Tracking and Detection of the Soccer Ball
Lab Vision Systems

Kannan Ravi, Manoj Prabhakar, Sundaram, Subbulakshmi

Rheinische Friedrich-Wilhelms-Universität Bonn
s6makann@uni-bonn.de, s6susund@uni-bonn.de, Matrikelnummer: 3069835, 3157792

Abstract. In ball tracking and detection, localization of the ball and tracking it is the most challenging task, and in particular when a sequence of frames of a video is fed as input to the model. In the recent years there has been significant improvement in performance of Fully Convolutional Neural Network, and in particular for object detection. However the performance has been significantly good only in the case of object detection in single frames. The performance is degraded when spatiotemporal data is provided to the model i.e., sequence of frames in this case. To overcome this drawback, Convolutional LSTM model is added to work with sequence of frames.

1 Introduction

In recent years there has been a growing interest towards RoboCup Soccer which is a soccer competition in which the robot has to detect objects like the ball and the other objects on the field like goal posts and the boundaries [1]. Various conditions like the pace of ball movement, differing lighting conditions require the model for detection to be fine tuned quite often. Hence algorithms which serve the purpose of detection and tracking of the ball to produce reliable and good results using object localization is posing a huge problem. Further to this, tracking the moving ball in the video makes it all the more difficult.

There has been a paradigm shift in the field of computer science from traditional edge detection and object localization algorithms using region of interest extraction to convolutional neural networks because of its better performance [2]. Due to this shift, convolutional neural networks are applied to the images for object localization spatially and also highlight it on the image. Convolutional Long Short-Term Memory has been chosen over LSTM for its efficiency in handling spatio-temporal data, unlike LSTM which handles only one-dimensional input when handling a sequence of frames of a video.

Convolutional Neural Network(CNN) is a neural network consisting of convolution layers. CNN's are used for object detection, image classification, document analysis and many more. Convolutional neural network is a special type of multilayer neural network which is designed to extract features from the input passed to it. CNN's have been successful in object classification tasks. In recent years they have also proved to be favourable in terms of the good results for object localization .

We have implemented the three SweatyNet models which are robust and efficient. Three variations of SweatyNet have been tested for ball detection. The model is further improvised by stacking it up with an additional Conv-LSTM layer that aids in predicting the frames which the SweatyNet model fails to predict.

2 Related Work

Fully convolutional neural networks(FCNN) serve the purpose of ball detection and tracking [3]. Unlike traditional fully connected neural network which uses a lot of weights as the network size grows larger for complex computer vision kind of tasks, FCNN convolves the image and extracts information from the input image from the initial layers of the network itself [4].

Convolutional Long-Short Term Memory(Conv-LSTM) is a recurrent neural network in which the network remembers the input fed to it, and predicts the future states. This is particularly useful for spatio-temporal data like a video, to predict the next frame with respect to the current input frame provided to it [5]. There are also similar approaches using the Convolutional Gated Recurrent Units(Conv-GRU) [6].

The models which have been developed in the field of vision are trained to work with static images but in the real world, most of the objects are in motion so visual objects are rarely seen as static frames. So to solve this kind of scenario deep learning algorithms are being designed to track objects in video sequences.

There has been a lot of research work in the field of deep learning to solve many supervised learning tasks. Amongst them predicting future frames in a video sequence is still state of the art research work.

3 Architecture

3.1 Fully Convolutional Neural Network(FCNN)- SweatyNet

The network architecture of the FCNN consists of an input image and a corresponding target image which are passed through the SweatyNet-1 model [1], and the resulting output from the model is an image of dimension 160*128*1 with the ball localized in it. And if there is an input image in which there is no ball present, it returns a blank image. Fig. 1 depicts the architecture of this model.

