposed methods against (a) keypoint and (b) part occlusion attacks. Part augmentations improve the baseline but does not solve occlusion.

V. Discussion and Conclusion

In this study, we investigate the sensitivity of human pose estimators to occlusion. Firstly, we introduce targeted keypoint and body part occlusion attacks to show how much occlusion affects the performance. Secondly, keypoint and part based data augmentation techniques against occlusion are investigated. The structured analyses indicate that deep pose estimators are not robust to occlusion. With all the bells and whistles, the current and proposed data augmentation methods do **not** bring significant improvements on the performance of the top-down pose estimators. Figure 12 also shows small improvements and failures of baseline and keypoint blurring augmentation. Our paper is important because it helps data scientists looking for improvements against occlusions to not work on data augmentation. Battling occlusions is still an open problem for human pose estimation.

Part based attacks and augmentation are applied as a rectangle shape. This fact can introduce unusual artefacts because the natural occlusions can have arbitrary shapes. The proposed occlusion attacks can be also applied to check the occlusion robustness of bottom-up methods.

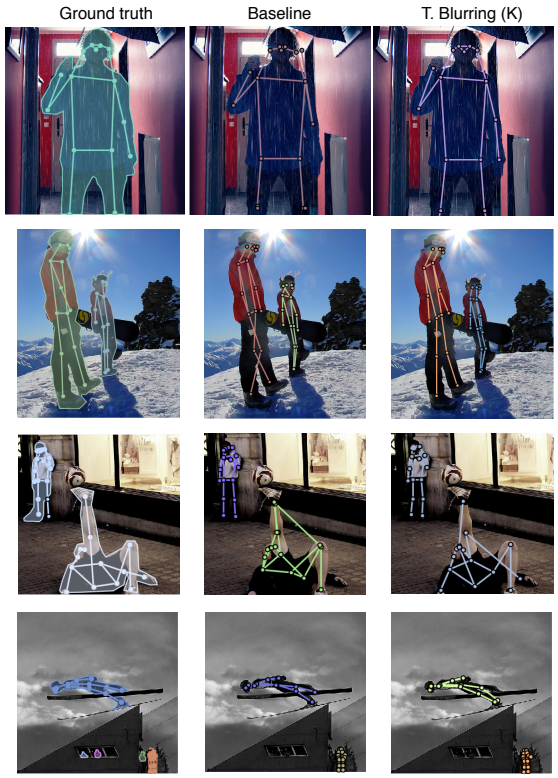


Fig. 12: Qualitative comparison between ground truth (left), baseline (middle) and keypoint Blurring (K) (right). 1st and 2st rows respectively - misplacement of left wrist keypoint and mismatch between knee keypoints in the baseline and keypoint blurring fixes the mistakes. 3rd row - both baseline and proposed method produce wrong keypoints. 4th row - baseline produces near-optimal keypoints whilst keypoint blurring makes mistake on left ankle keypoint. Data augmentation does not solve occlusion problem.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018.

[4] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742, 2016.

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] J. Choe, S. Lee, and H. Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[8] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. pages 5669–5678, 07 2017.

[9] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.

[10] Philipe Ambrozio Dias, Damiano Malafronte, Henry Medeiros, and Francesca Odone. Gaze estimation for assisted living environments. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 290–299, 2020.

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.

[12] Zhijie Fang and Antonio M. López. Intention recognition of pedestrians

and cyclists by 2d pose estimation. *ArXiv*, abs/1910.03858, 2019.
[13] Geoffrey French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham D. Finlayson. Consistency regularization and cutmix for semi-supervised semantic segmentation. *CoRR*, abs/1906.01916, 2019.
[14] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-Track: Efficient Pose Estimation in Videos. In *CVPR*, 2018.
[15] Thomas Golda, Tobias Kalb, Arne Schumann, and Jürgen Beyerer. Human pose estimation for real-world crowded scenarios. *CoRR*, abs/1907.06922, 2019.
[16] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. *CoRR*, abs/1809.02499, 2018.
[17] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[18] Ying Huang, Bin Sun, Haipeng Kan, Jiankai Zhuang, and Zengchang Qin. Followmeup sports: New benchmark for 2d human keypoint recognition. *ArXiv*, abs/1911.08344, 2019.
[19] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. *CoRR*, abs/1803.09894, 2018.
[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.
[21] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. *CoRR*, abs/1903.06593, 2019.
[22] L. Ladický, P. H. S. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3578–3585, 2013.
[23] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *ArXiv*, abs/1901.00148, 2019.
[24] Zhong Li, Xin Chen, Wangyiteng Zhou, Yingliang Zhang, and Jingyi Yu. Pose2body: Pose-guided human parts segmentation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 640–645. IEEE, 2019.
[25] Kevin Lin, Lijuan Wang, Kun Luo, Yinpeng Chen, Zicheng Liu, and Ming-Ting Sun. Cross-domain complementary learning using pose for multi-person part segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.
[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.
[28] Diogo Luvizon, David Picard, and Hedi Tabia. Multi-task deep learning for real-time 3d human pose estimation and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020.
[29] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.
[30] Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1293–1301, 2015.
[31] Štěpán Obdržálek, Gregorij Kurillo, Ferda Ofli, Ruzena Bajcsy, Edmund Seto, Holly Jimison, and Michael Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1188–1193. IEEE, 2012.
[32] Sai Perla, Sudip Das, Partha Mukherjee, and Ujjwal Bhattacharya. Cluenet : A deep framework for occluded pedestrian pose estimation. 12 2019.
[33] U. Rafi, J. Gall, and B. Leibe. A semantic occlusion model for human pose estimation from a single depth image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 67–74, 2015.
[34] N. D. Reddy, M. Vo, and S. G. Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7318–7327, 2019.
[35] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.
[36] Sahil Shah, Naman Jain, Abhishek Sharma, and Arjun Jain. On the robustness of human pose estimation, 08 2019.
[37] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
[38] K. K. Singh and Y. J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553, 2017.
[39] Anna Sokolova and Anton Konushin. Pose-based deep gait recognition. *CoRR*, abs/1710.06512, 2017.
[40] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Online localization and prediction of actions and interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:459–472, 2019.
[41] C. Summers and M. J. Dinneen. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270, 2019.
[42] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. 2019.
[43] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *ArXiv*, abs/1911.09070, 2019.
[44] H. L. Tavares, J. B. C. Neto, J. P. Papa, D. Colombo, and A. N. Marana. Tracking and re-identification of people using soft-biometrics. In *2019 XV Workshop de Visão Computacional (WVC)*, pages 78–83, 2019.
[45] Luke Taylor and Geoff Nitschke. Improving deep learning using generic data augmentation. *arXiv preprint arXiv:1708.06020*, 2017.
[46] Y. Tokozume, Y. Ushiku, and T. Harada. Between-class learning for image classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018.
[47] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.
[48] Jagannadan Varadarajan, Ramanathan Subramanian, Samuel Rota Bulò, Narendra Ahuja, Oswald Lanz, and Elisa Ricci. Joint estimation of human pose and conversational groups from social scenes. *International Journal of Computer Vision*, 126(2-4):410–429, 2018.
[49] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
[50] Sijia Wang, Fabian Flohr, Hui Xiong, Tuopu Wen, Bao feng Wang, Mengmeng Yang, and Diange Yang. Leverage of limb detection in pose estimation for vulnerable road users. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 528–534, 2019.
[51] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016.
[52] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L. Yuille. Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[53] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *CoRR*, abs/1804.06208, 2018.
[54] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.
[55] Huayong Xu, Yangyan Li, Wenzheng Chen, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. A holistic approach for data-driven object cutout. *CoRR*, abs/1608.05180, 2016.
[56] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. *CoRR*, abs/1708.01101, 2017.
[57] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019.
[58] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. *ArXiv*, abs/1910.06278, 2019.
[59] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.
[60] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *CoRR*, abs/1708.04896, 2017.

