Recognizing people in blind spots based on surrounding behavior

Kensho Hara[1], Hirokatsu Kataoka[1],
Masaki Inaba[2], Kenichi Narioka[2], and Yutaka Satoh[1]

[1] National Institute of Advanced Industrial Science and Technology (AIST),
Tsukuba, Ibaraki, Japan {kensho.hara, hirokatsu.kataoka}@aist.go.jp
[2] DENSO CORPORATION, Chuo-ku, Tokyo, Japan
{MASAKI_INABA, KENICHI_NARIOKA}@denso.co.jp

Abstract. Recent advances in computer vision have achieved remarkable performance improvements. These technologies mainly focus on recognition of visible targets. However, there are many invisible targets in blind spots in real situations. Humans may be able to recognize such invisible targets based on contexts (e.g. visible human behavior and environments) around the targets, and used such recognition to predict situations in blind spots on a daily basis. As the first step towards recognizing targets in blind spots captured in videos, we propose a convolutional neural network that recognizes whether or not there is a person in a blind spot. Based on the experiments that used the volleyball dataset, which includes various interactions of players, with artificial occlusions, our proposed method achieved 90.3% accuracy in the recognition.

Keywords: Action recognition · Convolutional Neural Networks.

1 Introduction

Performance of many computer vision tasks, such as object recognition in images and action recognition in videos, has been remarkably improved [3, 6, 12]. Most approaches try to recognize targets captured in images and videos. In other words, they mainly focus on recognition of targets visible in images and videos.

There are many invisible targets in blind spots in real situations. Such blind spots are caused by occlusions and angle of view of a camera. It is difficult to observe the blind spots without special equipments. Humans, however, may be able to recognize such invisible targets based on contexts (e.g. visible human behavior and environments) around the targets. For example, as shown in Fig. 1 (left), we can know there is at least one person on the left even though the left side is a blind spot because the man on the right looks and talks to others on the left. In addition, we can easily know such things if the scenes are given as videos, which capture dynamic information of human behavior. The recognition in blind spots is useful for various situations, such as avoiding traffic accidents to pedestrians running out from behind buildings, understanding crowded scenes with heavy occlusions, and achieving low cost surveillance by reducing the number of cameras. Because such recognition can be extended to various objects and environments besides recognizing people, it is important to develop this topic.

Fig. 1: This is scenes including interactions of some people (included in the AVA dataset [5]). Let us consider the black rectangles as blind spots. Based on the behavior, such as gestures and gaze directions, of people visible in the frames, we can know there is at least one person in the blind spot.

As the first step towards recognizing targets in blind spots, we propose a method that recognize whether or not there is a person in a blind spot captured in videos. We use a spatiotemporal 3D convolutional neural network (3D CNN) [6] to represent features of behavior of visible environments. Our proposed method uses a video with a blind spot as an input, and outputs a label, which indicate whether or not there is a person in the blind spot. Though our network cannot directly describes features in blind spots, it represents visible human behavior and environments around the blind spots, which include useful information for the recognition. We experimentally evaluated our proposed method using the volleyball dataset [9], which includes various interactions of players, with artificial blind spots.

2 Related Work

The use of large-scale datasets, such as ImageNet [4] and Kinetics [11], and deep CNNs [3,6,7,12,15] have contributed substantially to the creation of successful vision-based algorithms for images and videos. These technologies mainly focus on recognition of targets visible in images and videos whereas we focus on recognition of invisible targets in blind spots.

Some works tried to estimate information in blind spots. Bouman et al. and Baradad et al. proposed estimation methods that use observations of light [1,2]. Mak et al. tried to estimate location of sound source without visual information [13]. Zhao et al. and used radio frequency signals to estimate human pose over a wall [16]. In contrast, our recognition method is based on behavior of visible people described by a 3D CNN.

He et al. [8] experimentally showed recent action recognition method can recognize human actions based on background information without observations of humans. Motivated by this work, we propose a action recognition based method to recognize a person in a blind spot.

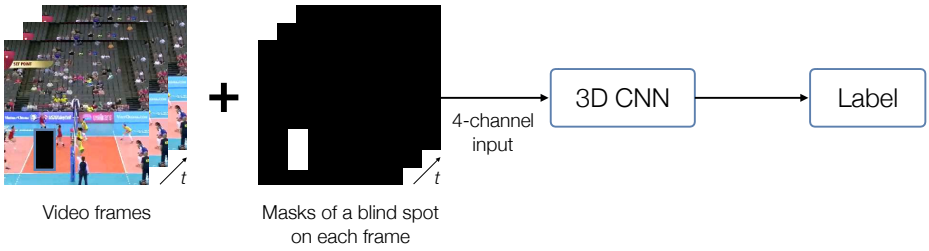

Fig. 2: Overview of our proposed method. The method uses the input video with bounding boxes of the blind spot in each frame, and outputs whether or not there is a person on the blind spot.

