Visualization of Convolutional Neural Networks for Monocular Depth Estimation

Junjie Hu[1,2] Yan Zhang[2] Takayuki Okatani[1,2]
[1] Graduate School of Information Sciences, Tohoku University, Japan
[2] Center for Advanced Intelligence Project, RIKEN, Japan
{junjie.hu, zhang, okatani}@vision.is.tohoku.ac.jp

Abstract

Recently, convolutional neural networks (CNNs) have shown great success on the task of monocular depth estimation. A fundamental yet unanswered question is: how CNNs can infer depth from a single image. Toward answering this question, we consider visualization of inference of a CNN by identifying relevant pixels of an input image to depth estimation. We formulate it as an optimization problem of identifying the smallest number of image pixels from which the CNN can estimate a depth map with the minimum difference from the estimate from the entire image. To cope with a difficulty with optimization through a deep CNN, we propose to use another network to predict those relevant image pixels in a forward computation. In our experiments, we first show the effectiveness of this approach, and then apply it to different depth estimation networks on indoor and outdoor scene datasets. The results provide several findings that help exploration of the above question.

1. Introduction

Enabling computers to perceive depth from monocular images has attracted a lot of attention over the past decades. It was shown recently [6] that employment of deep convolutional neural networks (CNNs) achieves promising performance. Since then, a number of studies [25, 5, 2, 3, 16, 40, 26, 8, 19] have been published on this approach, leading to significant improvement of estimation accuracy.

On the other hand, it is largely unknown why and how CNNs can estimate depth of a scene from its monocular image; they are basically black boxes as in other tasks. This will be an obstacle for this method to be employed in real-world applications, such as vision for self-driving cars and service robots, although it could be a cheap alternative solution to existing 3D sensors. In these applications, interpretability is essential for safety reasons.

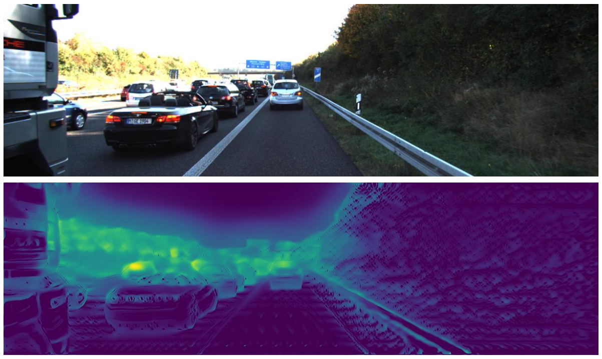


Figure 1. An example of the proposed visualization of single view depth estimation. Upper: an input image. Lower: a mask generated by our method showing relevant pixels for depth estimation.

Long-term studies in psychophysics have revealed that human vision uses several cues for monocular depth estimation, such as linear perspective, relative size, interposition, texture gradient, light and shades, aerial perspective, *etc*. [24, 13, 23, 32, 30, 18]. A natural question arises, *do CNNs utilize these cues?* Exploring this question will help our understanding of why CNNs can (or cannot) estimate depth from a given scene image. To the best of our knowledge, the present study is the first attempt to analyze how CNNs work on the task of monocular depth estimation.

It is, however, hard to find direct answers to the above questions; after all, it is still difficult even with human vision. Thus, as the first step toward this end, we consider visualization of CNNs on the task. To be specific, as in previous studies of visualization of CNNs for object recognition, we attempt to identify the image pixels that are relevant to depth estimation; see Fig. 1 for an example. To do this, we hypothesize that the CNNs can infer depths fairly accurately from only a selected set of image pixels. An underlying idea is an observation with human vision that most of the cues are considered to be associated with small regions in the visual field.

We then formulate the problem of identifying relevant pixels as a problem of sparse optimization. Specifically, we

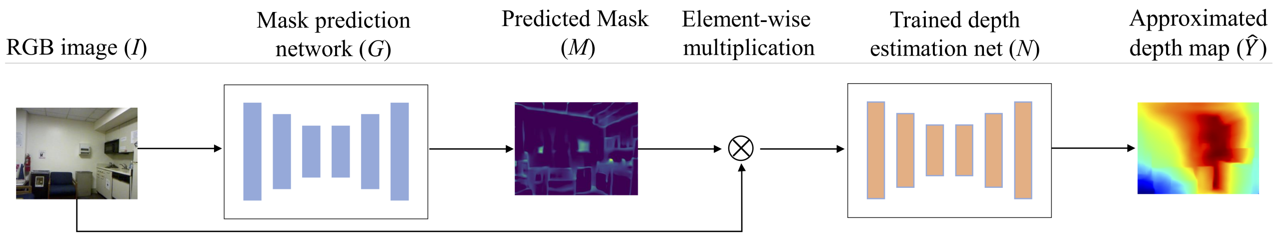

RGB image (I)
Mask prediction network (G)
Predicted Mask (M)
Element-wise multiplication
Trained depth estimation net (N)
Approximated depth map ($\hat{Y}$)

