Southern California Robotics Symposium
SCR 2019

Weakly supervised keypoint detection

Mo Shan[1]

***Abstract*— Keypoint detection using convolutional neural networks (CNNs) requires a large amount of annotations that are time consuming and labor intensive. In this work, it is shown that CNNs could merely rely on class labels to categorize images and locate the keypoints simultaneously. Specifically, keypoints are detected in a multiscale framework based on the relevance of features and high activations. The performance of the proposed pipeline is analyzed qualitatively.**

I. Introduction

Keypoint detection is a critical step in semantic scene understanding for autonomous driving, such as object detection and 3D reconstruction of cars. Hand-crafted keypoint detectors often lack the ability to maintain semantic consistency, whereas convolutional neural networks (CNNs) that have gained popularity for various recognition tasks could tackle semantic keypoint detection better. However, training CNNs usually require a massive amount of labeled data with high diversity to prevent over-fitting, and it is quite challenging to ensure the annotated keypoints are semantically consistent in different object instances.

Manual keypoint annotation suffers from several issues. First and foremost, the annotation process is quite time consuming and labor intensive, as an object often possesses many keypoints. Secondly, the labeling is subjective and the position of the keypoints are not well defined. Thirdly, it is problematic to ensure that the keypoints are semantically consistent for varied instances in a class. As a result, the number of annotated keypoints is often limited. For instance, there are only 14 keypoints for each car image in the popular PASCAL VOC dataset. Although it is possible to take advantage of 3D CAD models in ShapeNet to generate overfit-resistant training set for viewpoint prediction [1], how to define and locate the keypoints on those models for different object instances in a semantically consistent way still remains a challenging issue. A question arises naturally: is it really necessary to label each keypoint for CNNs?

In this work, it is found that keypoint annotations may not be necessary, as class labels could provide weak supervision that is sufficient for CNNs to figure out the locations of the important features in the image that are vital for accurate classification. The proposed pipeline is shown in Fig. 1. A model pretrained on classification task is used to detect the keypoints, which correspond to the features that are most relevant to the activations. A multiscale framework is formulated, in which the likelihoods of keypoints at coarse scale and fine scale are combined. The features with highest probability are refined with sub-pixel accuracy to locate the keypoints. To the best of our knowledge, the method proposed in this paper is the first work on weakly supervised keypoint detection, which neither requires keypoint annotations for training nor ground-truth class labels during testing.

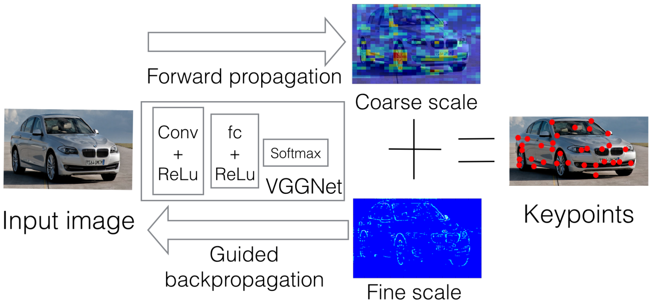

Fig. 1. Overview of weakly supervised keypoint detection pipeline. The input image is fed to a pretrained network on classification, with an occluder to obtain the coarse scale heatmap, and then guided backpropagation is performed to get the fine scale heatmap. The results are combined to produce the log-likelihood distribution for keypoints.

The rest of the paper is organized as follows: Section II reviews the previous works; Section III illustrates the pipeline of weakly supervised keypoint detection, and Section IV presents qualitative analysis; Section V contains the discussion.

II. Literature review

An early work [2] proposes a convolutional keypoint detection pipeline for generic objects. The viewpoint and keypoints are predicted using models adapted from VGGNet [3], and the viewpoint conditioned as well as the appearance based log likelihood keypoint distributions are combined to produce the resulting heatmaps. A multiscale structure is designed for keypoint detection model to strike a balance between accuracy and robustness. However, the number of keypoints is sparse since it is limited by the available annotations in the training set. Meanwhile, the ground-truth class labels have to be provided during testing. Instead of detecting class dependent keypoints, [4] investigates how to use CNNs for generic feature detection, by proposing a network that includes detection, orientation estimation and feature description. Nevertheless, the training relies on SIFT features.

[1] Existential Robotics Laboratory, University of California, San Diego, moshan@eng.ucsd.edu

Regarding semantic keypoint detection, [5] proposes a model capable of localizing landmarks for articulated objects such as faces and birds. The network uses modified VGGNet to extract features, which are fed into the Shape Basis Network to express the object shape as a linear combination of shapes bases, which act as global geometric priors. Next, the coarse shape is refined in the Point Transformer Network using thin-plate spline transformations to deal with pose variations. The training set also contains manually labeled landmark annotations.