3 Method

In this study, we propose a spatiotemporal 3D CNN based method to recognize a person in a blind spot in a video. We assume that an input video captures an activity scene with people, objects, and a blind spot. Our proposed method uses the input video with bounding boxes of the blind spot in each frame, and outputs whether or not there is a person on the blind spot. Fig. 2 shows an overview of our proposed method.

In this study, we use the volleyball dataset [9] for the evaluation experiment. Because the situation of the dataset is restricted, the people captured in videos interact with each other, and it includes bounding boxes of each person in videos, it is good for the experiment in this study. A dataset for this experiment should include ground truth labels in blind spots though the volleyball dataset does not include such labels. Therefore, we add two types of artificial occlusions, similar to [8], as blind spots for positive and negative samples. We fill bounding boxes of a randomly selected person in each frame of a video to generate positive samples, which mean there is a person in the blind spot, whereas we fill randomly selected regions that do not cover any people to generate negative samples, which mean there is not a person in the blind spot. We evaluate our method by recognizing such samples in the experiment as a first step towards recognition of people in blind spots. Though the configuration of this experiment is not the same as real situations because a blind spot that track a person does not exist, we expect that this experiment shows the effectiveness of our method, which tries to represent human behavior around the targets in order to recognize a person in a blind spot.

In this section, we first explain the network architecture of 3D CNN, and then describe the detail implementation of training and testing.

3.1 Network Architecture

We use 3D ResNet [6], which is a spatiotemporal extension of original 2D ResNet [7]. ResNet, which is one of the most successful architectures in image classification, provides shortcut connections that allow a signal to bypass one layer and move to the next layer in the sequence. Since these connections pass through the networks' gradient flows from the later layers to the early layers, they can facilitate the training of very deep net-

works. Because 3D ResNet achieved good performance in action recognition, as shown in [6], we adopt the ResNet architecture.

We use the 18-layer ResNet with basic blocks and deeper 50-layer ResNet with bottleneck blocks. A basic block consists of two convolutional layers, and each convolutional layer is followed by batch normalization [10] and a ReLU [14]. The sizes of convolutional kernels in the blocks are $3 \times 3 \times 3$. We use identity connections and zero padding for the shortcuts of the ResNet block (type A in [7]) to avoid increasing the number of parameters. Strides of first convolutional layers of conv3, conv4, and conv5 are set to two to perform down-sampling of the inputs. A max pooling layer in conv2 also down-samples the inputs with a stride of two. Different from other convolutional layers, the size of conv1 is $7 \times 7 \times 7$. The temporal stride of conv1 is 1 whereas the spatial one is 2, similar to C3D [15].

A bottleneck block consists of three convolutional layers. The kernel sizes of the first and third convolutional layers are $1 \times 1 \times 1$, whereas those of the second are $3 \times 3 \times 3$. The shortcut pass of this block is the same as that of the basic block. We use identity connections except for those that are used for increasing dimensions (type B in [7]). Other settings are the same as the basic blocks.

The number of dimensions of a video input is four, which includes one channel, one temporal, and two spatial dimensions. The channel dimension consists of three RGB and one binary mask channels. The mask channel is used to specify where the blind spot of the input is (1 for a pixel on the spot). The number of dimensions of the output layer is two, which indicates whether or not there is a person in the blind spot.

3.2 Implementation

Training We use stochastic gradient descent with momentum to train the networks and randomly generate training samples from videos in training data in order to perform data augmentation. First, we select a temporal position in a video by uniform sampling in order to generate a training sample. A 16-frame clip is then generated around the selected temporal position. Each clip is cropped around a center position with the maximum scale (i.e. the sample width and height are the same as the short side length of the frame). We spatially resize the sample at 112×112 pixels. We then randomly decide the sample as positive or negative ones. Note that we can arbitrary set class labels for each video because we add artificial occlusions and class labels are decided based on the positions of artificial occlusions in this experiment. If the label is positive, we randomly select a person in the sample and fill the bounding boxes of the person in each frame as a blind spot. If the label is negative, we fill a region that is randomly selected in each frame. Note that relative movements of the blind spot of negative samples in each frame are based on a person that is randomly selected to reduce the information based on the movements of blind spots. We generate a binary mask channel based on the blind spots on each frame. The size of each sample is 4 channels $\times$ 16 frames $\times$ 112 pixels $\times$ 112 pixels, and each sample is horizontally flipped with 50% probability. All generated samples retain the same class labels as their original videos.