Figure 2. Diagram of the proposed approach. The target of visualization is the trained depth estimation net N. To identify the pixels of the input image I that N uses to estimate its depth map Y, we input I to the network G for predicting the set of relevant pixels, or the mask M. The output M is element-wise multiplied with I and inputted to N, yielding an estimate $\hat{Y}$ of the depth map. G is trained so that $\hat{Y}$ will be as close to the original estimate Y from the entire image I and M will be maximally sparse. Note that N is fixed in this process.

estimate an image mask that selects the smallest number of pixels from which the target CNN can provide the maximally similar depth map to that it estimates from the original input. This optimization requires optimization of the output of the CNN with respect to its input. As is shown in previous studies of visualization, such optimization through a CNN in its backward direction sometimes yields unexpected results, such as noisy visualization [35, 37] at best and even phenomenon similar to adversarial examples [7] at worst. To avoid this issue, we use an additional CNN to estimate the mask from the input image in the forward computation; this CNN is independent of the target CNN of visualization. Our method is illustrated in Fig. 2.

We conduct a number of experiments to evaluate the effectiveness of our approach. We apply our method to CNNs trained on indoor scenes (the NYU-v2 dataset) and those trained on outdoor scenes (the KITTI dataset). We confirm through the experiments that

- CNNs can infer the depth map from only a sparse set of pixels in the input image with similar accuracy to those they infer from the entire image;
- The mask selecting the relevant pixels can be predicted stably by a CNN. This CNN is trained to predict masks for a target CNN for depth estimation.

The visualization of CNNs on the indoor and outdoor scenes provides several findings including the following, which we think contribute to understanding of how CNNs works on the monocular depth estimation task.

- CNNs frequently use some of the edges in input images but not all of them. Their importance depends not necessarily on their edge strengths but more on usefulness for grasping the scene geometry.
- For outdoor scenes, large weights tend to be given to distant regions around the vanishing points in the scene.

2. Related work

There are many studies that attempt to interpret inference of CNNs, most of which have focused on the task of image classification [1, 43, 37, 36, 44, 31, 33, 17, 7, 28, 38]. However, there are only a few methods that have been recognized to be practically useful in the community [11, 20, 21].

Gradient based methods [36, 28, 38] compute a saliency map that visualizes sensitivity of each pixel of the input image to the final prediction, which is obtained by calculating the derivatives of the output of the model with respect to each image pixel.

There are many methods that mask part of the input image to see its effects [42]. General-purpose methods developed for interpreting inference of machine learning models, such as LIME [31] and Prediction Difference Analysis [44], may be categorized in this class, when they are applied to CNNs classifying an input image.

The most dependable method as of now for visualization of CNNs for classification is arguably the class activation map (CAM) [43], which calculate the linear combination of the activation of the last convolutional layers in its channel dimension. Its extension, Grad-CAM [33], is also widely used, which integrates the gradient-based method with CAM to enable to use general network architectures that cannot be dealt with by CAM.

However, the above methods, which are developed mainly for explanation of classification, cannot directly be applied to CNNs performing depth estimation. In the case of depth estimation, the output of CNNs is a two-dimensional map, not a score for a category. This immediately excludes gradient based methods as well as CAM and its variants. The masking methods that employ fixed-shape masks [44] or super-pixels obtained using low-level image features [31] are not fit for our purpose, either, since there is no guarantee that their shapes match well with the depth cues in input images that are utilized by the CNNs.

3. Method

3.1. Problem Formulation

Suppose a network N that predicts the depth map of a scene from its single RGB image as

$$Y = N(I), \tag{1}$$

where Y is an estimated depth map and I is the normalized version of the input RGB image. Following previous studies, we normalize each image by the z-score normalization. This model N is the target of visualization.

Human vision is considered to use several cues to infer depth information, most of which are associated with regions with small areas in the visual field. Thus, we make an assumption here that *CNNs can infer depth map equally well from a selected set of sparse pixels of I, as long as they are relevant to depth estimation.* To be specific, we denote a binary mask selecting pixels of I by M and a masked input by $I \otimes M$, where $\otimes$ denotes element-wise multiplication. The depth estimate $\hat{Y}$ provided by our network N for the masked input is

$$\hat{Y} = N(I \otimes M). \tag{2}$$

Our assumption is that $\hat{Y}$ can become very close to the original estimate $Y = N(I)$, when the mask M is chosen properly.

Now, we wish to find such a mask M for a given input I that $\hat{Y} = N(I \otimes M)$ will be as close to $Y = N(I)$ as possible. As our purpose is to understand depth estimation, we also want M that is as sparse as possible (*i.e.*, having the smallest number of non-zero pixels). To do so, we relax the condition that M is binary, *i.e.*, its element is either 0 or 1. We instead assume each element of M to have a continuous value in the range of $[0, 1]$. We will validate this relaxation in our experiments, where we also check the validity of the above assumption of depth estimation from sparse pixels.

Finally, we formulate our problem as the following optimization:

$$\min_{M} \; l_{\text{dif}}(Y, \hat{Y}) + \lambda \frac{1}{n} \|M\|_1 \tag{3}$$

where l_{dif} is a measure of difference between Y and $\hat{Y}$; λ is a control parameter for the sparseness of M; n is the number of pixels; and $\|M\|_1$ is the ℓ_1 norm (of a vectorized version) of M.

