TrackNetV2: Efficient Shuttlecock Tracking Network

Nien-En Sun Yu-Ching Lin Shao-Ping Chuang Tzu-Han Hsu Dung-Ru Yu Ho-Yi Chung
Tsì-Uí İk*
Department of Computer Science, College of Computer Science
National Chiao Tung University
1001 University Road, Hsinchu City 30010, Taiwan
*Email: cwyi@nctu.edu.tw

***Abstract*—TrackNet, a deep learning network, was proposed to track high-speed and tiny objects such as tennis balls and shuttlecocks from videos. To conquer low image quality issues such as blur, afterimage, and short-term occlusion, some number of consecutive images are input together to detect an flying object. In this work, TrackNetV2 is proposed to improve the performance of TrackNet from various aspects, especially processing speed, prediction accuracy, and GPU memory usage. First of all, the processing speed is improved from 2.6 FPS to 31.8 FPS. The performance boost is achieved by reducing the input image size and re-engineering the network from a Multiple-In Single-Out (MISO) design to a Multiple-In Multiple-Out (MIMO) design. Then, to improve the prediction accuracy, a comprehensive dataset from diverse badminton match videos is collected and labeled for training and testing. The dataset consists of 55563 frames from 18 badminton match videos. In addition, the network mechanisms are composed of not only VGG16 and upsampling layers but also U-net. Last, to reduce GPU memory usage, the data structure of the heatmap layer is remodeled from a pixel-wise one-hot encoding 3D array to a real-valued 2D array. To reflect the change of the heatmap representation, the loss function is redesigned from a RMSE-based function to a weighted cross-entropy based function. An overall validation shows that the accuracy, precision and recall of TrackNetV2 respectively reach 96.3%, 97.0% and 98.7% in the training phase and 85.2%, 97.2% and 85.4% in a test on a brand new match. The processing speed of the 3-in and 3-out version TrackNetV2 can reach 31.84 FPS. The dataset and source code of this work are available at https://nol.cs.nctu.edu.tw:234/open-source/TrackNetv2/.**

***Index Terms*—Badminton, shuttlecock tracking, deep learning, heatmap**

I. Introduction

Badminton is one of the most popular sports in the world. The size of the badminton-related market is huge such as ticket revenue, broadcast rights, coaching market, sports equipment, and even sports lotteries, etc. Not surprisingly, this reflects that the professional competition becomes increasingly fierce, and apparently how to win has become a hot topic. In recent years, precision sport science has become a trend. Tactical analysis and scientific training often determine the outcome of matches [1]. Microscopic data is essential for tactical analysis. The game video retains a lot of detailed contest data and is an important material for collecting microscopic data.

The microscopic data of badminton provide shot-by-shot spatial, temporal, posture and ball usage information such as shuttlecock hitting times, players' postures and foot positions, and ball types. In the match video, shuttlecock trajectories would be the primary reference to compute hitting times and calculate other shot attributes. Hawk-Eye [2] was developed to precisely detect the ball trajectories in many professional ball games, including badminton, tennis balls, football, cricket, golf, and many other ball games. The system consists of multiple synchronized high-end cameras to build 3D ball trajectories, which can help both umpire's judgement and athlete's training. But Hawk-Eye is a proprietary system and with high cost, that is not affordable to most non-professional games and is not possible for widely deployed. Therefore, it is expected to compute the trajectory from video that is widely available such as broadcast video [3]. However in broadcast video, trajectory computing is not an easy task due to the blurry images of the tiny and high-speed shuttlecock as shown in Fig. 1.

Fig. 1. Badminton is the world's fastest racquet sport. The shuttlecock not only tiny but also can fly over 300 kilometres per hour, making it blurred in image and hard to locate accurately by broadcast videos.

Thanks to the recent advancement in deep learning, TrackNet [4] was developed to track the tennis ball or shuttlecock from broadcast match video. A heatmap that represents the probability distribution of the ball center is calculated from several consecutive images by a VGG16-based feature extraction network [5] followed by an upsampling network [6] to generate the heatmap. To conquer blurry of images, TrackNet takes multiple consecutive frames instead of a single one as

978-1-6654-0483-9/20/$31.00 ©2020 IEEE
DOI 10.1109/ICPAI51961.2020.00023

input to benefit detection implicitly from trajectory patterns. It was reported that the precision, recall, and F1-measure are respectively 95.3%, 75.7%, and 84.3% in a 10-fold cross validation. However, the processing speed of TrackNet is not fast enough for real-time applications, and the network design consumes many GPU memory. Moreover, although it has been demonstrated that the tennis model can be transferred for badminton, the accuracy is not as good as the original one due to faster speed and more blurrier images.