APPENDIX

MORE RESULTS ON COCO VAL SET

More results for HRNet

We provide results for higher (384x256) than default resolution (256x192) of input images (Figure 13). The training process is following the aforementioned scheme for COCO dataset.

According to our analysis of the performance across a variety of detection backbones, we notice that PartMix is slightly improving performance - with the greatest boost of 0.4% for Cascade R-CNN and 0.3% for Faster RCNN. For both Blurring (keypoint level) and Cutout (part level) we observe no significant improvement or even decrease in the performance - for Cutout using EfficientDet, Faster RCNN and RetinaNet and for Blurring using ReinaNet. All the presented augmentations show largest gain for Cascade RCNN as detector providing person bounding boxes.

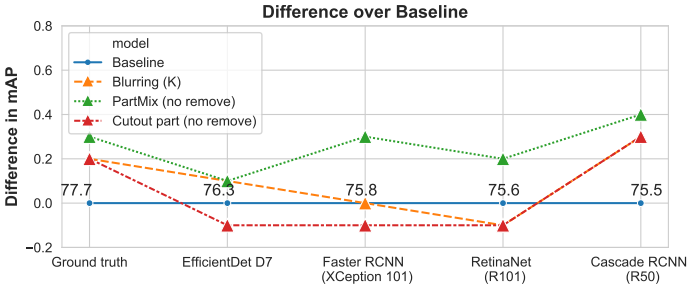


Fig. 13: Higher resolution input for HRNet 32: 384x256 (above) instead of 256x192(previously). The best performance across detection backbones is observed for PartMix.

Results for SimpleBaseline

The presented augmentations are not only limited to HRNet but can be used with the various architectures like SimpleBaseline (results in Figure 14). The training procedure was following the scheme from [53].

By checking the performance across the various detection backbones we observe either small or no improvement at all. PartMix show the most significant improvement across detection backbones, with 0.4 % boost in the performance for the Ground truth boxes and the boxes produced by Cascade RCNN, 0.2 % for EfficientDet and Faster RCNN and 0.1 % for RetinaNet. Cutout and Blurring give an improvement of at most 0.2 % across all the detection backbones, apart from 0.4 % for Cutout using Ground truth bounding boxes.

Results for Higher HRNet

Apart from investigating top-down approaches we also check the performance while applying the augmentations on bottom-up approach - Higher HRNet [6].

For this approach, we had to adapt our method to bottom-up approaches as we were operating on full images with multiple person instances per each image, not on bounding box with a single person, as in the case of top-down approaches.

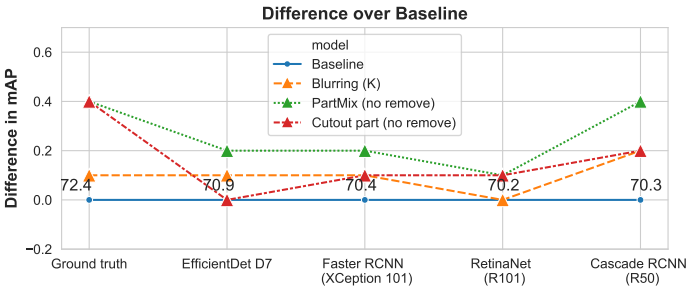


Fig. 14: Performance of chosen augmentations for SimpleBaseline (ResNet-50) on various detection backbones and ground truth boxes. The ground truth bounding boxes perform best.

	Evaluation results				
Augmentation	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Baseline	67.1	86.2	73.0	61.5	76.1
Blurring (K)	66.5	86.3	72.1	60.6	75.7
Cutout part (no remove)	66.6	86.4	72.9	60.7	75.6
PartMix (no remove)	67.0	86.4	73.0	61.3	75.8

TABLE III: Results for Higher HRNet (512x512 input image with HRNet-32 backbone). Proposed augmentations degrade the performance of bottom-up methods, with most significant for Blurring on keypoint level and Cutout on part level.

For bottom-up approaches per each person instance, there is a chance of applying augmentation, which simulates similar behaviour as in top-down approaches. This algorithm produces highly perturbed images which probably indicate degradation of performance after applying proposed augmentations.

We provide results for Higher HRNet (Table III), with HRNet-32 backbone, using 512x512 input images, following the training scheme from [6]. All the methods degrade the performance of the network, keypoint level Blurring producing 0.6 % worse results than Baseline, part level Cutout producing 0.5% worse results and PartMix showing small degradation of 0.1%.

Our results indicate that the adapted pipeline is producing very perturbed images for the bottom-up approach, which harm the learning process, resulting in worse performance compared to not using proposed augmentations.

Contents

1

Background about Deep Learning

This section introduces key concepts of Deep Learning.

1.1. Deep Learning

Deep Learning is a branch of machine learning, which learns from a raw representation of multiple examples. A recent development in hardware, and availability of excellent frameworks, enabled reshaping the State of the Art in multiple visual tasks [8, 11, 18].

The main advantage of deep learning over previous classical approaches in machine learning is an automated process of feature engineering, where instead of carefully creating features, the model learns useful representation from raw data.

As presented in Figure 1.1, input vector is pushed through a cascade of hidden layers of neural network. Each neuron in hidden layer is fully-connected with all the neurons from the previous layer. Output layer, which is the last layer of the network, produces either class scores (classification problem), or continuous value (regression problem).

1.2. Convolution Neural Network

Behind the success of Deep Learning in Computer Vision, stand Convolution Neural Networks. As the name suggests at its core convolution operation play an important role. CNN's are a combination of convolutions (Figure 1.2), pooling operation and non-linear activation with the output of each layer to be input the next one. Convolution filters slide over an image to extract features. Pooling operation removes inessential information, by decreasing spatial resolution. Non-linear activation function produces powerful function approximations.

1.2.1. Convolutions

Convolutions are the main building block of CNNs [12], so it is important to understand how they operate. Convolution layers aim to learn image features using small squares of input data, called kernels or filters.

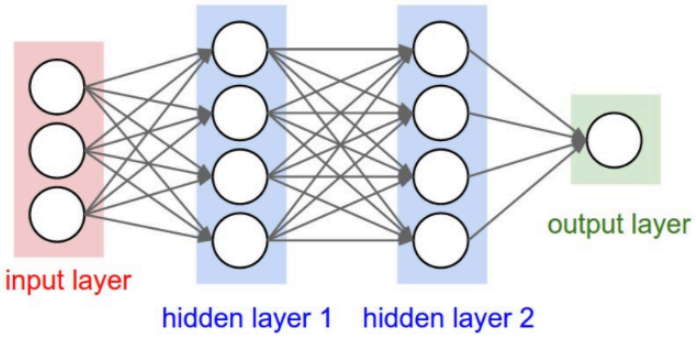


Figure 1.1: A regular 3-layer Neural Network [1]. Each neuron is connected to every neuron in the consecutive layer.