3.2. Learning to Predict Mask

Now we consider how to perform the optimization (3). The network N appears in the objective function through the variable $\hat{Y} = N(I \otimes M)$. We need carefully consider such optimization associated with the output of a CNN with respect to its input, because it often provides unexpected results, as is shown in previous studies.

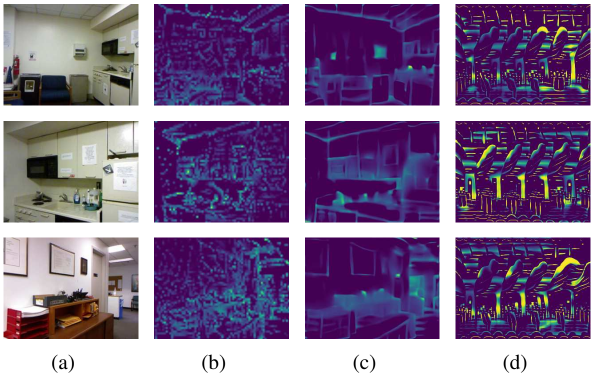


Figure 3. Form left to right, (a) RGB images (b) M obtained by solving (3), (c) M obtained by solving (4), (d) M obtained by solving (5).

In [35], the optimal inputs to CNNs trained on object recognition are computed that maximize the score of a selected object class for the purpose of visualization. Although they provide some insights into what the CNNs have learned, the images thus computed are unstable (*e.g.*, sensitive to initial values); they are distant from natural images and not so easy to interpret. To obtain more visually interpretable images, researchers have employed several constraints on the input images to be optimized, *e.g.*, the one making them appear to be natural images [10, 29]. In addition, optimization of (a function of) network outputs sometimes yield unpredictable results; typical examples are the adversarial examples [7].

Thus, instead of minimizing (3) with respect to individual elements of M, we use an additional network G to predict $M \approx G(I)$ that minimizes (3). More specifically, we consider the following optimization:

$$\min_{G} \; l_{\text{dif}}(Y, N(I \otimes G(I))) + \lambda \frac{1}{n} \|G(I)\|_1, \tag{4}$$

where $\|G(I)\|_1$ indicates ℓ_1 norm of vectorized $G(I)$. We employ the sigmoid activation function for the output layer of G, which constrain its output in the range of $[0, 1]$. The details of our method for training G are shown in Algorithm 1. Figure 3 shows comparison of M computed by different methods. It is seen that the direct optimization of (3) (Fig.3(b)) yields noisy, less interpretable maps than our approach (Fig.3(c)).

We have considered removing as many unimportant pixels of I as possible while maximally maintaining the original prediction $Y = N(I)$. There is yet another approach to identify important/unimportant pixels, which is to identify the most important pixels of I, without which the prediction will maximally deteriorate. This is formulated as the following optimization problem:

$$\min_{G} \; -l_{\text{dif}}(Y, N(I \otimes G(I))) + \lambda \frac{1}{n} \|(1 - G(I))\|_1. \tag{5}$$

This formulation is similar to that employed in [7], a study for visualization of CNNs for object recognition, in which the most important pixels in the input image are identified by masking the pixels that maximally lower the score of a selected object class. Unlike our method, the authors directly optimize M; to avoid artifacts that will emerge in the optimization, they employ additional constraints on M other than its sparseness[1]. The results obtained by the optimization of (5) are shown in Fig. 3(d). It is seen that this approach cannot provide useful results.

Algorithm 1 Algorithm for training the network G for prediction of M.

```
Input: $N$: a target, fully-trained network for depth estimation; $\psi$: a training set, i.e., pairs of the RGB image and depth map of a scene; $\lambda$: a parameter controlling the sparseness of $M$.
Hyperparameters: Adam optimizer, learning rate: $1e^{-4}$, weight decay: $1e^{-4}$, training epochs: $K$.
Output: $G$: a network for predicting $M$.
Freeze $N$;
for $j$ = 1 to $K$ do
    for $i$ = 1 to $T$ do
        Select RGB batch $\psi_i$ from $\psi$;
        Set gradients of $G$ to 0;
        Calculate depth maps for $\psi_i$:
                $Y_{\psi_i} = N(\psi_i)$;
        Calculate the value (L) of objective function:
        $L = l_{\text{dif}}(Y_{\psi_i}, N(\psi_i \otimes G(\psi_i)) + \lambda \frac{1}{n}\|G(\psi_i)\|_1$;
        Backpropagate $L$;
        Update $G$;
    end for
end for
```

4. Experiments

4.1. Experimental Setup

Datasets We use two datasets NYU-v2 [34] and KITTI datasets [39] for our analyses, which are the most widely used in the previous studies of monocular depth estimation. The NYU-v2 dataset contains 464 indoor scenes, for which we use the official splits, 249 scenes for training and 215 scenes for testing. We obtain approximately 50K unique pairs of an image and its corresponding depth map. Following the previous studies, we use the same 654 samples for testing. The KITTI dataset contains outdoor scenes and is collected by car-mounted cameras and a LIDAR sensor. We use the official training/validation splits; there are 86K image pairs for training and 1K image pairs from the official cropped subsets for testing. As the dataset only provides sparse depth maps, we use the depth completion toolbox of the NYU-v2 dataset to interpolate pixels with missing depth.