In this work, TrackNetV2 tuned for badminton is proposed to improve the performance of TrackNet from various aspects including processing speed, prediction accuracy, and GPU memory usage. The performance improvement is achieved by reducing the input image size from 640×360 to 512×288, and re-engineering the network design from a Multiple-In Single-Out (MISO) design to a Multiple-In Multiple-Out (MIMO) design that significantly boost the processing speed. To improve the accuracy on badminton video, a comprehensive badminton dataset is collected for training and testing. In addition, besides VGG and upsampling networks, the idea of skip connection of U-Net [7] is adopted. To reduce the demain on GPU memory, the heatmap data representation is remodeled from a pixel-wise one-hot encoding 3D array to a real-valued 2D array. To incorporate with the change, the loss function is redesigned from a RMSE-based version to a weighted cross-entropy based version.

Overall speaking, compared with TrackNet, the 3-in and 3-out version TrackNetV2 boosts the processing speed from 2.6 FPS to 31.8 FPS, provides higher accuracy, recall, precision, and reduces memory usage . TrackNetV2 has been integrated with a badminton video labeling tool to accelerate microscopic data collection. In summary, the main contributions of this work include:

1. TrackNetV2, an upgraded version of TrackNet, is proposed to predict the trajectory of the fast-moving and tiny shuttlecock from match video. The processing speed, accuracy, and memory usage are improved.
2. A comprehensive badminton dataset is published to benefit the community for future research.

The rest of this paper is organized as below. In Section II, literature review and background knowledge are given. In Section III, the proposed deep learning network and training details are introduced. In Section IV, performance evaluation from the aspect of processing speed, accuracy, etc. is provided. Conclusion for this work is given in Section V.

II. Preliminary

The Inertial Measurement Unit (IMU) sensors have been used to compute golf swing paths [8] and classify tennis or badminton stroke types [9]. In swing analysis and stroke type classification, peak detection or threshold based algorithms were proposed to point out the moment of hitting for sensor data segmentation. However, The accuracy of hitting time are not accuracy enough, and false detection happens from time to time. Later on, voice fingerprint of acoustic data was introduced to make up the weaknesses by applying acoustic sensors [9]. But wearable sensors are not allowed in some professional games or may impede the athletes' performance.

The Hawk-eye [2] system for ball tracking has been widely used in various sport such as football, tennis, baseball, etc. Since the system needs high-quality and well-deployed cameras, proprietary software, and professional operators, the operating cost is high. In recent years, computer vision based ball tracking from common quality video such as broadcast video or home video have been investigated [10]. It is an easy task since the image of the ball is small, with high-speed, and sometimes confused with the background.

TrackNet [4], a deep learning solution, was proposed to detect tennis ball and shuttlecock from broadcast video. This heatmap based deep learning network combines VGG16 [5] and the upsampling mechanism of FCN. The former is a classical way of feature map extraction; the latter decodes the feature map to generate a probability-like heatmap in original size as the input frames. The target's location will be indicated on the heatmap. Instead of single frame input, TrackNet takes 3 consecutive frames as input to learn trajectory patterns. In a 10-fold cross validation on 20844 frames from the broadcast video of men's tennis single final at the 2017 Summer Universiade, the results revealed 95.3% of precision, 75.7% of recall, and 84.3% of F1-measure. Benefited by the multiple frame approach, the location of object can refer not only current frame but also previous and future information, provides the capability of predicting ball's location even in the case of invisible, due to confusing background or foreground occlusion. However, the processing speed is not fast enough for realtime applications, and the performance on shuttlecock tracking is not as good as the tennis case.

We refer the Encoder-Decoder structure of TrackNet and add some concept from U-Net [7] which is convolution network architecture for fast and precise segmentation of images. This work is usually discussed in biomedical image segmentation, like [11] [12] [13]. But an ingenious idea from the U-Net is skip connection. The feature map obtained by each convolutional layer of the U-Net network will be concatenate to the corresponding upsampling layer. In this way, U-Net combines the low-level features and high-level features to improve the accuracy.