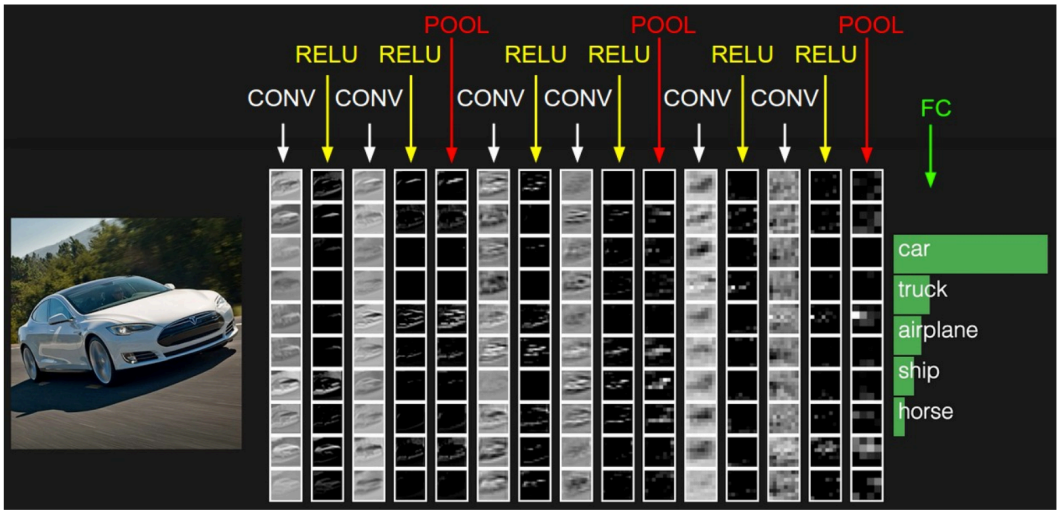
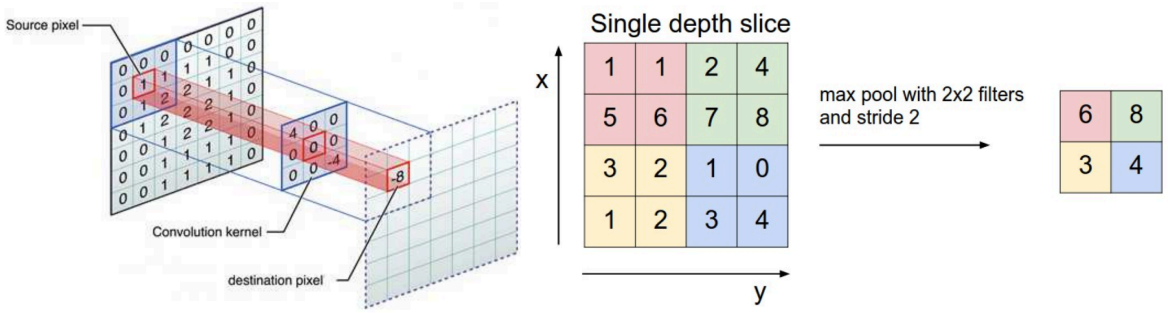


Figure 1.2: Network components example from [2]. CNN is cascade of multiple convolution (CONV) and pooling layers (POOL) with activation functions in between (RELU). At the end output is combined from all the available feature maps via fully-connected layer (FC) to produce class labels.



(a) Example of single convolution operation from [2]. Element-wise matrix multiplication of 3x3 region of the source image by convolution kernel followed by sum over all the elements of produced matrix.

(b) Example of max pooling operation from [1] - max pool with 2x2 filters and stride 2. Out of each colored area highest number is chosen to be a final product of the operation.

Figure 1.3: Example of two operation in CNNs: (a) convolution, (b) pooling.

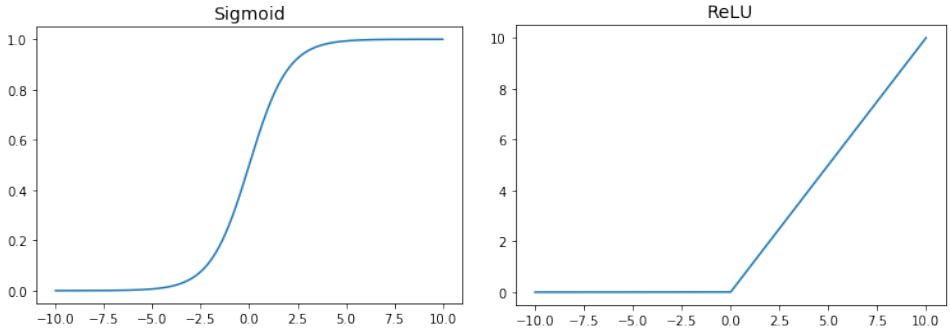
Filters are sliding over the image, where output is the sum of product between convolution filter and the corresponding area of the image. All the elements of a single feature map use the very same convolution kernel. Parameters of convolution kernel are learned during training.

We can observe how does sliding of kernel look like in Figure 1.3a, where to 7x7 input data 3x3 filter is applied. On the part of input data, element-wise multiplication with convolution kernel is applied. After obtaining 3x3 matrix sum over all values is applied which produces -8 at the output plane for a given feature map.

1.2.2. Pooling

Pooling layer is a function that reduces the spatial resolution of the feature map by max (Figure 1.3b) or average operation. After performing down-sampling model is more robust against overfitting and is able to focus on the presence of a feature regardless of its specific location. Every part of an output essentially corresponds to sub-region of an image, which translates to a reduction of spatial dimension. With compact representation provided by pooling layer, it is possible to learn a larger selection of features with a smaller number of parameter and computation load. One of the most commonly used pooling layers is Max Pooling, which is offering robustness via ignoring small change of non-maximum values.

Following the example from Figure 1.3b, we can see that after applying 2x2 MaxPooling filter on 4x4 input we obtain 2x2 output having only maximal values within the coloured areas.



(a) Sigmoid activation function

(b) ReLU activation function

Figure 1.4: Example of most widely used activation functions: (a) sigmoid, (b) ReLU

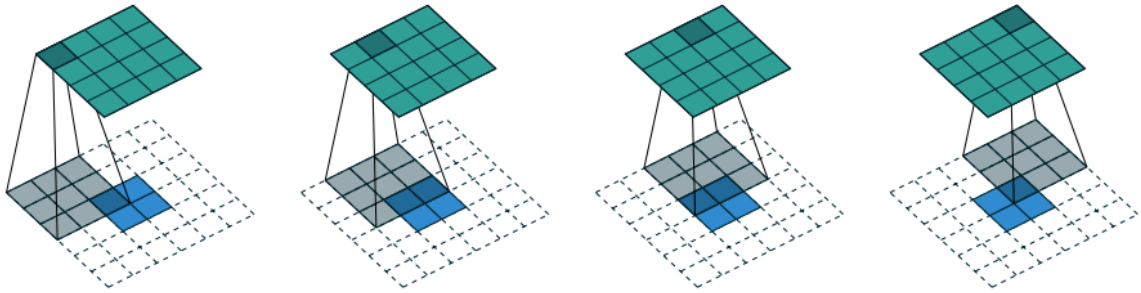


Figure 1.5: Example of transposed convolution from [7] with 3x3 kernel over a 4x4 input. Equivalent to 3x3 kernel over 2x2 input padded with a 2x2 border of zeros.

1.2.3. Activation Functions

The sequence of the only convolution and pooling layers can be represented as a simple weighted sum of input, regardless of a number of layers. To mitigate this problem, the non-linear activation functions are added at the end or between the layers.