Target CNN models There are many studies for monocular depth estimation, in which a variety of architectures are proposed. Considering the purpose here, we choose models that show strong performance in estimation accuracy with a simple architecture. One is an encoder-decoder network based on ResNet-50 proposed in [16], which outperforms previous ones by a large margin as of the time of publishing. We also consider more recent ones proposed in [14], for which we choose three different backbone networks, ResNet-50 [12], DenseNet-161 [9], and SENet-154 [15]. For better comparison, all the models are implemented in the same experimental conditions. Following their original implementation, the first and the latter three models are trained using different losses. To be specific, the first model is trained using ℓ_1 norm of depth errors[2]. For the latter three models, sum of three losses are used, *i.e.*, $l_{\text{depth}} = \frac{1}{n}\sum_{i=1}^{n} F(e_i)$, $l_{\text{grad}} = \frac{1}{n}\sum_{i=1}^{n}(F(\nabla_x(e_i)) + F(\nabla_y(e_i)))$, and $l_{\text{normal}} = \frac{1}{n}\sum_{i=1}^{n}(1 - \cos\theta_i)$, where $F(e_i) = \ln(e_i + 0.5)$; $e_i = \|y_i - \hat{y_i}\|_1$; y_i and $\hat{y}_i$ are true and estimated depths; and θ_i is the angle between the surface normals computed from the true and estimated depth map.

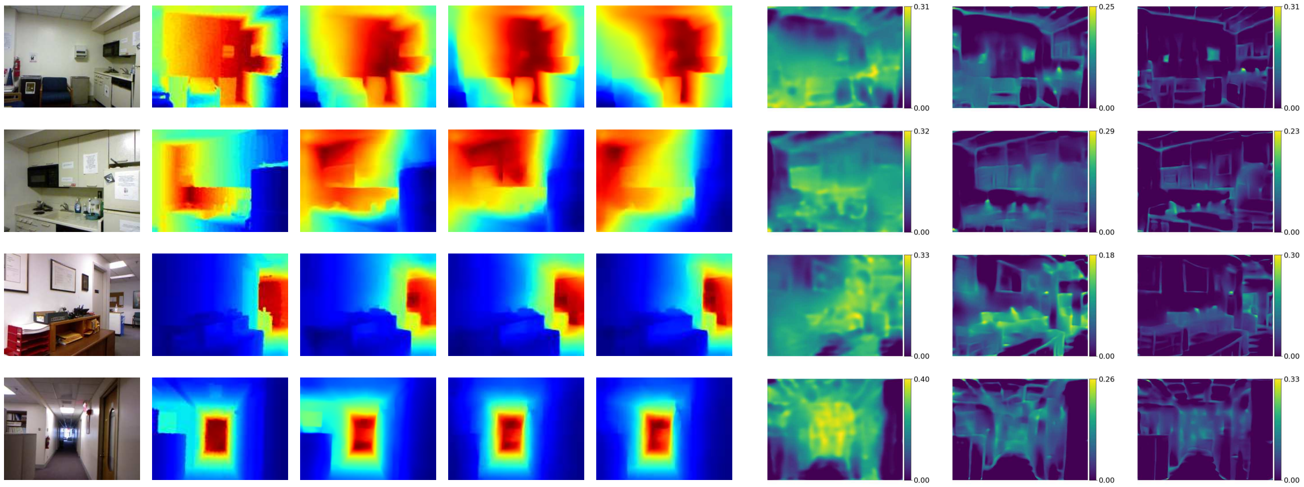
Network G for predicting M We employ an encoder-decoder structure for G. For the encoder, we use the dilated residual network (DRN) proposed in [41], which preserves local structures of the input image due to a fewer counts of down-sampling. Specifically, we use a DRN with 22 layers (DRN-D-22) pre-trained on ImageNet [4], from which we remove the last fully connected layer. It yields a feature map with 512 channels and $1/8$ resolution of the input image. For the decoder, we use a network consisting of three up-projection blocks [16] yielding a feature map with 64 channels and the same size as the input image, followed by a 3×3 convolutional layer outputting M. The encoder and decoder are connected to form the network G, which has 25.3M parameters in total. For the loss used to train G, we use $l_{\text{dif}} = l_{\text{depth}} + l_{\text{grad}} + l_{\text{normal}}$.

4.2. Estimating Depth from Sparse Pixels

As explained above, our approach is based on the assumption that the network N can accurately estimate depth from only a selected set of sparse pixels. We also relaxed the condition on the binary mask, allowing M to have continuous values in the range of [0,1]. To validate the assumption as well as this relaxation, we check how the accuracy

[1] In our experiments, we confirmed that their method works well for VGG networks but behaves unstably for modern CNNs such as ResNets.

[2] We have found that ℓ_1 performs better than the $berhu$ loss originally used in [16], which agree with [27].