Heatmap-based CNNs have been demonstrated useful in solving problems [14] [15]. In the TrackNet, the output of network is 255 channels and each channel denote the probability of the pixel. In this way, the network is hard to converge and trained quite slow. Therefore, in [16] which is multi-task deep learning network with machine-train-machine migration learning, it used weighted masks method to more amplification the human body keypoints. Referring to this method, we also weight the center point of the shuttlecock when calculating loss function.

III. TrackNetV2

The design of TrackNetV2 is illustrated in Fig. 2. Like TrackNet and many other CNNs, TrackNetV2 follows the

encoder-decoder structure. The encoder, as a feature extractor, captures the image clue by the convolution kernels and condenses the features by the max-pooling operation. On the other hand, the decoder expands the feature map to generate the prediction function. In the design, VGG16 is adopted as the encoder to generate the feature map of $512 \times 64 \times 36$, and then an upsampling structure corresponding to the downsmapling structure of the encoder follows to generate the prediction heatmaps with the same size of the input images. Compared to TrackNet, the input image size is reduced from 640×360 to 512×288 to speed up processing speed and reduce memory usage but keep similar accuracy and precision. Since this is a straightforward approach, it will not be further discussed. The major enhancements including the Multiple-In Multiple-Out (MIMO) design, introduction of the U-net structure, and renovation of the heatmap representation and loss function will be introduced below. The details of model is given in Table. I.

Layer	Details	Output size
input	npy file	$512 \times 288 \times 9$
conv2d_1	$3 \times 3 \times 64$; relu; batch norm	$512 \times 288 \times 64$
conv2d_2	$3 \times 3 \times 64$; relu; batch norm	$512 \times 288 \times 64$
max_pooling_1	2×2 max pool; stride 2	$256 \times 144 \times 64$
conv2d_3	$3 \times 3 \times 128$; relu; batch norm	$256 \times 144 \times 128$
conv2d_4	$3 \times 3 \times 128$; relu; batch norm	$256 \times 144 \times 128$
max_pooling_2	2×2 max pool; stride 2	$128 \times 72 \times 128$
conv2d_5	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
conv2d_6	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
conv2d_7	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
max_pooling_3	2×2 max pool; stride 2	$64 \times 36 \times 256$
conv2d_8	$3 \times 3 \times 512$; relu; batch norm	$64 \times 36 \times 512$
conv2d_9	$3 \times 3 \times 512$; relu; batch norm	$64 \times 36 \times 512$
conv2d_10	$3 \times 3 \times 512$; relu; batch norm	$64 \times 36 \times 512$
up_sampling_1	2×2	$128 \times 72 \times 512$
concatenate_1	with conv2d_7; axis = 1	$128 \times 72 \times 768$
conv2d_11	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
conv2d_12	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
conv2d_13	$3 \times 3 \times 256$; relu; batch norm	$128 \times 72 \times 256$
up_sampling_2	2×2	$256 \times 144 \times 256$
concatenate_2	with conv2d_4; axis = 1	$256 \times 144 \times 384$
conv2d_14	$3 \times 3 \times 128$; relu; batch norm	$256 \times 144 \times 128$
conv2d_15	$3 \times 3 \times 128$; relu; batch norm	$256 \times 144 \times 128$
up_sampling_3	2×2	$512 \times 288 \times 128$
concatenate_3	with conv2d_2; axis = 1	$512 \times 288 \times 192$
conv2d_16	$3 \times 3 \times 64$; relu; batch norm	$512 \times 288 \times 64$
conv2d_17	$3 \times 3 \times 64$; relu; batch norm	$512 \times 288 \times 64$
conv2d_18	$1 \times 1 \times 3$; relu	$512 \times 288 \times 3$

TABLE I
TRACKNETV2 MODEL STRUCTURE.

A. Skip Conncections

The skip connection mechanism from U-Net [7] is adopted. In the classical waterfall-like design, the features of tiny objects could be gradually diminished along the long pipeline of convolution layers and max-pooling layers. The skip connections that pass feature arrays in the encoder network directly to the layers in the encoder network provide shortcuts to preserve information of tiny objects . This feature helps track the tiny and fast-moving shuttlecock.