One of the most widely used activation function is ReLU (Rectified Linear Unit). In essence, this function is trying to find the maximum between the input value and zero. Due to its simplicity, the computation cost is quite low and can be performed quite fast. Unfortunately, due to its characteristics for all the negative values output will be essentially 0, which may result in dead neuron situation and gradient will also be 0, making it impossible to perform back-propagation. ReLU activation function is vulnerable against parameter initialization and learning rate.

$$f(x) = max(0, x) \tag{1.1}$$

Before the success of ReLU, prime was lead by Sigmoid function (Equation 1.2). This function is mapping a real number into (0,1). Performing exponential operation is quite time-consuming.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.2}$$

Another example of a popular activation function is Softmax (Equation 1.3), which can be used for a neuron that has more than one-dimensional output. For the classification task, the output of fully-connected layer - logits, are a real number. After using the Softmax function, the logits represent probabilities of different classes, which all sum to one.

$$f_i(\hat{x}) = \frac{e^{x_i}}{\sum_{i=1}^{j} e^{x_j}} \tag{1.3}$$

1.2.4. Transposed Convolution

Transpose convolution, also known under the inaccurate name deconvolution, is an operation where a low dimensional feature map is transformed into high dimensional output. This means that the output of this

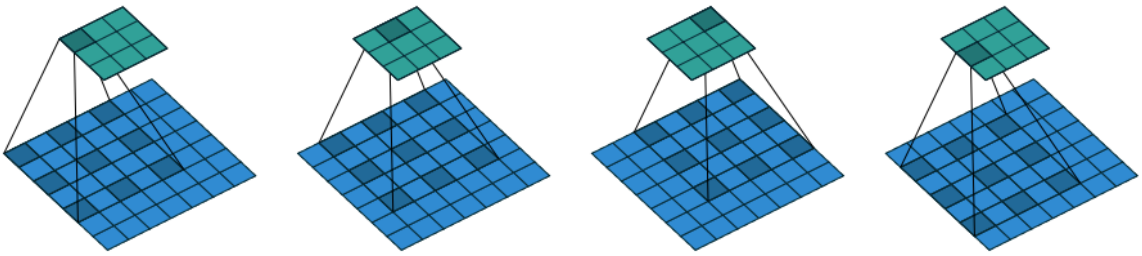


Figure 1.6: Example of dilated convolution from [7] with 3x3 convolution kernel over 7x7 input with a dilation factor of 2.

convolution outputs more detailed information than provided on the input. Transposed convolution is often used in upscaling or mapping features in an overcomplete autoencoder. Transposed Convolution can reverse dimension reduction applied by original convolution, with the flexibility to truly reverse the effect of convolution (deconvolution).

To understand the transpose convolution following example will be in order: consider a kernel K, an input i, and output o, with i and o being vectors. Convolution operation, will then be K x $i = o$. If we multiply both sides by the transpose of kernel K^t, we would get $i = K^t$ x o. This example shows that translation from a lower dimension to higher dimension is possible. Visualization of that process can be seen in Figure 1.5. Due to filter size and the limited input data, padding around the input is necessary.

1.2.5. Dilated Convolution

In dilated convolution [26] operation the kernel that is used is inflated by inserting a predefined spacing between the kernel values, controlled by a parameter l, known as dilation rate.

Dilatation rate is used for inserting $l - 1$ empty spaces between kernel values. This concept results in increasing receptive filed of the model while leaving the number of parameters unchanged. Visualization of that process can be seen in Figure 1.6.

An interesting fact about dilated convolution is that the receptive field increases exponentially while the number of parameters grows linearly. Dilated convolutions are often used for image segmentation, where each pixel is labelled by a corresponding class. To upscale an image, we can apply convolution and then deconvolution, however, this results in an increased number of parameters. To mitigate that, [26] applied dilated convolution to maintain the high resolution of the output without having to upsample.

1.2.6. Fully-convolutional Neural Networks

Representation of the input learned by CNN can be used by any conventional ML model. Typically, fully connected layers were used for the final classification, however, this step resulted in a significant increase in number parameters, affecting the generalization abilities of the network.

Fully convolutional variants have presented competitive results in object recognition tasks [19, 20], by replacing fully connected layers with convolutional ones, resulting in a decrease in a number of trainable parameters, the output format features map, which could be used in the more fine-grained image segmentation setting [15]. They are really popular in segmentation problems [4, 10, 15] and also in human pose estimation [16, 22, 25].

1.3. Training a CNN

The training process of neural networks can be broken into 3 main steps: forward pass, loss calculation and back-propagation pass to update learnable parameters.

1.3.1. Forward pass

The forward pass is simply pushing input data via the network to get the desired output. Consider the following scenario, where we have weights W and bias b, the output y is:

$$y = Wx + b \tag{1.4}$$

Then the output of the single layer is again used as an input to the next layer, following its way up to the final output.

1.3.2. Loss function

Aim of training a neural network is to optimize weights and biases of the filters, to minimize a loss function. After completing the forward pass, loss between output and provided target label is calculated. The common choice for loss function in the classification problem is cross-entropy loss (CE) where the output of the network corresponds to probability and is between 0 and 1.

$$L = -(label * log(output) + (1 - label) * log(1 - output)) \tag{1.5}$$

For regression tasks like object detection or human pose estimation, often used is L2 Loss (mean squared loss). In essence, it is a sum of squared distance between the predicted output and the ground-truth label.

$$L = \frac{1}{N}\sum_{i}^{N}(label - output)^2 \tag{1.6}$$

1.3.3. Backpropagation

After successfully getting an output from the network and calculating chosen loss, now there is a step to minimize the loss. Backpropagation is an algorithm building on the chain rule of calculus, to easily optimize learnable parameters of CNN in en-to-end fashion.

Let us consider that concept a bit more in-depth. Considering that our network has weights and biases, corresponding to w. To update the parameters, gradient descent is applied taking smaller or larger steps along the negative gradient $-\frac{\delta L}{\delta W}$. Thanks to chain rule in calculus, it is possible to reuse already calculated gradients and reduce computation cost. Example of that rule can be found below, where y is the output of the network, t is target label and x is an input:

$$\begin{gathered} z = wx + b, \quad y = \sigma(z), \quad L = \tfrac{1}{2}(y - t)^2 \\ \frac{\delta L}{\delta y} = y - t, \quad \frac{\delta L}{\delta z} = \frac{\delta L}{\delta y}\sigma'(z), \quad \frac{\delta L}{\delta w} = \frac{\delta L}{\delta z} \\ \bar{y} = y - t, \quad \bar{z} = \bar{y}\sigma'(z), \quad \bar{w} = \bar{z}w, \bar{b} = \bar{z} \end{gathered} \tag{1.7}$$

Following this idea, the calculated loss is backpropagated to every neuron through the corresponding gradients. Parameters are then updated according to the contribution (gradient) to the output.

1.3.4. Learning rate

Learning rate corresponds to the magnitude of the step taken in updating the parameters during back-propagation. Using too small learning rate, there is a risk of finishing within the local minimum. On the other hand, too large learning rate could not allow the network to converge to any minimum, because it will always overshoot. A common practice is to use an adaptive learning rate, with larger at the beginning, decrease as training progresses.

1.3.5. Optimizers - Gradient Descent

Deep Learning is an optimization problem which aims to minimize the loss function ($J(\theta)$, where θ are the parameter of the model. The algorithm used to optimize this process is called optimizer.

Gradient $\Delta_\theta J(\theta)$ is derivative of multi-variable function.

$$\Delta_\theta J(\theta) = \frac{dJ(\theta)}{d\theta} \tag{1.8}$$

Keeping in mind the definition of the gradient we know that function f(x) decreases the fastest in the direction of $-\Delta_\theta J(\theta)$. Following that, parameters are updated in the following manner:

$$\theta = \theta - \eta * \Delta_\theta J(\theta) \tag{1.9}$$

The disadvantage of that approach is speed and memory consumption. For one update, gradient over all dataset has to be calculated, which makes online training impossible.

1.3.6. Optimizers - Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is probably one of the most popular optimizers, where updates of the parameters are calculated for every pair of training sample x_i and corresponding label y_i. Having this behaviour, SGD is much faster and can be used in online fashion, however, introduces fluctuations for the value of loss function by updating with high variance.

$$\theta_{t+1} = \theta_t - \eta * \Delta_\theta J(\theta, x_i, y_i) \tag{1.10}$$

1.3.7. Optimizers - Mini-batch Gradient Descent

Instead of using either full dataset (Gradient Descent) or only one sample (SGD), this optimizer is performing the estimation of the gradient using mini-batch. Thanks to that gradient descent is still fast, with a more stable learning process.

$$\theta_{t+1} = \theta_t - \eta * \Delta_\theta J(\theta, x_{(i:i+n)}, y_{(i:i+n)}) \tag{1.11}$$

1.3.8. Optimizers - Adam

Adaptive Moment Estimation (Adam) is optimizer computing adaptive learning rates. Currently, it is proven to be best performing across many platforms, including Human Pose Estimation.