(a) RGB images (b) Ground truth (c) $\hat{Y}$ when λ=1 (d) $\hat{Y}$ when λ=3 (e) $\hat{Y}$ when λ=5 (f) M when λ=1 (g) M when λ=3 (h) M when λ=5

Figure 4. Visual comparison of approximated depth maps and estimated masks (M's) for different values of the sparseness parameter λ.

Table 1. Accuracy of depth estimation for different values of the sparseness parameter λ. Results on the NYU-v2 dataset by the ResNet-50 model of [14]. Sparseness in the table indicates the average number of non-zero pixels in M'.

λ	RMSE (M)	RMSE (M')	Sparseness
original	0.555	0.555	1.0
$\lambda = 1$	0.605	0.568	0.920
$\lambda = 2$	0.668	0.617	0.746
$\lambda = 3$	0.699	0.668	0.589
$\lambda = 4$	0.731	0.733	0.425
$\lambda = 5$	0.740	0.758	0.361
$\lambda = 6$	0.772	0.882	0.215

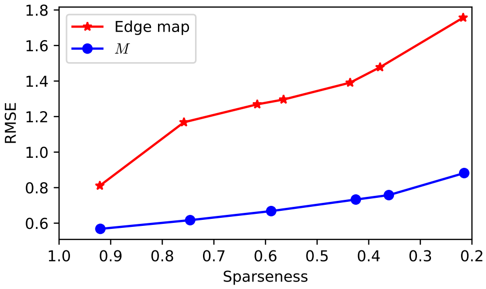


Figure 5. Comparison of accuracy of depth estimation when selecting input image pixels using M and using the edge map of input images.

of depth estimation will change when binarizing the continuous mask M predicted by G.

To be specific, computing $M = G(I)$ for I, we binarize M into a binary map M' using a threshold $\epsilon = 0.025$. We then compare accuracy of the predicted depth maps $N(I \otimes M')$ and $N(I \otimes M)$. As the sparseness of M is controlled by the parameter λ as in Eq.(4), we evaluate the accuracy for different λ's. We use the NYU-v2 dataset and a ResNet-50 based model of [14]. We train it for 10 epochs on the training set and measure its accuracy by RMSE.

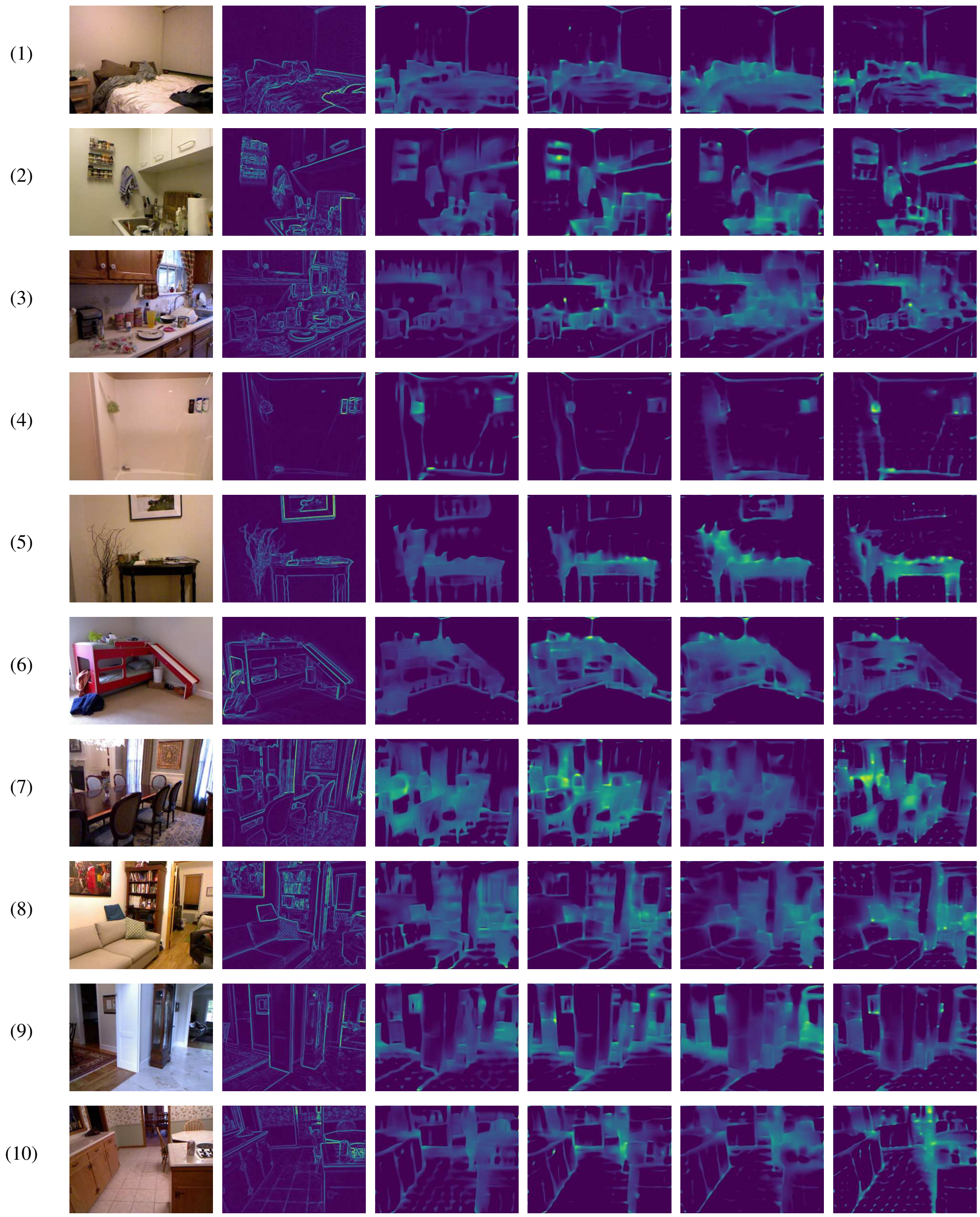
Table 1 shows the results. It is first observed that there is trade-off between accuracy of depth estimation and sparseness of the mask M. Please note that the RMSE values are calculated against the ground truth depths. The error grows from 0.555 ($\lambda = 0$) to 0.740 ($\lambda = 5$, the value we used in the subsequent experiments), which is only 33% increase. We believe this is acceptable considering the accuracy-interpretability trade-off that is also seen in many visualization studies.

