

PAPER REVIEW

TTNet: Real-time temporal and spatial video analysis of table tennis

Presenter: Nguyen Mau Dung

Advisor: Professor Jinwook Kim

Seoul, May 2020

- 1. Objectives and Challenges**
- 2. Proposed methods**
- 3. Experiments**
- 4. Results**
- 5. Limitations**

Objectives

- **Real-time** processing of high-resolution table tennis videos
- Events spotting (ball bounces and net touches)
- Ball detection
- Semantic segmentation (humans, table, and scoreboard)



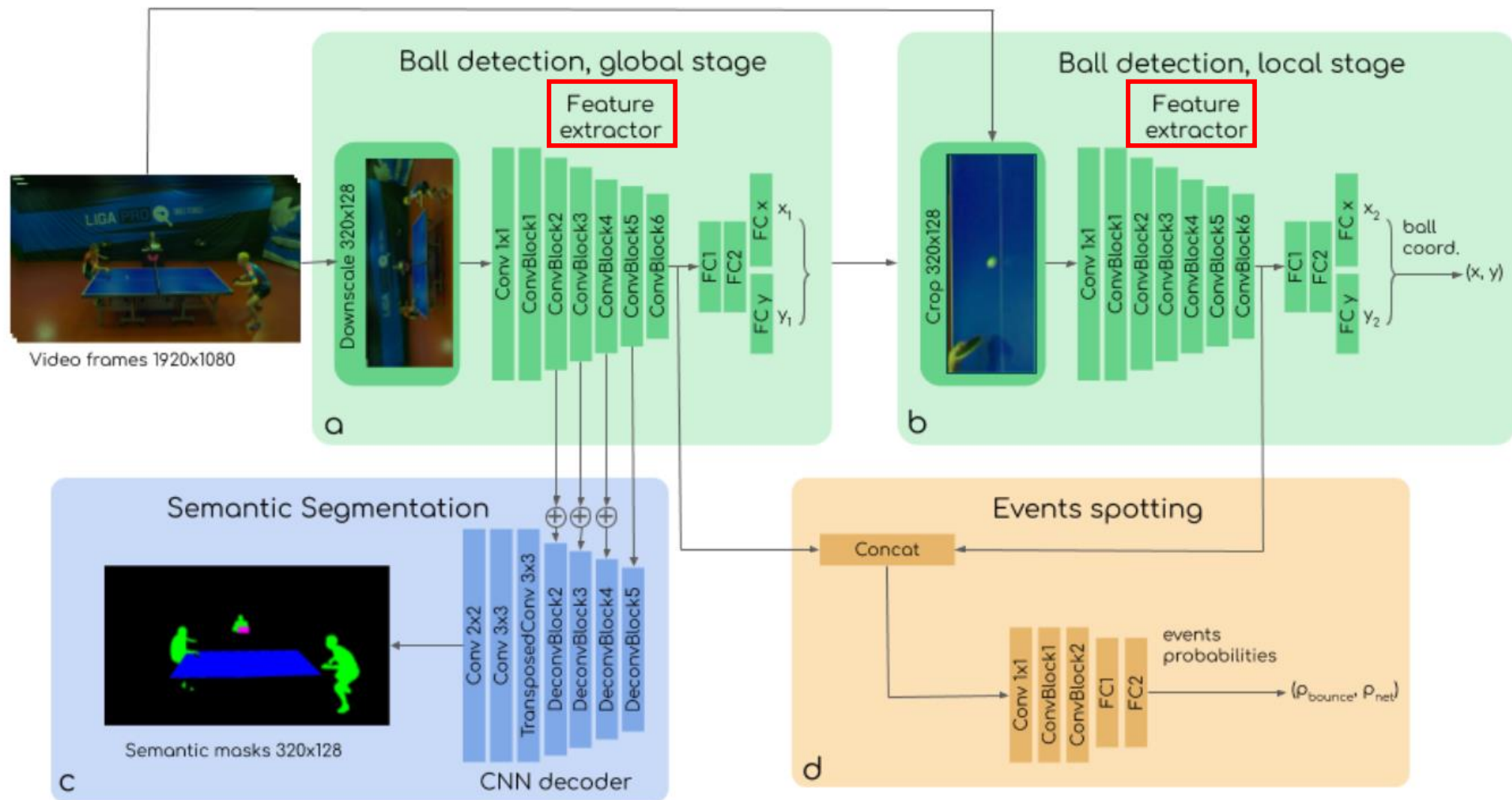
Sample frame with the TTNNet predictions overlaid