Firstly, using exponentially weighted averages of past and past squared gradients m_t and v_t are computed in the following way:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_1) g_t^2 \end{aligned} \tag{1.12}$$

where g_t stands for the gradients, β_1, β_2 are two hyper-parameters to be tuned. Then bias correction is used for both m_t and v_t:

$$\begin{aligned} \hat{m_t} &= \frac{m_t}{1=\beta_1^t} \\ \hat{v_t} &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{1.13}$$

Finally, the learning parameters of the model are updated in the following manner:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v_t}} + \epsilon} \hat{m_t} \tag{1.14}$$

1.4. Regularization

Similar to classical ML approaches, the goal of the deep neural networks is to generalize well on unseen data. Models, performing well on training but poorly on held-out data suffer from overfitting. The most straight forward way to reduce over-fitting is to increase the amount of data, however, in most cases that is not possible. Another possibility is to reduce the complexity of the model as a simpler model require fewer data to generalize better, however, at the cost of reduced representational power. In other words, for a simpler model, it is easier to have similar performance between train and test set, however, their performance may be worse on train set compared to more complicated alternatives. Regularisation methods punish model complexity, forcing the model to extract more representation, which generalizes better across the data. Lets now investigate some of these methods.

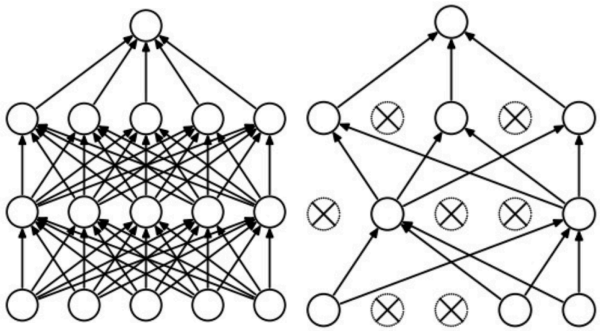
1.4.1. Early stopping

Early stopping is quite elegant and simple method exploiting validation loss, which decreases when the model is still improving. When validation loss is increasing with training loss decreasing indicate that the model is over-fitting, which is a clear signal to terminate the training process.

1.4.2. Dropout

Dropout [21] (Figure 1.7) is another regularization technique, which during each training step mutes output of fraction p of all activations. During repeated epochs, it happens that the same examples have a different part of the activation, which enforce a model to learn more generalizable features.

Different from the training process, during testing all activations are available but the output is scaled by $1-p$. Rise of Batch normalization, explained in next step, decreased popularity of dropout, as some works indicate that their simultaneous usage harms model performance [13].



(a) Standard neural network. (b) After applying dropout.

Figure 1.7: Illustration of dropout [21]

1.4.3. Batch normalization

Vanishing and exploding gradient are common problems in deep learning. When we consider the single layer in a deep neural network, parameters are always changing during gradient descent, consequently changing output distribution. The output of that layer is fed to another one, which indicates that input of the next layer again varies, which makes it a very difficult model to learn good representations. This process of changed distribution of activations within a certain layer is called Internal Covariate Shift.

To deal with that problem, activations distribution of each layer are scaled to have zero mean value μ and unit variance σ^2. Then to understand how does the output from batch normalization looks like, let's look below:

$$\begin{aligned} \mu &= \frac{1}{m}\sum_{i=1}^{m} x_i \\ \sigma_B^2 &= 1/m\sum_{i=1}^{m}(x_i-\mu_B)^2 \\ \hat{x_i} &= \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}} \\ y_i &= \gamma\hat{x_i}+\beta \end{aligned} \tag{1.15}$$

Applying batch normalization led to speed-up in the training process, as models using this method can apply larger learning rates with quicker learning rate decay. Moreover, usage of batch normalization acts as a regularization method, which reduces the need for using other regularization methods.

1.5. Data augmentations

As stated before, one way to overcome over-fitting is to gather more data. Data augmentations act in that direction creating perturbed copies of the original sample. Thanks to that, it should be easier for the network to model the distribution of data. Within this topic, we will investigate affine augmentations, channel transformation and some strong augmentations like Cutout and CutMix.

1.5.1. Affine transforms

This augmentations aim at robustness against examples which have been transformed using affine transforms, which are geometric transformations preserving line and parallelism. Good examples of that affine transform augmentations are zooming, flipping and rotation (Figure 1.8 b-d).

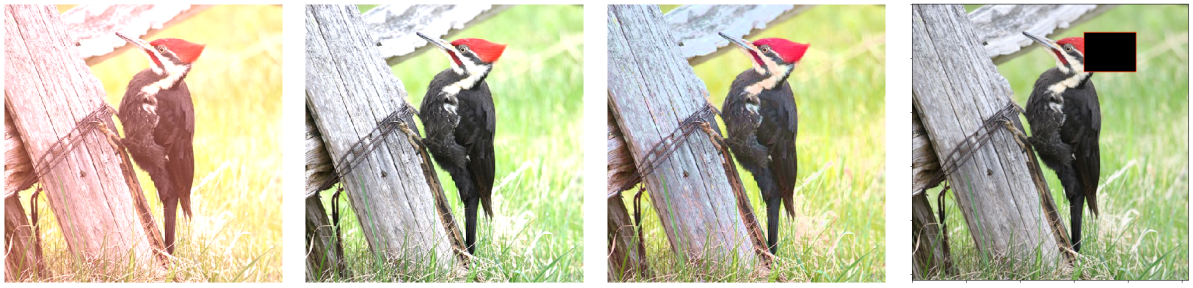
1.5.2. Channel transforms

Another branch of augmentations aims at robustness against colour channel perturbations. Apart from tweaking RGB channels, it is also common to perturb image in HSV and HSL colour representation (both in Figure 1.8 e-g). Good examples are shifts in brightness, RGB channels, hue and saturation or contrast.

1.5.3. Cutout

The cutout is another method which is also known under the name, Regional Dropout. The idea of this method is to cover with a box, certain part of an image to enforce model to learn beyond most discriminative features.

(a) Original image. (b) Zoom. (c) Fliping. (d) Rotation.



(e) RGB. (f) Brightness. (g) Hue and saturation. (h) Cutout.

Figure 1.8: Example of augmentations - affine transforms (b-d), channel transformations (e-g) and cutout (h)

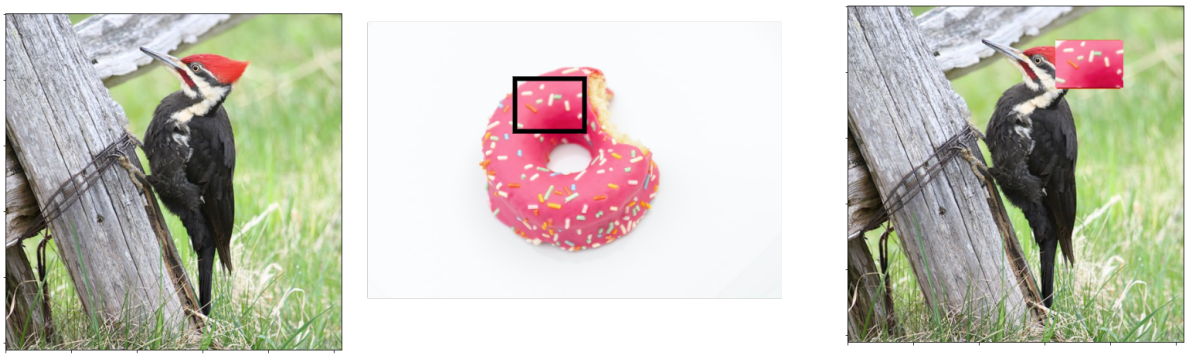


Figure 1.9: CutMix augmentation. Part of the Doughnut image is pasted on picture of a bird. Target label on final image is combination of bird and doughnut with confidence score corresponding to the pasted area.