Figure 4 shows examples of pairs of the mask M and estimated depth map $\hat{Y}$ for different λ's for four different input images. It is also observed from Table 1 that the estimated depth with the binarized mask M' is mostly the same as that with the continuous M when λ is not too large; it is even more accurate for small λ's. This validates our relaxation allowing M to have continuous values. Considering the trade-off between estimation accuracy and λ as well as the difference between prediction with M and M', we choose $\lambda = 5$ in the analyses shown in what follows.

4.3. Analyses of Predicted Mask

4.3.1 NYU-v2 dataset

Figure 6 shows predicted masks for different input images and different depth prediction networks. It is first observed that there are only small differences among different networks. This will be an evidence that the proposed visualization method can stably identify relevant pixels to depth



(a) RGB images (b) Edge maps (c) M for [16] (ResNet-50) (d) M for [14] (ResNet-50) (e) M for [14] (DenseNet-161) (f) M for [14] (SENet-154)

Figure 6. Predicted masks for different input images for different depth estimation networks, ResNet-50-based model of [16] and three models of [14] whose backbones are ResNet-50, DenseNet-161, and SENet-154, respectively. The edge map of the input I is also shown for comparison.

estimation. For the sake of comparison, edge maps of I are also shown in Fig. 6. It is seen from comparison with them that M tends to have non-zero values on the image edges; some non-zero pixels indeed lie exactly on image edges (*e.g.*, the vertical edge on the far side in (1)).

However, a closer observation reveals that there is also a difference between M and the edge map; M tends to have non-zero pixels over the filled regions of objects, not on

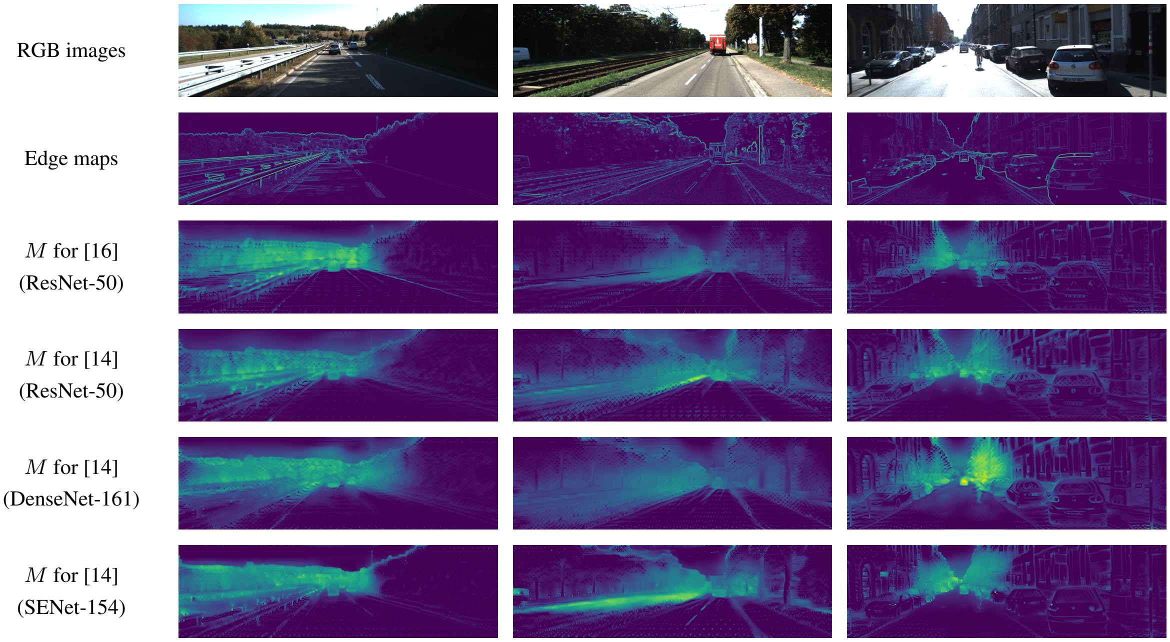


Figure 7. Predicted masks for different networks trained on the KITTI dataset for different input images from the test split.

their boundaries, as with the table in (5), the chairs in (7) *etc*. Moreover, very strong image edges sometimes disappear in M, as is the case with a bottom edge of the cabinet in (2); instead, M has non-zero pixels along a weaker image edge emerging on the border of the cabinet and the wall. This is also the case with the intersecting lines between the floor and the bed in (6); M has large values along them, whereas their edge strength is very weak.