There are three network architectures proposed by Fabian Schnekenburger et al [1] which are the SweatyNet-1, SweatyNet-2 and SweatyNet-3 models which are more or less the same with minor modifications. We have compared all the three models architecture. It was found that SweatyNet-1 model gave us the best results. The SweatyNet-1 model is a fully convolutional neural network(FCNN) consisting of encoders and decoders, which is trained to detect the ball in the image. The encoder consists of 12 convolutional layers. Relu activation function and Batch normalization are applied after each of the convolutions. And also, the number of filters applied to each layer starts at 8 in the first layer and is

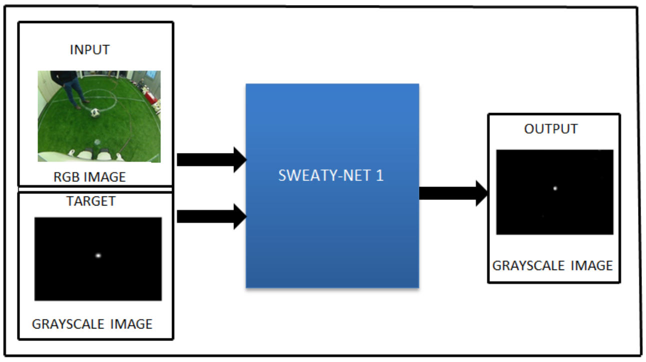

Fig. 1. Architecture of the model using SweatyNet-1 for ball localization

consequently doubled in the consecutive layers after every set of four max-pooling layers. The convolutional layers are compatible with a kernel of dimension 3x3. The decoder is shorter than the encoder with 6 convolutional layers, and bilinear up-sampling is used instead of transposed convolution. Also, two up-sampling layers are added along with the other convolutional layers. The image dataset consisting of 3000 images have a resolution of 640x480 which have been reshaped to 640x512 to match the input size of the SweatyNet-1 model.

Skip connections exist between the encoders and decoders to maintain high resolution of the images. Fig. 2 depicts the architecture of the SweatyNet-1.

3.2 SweatyNet-1 with Convolutional LSTM

The output from the SweatyNet-1 model is stacked up and passed as a sequence of frames to the Convolutional LSTM model. The Convolutional LSTM is a recurrent neural network which consists of gates that decide which information is to be retained and which information has to be discarded from the sequence passed to it. Convolutional LSTM is a regular LSTM in which the matrix multiplication is replaced by convolution operation at each of the gates. In Conv LSTM the future states are determined from the information retained from the previous states. The Convolutional LSTM consists of 5 key equations to perform the convolution operation at the gates where '*' denotes the convolution operator and 'o', denotes the Hadamard product [5].

$$\begin{aligned}
i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o\right) \\
\mathcal{H}_t &= o_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{1}$$

The output from the convolutional lstm model is the last hidden state and the list of all layers. The last hidden state is compared with the ground truth to

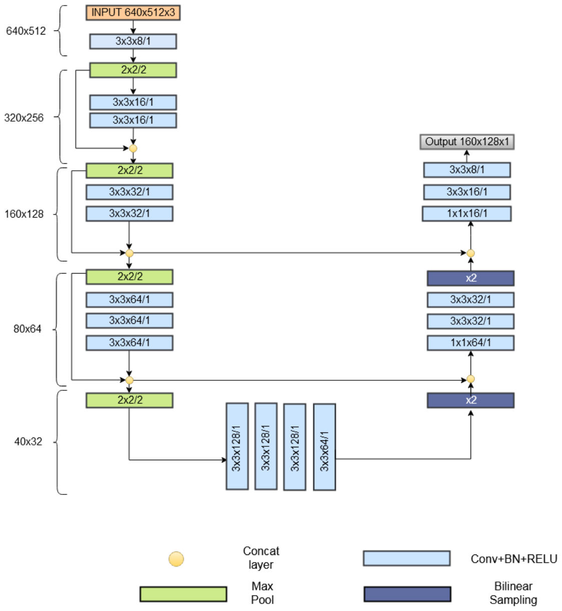

Fig. 2. SweatyNet-1 model

train the model and compute the loss. Suppose a sequence of 5 frames is passed to the model, the model predicts the 5th frame in the sequence and so on and so forth. By this approach the output from this model will detect images which have been missed out by the sweaty model. Fig 3 depicts this architecture of the SweatyNet-1 model stacked up with Conv-LSTM layer.

