Efficient grouping for keypoint detection

Alexey Sidnev[1,2], Ekaterina Krasikova[1], Maxim Kazakov[1,3]

[1]Huawei Research Center, Nizhny Novgorod, Russia
[2]Lobachevsky State University of Nizhny Novgorod, Russia
[3]National Research University Higher School of Economics, Nizhny Novgorod, Russia
{sidnev.alexey, krasikova.ekaterina, kazakov.maxim}@huawei.com

Abstract—The success of deep neural networks in the traditional keypoint detection task encourages researchers to solve new problems and collect more complex datasets. The size of the DeepFashion2 dataset [1] poses a new challenge on the keypoint detection task, as it comprises 13 clothing categories that span a wide range of keypoints (294 in total). The direct prediction of all keypoints leads to huge memory consumption, slow training, and a slow inference time. This paper studies the keypoint grouping approach and how it affects the performance of the CenterNet [2] architecture. We propose a simple and efficient automatic grouping technique with a powerful post-processing method and apply it to the DeepFashion2 fashion landmark task and the MS COCO pose estimation task. This reduces memory consumption and processing time during inference by up to 19% and 30% respectively, and during the training stage by 28% and 26% respectively, without compromising accuracy.

I. INTRODUCTION

Recent research shows that keypoints, which are also known as landmarks, are one of the most distinctive and robust representations of visual fashion analysis. The class of keypoint-based methods in computer vision includes the detection and further processing of keypoints. They can be utilized for object detection, pose estimation, facial landmark recognition, and more.

The performance of keypoint detection models strongly depends on the number of unique keypoints defined in a task, and this could be problematic for large datasets. One of the newest fashion datasets DeepFashion2 provides annotations for 13 classes, each of which is characterized by a certain set of keypoints, totaling 294. Therefore, the straightforward prediction of all keypoints requires a heavy CNN architecture, leading to huge memory consumption, slow training, and low inference speed (see Figure 1), which restricts the application areas of such a model.

Authors [5], [3] use semantic information from the task to manually merge certain keypoints into one group. This helps train the model more quickly while also consuming less memory. However, the manual grouping may contain human errors and in many cases may not be optimal.

In this paper, we propose and study various techniques of automatic keypoint grouping which help improve the CNN model in terms of training speed, inference time, and memory consumption without compromising accuracy. Additionally, we study a powerful post-processing technique specifically designed to improve the accuracy of keypoint detection after
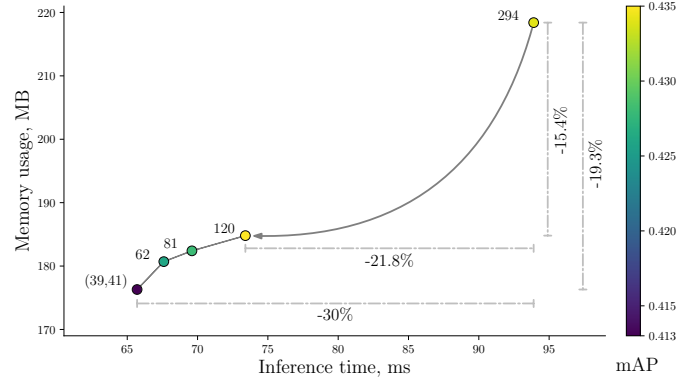

Fig. 1. The inference time and memory usage of the mDLA-34 model [3] on the DeepFashion2 dataset. Performance is shown for models with a direct prediction of 294 keypoints, and with groupings obtained through the proposed approach. Color scale represents the accuracy on the keypoint detection task. Performance is measured on Huawei Mate 20 (Kirin 980) with MNN v1.0.1 (number of threads = 4) [4].

grouping. We show the results of the clothing landmark detection task on the DeepFashion2 dataset [1] and those of the human pose estimation task on the MS COCO dataset [6].

II. RELATED WORK

In general, the keypoint detection task can be used in numerous application scenarios, such as to identify human poses [7], find facial landmarks [8], and estimate clothing landmarks [1], [9], [10].

DeepFashion2 [1] poses a new challenge on the keypoint detection task. It is a large-scale fashion dataset that contains consumer-commercial image pairs, and labels such as clothing attributes, landmarks, and segmentation masks. The public version of DeepFashion2 train set contains 191,961 rich images covering 13 popular clothing categories from both commercial shopping stores and consumers. It has about 320 thousand clothing items, and the number of keypoints for every category varies from 8 to 39, with 294 unique keypoints in total.