To further investigate (dis)similarity between M and the edge map, we compare them by setting the edge map to M and evaluate the accuracy of the predicted depth $N(I \otimes M)$. Figure 5 shows the results. It is seen that the use of edge maps yields less accurate depth estimation, which clearly indicates the difference of the edge maps and the masks predicted by G.

Not boundary alone but filled region is highlighted for small objects. We conjecture that the CNNs recognize the objects and somehow utilize it for depth estimation.

4.3.2 KITTI dataset

Figure 7 shows the predicted masks on the KITTI dataset for three randomly selected images along with their edge maps. More examples are given in the supplementary material. As with the NYU-v2 dataset, the predicted masks tend to consist of edges and filled regions, and are clearly different from the edge maps. It is observed that some image edges are seen in the masks but some are not. For example, in the first image, the guard rail on the left has strong edges, which are also seen in the mask. On the other hand, the white line on the road surface provides strong edges in the edge map but is absent in the mask. This indicates that the CNNs utilizes the guard rail but does not use the white line for depth estimation for some reason. This is also the same as the white vertical narrow object on the roadside in the second image.

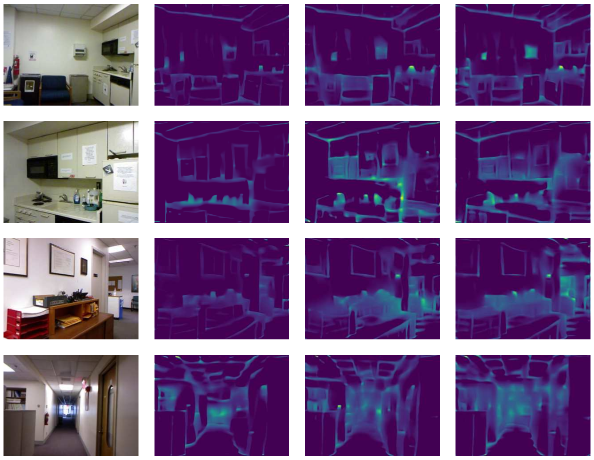
A notable characteristic of the predicted masks on this dataset is that the region around the vanishing point of the scene is strongly highlighted in the predicted masks. This is the case with all the images in the dataset, not limited to the three shown here. Our interpretation of this phenomenon will be given in the discussion below.

4.3.3 Summary and Discussion

In summary, there are three findings from the above visualization results.

Important/unimportant image edges Some of the image edges are highlighted in M and some are not. This implies that the depth prediction network N selects important edges that are necessary for depth estimation. The selection seems to be more or less independent of the strength of edges. We conjecture that those selected are essential for inferring the 3D structure (*e.g.*, orientation, perspective *etc.*) of a room and a road.

Attending on the regions inside objects As for objects in a scene, not only the boundary but the inside region of them tend to be highlighted. This is the case more with



(a) RGB images
(b) l_{depth}
(c) l_{depth} $+l_{\text{grad}}$
(d) l_{depth} $+l_{\text{grad}}$ $+l_{\text{normal}}$

Figure 8. Comparison of the estimated mask M for the three combinations of loss functions.

smaller objects, although this may be partly attributable to the use of sparseness constraint. Unlike the image edges providing the geometric structure of the scene, we conjecture that the depth estimation network N may 'recognize' the objects and use their sizes to infer absolute or relative distance to them.

Vanishing points In the case of outdoor scenes of KITTI, the regions around vanishing points (or simply far-away regions) are always highlighted almost without exception. This shows that these regions are important for N to provide accurate depths. This may be attributable to the fact that distant scene points tend to yield large errors because of the loss evaluating the difference in absolute depths; then such distant scene regions will be given more weights than others. Another possible explanation is that this is due to the natural importance of vanishing points; they are naturally a strong cue to understand geometry of a scene. Although these two explanations appear to be orthogonal, they could be coupled with each other in practice. A possible hypothesis is that CNNs (and/or human vision) learn to look at the vanishing points as they are distant and given more weights. Further investigation will be a direction of future studies.

