Ball Detection in Table Tennis Match Videos

Sruthi Meda
M03484920
vm94s@missouristate.edu

Charan Kumar Valluru
M03486006
cv375s@missouristate.edu

***Abstract*—In this research, we describe a two-stage ball detection model for table tennis matches, which is an essential component of the table tennis referee system. The proposed model is a lightweight model influenced by the work of Voeikov et al. (2020), which use a low-resolution image. Using global detection blocks, we first localise the ball, then utilize the localization result to forecast the exact coordinates of the ball. The model is trained using the OpenTTGames dataset, which contains frames captured at a rate of 120 frames per second. We used a sophisticated training strategy that allows our model to train quickly. Our testing results reveal that the global detection block detects the existence of a ball in a patch with an accuracy of 97%, while the local model predicts the center of the ball in the frame with an RMSE of roughly 7px.**

I. Introduction

The growing popularity of table tennis as a physical and cerebral workout has motivated researchers to investigate computer vision and deep learning breakthroughs to improve game analysis and refereeing. Notably, academics have used these technologies to create officiating systems that not only detect occurrences during table tennis matches but also provide in-depth analytical for fair decision-making.

Precision identification of the ball's position is critical in table tennis refereeing. The coordinates of the ball are critical for comprehending the sequence of events, distinguishing real bounces on the table, and filtering out false positives that occur beyond the play area. Detecting the fast-moving ball at speeds of up to 30 meters per second, on the other hand, causes difficulties. Because of the quick velocity, the ball's shape changes constantly, ranging from elliptical to fuzzy or ray-like, making exact pinpointing difficult.

Figure (1) depicts the various configurations that the ball takes while in motion, showing the difficulty that detection models encounter. These differences in motion need the use of complex algorithms to precisely find the ball, which is particularly difficult for older methods.

To make matters worse, the existence of background objects with similar qualities to the ball increases the possibility of false positives. Separating the ball from the backdrop features necessitates advanced algorithms capable of distinguishing visual characteristics even in difficult settings.

In summary, incorporating computer vision and deep learning into table tennis refereeing systems is a substantial technological advancement. Nonetheless, the complexities of detecting a swiftly moving and dynamically changing item highlight the need for powerful algorithms capable of dealing with motion blur, changing forms, and potential background interference. Addressing these problems will very certainly lead to more precise and dependable ball recognition systems in table tennis and other sports as technology advances.

Fig. 1. Ball shapes while in motion

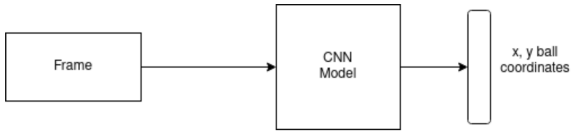

Fig. 2. Proposed Model

The study focuses on the significant challenges of ball detection in table tennis, where elements such as ball color and varying lighting conditions complicate computer vision models. In response, this research proposes a Convolutional Neural Network (CNN) architecture inspired on Voeikov et al.'s (2020) new deep neural network detection approach.

The CNN architecture's main design includes a two-stage detection mechanism. The first stage will most likely involve rough localization or feature extraction, which will provide a rudimentary understanding of the ball's position, while the second stage will refine this localization, assuring exact and accurate detection. This multi-stage method strikes a balance between computational economy and detection accuracy, which is critical for real-time applications like table tennis officiating systems.

The proposed model's attention for performance on consumer-grade GPUs is a critical component. This emphasis on cost-effectiveness assures greater accessibility and usefulness, especially in settings where high-end gear may be unavailable. The model is optimized to analyze single frames collected by a fixed-angle camera at a fast frame rate of 120 frames per second (FPS) while keeping to particular parameters (width = 1920 and height = 1080). This customized

configuration corresponds to the needs of table tennis settings, giving the necessary input for the CNN to successfully evaluate and extract information about the ball's position in each frame.

This work is significant because of its potential applications, notably in real-time table tennis officiating systems where quick and precise ball detection is critical. The suggested CNN architecture offers a viable method for obtaining precise ball coordinates in a timely manner during table tennis matches by overcoming the challenges provided by ball color, lighting circumstances, and the dynamic nature of the game.

II. Literature Review

The task of detecting a ball in a table tennis match has been extensively researched, and numerous ways to attain precise and efficient outcomes have been examined. Kulkarni et al. (2022) conducted a comparison study of various models for ball identification and tracking, taking into account varied speeds. According to their findings, the You Only Look Once (Yolo) paradigm outperforms alternative techniques in this setting.

