Weakly supervised keypoint detection

Mo Shan[1]

[1]Existential Robotics Laboratory, University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

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.

Motivation

1. Manual keypoint annotation process is quite time consuming and labor intensive.
2. The labeling is subjective and the position of the keypoints are not well defined.
3. Hard to ensure that the keypoints are semantically consistent for varied instances in a class.
4. A question arises naturally: is it really necessary to label each keypoint for CNNs?

Overview

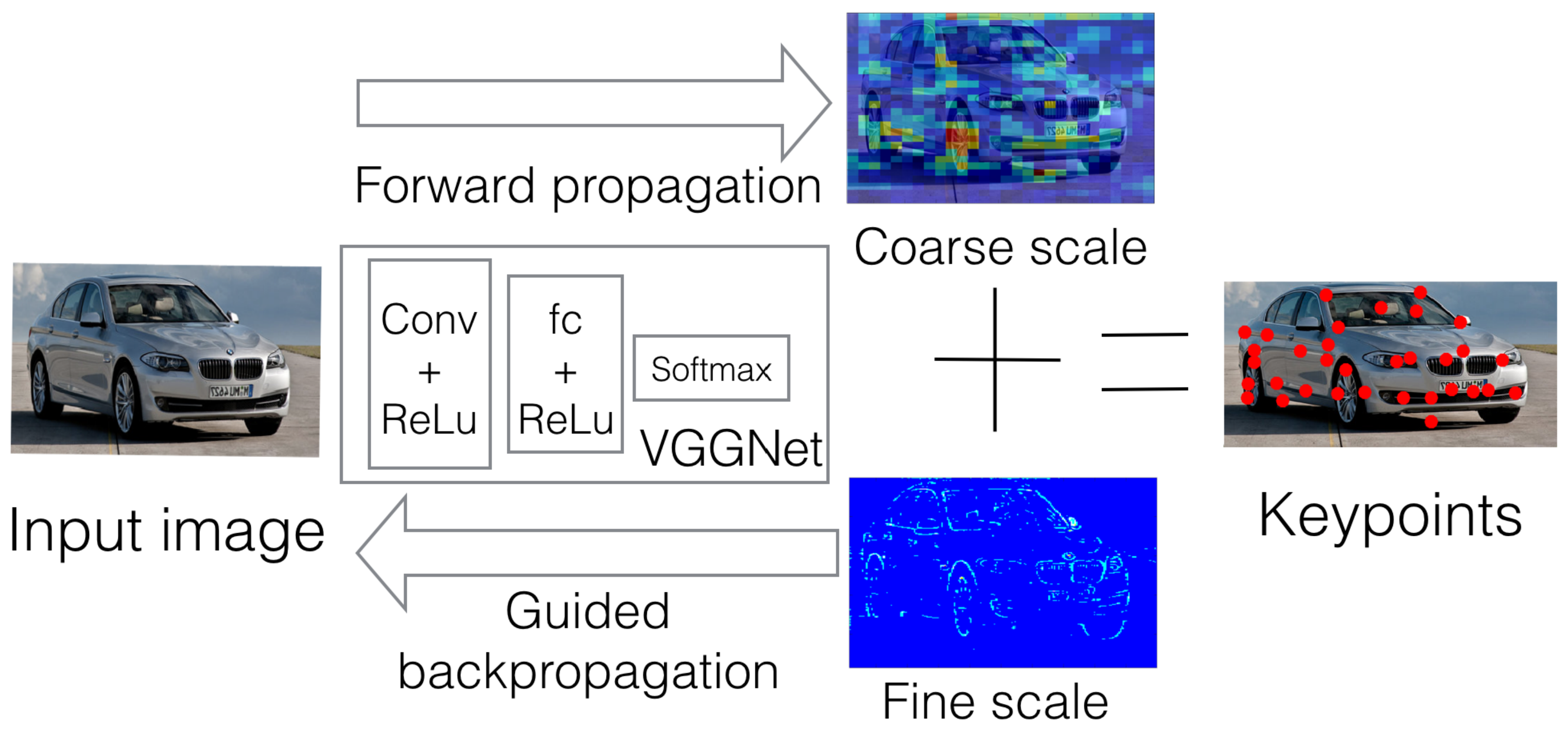

Figure 1: Overview of the proposed approach.

