Simultaneous Attack on CNN-Based Monocular Depth Estimation and Optical Flow Estimation

Koichiro YAMANAKA\(^{a}\), Member, Keita TAKAHASHI\(^{b}\), Senior Member, Toshiaki FUJII\(^{c}\), Fellow, and Ryuraroh MATSUMOTO\(^{d}\), Senior Member

SUMMARY Thanks to the excellent learning capability of deep convolutional neural networks (CNNs), CNN-based methods have achieved great success in computer vision and image recognition tasks. However, it has turned out that these methods often have inherent vulnerabilities, which makes us cautious of the potential risks of using them for real-world applications such as autonomous driving. To reveal such vulnerabilities, we propose a method of simultaneously attacking monocular depth estimation and optical flow estimation, both of which are common artificial-intelligence-based tasks that are intensively investigated for autonomous driving scenarios. Our method can generate an adversarial patch that can fool CNN-based monocular depth estimation and optical flow estimation methods simultaneously by simply placing the patch in the input images. To the best of our knowledge, this is the first work to achieve simultaneous patch attacks on two or more CNNs developed for different tasks.

**key words:** adversarial attack, monocular depth estimation, optical flow estimation, CNN

1. Introduction

Thanks to the excellent learning capability of deep convolutional neural networks (CNNs), CNN-based methods have achieved great success in computer vision and image recognition tasks. Among them, monocular depth estimation and optical flow estimation are two representative tasks that are intensively investigated for real-world applications such as autonomous driving. Meanwhile, it has been shown that CNN-based methods often have inherent vulnerabilities to deliberate attacks\(^{[1]}\)–\(^{[7]}\). We focus on adversarial patch attacks, in which an attack is conducted by overwriting a region on the input image with a specific pattern (adversarial patch)\(^{[3]}\)–\(^{[7]}\) because of the implementation feasibility in the real world, namely, we can deceive a target CNN by simply placing the adversarial patch in the target scene.

We previously reported\(^{[7]}\) that we can generate an adversarial patch that is effective against two CNNs for monocular depth estimation simultaneously, indicating the possibility that two networks for the same task (in this case, monocular depth estimation) would have common vulnerabilities regardless of the differences in the network architectures. In this paper, we extend the idea of simultaneous attack to two different tasks. In particular, we propose a method of simultaneously attacking monocular depth estimation and optical flow estimation by using an adversarial patch placed in the input images. To the best of our knowledge, this is the first work to achieve simultaneous patch attacks on two or more CNNs developed for different tasks.

2. Related Work

Monocular depth estimation refers to the task of predicting pixel-wise depth from a single image. This technique has attracted great attention due to its less demanding hardware requirements. Following the seminal work by Eigen et al.\(^{[8]}\), there has been rapid and significant progress in this technique. In particular, Lee et al.\(^{[9]}\) and Guo et al.\(^{[10]}\) achieved state-of-the-art results in the supervised and unsupervised frameworks, respectively. Their results are quite impressive and seemingly comparable to those with stereo depth estimation methods.

Optical flow estimation refers to the task of predicting apparent pixel-wise 2-D motion in a temporal sequence of images, which is represented by a vector field describing the displacement of each pixel. Dosovitskiy et al.\(^{[11]}\) proposed an end-to-end optical flow estimation CNN called FlowNet. Ilg et al.\(^{[12]}\) extended it by stacking multiple FlowNets and achieved state-of-the-art performance.

The adversarial patch attack was introduced by Brown et al.\(^{[3]}\), where an attack was conducted by placing a small artificial pattern on the input image, instead of adding perturbations on the entire image, as was done in previous works\(^{[1]}\),\(^{[2]}\). The patch can be placed anywhere in the image and independent of target scenes. Therefore, an adversarial patch attack is easily implemented in the real world, e.g., by placing printed patches in a target scene. Following a previous work\(^{[3]}\), adversarial patch attacks have been applied to several tasks such as face recognition\(^{[4]}\), object detection\(^{[6]}\), optical flow estimation\(^{[5]}\), and monocular depth estimation\(^{[7]}\).

The most similar methods to ours are those by Ranjan et al.\(^{[5]}\) (against optical flow estimation) and Yamanaka et al.\(^{[7]}\) (against monocular depth estimation). However, our adversarial patch is effective against both optical flow estimation and monocular depth estimation simultaneously, which was not achieved in previous works.
3. Proposed Method

Figure 1 illustrates the overview of our method. We adopted the CNNs of the methods by Guo et al. [10] (monocular depth estimation) and Ilg et al. [12] (optical flow estimation) as the target methods to deceive. Our goal is to derive a single adversarial patch (denoted as $P$) that can deceive (inducing incorrect estimates) the two target CNNs simultaneously. Our method is basically constructed on the basis of that of Yamanaka et al. [7].