In our training, we use cross-entropy losses and back-propagate their gradients. The training parameters include a batch size of 64, weight decay of 0.01, and 0.9 for momentum. When training the networks from scratch, we start from learning rate 0.01,

Table 1: Network Architectures. The dimensions of output sizes are $T \times Y \times X$, and the sizes are calculated based on a $16 \times 112 \times 112$-input. We represent $x \times x \times x, F$ as the kernel size, and the number of feature maps of the convolutional filter are $x \times x \times x$ and F, respectively. Each convolutional layer is followed by batch normalization and a ReLU. Spatio-temporal down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of two. A max-pooling layer (stride 2) is also located before conv2_x for down-sampling. In addition, conv1 spatially down-samples inputs with a spatial stride of two.

Layer Name	Output Size	Architecture	
		18-layer	50-layer
conv1	$16 \times 64 \times 64$	$7 \times 7 \times 7, 64$, stride 1 (T), 2 (XY)	
conv2	$8 \times 32 \times 32$	$3 \times 3 \times 3$ max pool, stride 2 $\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1 \times 1, 64 \\ 3 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	$4 \times 16 \times 16$	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1 \times 1, 128 \\ 3 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	$2 \times 8 \times 8$	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1 \times 1, 256 \\ 3 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5	$1 \times 4 \times 4$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1 \times 1, 512 \\ 3 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$
average pool, 2-d fc, softmax			

and divide it by 10 at 500 epochs. Training is done for 600 epochs. When performing fine-tuning, we start from a learning rate of 0.001 and divide it by 10 at 200 epochs. Training of fine-tuning is done for 300 epochs.

Testing We recognize people in blind spots in videos using the trained model. We adopt the sliding window manner to generate input clips, (i.e. videos are split into non-overlapped 16 frame clips.) Each clip is cropped around a center position and combined with a binary mask channel similar to the training step. We estimate class probabilities of each clip using the trained model, and average them to recognize people in blind spots in videos.

Table 2: Recognition accuracies of each method.

Model	*2D AlexNet*	*2D ResNet-18*	*3D ResNet-18*		*3D ResNet-50*	
Pretraining				✓		✓
Accuracy	80.7	88.3	89.4	89.2	90.3	89.2

4 Experiments

4.1 Dataset

In the experiments, we used the volleyball dataset [9]. The volleyball dataset contains 55 videos, which captures volleyball game scenes. 4,830 frames in the videos are annotated with players' bounding boxes, individual action labels, and group activity labels. The videos are separated based on the annotations frames into 4,830 sequences. Each sequence contains 41 frames, which consist of the annotated frame and 20 frames before and after it. We used one sequence as one video input. We followed the split of training, validation, and testing sets provided in [9]. We randomly selected positive and negative sequences with 50% probability in the testing set, and selected one of the bounding boxes in each sequence of the positives as the blind spot. We also randomly generated a blind spot for the negative sequences.

4.2 Results

Table 2 shows the recognition accuracies of ResNet-18 and -50 with or without pretraining on the Kinetics dataset [11]. In addition, we show the results of 2D AlexNet [12] and 2D ResNet-18 as baselines. The 2D models used each frame as an input, output recognition scores, and recognize a video based on the averaged the scores over all frames in the video.

The accuracies of 3D ResNets, which are around 90%, indicate that our method can correctly recognize a person on a blind spot in many video sequences in this experimental configuration. We can see that ResNet-50 trained from scratch achieved the highest accuracy among the methods. This result indicate that using a deeper model improves recognition accuracy. On the other hand, the pretraining on Kinetics did not improve the recognition accuracies. This result indicates that the recognition in this experiment requires different feature representations to action recognition. Compared with the baselines, our 3D ResNets achieved higher accuracies. This result indicates that spatiotemporal feature representations are effective to this task.