The keypoint detection task for multiple objects can be solved in a number of ways:

- Top-down: We first detect objects, and then estimate the keypoint position for each object [11], [12], [2].
- Bottom-up: We first detect all keypoints on the image, and then group the keypoints into objects [13], [14].

Output tensor	Number of channels	Channel type
Center heatmap	$C = 13$	Heatmap
Center offset	$2 = 1 \cdot 2$	Regression: $\triangle$x, $\triangle$y
Object size	$2 = 1 \cdot 2$	Regression: w, h
Keypoint regression	$588 = 294 \cdot 2$	Regression: $\triangle$x, $\triangle$y
Keypoint heatmap	$K = 294$	Heatmap
Keypoint offset	$2 = 1 \cdot 2$	Regression: $\triangle$x, $\triangle$y

Fig. 2. CenterNet architecture for keypoint detection.

Keypoint-based detection methods are becoming more popular in recent research as they are simpler, faster, and more accurate compared to anchor-based detectors. Previous approaches like [15], [16] require the manual designing of anchor boxes to train a detector. The subsequent approach involved a series of anchor-free object detectors, where the goal was to predict the keypoints of the bounding box, rather than trying to fit an object to an anchor. Law and Deng proposed a novel anchor-free framework CornerNet [17], which detects objects as a pair of corners. On each position of the feature map, class heatmaps, pair embeddings and corner offsets were predicted. Class heatmaps calculated the probabilities of pixels being corners of an object, and corner offsets were used to regress the corner locations while the pair embeddings grouped a pair of corners that belong to the same objects. Without relying on manually designed anchors to match objects, CornerNet significantly improved detection accuracy on the MS COCO dataset. Subsequently, there were several other variants of keypoint-based one-stage detectors, with one of them being CenterNet [2].

Methods that utilize hierarchical information to improve the accuracy and performance of the model were widely studied in the domain of image classification [18], [19], [20]. Some of them explicitly used hierarchical information or encoded the properties of such a class hierarchy into the probabilistic model, while others attempted to address severe mistakes by using graph distances in class hierarchies [21], [22], [23]. Visual hierarchies can be learned or used implicitly. The same general idea of leveraging external semantic information to improve performance may be applied to the keypoint detection task.

Authors [5], [3] use semantic information in the clothes landmark estimation task of the DeepFashion2 dataset to achieve grouping by manually merging certain keypoints into one group. Some clothing landmarks are evidently a subset of others. For example, shorts can be represented as a part of trousers; therefore, they do not need unique keypoints and can be merged into trousers. As such, we can formulate a rule for semantic grouping: the semantically identical keypoints (collar center, top sleeve edge, etc.) from different categories are merged into one group. This grouping method helps train the model faster, while using less memory.
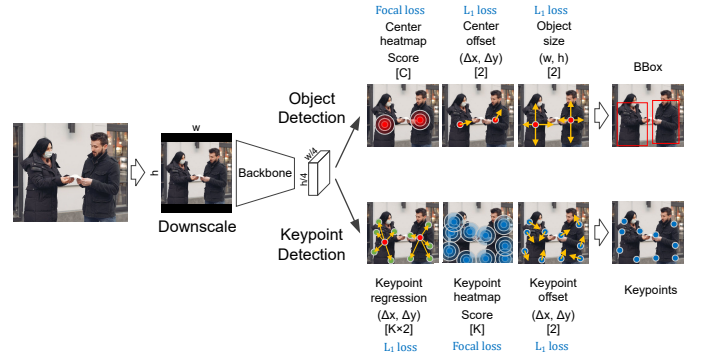
III. KEYPOINT GROUPING

A. CenterNet Architecture

CenterNet [2] architecture has proven to be effective for a wide range of tasks. In particular, it is able to carry out two tasks simultaneously: object detection and keypoint location estimation. Figure 2 illustrates this architecture, where a backbone network downsamples an image four times to generate a feature map, which is then processed to identify objects and corresponding keypoints. The proposed architecture has six output tensors, also called heads.

A center heatmap is used to predict the probability of each pixel being the object's center for each of C classes. The center of an object is defined as the center of a bounding box. A ground truth heatmap is generated by applying a Gaussian function at each object's center. Two additional channels in the output feature map $\triangle$x and $\triangle$y are used to refine the center coordinates, while both width and height are predicted directly.