Prior works [6], [7], [8], [9], [10] on weakly supervised localization have demonstrated that object detectors emerge in CNNs trained merely using class labels. Nevertheless, these works mainly focus on detecting and locating objects instead of keypoints.

III. Weakly supervised keypoint detection

At coarse scale, the contribution of each patch in the input image for object classification is analyzed by covering it and examining the change in the confidence of class prediction similar to [11]. If the confidence of the correct class drops dramatically due to the occlusion of a patch, then the probability of the patch containing a discriminative feature is very high.

Following the notation in [7], the network is denoted by a mapping $f : \mathbb{R}^N \mapsto \mathbb{R}^C$, $x \in \mathbb{R}^N$, $y \in \mathbb{R}^C$, where x in an image of N pixels, and $y = [y_1, ..., y_C]^T$ denotes the classification score of C classes, with y_i being the probability of the i th class. The pixels inside an occluder b of image x are replaced by a vector g, and this occlusion function is denoted by h_g. Hence the change in classification score is $\delta_f(x, b) = max(f(x) - f(h_g(x, b)), 0)$. To avoid creating edges, random colors are used as g instead of mono color, which is advisable according to [6]. Since only the class with maximum probability is considered, the decrease of score is $d(x, b) = \delta_f(x, b)^T \mathbb{I}^C$, where $\mathbb{I}^C \in \mathbb{N}^C$ is an indicator vector whose elements are zero except at the predicted class c.

One may argue that the occlusion could be conducted at every pixel, such that it measures the discriminativeness at a fine scale. The reasons for not using the occlusion densely are twofold. Firstly, it is time consuming to do forward passes repeatedly for all patches, especially for deep CNNs. Secondly, the change in activations is induced at patch level, and thus it is difficult to locate the exact feature within the patch.

For the fine scale, guided backpropagation [12] is performed on the unit that has maximum activation, whose results reflect the effect of the input image at pixel level. In other words, guided back-propagation from the softmax layer reveals which pixel positively influences the class prediction, by maximizing the probability of the predicted class while minimizing that of other classes. During back-propagation, the gradient of the predicted class with respect to the input is computed, which locates the pixel where the least modification has to be made in order to affect the prediction the most. The activation at layer $l + 1$ could be obtained from the activation at layer l through a ReLU unit as $f_i{}^{l+1} = ReLU(f_i{}^l) = max(f_i{}^l, 0)$. The back-propagation is $R_i{}^l = (f_i{}^l > 0) \cdot R_i{}^{l+1}$, where $R_i{}^{l+1} = \frac{\partial f^{out}}{\partial f_i{}^{l+1}}$. For guided back-propagation, not only the input is positive, but also the error, i.e. $R_i{}^l = (f_i{}^l > 0) \cdot (R_i{}^{l+1} > 0) \cdot R_i{}^{l+1}$. In this way the error is guided both by the input as well as the error. The coarse scale and fine scale are combined linearly as in [2], where sigmoid functions are used to transform the heatmaps into log-likelihood keypoint distributions. This values is used as the confidence score. Note that unlike [2] which needs ground-truth class labels both during training and testing, the class labels are not used in the proposed pipeline since the class with highest activations is considered.

After the log-likelihood map is obtained, Non Maximum Suppression is performed to prune the nearby keypoints. For each keypoint, the subpixel coordinates are determined using the $F\ddot{o}rstner$ operator [13] by solving a least squares solution for $Ax = b$, i.e. $\hat{x} = A^{-1}b$, where $x, \hat{x}$ are the original keypoints and keypoints with sub-pixel accuracy, w is the window about the pixel, whose size is very small, and I_x, I_y are the gradient images in the x and y direction. A, b are given by

$$A = \begin{bmatrix} \sum\limits_w I_x^2 & \sum\limits_w I_x I_y \\ \sum\limits_w I_x I_y & \sum\limits_w I_y^2 \end{bmatrix}, b = \begin{bmatrix} \sum\limits_w (I_x^2 x + I_x I_y y) \\ \sum\limits_w (I_x I_y x + I_y^2 y) \end{bmatrix}$$

IV. Experiment

The keypoint detection framework is implementated using FeatureVis library [14], and the pretrained model is VGG Net-E [3] provided in [15]. An alternative model is ResNet, but it does not seem to provide significant advantage over VGG Net, by comparing the correspondence between the patches that correspond to high activation changes and the groundtruth landmarks, and the range and standard deviation of the values for activation changes. Moreover, different patch sizes for occlusion are also compared, including 8×8 with stride size 8, 16×16 with stride size 8, 16×16 with stride size 16, and 32×32 with stride size 16. It is found that using smaller patch size produces more localized results, albeit the change in activation is smaller. To trade off the localization accuracy and the variation of activations, 16×16 and a stride size of 8 is chosen. To summarize, VGG Net is used with a patch size of 16×16 and a stride size of 8 in all the experiments.

