A Deep Learning-Based Approach for Camera Switching in Amateur Ice Hockey Game Broadcasting

Hamid Reza Tohidypour, Yixiao Wang, Mohsen Gholami, Megha Kalia, Kexin Wen, Lawrence Li, Mahsa T. Pourazad, and Panos Nasiopoulos

Abstract Switching the camera views in ice hockey plays an important role in television broadcasting. The traditional approach is to employ a broadcast director and professional crew responsible for making decisions about switching camera views, which is not an affordable option for amateur games. In this study we propose an automatic switching scheme of camera views that is based on a deep learning-trained model which detects important objects such as players, puck, and net. Our solution uses a Faster-RCNN object detection network which was trained using a dataset of 1000 high-definition (HD)-labeled frames of hockey games. Our Faster-RCNN achieves the average precision (AP) of 0.90, 0.88, and 0.61 for players, net, and puck, respectively, using a vgg-16 pretrained model. Our camera switching approach uses the confidence values of the detected objects to predict the best camera view for the current moment. Performance evaluations showed that our method achieved accuracy of 75% for choosing the most important camera view in real time.

Keywords Multi-camera switching · Object detection; Faster-RCNN · Convolutional neural network · Video and camera tracking

H. R. Tohidypour (✉) · Y. Wang · M. Gholami · M. Kalia · K. Wen · L. Li · P. Nasiopoulos
Department of Electrical & Computer Engineering, University of British Columbia, Vancouver, BC, Canada
e-mail: htohidyp@ece.ubc.ca; yixiaow@ece.ubc.ca; mgholami@ece.ubc.ca; mkalia@ece.ubc.ca; kwen04@ece.ubc.ca; panos@ece.ubc.ca

M. T. Pourazad
Department of Electrical & Computer Engineering, University of British Columbia, Vancouver, BC, Canada

TELUS Communications Inc., Vancouver, BC, Canada
e-mail: pourazad@ece.ubc.ca

C.-C. Wang, A. Nallanathan (eds.), *Proceedings of the 5th International Conference on Signal Processing and Information Communications*, Signals and Communication Technology, https://doi.org/10.1007/978-3-031-13181-3_1

1 Introduction

It is common practice in broadcasting live sports to utilize many cameras and switch between them, depending on how the game develops. This switching involves professional operators and expensive specialized equipment. Besides the expensive equipment, professional coverage involves highly skilled individuals such as camera operators and a director responsible for supervising and deciding the overall operation.

Unfortunately, such an expense is prohibitive when it comes to broadcasting amateur community or school sports. In this case, despite the fact that more than one camera may be used, real-time coverage involves only the main view, without offering the option of watching the view that better covers crucial moments during the game.

As a result, this monotonous coverage of regional sports may potentially hinder the viewership and be detrimental in the progress of school and amateur games. Thus, there is a need for a cost-effective, fully automated camera view switching system, which analyzes the importance of the scene covered by each camera and then switches the view in a manner that is pleasant to the viewer.

To the best of our knowledge, there is only one existing work for designing automatic camera switching for ice hockey which is presented in [1]. This method performs an automatic camera selection using the hidden Markov model to create personalized video programs for users that are more interested in the performance or positions of the players from different perspectives than the game itself. In that respect, this method was player-centered rather than puck-centered or play-centered. However, our task is to design an automatic play-centered camera switching approach for amateur ice hockey games.

To this end, in this chapter, we propose a play-centered solution that is based on deep learning, namely, the Faster-RCNN architecture [2], to optimize view switching in regional ice hockey games. Our deep learning-based object recognition network receives video feeds from the two primary camera views of ice hockey and detects the players, net, and the puck in real time with very good precision. Then, based on the predicted confidence values for the different objects, our algorithm decides which camera view should be broadcasted.

The rest of this chapter is organized as follows. Section II presents our approach and explains the dataset selection and labeling of our dataset. Section III presents the performance evaluation of our method and discusses the results. Finally, Section IV concludes our chapter.