Another branch handles keypoint estimation. This task involves estimating 2D keypoint locations for each object in one image. The coarse locations of the keypoints are regressed as relative displacements from the center of the box (keypoint regression in Figure 2). Consequently, if a certain pixel has already been classified as an object's center, you can take the values in the same spatial location from this tensor and interpret them as vectors to keypoints. Keypoint positions obtained through regression are not entirely accurate, and as such, an additional heatmap with probabilities is used for each keypoint type to refine the corresponding locations. Here, a local maximum with high confidence in the heatmap is used as a refined keypoint position. Like the detection case, two additional channels ($\triangle$x and $\triangle$y) are used to obtain more precise keypoint coordinates. During model inference, the location of each coarse keypoint is replaced with the closest refined keypoint position. In this way, we can group keypoints belonging to the same object.

During the training stage, CenterNet uses focal loss for heatmaps and L1 loss for each regression feature map. The loss function for keypoint regression is computed only for keypoints that are presented in the ground truth.

Encoder	Weights (MB)	Activations (MB)	Output tensors (MB)	Output tensors (%)
DLA-34 128×128	74.4	17.0	3.5	3.8
DLA-34 256×256	74.4	68.1	14.1	9.8
DLA-34 512×512	74.4	272.6	56.3	16.1
ResNet-50 128×128	115.2	16.9	3.5	2.7
ResNet-50 256×256	115.2	67.8	14.1	7.7
ResNet-50 512×512	115.2	271.1	56.3	14.5
Hourglass 128×128	743.4	42.4	3.5	0.4
Hourglass 256×256	743.4	169.5	14.1	1.5
Hourglass 512×512	743.4	677.9	56.3	4.0

Fig. 3. GPU memory consumption and training iteration time on RTX 2080ti. The input resolution is 256×256, the batch size is 32 for both DLA-34 and ResNet-50, and 8 for Hourglass. Time in ms was measured for one optimization step: batch loading to GPU, forward pass, and backward pass. GPU memory was measured by using the nvidia-smi tool. The image is reproduced with permission from [3].

B. Grouping Analysis

One of the first steps to overcoming the challenge in keypoint detection involves defining the model output. A straightforward approach is to concatenate keypoints from every category and deal with them separately. However, this is not the best solution because of the huge size of the model output. For instance, directly predicting 294 keypoints leads to a huge number of output channels: $901 = 13 + 2 + 2 + 588 + 294 + 2$. In this case, two tensors for keypoint detection occupy 97.9 % of output channels and computations.

To store FP32 output activations for an input resolution of 512×512, 56 MB is needed ($512/4 \cdot 512/4 \cdot 901 \cdot 4$). This contributes to over 20% of the total activation size for ResNet-50 and DLA-34. In Table II, we can assume that ReLU and BatchNorm operations do not occupy extra memory, while the last column shows how much memory is occupied by the output tensors as a percentage of the total memory consumption (activations + weights + input).

Memory consumption increases during neural network training (see Figure 3) because ground truth and loss computation are needed for every output channel. In CenterNet implementation, focal loss for keypoint centers requires 18 extra operations and potential memory allocation, while regression loss for keypoint offsets requires only two extra operations.

It is clear that certain clothing landmarks are a subset of others. Therefore, for example, shorts do not require unique keypoints because they can be represented by a subset of trouser keypoints. A manual grouping from [3] allows 62 groups to be formed and reduces the number of output channels from 901 to $205 = 13 + 2 + 2 + 124 + 62 + 2$. Figure 3 illustrates the findings of the experiments, and reveals that such an approach can reduce memory consumption (during the training stage) and training time by up to 28% and 26%, respectively. Th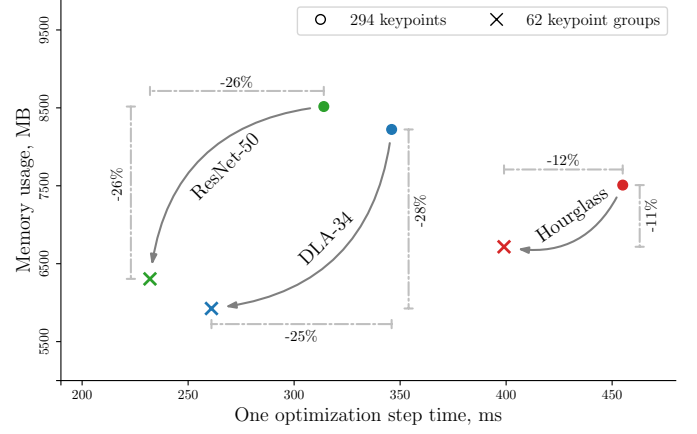eoretical evaluation (Table II) and experiment findings (Figure 3) reveal promising results for the keypoint grouping approach.