The dataset in [16] is used for evaluation purposes, where 64 landmarks are manually selected to cover salient features for car images. Since the groundtruth bounding box is not provided, they are manually defined by using the positions of the top left and bottom right landmark annotations that are visible, plus a margin to ensure that the entire body is located inside the bounding box.

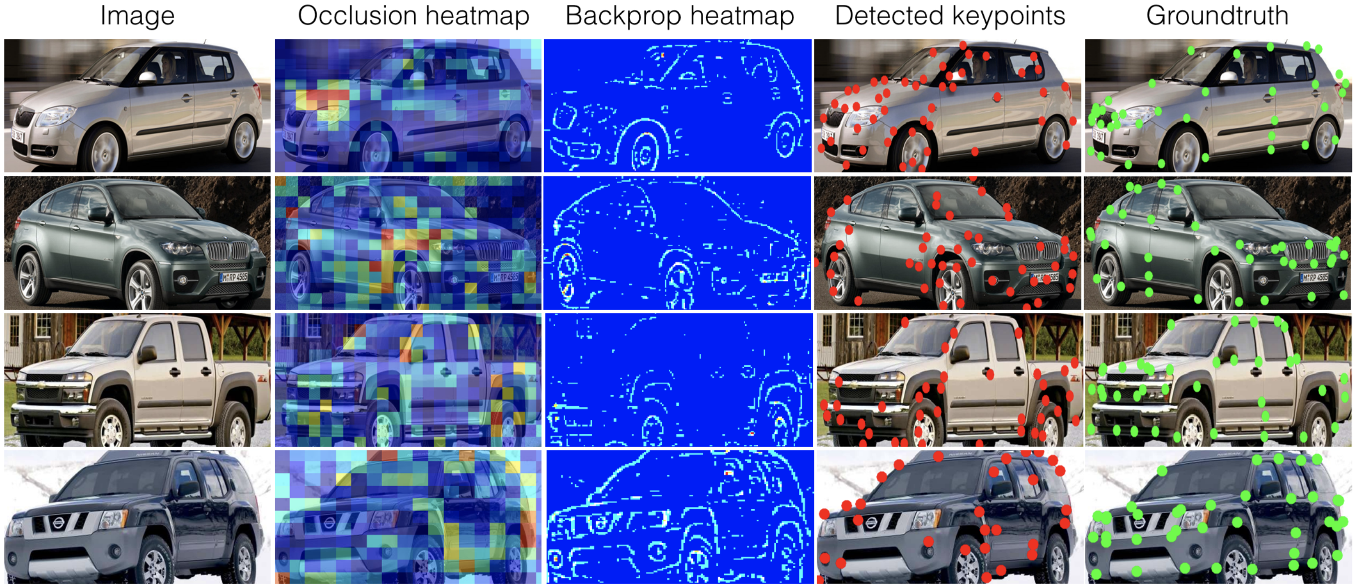

Fig. 2. Keypoints and contribution heatmaps. First column: input images. Second column: heatmaps that indicate the importance of different patches for classifying the image using VGG Net-E. The warmer the color, the larger the change in activations when the patch is covered. Third column: heatmaps of guided backpropagation. The warmer the color, the larger the gradient. Fourth column: detected keypoints using the proposed framework marked in red. Fifth column: keypoints annotations [16] marked in green. Best viewed electronically.

A. *Keypoint prediction*

The proposed keypoint detection is evaluated qualitatively in this section. A soft threshold is used for pruning the keypoints, where only the top 40% based on confidence score is kept. The car images used in Fig. 2 from first to last rows are Skoda Fabia 2007, BMW X6 2009, Chevrolet Colorado-LS 2004, Nissan Xterra 2005, where the car size increases.

As can be seen from Fig. 2, the most important patches are usually those centered around the keypoints, such as those near the rear view mirrors, head lights as well as the wheels, which are semantically consistent. Occluding these patches with a randomly colored square box leads to significant drop in activations, which indicates that regions close to the keypoints are critical for classification. Moreover, the rear view mirrors as well as car logos are always highlighted in the gradient images from guided back-propagation, which confirms the close relevance of keypoints and high activations. From the last two columns, the keypoints detected using weak supervision is comparable to the ground-truth annotations. For instance, the positions of the detected keypoints on the car logos and rear view mirrors are almost identical to the ground-truth, which demonstrates the effectiveness of the proposed framework without relying on manual annotations.