Thanks to that model should be more robust against partial occlusions as with occlusions or missing features model could still provide good labels. In the original paper, authors are using uniform distribution to choose position and dimension parameters of a bounding box. Example of Cutout can be observed in Figure 1.8 h.

1.5.4. CutMix

CutMix is an interesting enhancement over Cutout wherein the affected area of Cutout, part of the different image is pasted. Apart from image changes, it is crucial to produce mixed label: consisting of combination from two of the images, corresponding to their area.

Having our example (Figure 1.9) we have two images, one of a bird another of a doughnut. After applying the Cutmix and pasting 0.1 of doughnut picture on the bird picture, our mixed target label is actually: 0.9 bird, 0.1 doughnut. By doing so the network is enforced to recognize both of the classes in the final image.

2

Human Pose Estimation

The task of human pose estimation is a popular branch of Computer Vision for more than 20 years. Aim of this domain is to localize human joins (also known as keypoints - elbows, wrists, etc) in images or videos. In this section closer view on this area will be presented, starting with a description of the problem and most famous deep learning approaches.

Visualization of joint location for two of the most popular human pose estimation is available in Figure 2.1.

2.1. Description of a problem

Main problems in Human Pose Estimation are occlusions, unusual poses, missing key points and changes in lighting and clothing. Example of failure cases for currently most advanced architecture - HRNet can be seen in Figure 2.2.

2.2. Metrics

In the premise of human pose estimation and multiple datasets, there is a variety of metrics. For the sake of this introduction to Human Pose Estimation, the main metrics for currently most popular datasets - MPII [3] and COCO [14].

2.2.1. PCK - Percentage of Correct Keypoints

A detected joint is considered correct when the distance between the predicted location and true location is within a certain threshold. The threshold is often defined by a fraction of head bone link. In MPII often used metric is PCK@0.5, so then joints only within a distance of $0.5 * head\_bone\_lenght$ are considered as correctly detected. Having such a personalized distance definition it is possible to both accommodate larger and smaller instances.

2.2.2. OKS - Object Keypoint Similarity

OKS is the main metric for COCO dataset. The formula for this metric is:

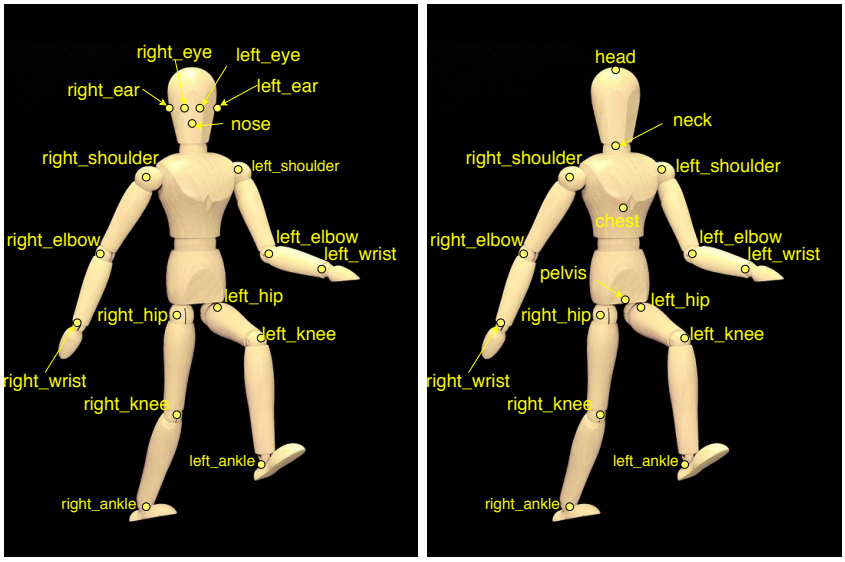
$$OKS = \frac{\sum_i exp(-\frac{d_i^2}{2s^2 * k_i^2}) * \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{2.1}$$

where d is the distance between predicted and the true location of keypoint,v_i is visibility flag of keypoint i and s is an object scale and k_i is a per-keypoint constant that controls falloff, calculated by COCO researchers.

In simple words, OKS acts similarly as IOU in Object Detection or Image Segmentation. Typically metric is analyzed via Average Precision (AP@50, AP@75, the average across 10 points between 50 and 95, performance on medium and large instances) and Average Recall (same steps as for AP).

2.3. Deep Learning approaches

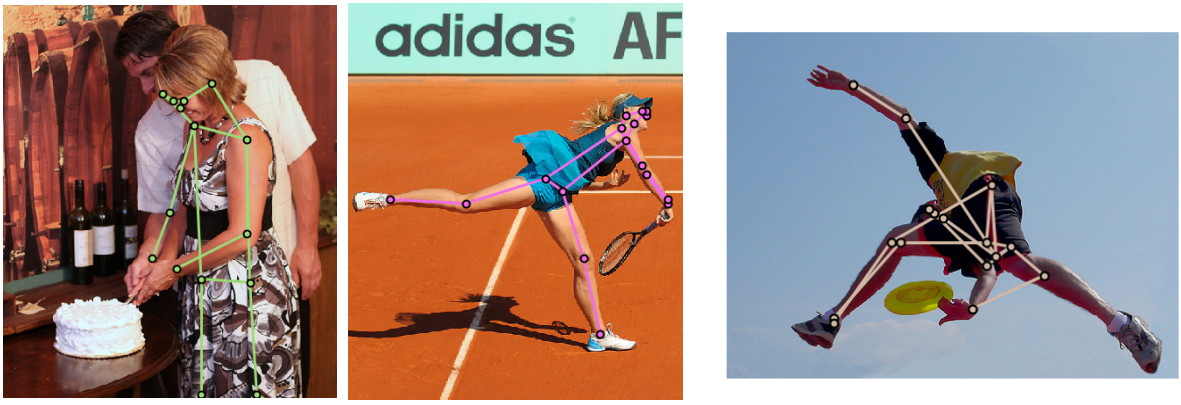
Currently, the world of Human Pose Estimation is fully dominated by Deep Learning approaches with interesting architectures often inspired from different domains [5, 9, 17], but also then ones which started Human



(a) COCO annotations, total of 17 keypoints. Detailed annotation of head with 5 keypoints (nose, ears and eyes).

(b) MPII annotations, 16 keypoints. Very similar to COCO annotations, with fewer keypoints for head annotations and additional keypoints for chest and pelvis.

Figure 2.1: Visualization of keypoint mapping in two most common human pose estimation datasets (a) COCO, (b) MPII.



(a) With overlapping person instances network wrongly annotates arm keypoints of women on the men instance.

(b) Due to self occlusion of left arm, keypoints of that limb are wrongly located in position of right arm.

(c) Due to unusual pose and self occlusions model fails to localize keypoints correctly of the frisbee player.

Figure 2.2: Example of failure cases for HRNet-32 for problematic images: (a) overlapping instances, (b) self-occlusions and (c) unusual pose.