The key problem of manual grouping involves the human aspect. To maintain accuracy, a smart, automatic approach is necessary. An accurate grouping approach will benefit the generalization ability of the model as each group will receive more diverse training samples. Meanwhile, the same grouping can be used as a tool for solving the class imbalance problem.

C. Automatic Grouping Approach

We view the keypoint grouping task as a clustering problem, which is defined as follows: k_i is a unique keypoint type where $i = \overline{1, n}$; and g_j is a group label where $j = \overline{1, m}$. For the DeepFashion2 dataset, n is equal to 294; and for MS COCO Human Pose, n is equal to 17. The keypoint grouping task involves assigning a group label g_j for every keypoint type k_i.

The first thing we must determine to solve the clustering problem is the dissimilarity measure between different keypoint types. We propose a method that uses the following information about keypoints to measure distance:

- Ground truth location of keypoints.
- Weights of the last convolution layer.

Ground truth location of keypoints can be directly used to analyze the spatial location of keypoints. We evaluate each keypoint's offset from the center of an object, and then use this offset to estimate the mean location of every keypoint type. Finally, we measure the dissimilarity between two keypoint types by calculating the Euclidean distance between mean locations.

Such an approach is very simple but does not utilize information about the content of the input image. Moreover, this approach can only be used for solving keypoint regression task.
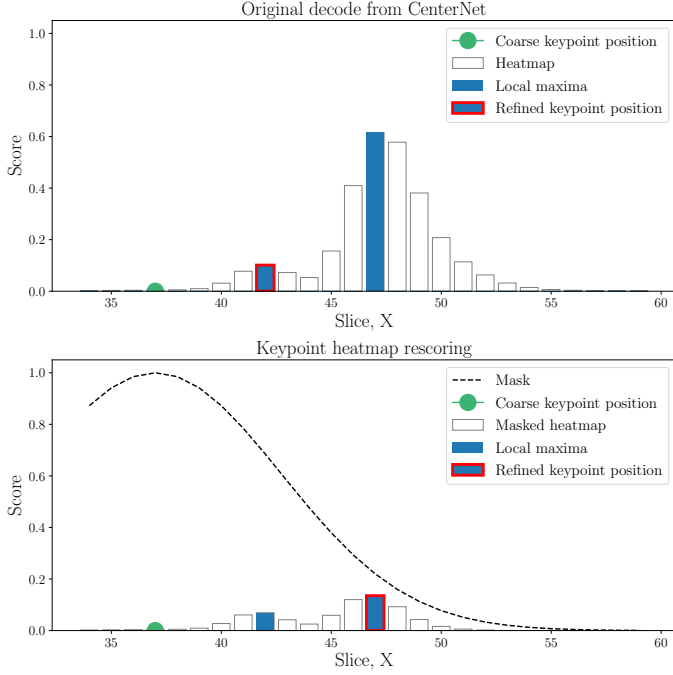
Fig. 4. In CenterNet, the refined location is defined as the closest point to the coarse keypoint position, which in turn is defined as a local maximum on the heatmap. This can lead to errors when the closest prediction is not actually the best one (as evident in the diagram above). In the proposed technique, the refined location is determined as a global maximum of the keypoint heatmap rescored by the Gaussian mask, improving keypoint localization. Image is reproduced with permission from [3].
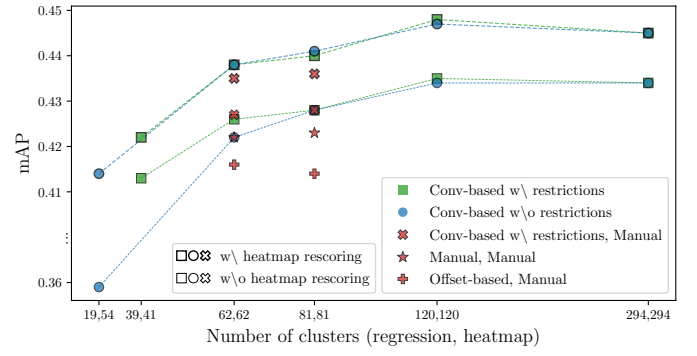

Fig. 5. Comparison of grouping strategies on training from scratch. Manual groupings from [5], [3] are used as-is and in combination with keypoint regression groupings: offsets-based and convolution-based (red points). Only the best results with heatmap rescoring for manual grouping are shown to avoid cluttering the chart.