Tamaki and Saitô (2013) used a contour-based technique to recognize balls. They did, however, highlight limits in instances when similar-looking objects are present in the background, potentially leading to detection failures. This emphasizes the significance of resilience and adaptability in ball identification systems, particularly when dealing with complex and dynamic contexts such as a table tennis match.

To segment the ball in stereo pictures, Myint et al. (2015) used adaptive color thresholding and background subtraction algorithms. This method provides an alternative option for dealing with background interference problems. Such strategies are critical for improving the accuracy of ball detection algorithms, especially in cases when the backdrop contains items that resemble the ball in appearance. Gómez-González et al. (2019) proposed using MobileNet with Single Shot multibox detector (SSD) architecture and four cameras in a robotics scenario for ball detection. While beneficial, the use of several cameras may not be appropriate in all scenarios. In the context of table tennis, the emphasis is on using a single camera; hence, such systems must be modified.

Voeikov et al. (2020) proposed a multi-modal architecture that includes a CNN-based component for ball detection in a sequence of frames. This stresses the significance of considering not only individual frames but also the temporal element of the ball's movement in order to identify it accurately. Such an approach can be especially useful in situations when the ball trajectory must be tracked throughout time.

Due to the fast pace and diverse shapes of the ball in other sports, like as football, SSD and Yolo models have been used for ball detection (Komorowski et al., 2020). DeepBall, a deep learning architecture specifically designed for ball detection in football games, was created by Komorowski et al. (2019). However, its operational speed of 190 frames per second (fps) on full HD images may be prohibitively slow for table tennis, where the ball moves at a much faster rate.

III. Methodology

My model architecture is inspired from the (Voeikov et al., 2020) proposed model for ball detection.

Below is our model:

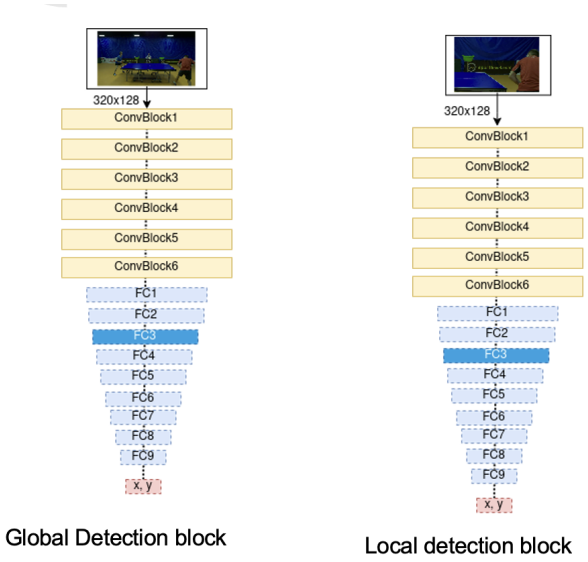

Fig. 3. Architecture

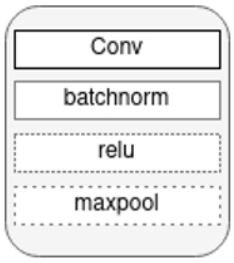

Fig. 4. Layers in Conv Block

Below are the table with layers and its filters:

Both the global and local detection blocks in our model follow a uniform structure, encompassing sequences of six layers characterized by convolutional layers, batch normalization, Rectified Linear Unit (ReLU) activation, and max-pooling. As information moves across the network, this consistent layer composition allows for the hierarchical extraction of progressively complex spatial properties. The model can recognize complicated patterns and representations within the input data because to the progressive increase in the amount of features at each layer.

However, our model's post-sixth layer design deviates significantly from the model proposed by Voeikov et al. (2020). Voeikov et al. used only two linear layers in their study to build a probability distribution for the ball's coordinates along the x and y axes. In comparison, our model adds an additional eight

TABLE I
LAYER INFORMATION FILTERS

Layer	Input Channels * Output Channels
conv1	3 * 64
batchnorm	64 * 64
relu	64 * 64
convblock1	64 * 64
convblock2	64 * 64
dropout2d	64 * 64
convblock3	64 * 128
convblock4	128 * 128
dropout2d	128 * 256
convblock5	128 * 256
convblock6	256 * 256
dropout2d	256 * 256
fc1	2560 * 1792
relu	1792 * 869
dropout1d	1792 * 869
fc2	1792 * 869
relu	1792 * 869
dropout1d	1792 * 869
fc3	869 * 448
fc4	448 * 256
fc5	256 * 128
fc6	128 * 64
fc7	64 * 32
fc8	32 * 16
fc9	16 * 8
fc10	8 * 2

levels, culminating in a dedicated output layer, for a significant enhancement. This planned granularity increase tries to capture complex geographical intricacies. By increasing the network's depth, we hope to improve the model's ability to distinguish finer details, allowing for more exact localization of the ball within the picture. This architectural modification corresponds with our goal of achieving improved performance and increased accuracy in ball localization, hence establishing our model.