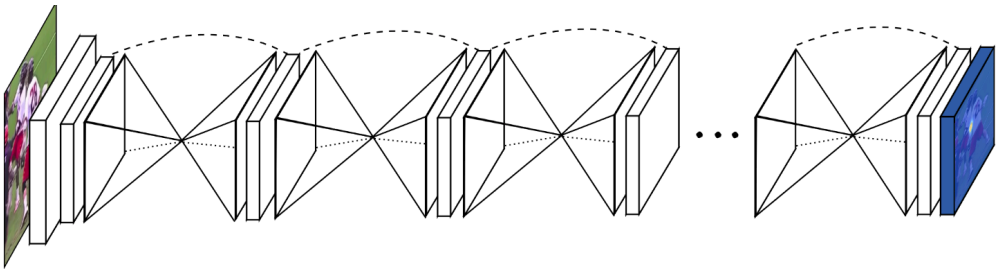


Figure 2.3: Scheme presenting high level view on Hourglass structure - sequence of same structure Hourglass modules, continuously reducing and then increasing resolution via strided convolutions and nearest neighbor upsampling.

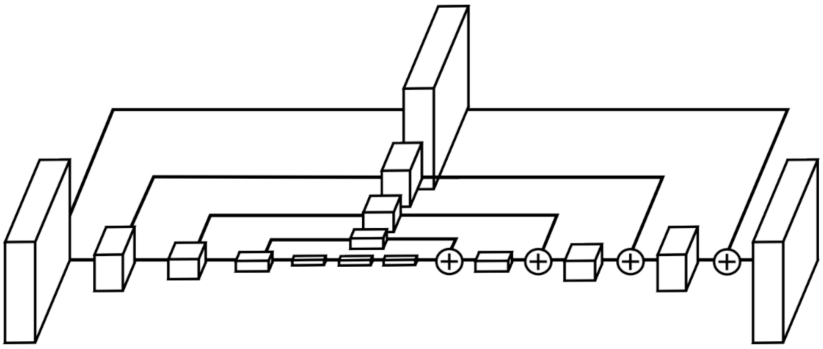


Figure 2.4: Representation of single Hourglass module. Each box is essentially a residual block consisting of 3 layers (128x1x1, 128x3x3, 256x1x1) and a skip connection. For step increasing resolution nearest neighbour upsampling is used. Element-wise addition is performed between corresponding upsampled version and the pre-pooled version of a residual block.

Pose Estimation and proved to be robust enough to conquer other domains - like HRNet [22, 24].

In human pose estimation, we can distinguish two branches of approaches: bottom-up and top-down. First ones are localizing identity-free keypoints and group them into person instances, while the latter is firstly performing person detection step and then regress keypoints within single person bounding box. Due to better performance on most of the human pose estimation benchmarks, we will focus primarily on top-down approaches - Hourglass [16], SimpleBaseline [25], HRNet [22], but also one single Bottom-up approach - Higher HRNet [6].

Within top-down approaches 2 categories can be distinguished:

1. regressing direct location of each keypoint [23]
2. keypoint heatmap estimation followed by the choosing the location with the highest heat values.

More recently most of the approaches [5, 16, 22, 25] follow heatmap estimation, as it proves to be more robust.

2.3.1. Stacked Hourglass Networks for Human Pose Estimation

Proposed approach (Figure 2.3), called stacked hourglass, is essentially a sequence of blocks which each is doing pooling and sampling. Process of pooling and up-sampling looks as the hourglass, hence the name. Design of hourglass is motivated by the need to capture information on all the scales. As the local position of the wrist or ankle is needed it is also important to capture overall context, like person orientation, the position of other limbs etc. These properties and more are well extracted by using different scales, where higher resolutions capture more general features, while smaller resolutions can extract more specific features.

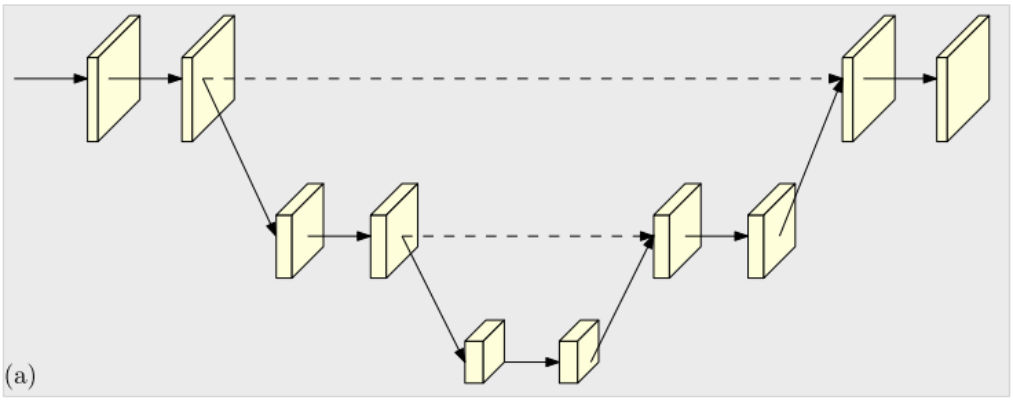
As stated before, the hourglass model is a sequence of modules following the same structure (Figure 2.4). Convolutional and max-pooling layers are processing features down to low resolution. At each max pooling step, the network has a fork on which additional convolutions are applied on the pre-pooled resolution. Once the network reaches the lowest resolution it starts to increase resolution via up-sampling and element-wise addition of up-sampled output and the aforementioned fork of the matching depth. The topology of the hourglass is symmetric, so for every layer decreasing resolution, there is a corresponding upsampling step.

To boost learning capabilities, the hourglass is using intermediate supervisions after each hourglass module, comparing the prediction of heatmaps to their regressed true position.

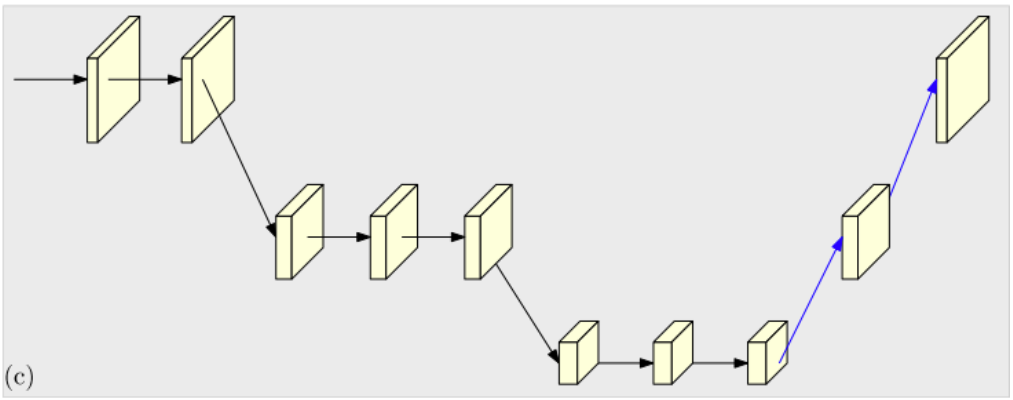
2.3.2. Simple Baseline for Human Pose Estimation and Tracking

This approach tries to solve the problem of human pose estimation with as simple as possible architecture and surprisingly outperforming more complicated previous architectures, including Hourglass. This network is consisting of ResNet architecture and few deconvolutional layers.