In fully convolution neural networks like CenterNet [2], the last layer produces separate channels for each keypoint type. For the regression task CenterNet provides each keypoint type with two channels ($\triangle$x and $\triangle$y), while the heatmap-based branch uses only one channel. The last layer contains weights that are applied to the same activation feature map, and it produces results for different keypoint types. We follow the intuition that similar keypoints should have convolution weights that are almost the same, and use those weights to measure the Euclidean distance between different keypoint types.

The simplest way to implement keypoint grouping involves merging keypoints from different object classes into one group (for example, keypoints for shorts and trousers). This can be achieved by setting a large distance value between keypoints from the same class, and will be referred to as grouping with restrictions. Hierarchical clustering approaches can handle such keypoint merging well, and we use agglomerative clustering with linkage "average".

To achieve the keypoint detection task, CenterNet relies on two branches: keypoint location regression to estimate approximate keypoint locations, and keypoint heatmap to refine regressed locations. Given this, we can allow two keypoints from the same class (the same object) to be merged into a single cluster by regression, provided that they stay in different clusters by heatmap, and vice versa. In this way, we can decode the original keypoints from clusters unambiguously. This type

of grouping method will be referred to as grouping without restrictions.

We discovered that decoding substantially affects accuracy, and provide the following modification to keypoint refinement from [3].

By rescoring the keypoint heatmap, we propose adding a penalty to the keypoint score in proportion to the distance from a coarse keypoint position with Gaussian function. The final keypoint position is determined as a location of the maximum value in the rescored keypoint heatmap.

Assuming $mask$ is a heatmap with zero values by default, we set 1 into the $mask$ in the coarse keypoint position and fill neighbor values with 2D Gaussian function with standard deviation $sigma$ (see the second image in Figure 4). The parameter $sigma$ is estimated using a subset of validation dataset for every model.

The keypoint heatmap is rescored for each coarse keypoint position through the following formula:

$$\hat{H}_{kps} = H_{kps} \cdot mask \qquad (1)$$

This modified decoding is particularly important when we use grouping from the same class for keypoint regression. The more points from the same class we merge, the further away the coarse keypoint position is from the correct point on the heatmap, and the more likely the scenario in Figure 4 is to occur. As we show in the next section, decoding with keypoint heatmap rescoring allows keypoints from the same class to be merged without significantly affecting accuracy.

IV. EXPERIMENTS

A. Setup

To demonstrate the effectiveness of the proposed approach and investigate its properties, we conducted experiments on the DeepFashion2 dataset [1]. It provides annotations for 13 classes of clothing, with each class annotated by a unique set of keypoints (294 keypoints in total). We also present the results for human pose estimation on the MS COCO dataset [6].
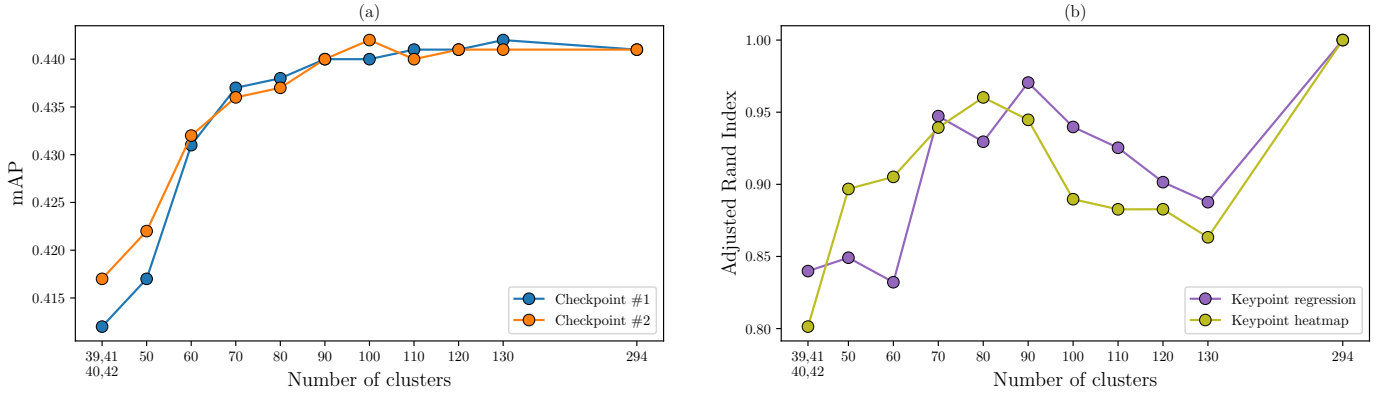
Fig. 6. Grouping robustness. Figure (a) presents keypoint detection accuracy achieved after fine-tuning the initial model for 15 epochs with corresponding grouping obtained from two different checkpoints. Figure (b) illustrates a consensus between two groupings from different checkpoints: one for keypoint regression, and the other for keypoint heatmap. In almost all cases, the same number of clusters for regression and heatmap are used. The leftmost point presents a minimal number of clusters.

