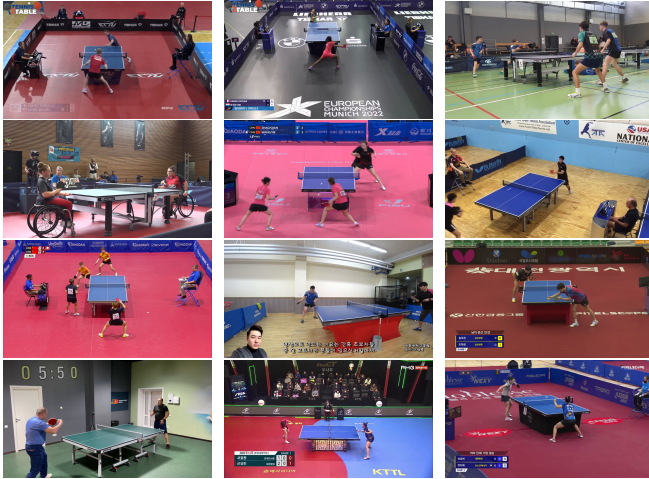


# BlurBall: a ball detector with blur estimation

## Supplementary Material

### Dataset

The dataset was generated using publicly available online videos under a Creative Commons license to ensure compliance with copyright regulations. We aimed to cover a diverse range of scenarios, including different playing conditions, camera angles, and lighting variations, to enhance the robustness of our models. However, our dataset does not include orange table tennis balls, which are also officially approved for competition. This omission is unlikely to impact performance significantly, as white balls are overwhelmingly preferred and almost exclusively used in official matches.



**Fig. 7:** Example scenes from our dataset, showcasing a diverse range of contexts to ensure comprehensive coverage.

The dataset can be downloaded using this link: <https://cloud.cs.uni-tuebingen.de/index.php/s/C3pJEPKWQAkono7>. For every video, there is an attached csv file containing:

Frame	Visibility	X	Y	theta	l
000049	1	581.62	295.26	-152.5	2.8
000050	1	572.985	292.86	171.8	2.1

**Table 4:** CSV description

The angles are in degrees and are defined as shown in Figure 2.

### Model training

We followed the training recommendations from WASB [12] to ensure consistency across models. Each model was trained

for 30 epochs using a suitable loss function and optimizer to achieve optimal performance.

- **DeepBall** [18] and **DeepBall-large** were both trained using Binary Cross-Entropy (BCE) loss and the Adam optimizer with a learning rate of  $3 \times 10^{-4}$ .
- **BallSeg** [19] was trained with focal loss ( $\gamma = 2$ ) and the Adam optimizer, also using a learning rate of  $3 \times 10^{-4}$ .
- **TrackNetV2** [10] and **TrackNetV2** were trained using focal loss and the AdaDelta optimizer with a learning rate of 1.0.
- **MonoTrack** [11] was optimized using Combo loss and trained with the AdaDelta optimizer at a learning rate of 1.0.
- **WASB** [12] was trained with quality focal loss and the Adam optimizer, maintaining a learning rate of  $3 \times 10^{-4}$ .

This setup ensures that each model is trained with an appropriate optimization strategy, balancing accuracy and convergence stability.