2 Our Approach

To keep viewers of an amateur ice hockey event engaged, there are primarily two types of views that are important; first, the side view that shows the arena (please see left image in Fig. 1) and gives a wide view of the field, and second, the goalie views (please see right image in Fig. 1) that show a closer view of the nets. It is common practice to use these cameras in amateur community or school ice hockey games. Since, most of the action is taking place away from the net, the primary view is the arena (side) view, while the goalie views are the secondary views. Thus, the natural question that arises here is when to show the goalie view, in other words, what are the criteria that will lead to switching from showing the side view to a goalie view. To address this challenge, we first propose the use of a deep learning-based object recognition approach that receives the video feeds from all the views and detects the players, puck, and the nets. Then, we use the weighted sum of the confidence values of the detected objects to decide which view to broadcast. Figure 1 shows our proposed scheme. Details regarding the dataset that we used to train our network and the criteria we introduced for switching camera views are presented in the following subsections.

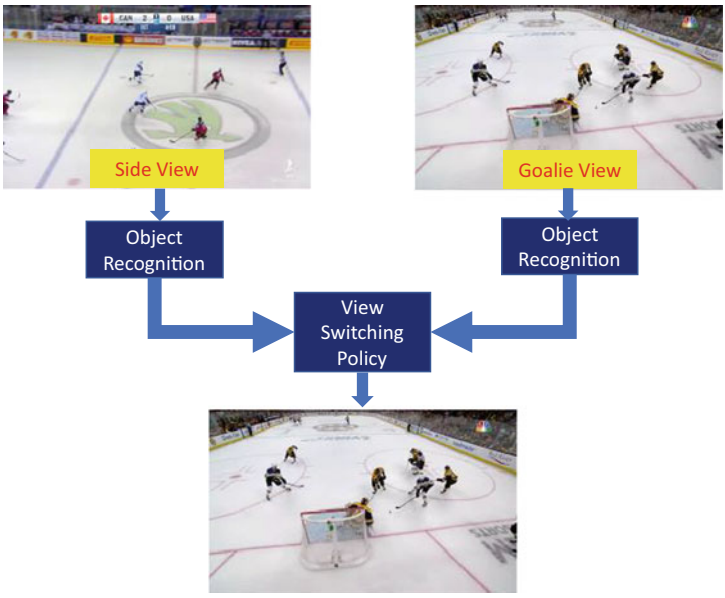

Fig. 1 Our proposed scheme for automatic camera view switching

2.1 Data Collection

In order to build a comprehensive dataset for our application, we downloaded several hockey videos of the resolution of 1920 × 1080 from YouTube [3]. The reason for turning to YouTube and not using amateur content was that we did not have access to the latter due to covid-19 restrictions, which did not allow community games to take place and could not find recorded content of previously played games. From those videos, 1000 representative frames were selected for the training-validation phase, skipping redundant frames and considering only frames with significantly different content to avoid overfitting, preferably including the puck and of high visual quality – avoided blurry, fast-moving puck frames. As already mentioned, in this study the objects of interest were the players, net, and the puck, while the referees and audience were excluded. The location of these three types of objects can be used to determine the best camera view for the current situation. An example of a labeled training frame from the side view is shown in Fig. 2a. Figure 2b shows a different example from the goalie view. For the test phase, we used four ice hockey video streams from YouTube, which were very different from the training videos [3]. These videos had the same resolution with the training videos.

2.2 Our Deep Learning Network

We chose the Faster-RCNN architecture [2] as our deep learning-based classification and object recognition network. The main reason for this choice is that Faster-RCNN is proven to be more accurate and much faster compared to its predecessors [4–6], making it an ideal approach for real-time object detection of the ice hockey fields [2]. Moreover, it also showed very promising results in detecting small objects. We trained this network to detect players, net, and the puck. Details about the network configuration and the training platform used are explained in the evaluation and discussion section.

a) b)

Fig. 2 Examples of labeled frames from our dataset (**a**) from the side view and (**b**) from the goalie view