Fig. 3 shows recognition examples of 3D ResNet-50 trained from scratch. The example of top row, which is a true positive, is a attack scene from right. Because the player on the right attacked but the visible players on the left did not receive a ball, we can estimate there is a player on the blind spot. The result indicate that the model could understand such activities in the scene. The middle row in Fig. 3 shows the example, which is a crowded scene near the net. The model wrongly recognized the video as a positive. We can see that it is difficult to recognize there is other people in a crowded

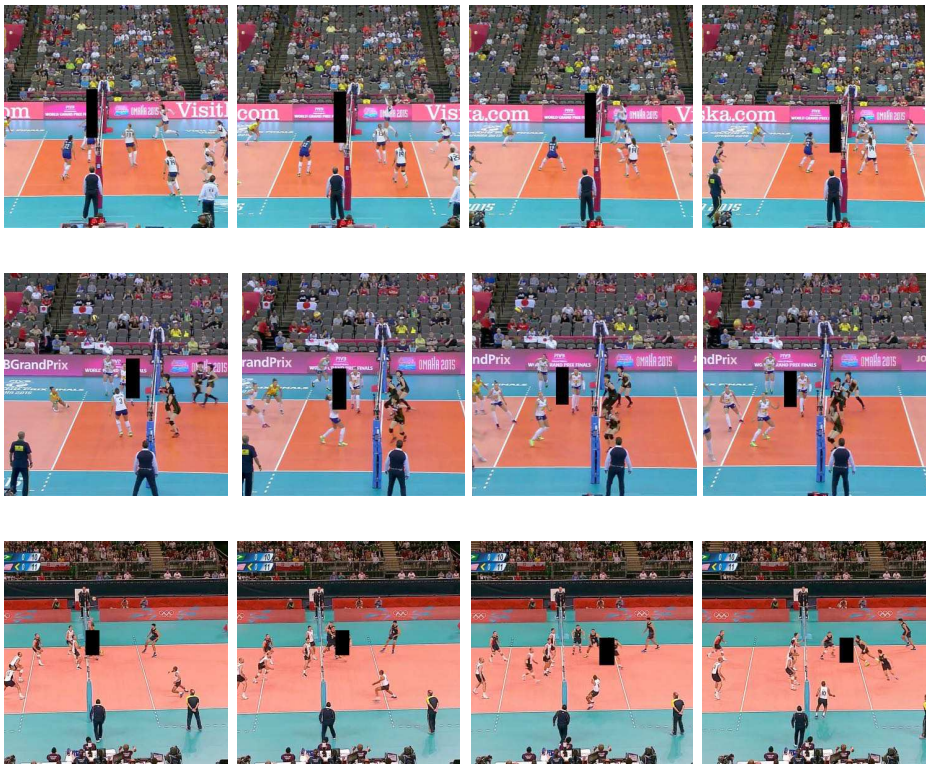

Fig. 3: Recognition examples. (top) True positive. (middle) False positive. (bottom) False negative.

scene. The bottom row shows a false negative example. Though this example is a positive one, the blind spot did not locate on a player in many frames, which indicate a negative sample. Because the annotations of volleyball dataset include some noise, annotations of some samples in this experiment also include noises.

5 Conclusion

In this study, we proposed a method that recognize whether or not there is a person in a blind spot in videos, as the first step towards recognizing targets in blind spots. The proposed method adopts a spatiotemporal 3D CNN to learn features of videos with blind spots. We confirmed the effectiveness of the proposed method using the volleyball dataset [9] with artificial blind spots.

In our future work, we will further investigate more natural experimental settings to achieve recognition in blind spots in the wild.

References

1. Baradad, M., Ye, V., Yedidia, A.B., Durand, F., Freeman, W.T., Wornell, G.W., Torralba, A.: Inferring light fields from shadows. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6267–6275 (2018) 2
2. Bouman, K.L., Ye, V., Yedidia, A.B., Durand, F., Wornell, G.W., Torralba, A., Freeman, W.T.: Turning corners into cameras: Principles and methods. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 2289–2297 (2017) 2
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the Kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4733 (2017) 1, 2
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 2
5. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: AVA: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6047–6056 (June 2018) 2
6. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imageNet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6546–6555 (2018) 1, 2, 3, 4
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 2, 3, 4
8. He, Y., Shirakabe, S., Satoh, Y., Kataoka, H.: Human action recognition without human. In: Proceedings of the ECCV Workshop on Brave New Ideas for Motion Representations in Videos. pp. 11–17 (2016) 2, 3
9. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1971–1980 (2016) 2, 3, 6, 7
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the International Conference on Machine Learning. pp. 448–4 56 (2015) 4
11. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The Kinetics human action video dataset. arXiv preprint **arXiv:1705.06950** (2017) 2, 6
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS). pp. 1097–1105 (2012) 1, 2, 6
13. Mak, L.C., Furukawa, T.: Non-line-of-sight localization of a controlled sound source. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics. pp. 475–480 (2009) 2
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the International Conference on Machine Learning. pp. 807–814. Omnipress (2010) 4
15. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015) 2, 4
16. Zhao, M., Li, T., Alsheikh, M.A., Tian, Y., Zhao, H., Torralba, A., Katabi, D.: Through-wall human pose estimation using radio signals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7356–7365 (2018) 2