Challenges

- The **size of the ball** in a full HD video is quite **small** (about 15 pixels on average).
- The ball may be not the only small white object in the image because of the parts of player clothing or background elements
- **The ball speed**: May be more than 30 m/s (~100 km/h)
 - video data with a high frame rate is required to detect the trajectory of the ball and corresponding events

2. Proposed methods

TTNet architecture Use *VGG-style* feature extractor



2. Proposed methods

Ball detection

Operator	In-channels/ Out-channels	stride	padding
Conv 3x3	a/b	1	1
BatchNorm	b/b	-	-
ReLU	b/b	-	-
MaxPool 2x2	b/b	2	0

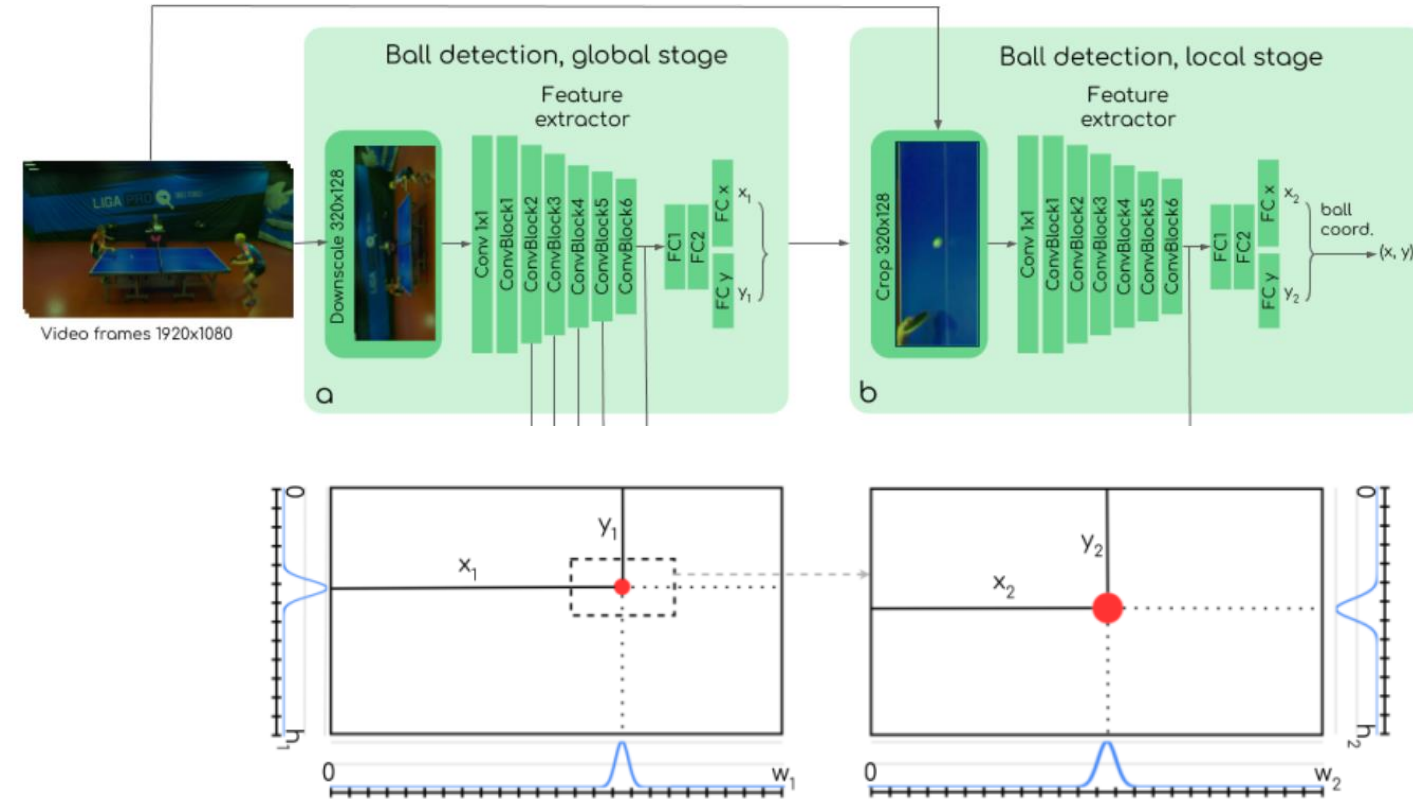
Structure of the ConvBlock

Input size	Operator	In-channels/ Out-channels
320 x 128	Conv 1x1	27/64
320 x 128	BatchNorm	64/64
320 x 128	ReLU	64/64
320 x 128	ConvBlock	64/64
160 x 64	ConvBlock	64/64
80 x 32	DropOut	64/64
80 x 32	ConvBlock	64/128
40 x 16	ConvBlock	128/128
20 x 8	DropOut	128/128
20 x 8	ConvBlock	128/256
10 x 4	ConvBlock	256/256
5 x 2	DropOut	256/256
5 x 2	Flatten	256/-
2560	FC	-
1792	ReLU	-
1792	DropOut	-
1792	FC	-
640/256	ReLU	-
640/256	DropOut	-
320/128	FC	-
320/128	Sigmoid	-

Structure of the Ball Detection part

2. Proposed methods

Ball detection



Target construction for two stages of the ball detection

The target is two **1D Gaussian distribution** curves with means associated with the x and y of the ball center, respectively.

The final ball coordinates are derived from the two stages

$$x = x_1 \frac{w_0}{w_1} - \frac{w_2}{2} + x_2; \quad y = y_1 \frac{h_0}{h_1} - \frac{h_2}{2} + y_2$$

$$w_0 = 1920px; h_0 = 1080px$$

$$w_1 = 320px; h_1 = 128px$$

$$w_2 = 320px; h_2 = 128px$$