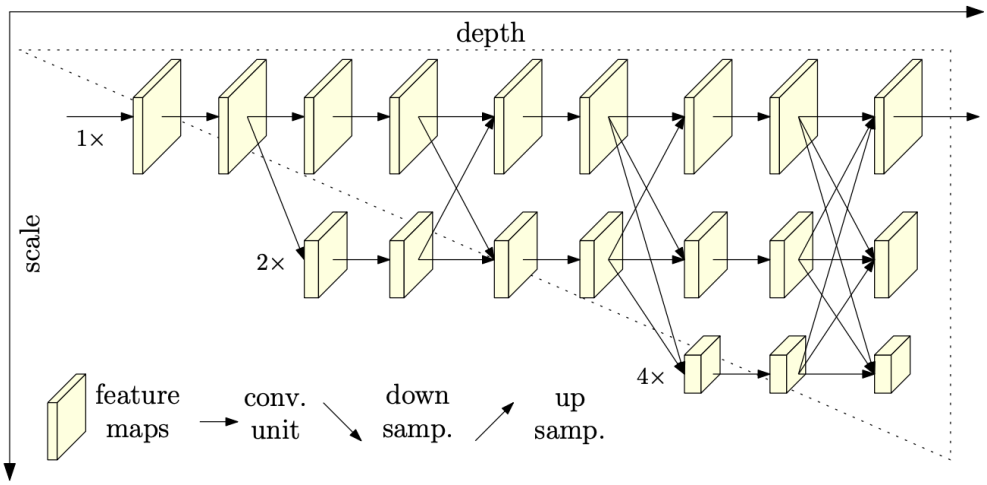
Ilustration of architecture of SimpleBaseline can be found in Figure 2.5b. In contrast to Hourglass (Figure 2.5a) which uses up-sampling to increase the resolution of the feature maps, this architecture uses transposed convolution layers, which enables to produce high-resolution features. Moreover, SimpleBaseline is not using skip connections between blocks. Lastly, Hourglass network is using multiple consecutive modules of a very same structure, while SimpleBaseline does not require to perform repeated decrease and increase in feature maps resolution.



(a) Hourglass [16]. Simple arrow is regular convolution, arrow down is strided convolution, dotted arrows correspond to skip connection between blocks and arrow up is upsampling.



(b) SimpleBaseline [25]. Simple arrow is regular convolution, arrow down is strided convolution and blue arrow corresponds to transposed convolutions.



(c) HRNet [22]. Simple arrow is convolution, arrow down is strided convolution, arrow up is upsampling.

Figure 2.5: Comparison between Hourglass, SimpleBaseline and HRNet from [22]. Legend: simple arrow = regular convolution, arrow down = downsample corresponds to strided convolution, arrow up = upsample, blue arrow up = transposed convolution, dotted arrow = skip connection between the blocks.

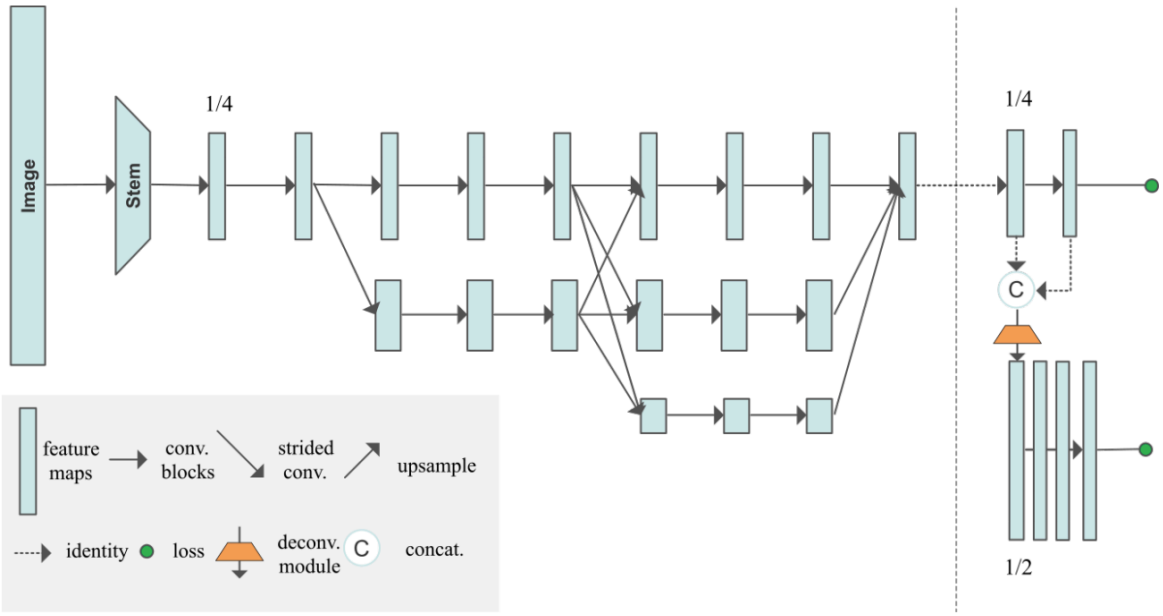


Figure 2.6: High-level view on Higher HRNet from [6]. On the left side, there is a visualization of HRNet backbone, producing two outputs 128x128 and 256x256 (on the right). After HRNet backbone additional two residual blocks are added to produce lower resolution output and deconvolution module with additional 4 residual modules (higher resolution).

2.3.3. HRNet

HRNet(High-resolution Net) outperformed all the previous solutions for Keypoint detection, multi-person pose estimation and pose estimation task on COCO dataset. Previous solutions were going from high->low-> high resolution, while this approach maintains high resolution throughout the whole process.

Model (Figure 2.5c) starts from a high-resolution subnetwork as the first stage and adds high to low-resolution subnetworks one by one to form more stages to connect the multi-resolution subnetworks in parallel. Repeated multi-scale fusions are conducted by exchanging information across parallel multi-resolution subnetworks over and over through the whole process. Then keypoints are estimated over high-resolution representations output by the presented network.

The architecture consists of a few types of subnetworks:

1. sequential multi-resolution subnetworks: each subnetwork, forming a stage is composed of a sequence of convolutions with down-sample layer across adjacent subnetworks to halve the resolution.

2. parallel multi-resolution networks: starting from high-resolution subnetwork as the first stage, by adding high to low-resolution subnetworks in parallel. This results in resolutions for the parallel subnetworks of a later stage consisting of the resolutions from the previous stage.

3. repeated multi-scale fusion: point of introducing exchange blocks is that subnetworks repeatedly receive the information from other parallel subnetworks.

4. heatmap estimation: heatmap regression from the high-resolution representation is performed from the output of the last exchange block.

2.3.4. Higher HRNet

Higher HRNet [6] is one of the best performing bottom-up approaches, which uses HRNet as its backbone architecture. Bottom-up approaches are said to perform better in estimation human pose within crowd instances. Model is providing 2 outputs - low (128x128) and high (256x256) resolutions.

While interference, both heatmaps are mean aggregated to higher resolution and highest valued points are chosen ad keypoint location. Deconvolution layer, which is represented as a trapezoid (Figure 2.6), outputs 2 times higher resolution, followed by 4 residual blocks.

For every keypoint, output scalar is calculated, with close values forming a group of keypoints belonging to single person instance, while distant tag values indicating different instances. Aforementioned tags are

calculated using ”Associative Embedding”. Based on empirical experiments, the tag values are only trained for lower resolution heatmap, as higher resolution do not learn to predict tags well enough.

Bibliography

[1] URL https://cs231n.github.io/convolutional-networks/.

[2] Alexandros Agapitos, Michael O'Neill, Miguel Nicolau, David Fagan, Ahmed Kattan, Anthony Brabazon, and Kathleen Curran. Deep evolution of image representations for handwritten digit recognition. pages 2452–2459, 05 2015. doi: 10.1109/CEC.2015.7257189.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *CoRR*, abs/1707.03718, 2017. URL http://arxiv.org/abs/1707.03718.

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *ArXiv*, abs/1603.07285, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

[9] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[10] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, abs/1611.09326, 2016. URL http://arxiv.org/abs/1611.09326.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] X. Li, S. Chen, X. Hu, and J. Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2677–2685, 2019.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[17] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. pages 4929–4937, 06 2016. doi: 10.1109/CVPR.2016.533.

[18] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL `http://arxiv.org/abs/1506.01497`.

[19] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[20] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

[22] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019.

[23] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013. URL `http://arxiv.org/abs/1312.4659`.

[24] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[25] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *CoRR*, abs/1804.06208, 2018. URL `http://arxiv.org/abs/1804.06208`.

[26] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions, 2015.