3.1 Modeling Attack with Adversarial Patch

Let $I_t$, $I_{t+1}$ be the original image taken at time $t$, $t + 1$, $F_d$ be the target monocular depth estimation CNN, and $F_o$ be the target optical flow estimation CNN. The estimated depth map and optical flow are represented as $D = F_d(I_t)$ and $O = F_o(I_t, I_{t+1})$, respectively.

The adversarial patch $P$ should be robust against various shooting conditions. Therefore, we apply various transformations (denoted as $T_\theta$) for $P$ similarly to what Yamanaka et al. [7] did: the transformations include random brightness shifts, contrast shifts, addition of noise, scaling, rotation, and perspective transformation, and we randomly change $T_\theta$ for each mini-batch during the training stage. Let $R_\theta$ denote the region in which $P$ falls after receiving transformations $T_\theta$.

The attacked image $\hat{I}_t$ is represented as $\hat{I}_t = I_t + R_\theta T_\theta(P)$, where $+_R$ refers to the pixel overwriting on the region $R_\theta$. The $\hat{I}_{t+1}$ is also represented as $\hat{I}_{t+1} = I_{t+1} + R_\theta T_\theta(P)$ with the same $\theta$, assuming that the position of $P$ is time-invariant, as was also assumed by Ranjan et al. [5].

The process mentioned above is implemented as fully differentiable, which allows us to update $P$ by the framework of back-propagation. During the training stage, we keep the weights for $F_d$ and $F_o$ unchanged but update $P$ alone.

3.2 Design of Loss Function

Before introducing our loss function, we discuss the difference between the two target CNNs [10, 12]. Yamanaka et al. [7] presented a method for visualizing the regions that are potentially affected by a patch located at a specific position. As shown in Fig. 2, we apply this visualization to the target CNNs with the same patch location $R_\theta$ (denoted with the red circle). We observe that the effect on monocular depth estimation is localized only around $R_\theta$, while the effect on optical flow estimation spreads over the entire image. This observation conforms to the results reported in previous works; an adversarial patch was designed to affect only the local region for monocular depth estimation [7], while it was designed to affect the entire image for optical flow estimation [5]. Considering what is mentioned above, we adopt different loss terms for the two target CNNs in accordance with the previous works [5, 7].

The entire loss function, defined for $P$, is composed of four terms,

$$L = L_{\text{depth}}(P) + \alpha L_{\text{flow}}(P) + \beta L_{\text{NPS}}(P) + \gamma L_{TV}(P)$$  \hspace{1cm} (1)

where $\alpha$, $\beta$, and $\gamma$ are weighting coefficients determined experimentally.

The depth loss $L_{\text{depth}}$ is designed with a pre-specified target depth $d_t$ and described as

$$L_{\text{depth}}(P) = ||M_{R_\theta} \odot (d_t - F_d(I_t + R_\theta T_\theta(P)))||_1.$$  \hspace{1cm} (2)

where $M_{R_\theta}$ is a binary mask vector that takes 1 only for the pixels in $R_\theta$, $\odot$ denotes element-wise multiplication, and $||.||_1$ denotes $L_1$ norm. With this term, we aim to mislead the depth estimates of the target CNNs only in the region $R_\theta$ into the target depth $d_t$. Meanwhile, the optical flow loss $L_{\text{flow}}$ is defined as

$\text{Fig. 1} \quad \text{Overview of proposed method.}$

$\text{Fig. 2} \quad \text{Results on visualizing potentially affected regions. Red circles indicate the region of } R_\theta.$
\[ L_{\text{flow}}(P) = 1 - \| F_0(I_t, I_{t+1}) - F_0(I_t + R_t \cdot T_0(P), I_{t+1} + R_t \cdot T_0(P)) \|_1, \]  

(3)

with which the effect of \( P \) should cover the entire image.

Following Yamanaka et al. [7], we also use a non-printability score (NPS) loss [13], \( L_{\text{NPS}} \), to limit the color space of the patch within the printable color gamut, and a total variation (TV) loss, \( L_{\text{TV}} \), which encourage the spatial smoothness of the patch, to reduce aliasing artifacts when captured with a camera.

4. Experiments

4.1 Training and Implementation Details

The two target CNNs [10], [12] were pre-trained on the KITTI dataset, which are for depth estimation [14] and for optical flow estimation [15], respectively.
For training our adversarial patch $P$, we used 69,950 stereo pairs taken from the KITTI2015 training dataset [15]. We used stereo pairs as input for the optical flow estimation CNN [12], whereas we used only the left views for the monocular depth estimation CNN [10]. The width and height of the input images were set to 512 and 256 pixels, respectively, to fit the input size of the monocular depth estimation CNN [10]. For data augmentation, the stereo pairs were randomly swapped, zoomed with a factor of [0.8, 1], gamma-corrected in the range of [0.8, 1.2], illumination-changed with a factor of [0.8, 1.2], and color-jittered in the range of [0.95, 1.05]. The resolution for $P$ was set to $256 \times 256$ pixels, but its apparent size in the input images was changed by the patch transformer $T_b$.