The loss function

$$L_{ball_{1,2}} = -\frac{1}{w_{1,2}} \sum_{i=1}^{w_{1,2}} \hat{p}_{x_{1,2}}^i \log p_{x_{1,2}}^i - \frac{1}{h_{1,2}} \sum_{i=1}^{h_{1,2}} \hat{p}_{y_{1,2}}^i \log p_{y_{1,2}}^i$$

2. Proposed methods

Events spotting

- The event spotting branch acts on **concatenated** feature maps from **global and local detectors**.
- The last activation layer (Sigmoid) allows both events to occur simultaneously.
- The target values were constructed as $\sin(n\pi/8)$ to be in $(0,1)$ range and act as events probabilities, where n is the *number of frames* between the *considered frame* and the *manually labeled event frame*, n is in $(-4, 4)$.

➡ The target is nonzero if n in $(-4; 4)$ and 0 otherwise.

Loss function

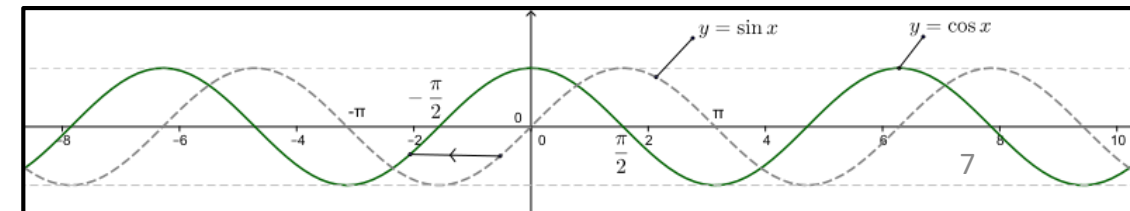
$$L_{event} = -\frac{1}{N_{events}} \sum_{i \in \{events\}} \beta_i \hat{p}^i \log p^i$$

N_{events} - number of possible event types

Input size	Operator	In-channels/ Out-channels
5 x 2	Conv 1x1	512/64
5 x 2	BatchNorm	64/64
5 x 2	ReLU	64/64
5 x 2	DropOut	64/64
5 x 2	ConvBlock (w/o MaxPool)	64/64
5 x 2	DropOut	64/64
5 x 2	ConvBlock (w/o MaxPool)	64/64
5 x 2	DropOut	64/64
5 x 2	Flatten	64/-
640	FC	-
512	ReLU	-
512	FC	-
2	Sigmoid	-

Structure of the Events Spotting part

A weighted ($1:3 \sim \text{bounce:net hit}$) cross-entropy loss



2. Proposed methods

Semantic Segmentation

Operator	In-channels/ Out-channels	stride	padding
Conv 1x1	$a/\frac{a}{4}$	1	0
BatchNorm	$\frac{a}{4}/\frac{a}{4}$	-	-
ReLU	$\frac{a}{4}/\frac{a}{4}$	-	-
TConv 3x3, op=1	$\frac{a}{4}/\frac{a}{4}$	2	1
BatchNorm	$\frac{a}{4}/\frac{a}{4}$	-	-
ReLU	$\frac{a}{4}/\frac{a}{4}$	-	-
Conv 1x1	$\frac{a}{4}/b$	1	0
BatchNorm	b/b	-	-
ReLU	b/b	-	-