B. Heatmap Representation

TrackNet generates a probability-like detection heatmap. The ground truth of the heatmap is an amplified 2D Gaussian distribution function centered at the position of the shuttlecock. Fig. 3 illustrates the distribution function. The heatmap is represented by a 256-channel boolean array in the pixel-wise one-hot encoding convention, and then the $argmax$ operation is applied to have the final prediction heatmap. This is inefficient. Alternatively, TrackNetV2 directly generates an 1-channel real-valued array in the last sigmoid layer. The modification decreases memory usage and increases run-time speed.

C. Loss function

In TrackNet, a softmax layer pixel-wisely generates the probability distribution of the heatmap value. See the subfigure in the top of Fig. 3. In the backpropagation, the pixel-wise cross entropy is adopted as the loss function. The softmax operation provide the normalization functionality that help the convergence. In TrackNetV2, the heatmap is generated by a sigmoid layer and the softmax layer no longer exists. See the subfigure in the bottom of Fig. 3. Without the normalization mechanism and due to the extreme imbalance between the small object and large scene, the model tends to predict each pixel as part of the background scene.

To guide the model more focusing on the shuttlecock, a weighted loss function $WBCE$ defined by Eq. (1) is proposed .

$$WBCE = -\sum_{i=1}^{n}(1-w)^2\hat{y}_i \log(y_i) + w^2(1-\hat{y}_i)\log(1-y_i) \quad (1)$$

In the equation, n is the index of pixels, $\hat{y}$ is the ground truth label that is 1 for the ball image pixels and 0 for the scene image pixels, y is the model prediction that is a value between 0 and 1, w is the weighted coefficient given by $w = y$. The coefficient w will encourage the model to focus more on the pixels with larger residual. Fig. 4 is an example that shows the difference between before and after applying the weighted mechanism.

Besides the WBCE loss function, weighted L2 and L1 loss functions were investigated, but the results revealed that both L2 and L1 versions loss can not have stable convergence.

D. MIMO Network

Benefitted by the streamlined heatmap representation, it is possible to generate multiple heatmaps for all input images at the same time instead of only one for one of the input images. Although MIMO design slightly reduce the processing speed, since the overhead is only in the last layer, the actual throughput of the model is multiples time increased. For example, a 3-in 3-out design almost triples the processing speed of an 3-in 1-out design. This design boosts the performance significantly.