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. An overview is shown in Fig. 1.

Coarse-scale 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 [2]. 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.

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. 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.

Fine-scale keypoint detection

For the fine scale, guided backpropagation [1] 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 backpropagation 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, where sigmoid functions are used to transform the heatmaps into log-likelihood keypoint distributions. This values is the confidence score.

Post-processing

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 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

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

Results

VGG Net is used with a patch size of 16×16 and a stride size of 8 in all the experiments.

Keypoint prediction

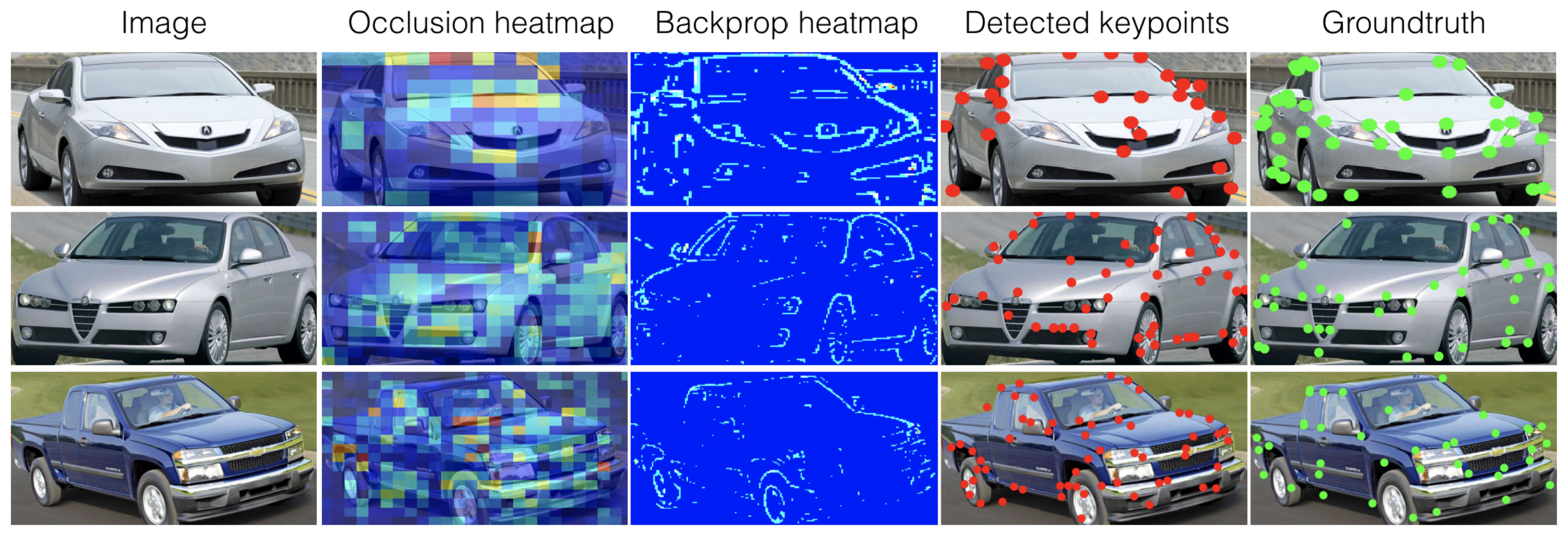

Figure 2: Keypoints and contribution heatmaps. First column: input images. Second column: heatmaps that indicate the importance of different patches for classifying the image using VGG Net-E. The warmer the color, the larger the change in activations when the patch is covered. Third column: heatmaps 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 marked in green.

1. Only the top 40% based on confidence score is kept.
2. 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.
3. 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.

Salient feature prediction

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

Figure 3: Salient feature for each type of cars.

1. 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 3.
2. 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.

Key insights

1. Keypoint detection using CNNs could be achieved via weak supervision, using classification as an auxiliary task
2. For future work, viewpoint annotations may be a more effective supervision than class labels.

References

[1] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[2] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Acknowledgements

I would like to thank Prof. Nikolay Atanasov and Prof. Manmohan Chandraker for their kind support and insightful advises.