Structure of the DeconvBlock

Input size	Operator	In-channels/ Out-channels
10 x 4	DeconvBlock	256/128
20 x 8	DeconvBlock	128/128
40 x 16	DeconvBlock	128/64
80 x 32	DeconvBlock	64/64
160 x 64	TConv 3x3, s=2, p=0, op=0	64/32
321 x 129	ReLU	32/32
321 x 129	Conv 3x3, s=2, p=0	32/32
319 x 127	ReLU	32/32
319 x 127	Conv 2x2, s=2, p=1	32/3
320 x 128	Sigmoid	3/3

Structure of the Semantic Segmentation part

Loss function

$$L_{segm} = (1 - DICE_{smooth}) + BCE$$

$$DICE_{smooth} = \frac{2|\hat{P} \cap P| + \epsilon}{|\hat{P}| + |P| + \epsilon}$$

$$DICE(A,B) = \frac{2 \times \text{Intersection}(A,B)}{\text{Area}(A) + \text{Area}(B)}$$

2. Proposed methods

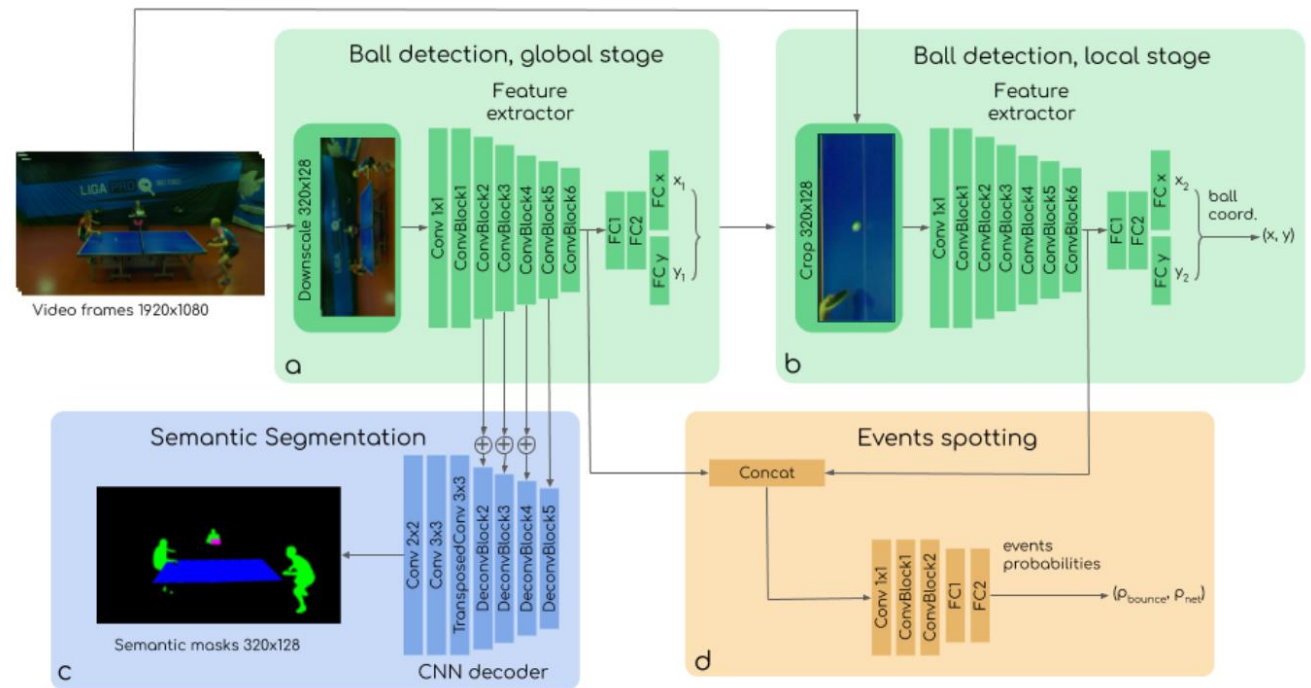
Multi-task loss function

$$L_{ball_{1,2}} = -\frac{1}{w_{1,2}} \sum_{i=1}^{w_{1,2}} \hat{p}_{x_{1,2}}^i \log p_{x_{1,2}}^i - \frac{1}{h_{1,2}} \sum_{i=1}^{h_{1,2}} \hat{p}_{y_{1,2}}^i \log p_{y_{1,2}}^i$$