2.3 Our Camera Switching Approach

The first task of our scheme for switching camera views is to receive the detection information of the objects of interest, i.e., the players, net, and puck, that comes out of our Faster-RCNN object recognition model. Then, our algorithm considers the position and confidence level of detection of all the objects, as each one has different roles to play in determining the best camera view for the current moment of the game. It is important to note that designing our algorithm to be biased toward the importance of objects to the fans, will allow our solution to be focused on the action. Driven by professional game coverage, we assume that the most important object/event in hockey broadcasting involves the puck, as the audience tries to follow its location when watching a hockey game. Following the above observation and the outcome of many trials asking subjects to validate the validity of our switching scheme, we assigned a weight to the confidence values predicted for each object type according to its importance: 20 for the puck, 1 for the net, and 1 for the player. More precisely, the confidence of each detected object in the current camera view is weighted according to its object type, and the weighted values are summed up to calculate the score for the current camera view. Please note that our method only considers objects with confidence values greater than 20%. In addition, we decided to add 10 to the weighted score calculated for the goalie view if the puck is present in that camera view.

Figure 3 shows the block diagram of our proposed camera switching scheme. To prevent any high-frequency camera switching, we built in a small delay of ten-frame duration (one-third of a second) before switching again after the last camera view change.

3 Evaluation and Discussion

For training, we used a PyTorch implementation of Faster-RCNN [7]. The fully connected layer of the model was changed to detect the three classes required for our application. Ninety percent of the training-validation dataset was randomly selected as the training dataset and the remaining 10% was considered for the validation phase. Horizontal data augmentation was used to augment the dataset for the training phase. For the training phase, we aimed to achieve the best performance by testing different combinations of the network configurations for this phase. To this end, two different pretrained models, namely, VGG16 and ResNet-101, were used as the backbones for the Faster-RCNN. Four batch sizes, namely, 1, 6, 12, and 24, were tried. Three different learning rates were used to achieve the best training performance: 0.0001, 0.001, and 0.01. We trained our Faster-RCNN using the Nvidia V100 Volta GPU, with 32 GB of HBM2 memory available on a state-of-the-art