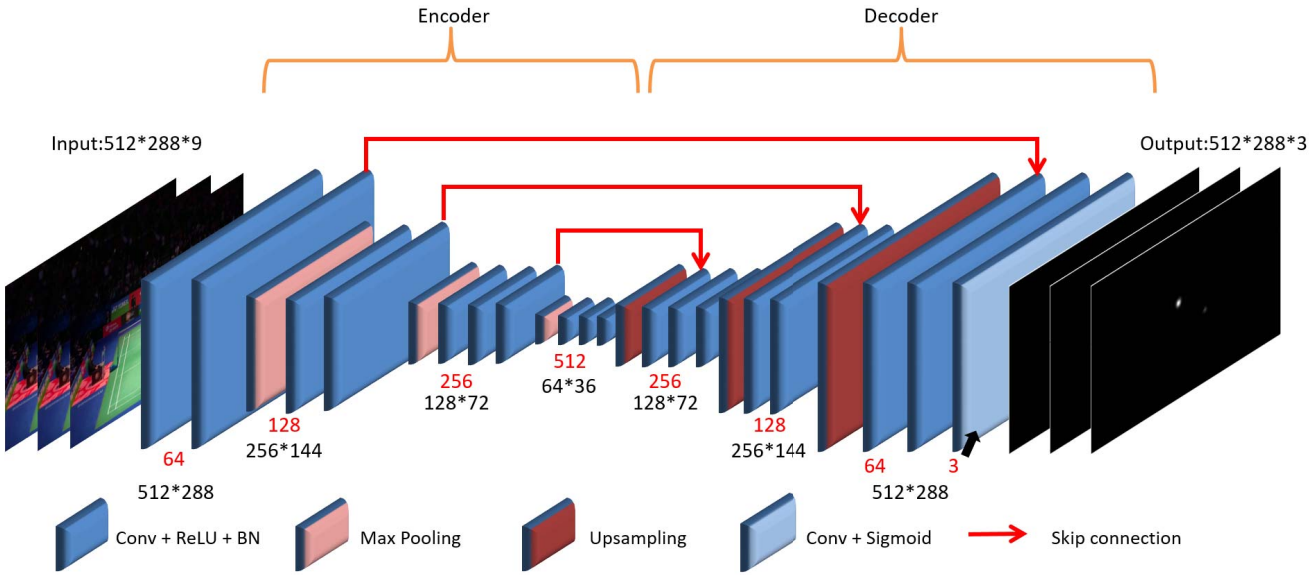

Fig. 2. The architecture of TrackNetV2. The encoder-decoder structure basically follows the design of TrackNet. Compared with the orignal TrackNet, the innovation includes the downsizing of input images, skip connections enlightened by U-net, concise heatmap representation, and MIMO network design.

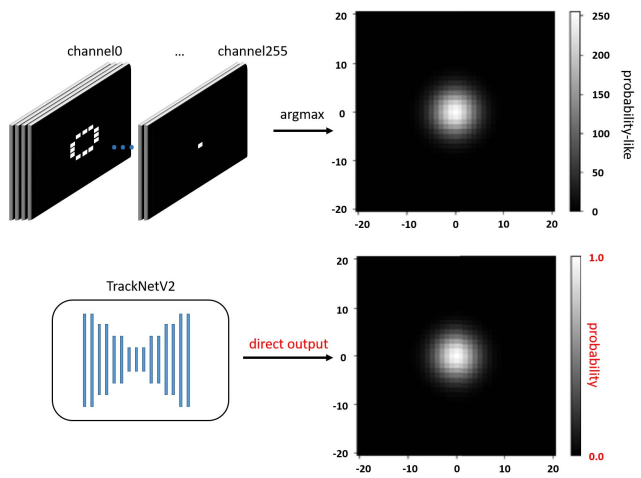

Fig. 3. TrackNet generates a 256-channel boolean array for the ball detection. Instead, TrackNetV2 directly generates a real-valued array to have the heatmap.

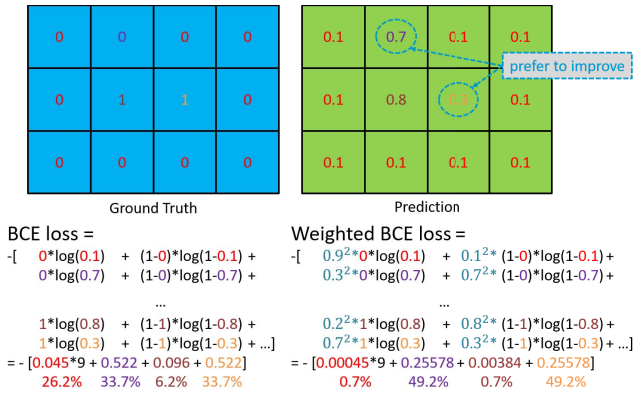

Fig. 4. Weighted Binary Cross Entropy. This figure gives an example of that without the weighting coefficient, a large amount of low residual shows in red numbers may account for quite a lot percentage of the final loss value. Using weighted binary cross entropy, the model can focus on the large residual part and increase the performance.

E. Comprehensive Training Dataset

To avoid the overfitting problem that frequently happens in deep learning training and further improve the performance, a comprehensive dataset was collected from 15 CHOU Tien Chen's and TAI Tzu Ying's competitions from 2018 to 2020. The videos were taken from various view angles, and the courts may have different colors. Fig. 5 illustrates 15 snapshots from 15 of the videos. Not only the professional games, we also collected 3 amateur games to make our dataset more general. The total frames of dataset are 55563, including 46038 frames from professional game and 9525 frames from amateur.

IV. Experiment Results

A. Dataset

The dataset contains 55563 frames used in the experiments consists of 15 broadcast videos of professional games and 3 amateur games . Since the networks need to take consecutive frames, videos are segmented into rallies, and the dataset is composed of clips of rallies. To prevent overfitting, there are 125 rally videos with different backgrounds and filming angles are collected in the dataset, and 2500 to 3000 frames from each video were included. The label data include the coordinate (x, y) and visibility of the shuttlecock. The shuttlecock coordinate will be used to generate the ground truth of the detection heatmaps, and the visibility will be used to categorize

Fig. 5. To avoid overfitting, the dataset contains rally clips from 15 broadcast video taken from various view angles and played on courts of various colors.

the hardness of the detection task. In TrackNet, the input image size is 640×360, but in TrackNetV2, it is further down to 512×288 to speed up both training process and inference process.