B. *Salient feature prediction*

The proposed keypoint detection could also be used to predict which features are salient. The detected keypoints are compared with the visible landmarks in each image contained in the dataset of [16], which has 30 different models and 300 images in total. PCK (probability of correct keypoint) is used as the metric to determine whether a landmark is also considered as a salient keypoint, which means that a landmark is found if there is a predicted keypoint that lies within $\alpha max(h, w)$ of it, where h, w is the size of the bounding box, and $\alpha = 0.2$.

A histogram could be built based on the occurrence of the detected landmarks for each car, and the ones that appear most frequently are reported in Table I. The landmark index and the location of the landmarks are also indicated. The landmark locations are roughly divided into several salient parts, including bonnet, headlight, windscreen, wheel, and the rest of the body. Since the car is symmetric, there is no distinction for left and right corresponding parts.

From Table I, it could be observed that the salient landmarks are still different for different models, even though the make is the same. In addition, headlight remains salient across different makes and models. Moreover, the most frequently detected landmarks for all the cars is 52, which is located at the right part of the bonnet. This indicates that the bonnet is important for CNNs to make classifications. To summarize, the car manufactures could consider focusing more on the design of headlight and bonnet to make their cars stand out from others.

V. Discussion

This work tackles the keypoint detection task, which is vital for semantic scene analysis for self driving cars. A weakly supervised keypoint detection pipeline is proposed to deal with the keypoint scarcity in the training set induced by the time consuming annotation process. The heatmaps produced from occlusion method and guided back-propagation are

Car	Acura ZDX	Alfa 159	Audi Q7	BMW 5-Series	BMW X6
Feature	50, Body	132, Body	52, Bonnet	42, Body	182, Headlight
Car	Chevrolet Colorado-LS	Dodge Ram	Ford F 150	Ford Mondeo	Honda CR-V
Feature	1, Wheel	179, Body	30, Windscreen	158, Body	54, Body
Car	Honda Odyssey	Honda Pilot	Jeep Commander	Lexus LS460	Mazda 6-US-spec
Feature	51, Headlight	29, Bonnet	158, Body	50, Body	30, Windscreen
Car	Mazda 6-Wagon	Mercedes-Benz CL-600	Mercedes-Benz GL450	Nissan Titan	Nissan Xterra
Feature	49, Body	145, Body	53, Headlight	158, Body	52, Bonnet
Car	Opel Corsa	Saab 93	Skoda Fabia	Skoda Octavia	Toyota Corolla
Feature	30, Windscreen	52, Bonnet	181, Headlight	42, Body	30, Windscreen
Car	Toyota Prius	Toyota Yaris	Vauxhall Zafira	Volkswagen Golf-GTI	Volvo V70
Feature	17, Body	42, Body	53, Headlight	17, Body	54, Headlight

TABLE I

SALIENT FEATURE FOR EACH TYPE OF CAR IN DATASET OF [16].

unified in a multiscale framework to produce a log-likelihood distribution of keypoints. The proposed method does not require full supervision during training and the ground-truth class labels are unnecessary for testing. The effectiveness of the proposed framework are demonstrated qualitatively on keypoint prediction and salient feature prediction using car images.

For future work, viewpoint annotations may be a more effective supervision than class labels. The viewpoint estimation is formulated as a fine-grained classification task using real and synthetic images in [1]. As a follow up work, [17] integrates classification and viewpoint estimation in a unified framework. Hence it is interesting to explore the additional benefits brought by viewpoint annotations compared with merely providing class labels, by using the models trained in [17].

REFERENCES

[1] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2686–2694.

[2] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1510–1519.

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

[4] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483.

[5] X. Yu, F. Zhou, and M. Chandraker, "Deep deformation network for object landmark localization," in *European Conference on Computer Vision*. Springer, 2016, pp. 52–70.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[7] A. Bergamo, L. Bazzani, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," *arXiv preprint arXiv:1409.3964*, vol. 2, 2014.

[8] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 685–694.

[9] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[13] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, 1987, pp. 281–305.

[14] N. N. F. T. Felix Grün, Christian Rupprecht, "A taxonomy and library for visualizing learned features in convolutional neural networks," in *ICML Visualization for Deep Learning Workshop*, 2016. [Online]. Available: http://arxiv.org/abs/1606.07757

[15] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692.

[16] Y.-L. Lin, V. I. Morariu, W. H. Hsu, and L. S. Davis, "Jointly optimizing 3d model fitting and fine-grained classification." Citeseer.

[17] F. Massa, R. Marlet, and M. Aubry, "Crafting a multi-task cnn for viewpoint estimation," *arXiv preprint arXiv:1609.03894*, 2016.