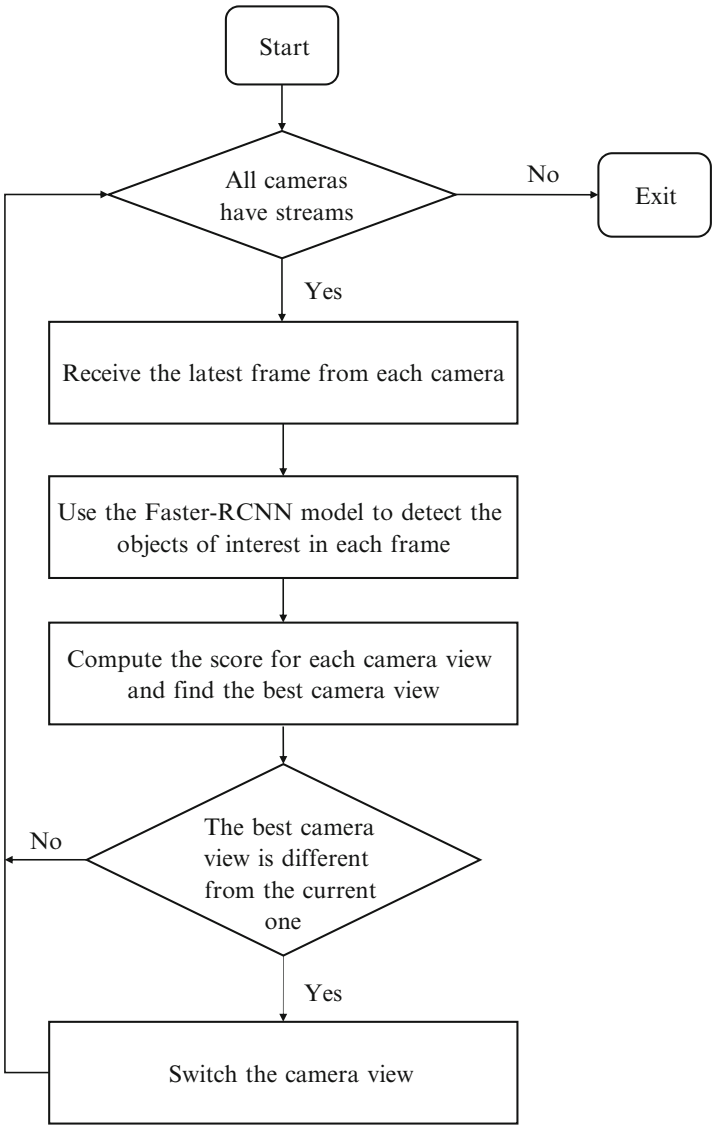

Fig. 3 Our proposed camera switching scheme

advanced research computing network [8]. Tables 1 and 2 show the average precision (AP) of the player, puck, and net classes that were achieved for the validation frames for each of the training settings. The batch sizes (bs) and learning rates (lr) used for each training setting are also reported in these tables. As can be seen in Table 1, the best AP values that were obtained by Faster-RCNN with VGG-16 backbone were 0.616, 0.876, and 0.9 for the puck, player, and net, respectively. The

Table 1 Detection results of Faster-RCNN using different settings with VGG-16 pretrained model

bs	lr	epoch	AP puck	AP player	AP net	mAP
1	0.001	20	0.205	0.821	0.518	0.518
6	0.001	20	0.565	0.876	0.896	0.779
12	0.001	20	0.382	0.876	0.875	0.711
24	0.01	20	0.52	0.88	0.875	0.758
12	0.01	20	**0.616**	**0.876**	**0.9**	**0.789**
12	0.0001	20	0.205	0.821	0.518	0.518

Table 2 Detection results of Faster-RCNN using different settings with ResNet-101 pretrained model

bs	lr	epoch	AP puck	AP player	AP net	mAP
1	0.001	15	0.489	0.865	0.844	0.732
6	0.001	20	0.565	0.876	0.896	0.779
12	0.001	20	0.382	0.876	0.875	0.711
24	0.01	20	0.489	0.865	0.844	0.732
12	0.01	20	**0.587**	**0.878**	**0.87**	**0.778**
12	0.0001	20	0.347	0.86	0.7	0.636

mean average precision (mAP) was 0.789. This performance was achieved using the learning rate of lr = 0.01 and batch size bs = 12. According to Table 2, the best AP obtained with the ResNet-101 pretrained model was 0.587, 878, and 0.778 for puck, player, and net, respectively. The mAP for this case was 0.778. This clearly shows that VGG-16 pretrained model, batch size bs = 12, and learning rate lr = 0.01 achieved the best performance among all the training settings examined in our study. Therefore, for the test phase, we used this model and we call it our model, hereafter.

In order to evaluate the performance of the trained model, we examined the trained model on the test videos with unseen frames. Our results showed that for most of the frames, our model detected the players, net, and the puck correctly. Figure 4 shows the predicted objects and the probability values assigned to the bounding boxes for four successful examples. However, there were some false positives and false negatives for the puck during the test phase. Some examples are shown in Fig. 5. Our results showed that our model detected the puck correctly when it was fully visible to the camera (see Fig. 5a). Since puck is a very small object compared to the players and the net, this contributes to the low AP of the puck. More precisely, our model could not detect the puck when it was not fully visible or blurry to the camera (Fig. 5b). Figure 5a shows an example of false positive for the puck, a case we observed only for few frames. In this rare case, the toe of the hockey stick was detected as a puck since its color was the same as the puck and the color of the hockey stick's blade was the same as the ice hockey field. As can be seen, in this case, two objects were detected as the puck. Thus, we decided to only consider the object with the highest predicted confidence. This approach significantly improved our accuracy. This issue may be resolved by adding frames with a similar scenario to the original training frames. Finally, we used the information of the objects detected