The strategic choice to add layers to our model was driven by a deliberate desire for more granularity in predicting capabilities. Our goal in incorporating these supplemental layers into the model architecture was to probe deeper into the precise details of the ball's placement within the frame. This enhancement reflects a deliberate attempt to extract more nuanced and detailed information, increasing the model's ability to detect subtleties that lead to a more accurate depiction of the ball's coordinates.

As described by Voeikov et al. (2020), this architectural divergence from the prior model provides a specific purpose in aligning our model with the broader goal of gaining finer localization accuracy. The addition of these extra layers is an intentional decision to provide the model better precision in capturing and interpreting spatial complexities. This divergence from conventional architecture reflects our commitment to pushing the limits of ball tracking precision. Finally, this architectural revision considerably helps to our dual detection model's overall performance, presenting it as a resilient and intelligent solution for exact ball localization in dynamic table tennis scenarios.

$$MSE = \frac{1}{N}\sum_{i}^{N}(Yi - \hat{Y}i)^2$$

I trained the model's global detection block in the first phase of training. It accepts the scaled video frame with dimensions of width = 320px and height = 128px. According to Voeikov et al. (2020), this dimension was used to have at least 2px of ball diameter in frames. This model predicts the ball's coordinates across the whole video frame. Because the image is so little, making the ball even smaller, I refer to this model as our ball localization model, and the first phase as our ball approximation phase. The model provides us with the coordinates, which are an estimate of the ball coordinates. To compare the predicted x and y coordinates, we employed the MSE loss function.

$$MSE = \frac{1}{N}\sum_{i}^{N}(Yi - \hat{Y}i)^2$$

In the second phase of training, I used the trained model weights to initialize our local detection model while freezing our global detection model. This allows us to save training time while still utilizing the features learnt in the previous phase. The global detection model's output was used to generate the input for the local detection model. I cut a patch from the original image using the coordinates from our global detection module, assuming the xy coordinates as my center. The reduced patch's dimensions were width = 320px and height = 128px, assuming the ball was present in the patch.

If at some point our global detection model makes any mistakes we simply make our ground truth coordinates x = -1 and y = -1 and crop a patch from the centre of the image. For local blocks, I have used the MSE loss function.

In the final phase of training, a simultaneous training strategy was used for both the global and local detection models to improve coordination and alignment. The global detection model was developed to produce estimated ball coordinates, which served as an initial estimate of the ball's position within the frame. Following that, the local detection model entered the picture, strengthening the prediction by providing more precise coordinates for the discovered ball. The major goal was to improve alignment and synergy between the two models by harnessing their complimentary capabilities.

During the training phase, we diligently designed a unified and comprehensive loss function to assess the performance of both the global and local detection blocks in our dual detection model. This loss function was created expressly to account

for the difference between predicted coordinates and ground truth values for both models. The loss function was designed to capture the overall error throughout the entire model by including the prediction outcomes from both detection blocks.

The sum of losses from both the global and local detection blocks was used in the combined optimization process. Given the model's capacity to provide both approximate and refined ball coordinates, this combined loss metric gave a unique and unified evaluation of its effectiveness. The coordinated propagation of the computed loss over both networks guaranteed that model parameters were updated in sync. The convergence of both detection blocks was facilitated by this strategic training method, which was supported by a single loss function. The end result was a meticulously aligned and exact detection system capable of producing precise and nuanced ball coordinates, demonstrating the usefulness of our dual detection model in capturing the nuances of ball movement in the dynamic setting of table tennis.

Lt = Lg + Ll

This is the output of our local detection block.

IV. Experimentation and Results

The global detection block was crucial in the training process because it is the first stage in our dual detection model. This block was trained using 130 epochs, which included the full training dataset. The importance of this phase stems from the fact that successful ball localization by the global block is required for subsequent predictions. Thorough training for this block was thought necessary in order to provide a solid foundation for the overall model.

Following the training of the global detection block, the next block was trained for 25 epochs with the weights from the previously trained global block loaded. The global block was frozen at this step to maintain its learned features. This strategic strategy meant that the future block could refine the ball's localization based on the global block's initial estimates. To maximize their combined performance, the synergy between the two blocks was meticulously arranged.