We also prepared the testing data to evaluate the performance, which is the match never seen in training phase consist of 13200 frames, in order to measure if our new network has the ability to do inference on frames that the backgrounds have never been seen. The testing data were prepared from 10 minutes video of badminton games, that is Tai Tzu Ying versus Chen Yufei 2018 All England Open Semi Final.

B. Training

The weighted binary cross entropy defined by Eq. (1) is the loss function used in the model training, and Adadelta optimizer with initial learning rate 1.0 is used to optimize the network parameters. The hyper parameters of TrackNetV2 is given in Table.II.

Hyperparameters	Values
Min. Kernels	64, 128, 256, 512
Kernel size	3
Kernel initializer	uniform
Pool method	max
Activations	relu
Optimizer	Adadelta
Learning rate	1.0
Epochs	30
Tolerance variable	4

TABLE II
TRACKNETV2 HYPERPARAMETERS.

C. Prediction Accuracy

In the inference, the last convolution layer with the sigmoid activation function outputs the heatmap of values in the range of 0 to 1. Then, 0.5 is selected as a threshold to turn each value to either 0 or 1. The ball location is considered as the center of the largest area in the 0-1 heatmap. If the Euclidean distance between the predicted coordinate and the ground truth is smaller than a given threshold, that is 4 pixels in the experiments, the prediction is considered as a true positive.

Table.III and Table.IV show the performances of TrackNetV2 under 3 in 1 out and 3 in 3 out structure, respectively. , FP1 represents the situation that both prediction and ground truth are ball existing, but the distance between two balls is out of tolerance value. FP2 represents the situation that the prediction is ball existing, but the ground truth is no ball. By doing inference under the 3 in 1 out structure, we got accuracy 83.9%, precision 90.7% and recall 89.2% on testing data, which the court backgrounds unseen by TrackNetV2. The performances under 3 in 3 out structure are similar to 3 in 1 out structure, getting accuracy 85.2%, precision 97.2% and recall 85.4% on testing data. Note that the quantity of the predicted frames under 3 in 3 out structure is three times as much as that under 3 in 1 out structure because each time 3-in and 1-out structure predicts a frame, 3-in and 3-out structure predicts three frames.

Model	Total	TP	TN	FP1	FP2	FN
3 in 1 out	13064	9447	1514	751	218	1134
3 in 3 out	39192	29129	4264	468	358	4973

TABLE III
DETAILS ABOUT THE CONFUSION MATRIX REPRESENTING THE PREDICTIONS MADE BY TRACKNETV2 .

Model	Acc.	Prec.	Rec.
3 in 1 out	83.9	90.7	89.2
3 in 3 out	85.2	97.2	85.4

TABLE IV
THE ACCURACY, PRECISION AND RECALL OF TRACKNETV2.

D. Processing Speed and Network Size

The speedup of processing speed comes from remodeling of heatmap representation, downsizing of input images downsizing, and MIMO network design. Table V shows the processing speed in FPS of TrackNet 640×360 3-in 1-out, TrackNetV2 640×360 3-in 1-out, TrackNetV2 512×288 3-in 1-out, and TrackNetV2 512×288 3-in 3-out.

We can see that the computing speed has improved a lot in TrackNetV2, especially under 3 in 3 out structure, the computing speed can reach 31.84 FPS. At the same time, comparing to Table.III, We can see that the processing speed is increased, but the accuracy is not downgraded. TrackNetV2 not only improve the speed during inference, but also retain the performance to some extent.

Although we thought that model changed the second to last layer output from the probability distribution of depth from possible 256 grayscale values(0-255) to confidence score(0-1) at each pixel were able to reduce the usage of GPU memory, the result was non-significant. This is perhaps we increase model's parameters by MIMO design.