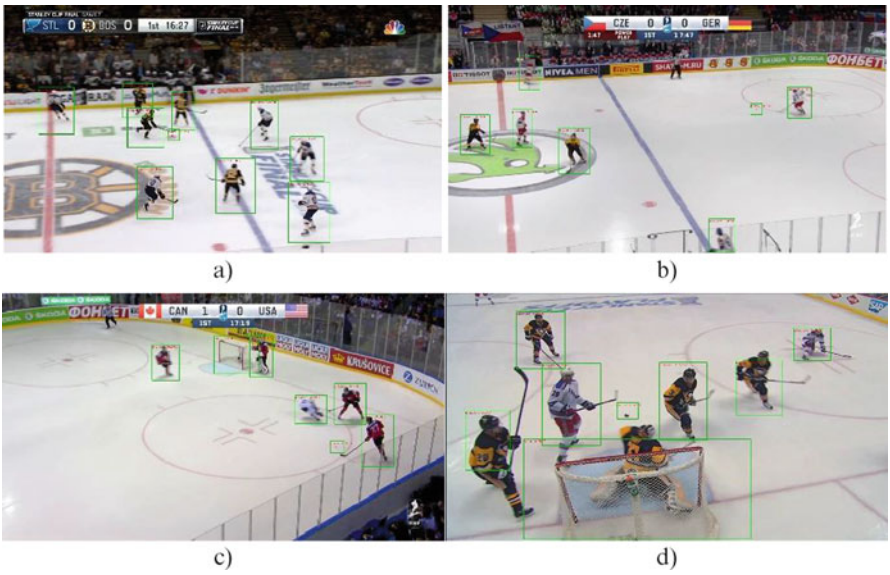

Fig. 4 Example frames from the test set: (**a**, **b**) players and the puck that were detected correctly by our model; (**c**, **d**) players, the puck, and one of the nets that were detected correctly by our model

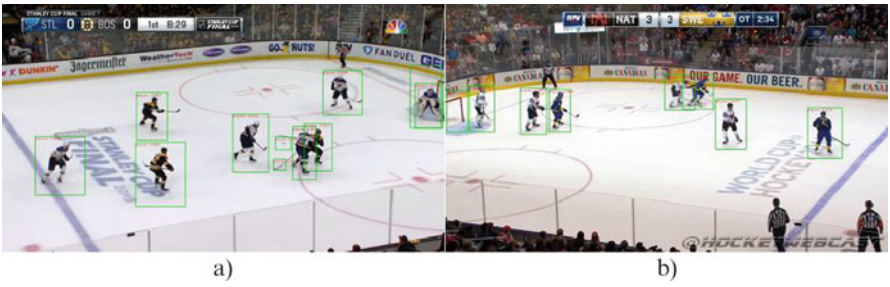

Fig. 5 Example frames from the test set: (**a**) players, the net, and the puck that were detected correctly by our model as well as a false positive puck; (**b**) players and the net that were detected correctly by our model as well as a non-visible puck (false negative)

by our model for our automatic camera view switching approach to detect the instances for which the view switching was needed. Our results show an accuracy of 75% for our camera switching method in real-time. Considering the fact that only 1000 frames were used for the training and validation phases, our camera switching approach achieved a great performance.

Acknowledgments This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC – PG 11R12450) and TELUS (PG 11R10321). This research was enabled in part by support provided by WestGrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca).

References

1. L. Wu, Multi-view hockey tracking with trajectory smoothing and camera selection. Thesis, University of British Columbia. Retrieved from https://open.library.ubc.ca/collections/ubctheses/24/items/1.0051270 (2008)
2. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn. Towards real-time object detection with region proposal networks. Adv. Neural Inform. Proc. Syst., 91–99 (2015)
3. 2019 IIHF Ice Hockey World Championship, IHF Worlds 2021, YouTube, https://www.youtube.com/c/IIHFWorlds/videos. Last accessed 2021/11/17
4. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. IEEE conference on computer vision and pattern recognition (ICCV)*, (Columbus, 2014), pp. 580–587
5. R. Girshick, Fast r-cnn, in *Proc. IEEE international conference on computer vision*, (2015), pp. 1440–1448
6. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
7. S. Ren, Faster-RCNN, GitHub repository, https://github.com/ShaoqingRen/faster_rcnn. Last accessed 2021/11/17
8. Compute Canada state-of-the-art advanced research computing network. https://www.computecanada.ca. Last accessed 2021/11/17