$$L_{event} = -\frac{1}{N_{events}} \sum_{i \in \{events\}} \beta_i \hat{p}^i \log p^i$$

$$L_{segm} = (1 - DICE_{smooth}) + BCE$$

$$DICE_{smooth} = \frac{2|\hat{P} \cap P| + \epsilon}{|\hat{P}| + |P| + \epsilon}$$



$$\Rightarrow L = \sum_{i=1}^4 \frac{L_i}{\sigma_i^2} + \sum_{i=1}^4 \log(\sigma_i) \quad \text{where } L_i - \text{one of four: } L_{ball_{1,2}}, L_{events} \text{ or } L_{segm}$$

The relative weights of the losses adaptively and treats them as *trainable parameters*.

OpenTTGames Dataset


- Full HD videos of table tennis games recorded at 120 fps
 - ✓ Training: 5 videos of different matches
 - ✓ Testing: 7 short videos from other matches
- Number of images:
 - ✓ Training: **38752** images
 - ✓ Validation set: **9502** images
 - ✓ Testing set : **7328** images
- Manually labeled:
 - In-game events (ball bounces, net hits, or empty event targets)
 - Ball coordinates
 - Segmentation masks

Implementation Details and Training

- Adam optimizer with default parameters
- Initial learning rate ***0.001***, reduce 2 times after 3 epochs without the loss value decrease.
- Early stopping after ***12*** epochs of no decrease on validation loss
- Data augmentation:
 - Random cropping with width and height reduction up to ***15%***
 - Random frame rotation ($\pm 15^\circ$)
 - Horizontal frames flip
 - Random brightness, contrast, and hue shifts

Evaluation metrics

- **Ball detection:** Root Mean Square Error (**RMSE**)
- **Event spotting:**
 - Percentage of Correct Events (**PCE**): An event was considered as correct if the predicted value is equal to the ground truth after *0.5 level thresholding of the predicted and target values*.
 - Smooth Percentage of Correct Events (**SPCE**): an event treated as a correct one if the *difference with the target* is less than a threshold (0.25 in the experiments)
- **Semantic segmentation:** Intersection Over Union (**IoU**)


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Feature extractor

Feature extractor architecture	ResNet-18	TTNet-encoder
Encoder GFLOPs	2.260	2.340
Parameters, M	11.250	1.180
Inference time, ms	7.6	6.0
Global RMSE, px	12.11	6.79
Local RMSE, px	1.99	1.97
Global accuracy	0.973	0.975
Local accuracy	0.973	0.978
IoU	0.943	0.928
PCE	0.971	0.977
SPCE	0.970	0.970

Target sequence length

Input sequence length optimization for the ball detection part of TNet.

Target width, n frames	1	3	5	7	9
Global RMSE, px	30.93	3.56	3.08	3.47	3.60
Local RMSE, px	6.64	1.23	1.36	1.46	1.46
Global accuracy	0.955	0.982	0.982	0.981	0.980
Local accuracy	0.912	0.983	0.983	0.983	0.981

The influence of the length of the input sequence on the event spotting metrics

Target width, n frames	1	3	5	7	9
PCE	0.947	0.940	0.965	0.970	0.979
SPCE	0.931	0.923	0.954	0.965	0.975
Global RMSE, px	23.65	7.02	5.81	5.79	5.27
Local RMSE, px	5.53	2.27	1.71	2.40	2.03
Global accuracy	0.962	0.978	0.981	0.979	0.981
Local accuracy	0.928	0.979	0.983	0.981	0.982

Loss balancing

- Comparison of training results of the TTNet with different loss aggregation strategies

Loss	Unbalanced	Manually weighted	Balanced
Global RMSE, px	9.34	7.74	6.79
Local RMSE, px	2.93	2.38	1.97
Global accuracy	0.977	0.977	0.975
Local accuracy	0.975	0.978	0.978
IoU	0.938	0.902	0.928
PCE	0.976	0.968	0.977
SPCE	0.966	0.963	0.970

- The adaptive balancing performs better on most tasks and metrics

- Every video is recorded in similar conditions with slight variations in camera angle
- There are only 3 people per images

THANK YOU!