During the final training phase, the entire model was subjected to an additional 10 epochs, allowing both the global and local detection blocks to align harmoniously. This phase was critical in ensuring that the two blocks effectively complemented each other, refining their predictions to build a coherent and accurate model.

Fig. 5. Output of Local detection block

A normalizing step was included during training to prevent potential difficulties caused by big coordinate values. Normalizing coordinates helped reduce significant weight shifts caused by big x and y values, resulting in a smoother learning process and improved model generalization throughout the dataset. This careful consideration of training procedures and normalizing approaches helps to our dual detection model's general resilience and dependability, placing it as a well-optimized model.

My global detection blocks give the following output:

The original ball position is highlighted in yellow in our visual representation of the ball identification process, while the model-predicted coordinates are delimited in red. The model may occasionally predict incorrect coordinates, even when the visual features do not closely reflect those of the ball. This phenomena is caused by two basic elements. First, we normalize the ball coordinates by scaling them from 0 to 1. Because of the use of floating-point representations, this

normalization introduces some imprecision, which affects the precision of model predictions. Furthermore, the model may struggle to properly forecast coordinates within the confined spatial context due to the small size of the actual image fed to the model. Despite these difficulties, the closeness of the predicted coordinates to the real coordinates demonstrates the model's ability to use ball properties to make reasonably accurate predictions.

Moving on, we use the output from the global detection block to crop an area from the image that the model detects as containing the ball using the methods given in (Voeikov et al., 2020). This patch is subsequently sent to the local detection block, which now contains the ball. We may retrieve the real coordinates of the ball within the frame using this sequential process. Our model overcomes the obstacles provided by image size and floating-point precision by applying this two-stage technique, resulting in a refined and precise assessment of the ball's position. This methodological clarity improves the dependability and accuracy of our ball identification algorithm, which contributes to its performance in table tennis matches.

We use two separate measures to evaluate the success of our ball detection model, each adapted to the individual responsibilities of the global and local detection blocks. We use the accuracy metric for the global detection model, which is responsible for identifying patches containing the ball. Our global detection approach successfully identifies patches where the ball is present, accounting for frames when the ball is genuinely detected and removing false predictions and instances where the ball is absent, with an amazing accuracy rate of 86%. This statistic provides an in-depth assessment of the model's ability to detect the existence of the ball within certain patches.

The Root Mean Squared Error (RMSE) statistic is used to evaluate the local detection model, which is tasked with precisely localizing the ball within the indicated patch. The RMSE score of the local model is 77 pixels, suggesting an average error between the predicted and real ball coordinates. A complex flaw in the dataset was discovered after a thorough study. Around 200 frames from the test data were recognized as having incorrect ball centers, adding inaccuracies into the evaluation results. This thorough examination emphasizes the significance of dataset integrity and its direct impact on model performance evaluation. The identification and comprehension of these nuances contributes to a more nuanced perception of the model's strengths and potential areas for improvement, which is a critical feature in the development process.

And for the training data, the global detection block is giving an accuracy of around 97.67% and the RMSE score is approximately 7px.

V. Conclusion

In conclusion, the developed model performs well using a two-stage detection method that consists of a global block that localises the ball and gives me the approximate coordinates of the ball, followed by a local detection block that uses the output of the previous block and refines it to give me the exact coordinates of the ball. There is a need for data refinement, which we identified during the testing phase after charting the findings. Further improvement of the model will enable us to merge it into a robust model that can become an integral part of the autonomous table tennis referee system.

References

[1] Osai AI Lab. (n.d.). Home — Osai AI Lab. https://lab.osai.ai/

[2] Voeikov, R., Falaleev, N., Baikulov, R. (2020). TTNet: Real-time temporal and spatial video analysis of table tennis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 884-885).

[3] Naik, B. T., Hashmi, M. F., Keskar, A. G. (2022, November). Modified Scaled-YOLOv4: Soccer Player and Ball Detection for Real Time Implementation. In International Conference on Computer Vision and Image Processing (pp. 154-165). Cham: Springer Nature Switzerland

[4] Tamaki, S., Saitô, H. (2013). Reconstruction of 3D Trajectories for Performance Analysis in Table Tennis. IEEE. https://doi.org/10.1109/cvprw.2013.148

[5] Myint, H., Wong, P., Dooley, L. S., Hopgood, A. A. (2015). Tracking a table tennis ball for umpiring purposes. Fourteenth IAPR International Conference on Machine Vision Applications. https://doi.org/10.1109/mva.2015.7153160