Model	Input size	heatmap	skip connection	Speed
TrackNet 3-1	640 * 360			2.64 FPS
TrackNetV2 3-1	640 * 360	√		10.42 FPS
TrackNetV2 3-1	512 * 288	√		14.15 FPS
TrackNetV2 3-1	512 * 288	√	√	12.88 FPS
TrackNetV2 3-3	512 * 288	√	√	31.84 FPS

TABLE V
SPEED COMPARISONS BETWEEN TRACKNET AND TRACKNETV2

V. CONCLUSION

In this paper, we proposed TrackNetV2, a improve version of TrackNet which is a deep learning network combine with VGG16 and upsampling layers. Compared with TrackNet, TrackNetV2 not only inherit the advantage, such as network architecture and consecutive images input, but also add some other features. Skip connections keep boundaries feature after deep convolution network. Also TrackNetV2 change softmax layer to sigmoid layer in order to reduce memory space by reducing the output range, and using weighted cross entropy to compute the loss of the model. Finally, the output of the network is a heatmap of object location. For the promotion given above, TrackNetV2 become much more resource efficiency and easily to train. Furthermore, we build a new dataset of badminton which consists 55563 training images. Result shows that our model can achieve accurately prediction by learning trajectory information and reach realtime image processing by changing output images from one to three. Future work will include improving the speed by changing 3-in and 3-out structure to 5-in and 5-out, and evaluating more modern loss function. Hopefully we'll be able to reach more than real-time applications, and overcome different backgrounds problem to locate high-speed and tiny object's more precisely.

ACKNOWLEDGMENT

This work of T.-U. İk was supported in part by the Ministry of Science and Technology (MOST), Taiwan under grant MOST 109-2627-H-009-001. This work was financially supported by the Center for Open Intelligent Connectivity from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

[1] T.-H. Hsu, C.-H. Chen, N. P. Ju, T.-U. İk, W.-C. Peng, C.-C. Wang, Y.-S. Wang, Y.-H. Lin, Y.-C. Tseng, J.-L. Huang, and Y.-T. Ching, "CoachAI: A project for microscopic badminton match data collection and tactical analysis," in *The 20th Asia-Pacific Network Operations and Management Symposium (APNOMS 2019)*, Matsue, Japan, 18-20 September 2019.

[2] Wikipedia Contributors, "Hawk-Eye - Wikipedia, the free encyclopedia," 2019, [Online; accessed 20-May-2019]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hawk-Eye&oldid=894277089

[3] H. C. Shih, "A survey of content-aware video analysis for sports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 1212–1231, May 2018.

[4] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng, "TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications," in *The 1st IEEE International Workshop of Content-Aware Video Analysis (CAVA 2019) in conjunction with the 16th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS 2019)*, Taipei, Taiwan, 18-21 September 2019.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*, Santiago, Chile, 11-18 December 2015, pp. 1520–1528.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[8] Y.-C. Huang, T.-L. Chen, B.-C. Chiu, C.-W. Yi, C.-W. Lin, Y.-J. Yeh, and L.-C. Kuo, "Calculate golf swing trajectories from IMU sensing data," in *Proceedings of the 41st International Conference on Parallel Processing Workshops (ICPPW 2012)*, Pittsburgh, PA, USA, 10-13 September 2012, pp. 505–513.

[9] J. Lin, C.-W. Chang, C.-H. Wang, H.-C. Chi, C.-W. Yi, Y.-C. Tseng, and C.-C. Wang, "Design and implement a mobile badminton stroke classification system," in *Proceedings of the 19th Asia-Pacific Network Operations and Management Symposium (APNOMS 2017)*, Seoul, Korea, 27-29 September 2017, pp. 235–238.

[10] M. Archana and M. K. Geetha, "Object detection and tracking based on trajectory in broadcast tennis video," *Procedia Computer Science*, vol. 58, pp. 225–232, 2015.

[11] T. Falk, D. Mai, R. Bensch, u. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Bẗohm, J. Deubner, Z. Jaeckel, K. Seiwald, O. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. Tay, M. Prinz, K. Palme, M. Simons, and O. Ronneberger, "U-net: deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, 01 2019.

[12] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation," 2018.

[13] S. Li and G. K. F. Tso, "Bottleneck supervised u-net for pixel-wise liver and tumor segmentation," 2018.

[14] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 468–475.

[15] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 717–732.

[16] B.-W. Shih, "Multi-task deep learning networks with machine-train-machine migration learning for pose estimation and depth prediction," M. Eng. Thesis, National Chiao Tung University, Hsinchu City, Taiwan, 2019, advised by Tsì-Uí İk.