TABLE III
DEEPFASHION2 KEYPOINT DETECTION ACCURACY WITH VARYING
NUMBERS OF CONVOLUTION CHANNELS BEFORE HEADS

Number of channels	256	64	32	16
mAP	0.401	0.412	**0.431**	0.423

All experiments are performed with an input image resolution of 256×256, using a batch of 64 images and Adam optimizer. Like the original CenterNet architecture, DLA-34 [24] is used as a backbone, but Deformable Convolution is replaced with conventional convolution layer for faster training. Backbone weights are initialized by the model pretrained on ImageNet.

An initial model was trained based on the following schedule. The first 35 epochs with a constant learning rate of 3e-3 were trained, and then a further 25 and 55 epochs for the DeepFashion2 dataset and the MS COCO dataset, respectively, were trained, with the learning rate decaying exponentially to 1e-5.

Since detection quality varies slightly from epoch to epoch, the model was trained for another 20 short epochs of 250 iterations with a constant learning rate of 1e-5. The best checkpoint on mini-val dataset was used for evaluation.

B. Convolution-based grouping with restrictions

In our experiments on the DeepFashion2 dataset, we use the modified loss function for learning the keypoint heatmap: supervision is present only at the keypoint heatmap's channels corresponding to the points present in the image. Therefore, we no longer penalize the network for non-zero values in the channels corresponding to other classes, which in turn enables us to obtain closer weights for similar points (collar of classes three and four, for example). In this way, we obtain weights that reflect the semantic proximity of keypoints from different classes.

Furthermore, using the proposed loss function allows us to attain a more accurate model (with any grouping), where the accuracy of the model with the original number of keypoints is 0.424 mAP and 0.434 mAP for the base and modified loss functions, respectively. Meanwhile, the accuracy of the model with manual grouping by 62 groups is 0.417 mAP and 0.422 mAP for the base and modified loss functions, respectively. The proposed loss function is only relevant for multi-class keypoint detection tasks such as DeepFashion2. Experiments on the single class MS COCO Human Pose dataset did not yield any noticeable increase in accuracy.

Additionally, we discovered that the number of channels in the convolution layers before heads significantly affects the quality of grouping approximation. Originally, 256 channels were used to achieve higher detection accuracy, while there were numerous channels. This means, convolution weights may consist of noise, and consequently affects keypoint approximation, ultimately leading to inefficient grouping. Conversely, a significantly small number of channels cannot be adequate for providing accurate keypoint representations. In Table III, quantitative results are presented for grouping $(60, 60)$ (the first value is a number of groups for keypoint regression, and the second is a number of groups for keypoint heatmap) after fine-tuning for 15 epochs. We found that 32 channels provide the most efficient grouping, and used this configuration for training. Note that this configuration is only used to train a checkpoint to gather convolution weights for keypoints clustering.

To train a model with grouping we considered these two strategies:

- Training from scratch with the parameters and schedule described in the beginning of this section;
- Fine-tuning from the original model for 15 epochs with a learning rate exponential decay from 4.0265e-4 to 1e-5. After keypoints are clustered, we cannot load head convolution weights directly. To deal with this issue we must initialize weights for each cluster g as $W_g = \frac{1}{|g|} \sum_{k \in g} W_k$, where k corresponds to the original keypoints.
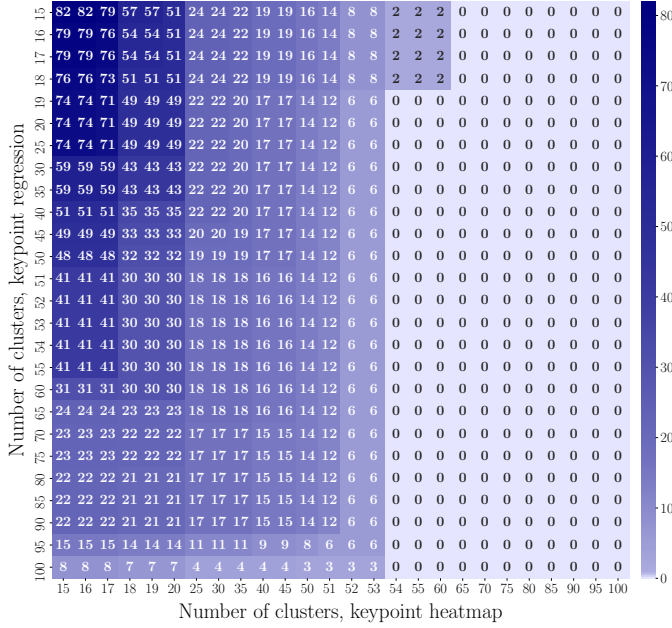
Fig. 7. Matrix of ambiguous decoding. Contains the number of keypoint pairs from the same class that was merged for regression and heatmap simultaneously.