4.4. Evaluation of Training Losses

There are several discussions in recent studies on how we should measure accuracy of estimated depth maps [22, 14] and what losses we should use for training CNNs [14]. We compare the impact of losses by visualizing a network N trained on different losses. Following [14], we consider three losses, l_{depth} (the most widely used one measuring difference in depth values); l_{grad} (difference in gradients of scene surfaces); and l_{normal} (difference in orientation of normal to scene surfaces). We train a ResNet-50 based model of [14] on NYU-v2 using different combinations of the three losses, *i.e.*, l_{depth}, $l_{\text{depth}} + l_{\text{grad}}$, and $l_{\text{depth}} + l_{\text{grad}} + l_{\text{normal}}$. Figure 8 shows the generated masks for networks trained using the three loss combinations. It is observed that the inclusion of l_{grad} highlights more on the surface of objects. The further addition of l_{normal} highlight more on small objects and makes edges more straight if they should be.

5. Summary and Conclusion

Toward answering the question of how CNNs can infer the depth of a scene from its monocular image, we have considered their visualization. Assuming that CNNs can infer a depth map accurately from a small number of image pixels, we considered the problem of identifying these pixels, or equivalently a mask concealing the other pixels, in each input image. We formulated the problem as an optimization problem of selecting the smallest number of pixels from which the CNN can estimate a depth map with the minimum difference to that it estimates from the entire image. Pointing out that there are difficulties with optimization through a deep CNN, we propose to use an additional network to predict the mask for an input image in forward computation.

We have confirmed through several experiments that the above assumption holds well and the proposed approach can stably predict the mask for each input image with good accuracy. We then applied the proposed method to a number of monocular depth estimation CNNs on indoor and outdoor scene datasets. The results provided several findings, such as i) the behaviour of CNNs that they seem to select edges in input images depending not on their strengths but on importance for inference of scene geometry; ii) the tendency of attending not only on the boundary but the inside region of each individual object; iii) the importance of image regions around the vanishing points for depth estimation on outdoor scenes. We also show an application of the proposed method, which is to visualize the effect of using different losses for training a depth estimation CNN.

We think these findings contribute to moving forward our understanding of CNNs on the depth estimation task, shedding some light on the problem that has not been explored so far in the community.

Acknowledgments: This work was partly supported by JSPS KAKENHI Grant Number JP15H05919 and JP19H01110 and by JST CREST Grant Number JPMJCR14D1.

References

[1] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, Deva Ramanan, and Thomas S. Huang. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *ICCV*, pages 2956–2964, 2015.

[2] Ayan Chakrabarti, Jingyu Shao, and Gregory Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, pages 2658–2666, 2016.

[3] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NIPS*, pages 730–738, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[5] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV*, pages 2650–2658, 2015.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[7] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *ICCV*, pages 3449–3457, 2017.

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[9] Huang Gao, Liu Zhuang, Weinberger Kilian Q, and van der Maaten Laurens. Densely connected convolutional networks. *CVPR*, 2017.

[10] Google. `https://deepdreamgenerator.com`.

[11] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Ian P Howard. *Seeing in depth, Vol. 1: Basic mechanisms.* University of Toronto Press, 2002.

[14] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019.

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[16] Laina Iro, Rupprecht Christian, Belagiannis Vasileios, Tombari Federico, and Navab Nassir. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248, 2016.

[17] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. Learn to pay attention. *CoRR*, abs/1804.02391, 2018.

[18] DH Kelly. Visual contrast sensitivity. *Optica Acta: International Journal of Optics*, 24(2):107–129, 1977.

[19] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.

[20] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods. *CoRR*, abs/1711.00867, 2017.

[21] Pieter-Jan Kindermans, Kristof T. Schutt, Maximilian Alber, K. Muller, Dumitru Erhan, Been Kim, and Sven Dahne. Learning how to explain neural networks: Patternnet and patternattribution. *CoRR*, 2017.

[22] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of cnn-based single-image depth estimation methods. *CoRR*, abs/1805.01328, 2018.

[23] Michael S Landy, Laurence T Maloney, Elizabeth B Johnston, and Mark Young. Measurement and modeling of depth cue combination: In defense of weak fusion. *Vision research*, 35(3):389–412, 1995.

[24] Pierre R. Lebreton, Alexander Raake, Marcus Barkowsky, and Patrick Le Callet. Measuring perceived depth in natural images and study of its relation with monocular and binocular depth cues. In *Stereoscopic Displays and Applications XXV*, volume 9011, page 90110C. International Society for Optics and Photonics, 2014.

[25] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. *CVPR*, pages 1119–1127, 2015.

[26] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *CVPR*, pages 3372–3380, 2017.

[27] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *ICRA*, 2018.

[28] Aravindh Mahendran and Andrea Vedaldi. Salient deconvolutional networks. In *ECCV*, 2016.

[29] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *CoRR*, abs/1602.03616, 2016.

[30] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. Depth cues in human visual perception and their realization in 3d displays. In *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, volume 7690, page 76900B. International Society for Optics and Photonics, 2010.

[31] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[32] Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *IJCAI*, volume 7, 2007.

[33] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Ba-

tra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV*, pages 618–626, 2017.

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[36] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017.

[37] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[39] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, 2017.

[40] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *CVPR*, pages 161–169, 2017.

[41] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017.

[42] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[43] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CVPR*, pages 2921–2929, 2016.

[44] Luisa M. Zintgraf, Taco Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *ICLR*, 2017.