We used a Linux-based PC equipped with a NVIDIA GeForce GTX 1080 Ti. The CNNs were implemented using Python version 3.6.9 and PyTorch version 1.1.0. We used a built-in Adam optimizer with a learning rate $10^{-3}$ and set the batch size to 8. The number of epochs was 50. The weighting coefficient $\alpha$ was set to 0 for the first 40 epochs, and 0.2 for the final 10 epochs, while $\beta$ and $\gamma$ were kept fixed to 0.01 and 2.5, respectively.

The resulting adversarial patches with the target depths $d_t$ of 3 and 150 meters are denoted as $P_n$ and $P_l$, respectively, and are shown in Fig. 4. For an ablation study, we prepared three other patches, $P_{depth}$: trained without $L_{flow}$, $P_{flow}$: trained without $L_{depth}$, and $P_{rand}$: a randomized pattern.

4.2 Result

We conducted simultaneous attacks on the two target methods [10], [12] by superimposing the two patches, $P_n$ and $P_l$ shown in Fig. 4, onto input images. Figure 3 shows the input images (w/ and w/o the attack), estimated depth maps, and estimated optical flow. We observed that our attack worked as designed; the depth values were incorrectly estimated (in accordance with $d_t$) in the regions where the patches were located, while the optical flow maps collapsed over the entire image. To highlight the effect of our attack, we also present the differences between the results with and without the attack (using patch $P_n$) in Fig. 5. We also observed that the regions with large differences in Fig. 5 were almost identical to the potentially affected regions visualized in Fig. 2. Furthermore, we compared results obtained with $P_n$, $P_{depth}$, $P_{flow}$, and $P_{rand}$ in Fig. 6. We can see that $P_{depth}$ and $P_{flow}$ had little effects on the estimated flow and depth, respectively. Meanwhile, $P_n$ achieved simultaneous attack on depth and optical flow. Note again that the target monocular depth estimation CNN and optical flow estimation CNN were deceived by a single adversarial patch, which indicates that both methods have common vulnerabilities despite the difference in the tasks and network architectures.

5. Conclusion

We proposed a method of simultaneously attacking a monocular depth estimation CNN and an optical flow estimation CNN by placing an adversarial patch on the input images. We reported several promising results through computer-simulated experiments (the patches were superimposed on the input images in a digital manner). For future work, we plan to conduct a more extensive evaluation of our method, which would include real-world experiments using printed adversarial patches and an on-vehicle camera. We hope that our work will lead to a wider recognition of the vulnerabilities of CNN-based methods; thus, lead to the development of safer technologies for real-world applications such as autonomous driving.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” Proc. Int. Conf. Learn. Represent. (ICLR), arXiv:1312.6199, 2014.
[2] I.J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” Int. Conf. Learn. Represent. (ICLR), arXiv:1412.6572, 2015.
[3] T. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” arXiv:1712.09665, 2017.
[4] S. Komkov and A. Petiushko, “Advhat: Real-world adversarial attack on arcface id system,” arXiv:1908.08705, 2019.
[5] A. Ranjan, J. Janai, A. Geiger, and M.J. Black, “Attacking optical flow,” Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp.2404–2413, 2019.
[6] S. Thys, W. Van Ranst, and T. Goedemé, “Fooling automated surveillance cameras: Adversarial patches to attack person detection,” Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp.1–7, 2019.
[7] K. Yamanaka, R. Matsumoto, K. Takahashi, and T. Fujii, “Adversarial patch attacks on monocular depth estimation networks,” IEEE Access, pp.179094–179104, 2020.
[8] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” Proc. Adv. Neural Inf. Process. Syst. (NIPS), pp.2366–2374, 2014.
[9] J.H. Lee, M.K. Han, D.W. Ko, and I.H. Suh, “From big to small: Multi-scale local planar guidance for monocular depth estimation,” arXiv:1907.10326, 2019.
[10] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, “Learning monocular depth by distilling cross-domain stereo networks,” Proc. Eur. Conf. Comput. Vis. (ECCV), pp.484–500, 2018.
[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp.2758–2766, 2015.
[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp.2462–2470, 2017.
[13] M. Sharif, S. Bhagavatula, L. Bauer, and M.K. Reiter, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (ACM SIGSAC), pp.1528–1540, Oct. 2016.
[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” Int. J. Robot. Res., vol.32, no.11, pp.1231–1237, 2013.
[15] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.3061–3070, 2015.