Number of clusters	19,54	62,62	81,81	120,120
Keypoint regression	362	66	32	6
Keypoint heatmap	13	5	2	0

Note that in both strategies, the model was trained with the original 256 channels to achieve a high detection score.

Furthermore, we investigated the robustness of convolution-based grouping in terms of detection accuracy and clustering consensus. To achieve this, we evaluated various groupings from two independent checkpoints trained with the same parameters and schedule. To evaluate accuracy, we fine-tuned models with each grouping for 15 epochs from the same checkpoint. To measure the consensus between equivalent groupings, we used the Adjusted Rand Index [25], and the corresponding results are presented in Figure 6.

While at a lower number of clusters, one can observe variations in accuracy and keypoint partitioning, and groupings of 70 clusters become robust and able to achieve accuracy comparable to the original training. A larger number causes the clusters to become very dense and small, and further partitioning is affected by variations in convolutional weights. Therefore, the clustering consensus experiences a decrease, but this should not affect the effectiveness of grouping, which is supported by robust behavior in detection accuracy. As such, the consensus measure can be used as a criterion to choose the number of clusters.

Finally, we trained the models with groupings from scratch and compared different grouping strategies, and the results are presented in Figure 5. Grouping through the proposed approach is slightly more effective than manual grouping. This improvement is driven mostly by the effective grouping for regression head since manual grouping has already been designed in a heatmap manner. At the same time, straightforward clustering based on offsets provides less effective grouping which stresses the importance of keypoint approximation for

clustering. The heatmap rescoring technique increases accuracy in all cases. This is particularly true when the high accuracy gain of 6 mAP is achieved for a very small number of clusters. The best model with 120 clusters achieves slightly better accuracy compared to the naive approach.

We also trained lighter models – DLA-34 with an input image resolution of 128×128 and MobileNet_v2 [26] – to gather weights and perform clustering. We then used groupings $(62, 62)$ and $(81, 81)$ to train our standard model and achieved a detection accuracy very close to our predictions. Moreover, we used groupings $(62, 62)$ to train model DLA-34 with an input image resolution of 512×512 and achieved accuracy comparable to the original 294 keypoints model, yet almost twice as fast while using 1.5 times less GPU memory for training. These results demonstrate that the proposed grouping approach can be used on models that are small and geared towards training time to perform clustering. The grouping can then be leveraged in the heavy model to accelerate the training process without compromising accuracy.

C. Convolution-based grouping without restrictions

The accuracy results of the models that implement convolution-based grouping without restrictions on the keypoint detection task of the DeepFashion2 dataset are presented in Figure 5. The permitted number of clusters for each output is determined by analyzing a matrix of ambiguous decoding that contains the number of pairs from the same class that was merged for regression and heatmap simultaneously (Figure 7). The minimum grouping includes 19 and 54 clusters for the keypoint regression and keypoint heatmap, respectively. The number of inconsistent pairs (pairs of keypoints corresponding to the same object category that were merged into a single cluster) is shown in Table IV. As the number of clusters decreases, the number of inconsistent pairs for the keypoint regression increases, which leads to a drop in accuracy relative to grouping with restrictions. As shown in Figure 5, the keypoint heatmap rescoring technique prevents this drop from happening.

Table V shows the accuracy results for the keypont detection task on the MS COCO Human Pose dataset of the models that implement convolution-based grouping without restrictions. In these experiments, groupings were performed either for the keypoint regression or the keypoint heatmap. Table V also shows that the keypoint heatmap rescoring technique can significantly reduce the keypoint detection accuracy drop caused by a considerably smaller number of keypoint regression groups. Our best grouping with 14 clusters achieves slightly better accuracy than the model with a full set of keypoints.

TABLE V
MS COCO Human Pose accuracy, AP

Number of clusters		10	14	17
Keypoint regression, conv-based grouping	Base decode	0.322	0.349	0.359
	Rescoring technique	0.360	0.369	0.368
Keypoint heatmap, base decode	Conv-based grouping	0.281	0.334	0.359
	Anti-offsets grouping	0.356	0.356	0.359

Let us analyze the results obtained for models with grouping for the keypoint heatmap. If we merge the close points from the same class into one cluster, they will probably be in close proximity on the heatmap. Close points on the same heatmap reduce accuracy; therefore, if we change the grouping strategy and merge distant points in one group will accuracy increase? To obtain the specified grouping we used clusterization by referring to the negative distances between the average offsets of keypoints from the objects' centers from annotations (agglomerative clustering with "complete" linkage was used), which will be referred to as "anti-offsets" grouping. As shown in Table V, this strategy allows us to obtain models with grouping for the keypoint heatmap featuring accuracy close to that of the model with a full set of keypoints.

We applied 14 groups for keypoint regression to the human pose estimation model provided by the authors of Center-Net [2]. We fine-tuned the original DLA-34 512×512 model (with Deformable Convolution) for 15 epochs using the batch of 16, with a learning rate exponential decay equal to 0.8 from 1.008125e-4. The models have been tested with original image resolution and flip testing. The accuracy of the model from the CenterNet paper is 0.589 AP with base decoding, and 0.596 AP with the keypoint heatmap rescoring technique; whereas the accuracy of the obtained model is 0.579 AP with base decoding, and 0.592 AP with the keypoint heatmap rescoring technique.

To obtain convolutional weights for grouping, we trained the full network with six different outputs (heads) and losses. However, this might not have been necessary and we can obtain good keypoint representation by training only one head – regression or heatmap for corresponding grouping. We trained two models with one head by utilizing the same parameters we used to train the full network. With groupings obtained from those weights, we were able to achieve a detection accuracy that is lower by only 1.5 mAP compared to grouping from the full model. Meanwhile, the chosen values of training parameters are not optimal for models with only one head, which resulted in over-fitting. To improve the results, more appropriate parameter values can be used.

V. Conclusion

We have shown that the proposed automatic grouping approach with the special post-processing technique works with the CenterNet architecture on human pose estimation and fashion landmark detection tasks. It boosts training speed, accelerates inference time, and reduces memory consumption without compromising accuracy.

The proposed grouping approach can be used on models that are small and geared towards training time to perform clustering, and then the grouping can be used in the heavy model to accelerate the training process.

Heatmap is the standard coordinate representation in keypoint detection tasks [27]. For this reason, the proposed approach can also be applied to almost any keypoint detection architecture, for example, HRNet [28], PoseFix [29], and Mask R-CNN [12].

References

[1] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5337–5345.

[2] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019. [Online]. Available: http://arxiv.org/abs/1904.07850

[3] A. Sidnev, A. Krapivin, A. Trushkov, E. Krasikova, M. Kazakov, and M. Viryasov, "Deepmark++: Real-time clothing detection at the edge," *CoRR*, vol. abs/2006.00710, 2020. [Online]. Available: https://arxiv.org/abs/2006.00710

[4] X. Jiang, H. Wang, Y. Chen, Z. Wu, L. Wang, B. Zou, Y. Yang, Z. Cui, Y. Cai, T. Yu, C. Lv, and Z. Wu, "Mnn: A universal and efficient inference engine," in *MLSys*, 2020.

[5] T. Lin, "Aggregation and finetuning for clothes landmark detection," *CoRR*, vol. abs/2005.00419, 2020. [Online]. Available: https://arxiv.org/abs/2005.00419

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[7] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016.

[8] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European conference on computer vision*. Springer, 2014, pp. 94–108.

[9] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.

[10] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "Modanet: A large-scale street fashion dataset with polygon annotations," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1670–1678.

[11] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[13] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[14] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4929–4937.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[17] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[18] M. Marszalek and C. Schmid, "Semantic hierarchies for visual object recognition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7.

[19] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *European conference on computer vision*. Springer, 2014, pp. 48–64.

[20] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1114–1123.

[21] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair, "Learning hierarchical similarity metrics," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2280–2287.

[22] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" in *European conference on computer vision*. Springer, 2010, pp. 71–84.

[23] B. Zhao, F. Li, and E. P. Xing, "Large-scale category structure aware image categorization," in *Advances in Neural Information Processing Systems*, 2011, pp. 1251–1259.

[24] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[25] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[27] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," *arXiv preprint arXiv:1910.06278*, 2019.

[28] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[29] G. Moon, J. Y. Chang, and K. M. Lee, "Posefix: Model-agnostic general human pose refinement network," 2018.