4 Implementation

4.1 Pre-Processing

Multiple videos of the ball movement have been captured and broken down into a sequence of frames. The ball in the frames have been annotated using Image-tagger and also a few of them were done manually when the software failed to do so. The annotations for all the images have been saved in a text file. The ground truth for each of the frames has been created by fitting the normal distribution with a standard deviation of 4 at the location of the ball. The images have been re-scaled to 640x512 dimension to match the input to the model.

4.2 Training

The dataset consists of 3000 images with the train-test split ratio to be 80 and 20. The training data images which have been rescaled are shuffled and passed into the SweatyNet models for training on the the Google Colab Tesla K80

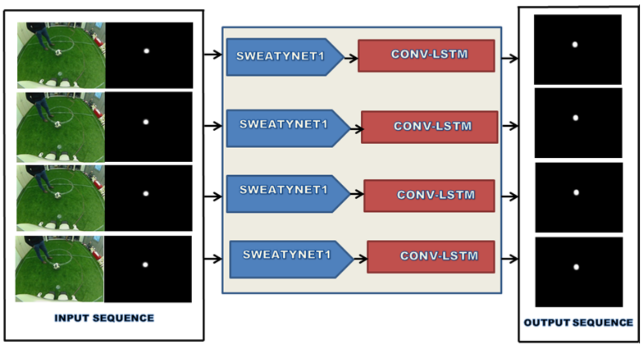

Fig. 3. SweatyNet-1 model with Conv-LSTM

GPU. It was observed that the network differentiates between the background and foreground in the initial few epochs itself. The network has been trained for about 100 epochs, and it was found that SweatyNet-1 gave the best results amongst the three SweatyNet models. The output from the SweatyNet-1 model is sequenced and passed to Conv-LSTM. It is ensured that the shuffling of images is turned off before passing the output from SweatyNet-1 to Conv-LSTM. The Adam optimizer has been used [7] and with a learning rate of 0.01 fixed in both cases. The Mean Squared Error(MSE) has been used for the loss calculation in both cases.

4.3 Post processing

We have eroded the image with a kernel size of 4 to remove noise from the image. We determine the pixel coordinates of the center of the ball by calculating the local maximum around the center coordinates using a peak detection algorithm. This is done because the network doesn't predict the coordinates of the pixels on its own. It only predicts them from the 2 dimensional probability map which we create. We find the contour centers from the output of the network and compare it against the target center coordinates to measure the accuracy of the prediction of the model [1].

5 Result

The major contribution has been made towards implementing,training and testing the three variants of SweatyNet model. Further minor modifications to the architecture have been made by adding an additional Convolutional LSTM layer to SweatyNet-1 model architecture so that the model tracks the ball in a sequence of frames in a video.

Experiments have been performed to evaluate the performance of the model with and without the use of the additional layer. In the sections following this, we will discuss the various metrics used for evaluating the performance of the model.

5.1 Metrics

The network has been trained for 100 epochs and various metrics have been calculated to check the performance of the model with the presence and absence of the additional Convolutional LSTM layer. IOU is defined as the number of true positives(TP) divided by the sum of TP , false positive(FP) and false negative(FN) .

$$IOU = \frac{TP}{TP + FP + FN} \tag{2}$$

Recall has been defined to be the number of true positives(TP) divided by the sum of TP and false negatives(FN) .

$$RC = \frac{TP}{TP + FN} \tag{3}$$

False detection rate (FDR) is defined as the number of false positives(FP) divided by the sum of FP and true positive(TP)

$$FDR = \frac{FP}{FP + TP} \tag{4}$$

Precision is defined as the number of true positive(TP) divided by the sum of TP and false positive(FP).

$$Precision = \frac{TP}{FP + TP} \tag{5}$$

Accuracy is defined as the sum of true positive(TP) and true negative(TN) divided by the sum of TP, TN, false positive(FP), false negative(FN).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

The local maximums calculated from the peak detection algorithm have been used for the calculation of the various metrics. A detection is considered to be true positive if the local maximum is detected within the range of 5 pixels from the coordinates of the ball .

5.2 Comparison of the models

We have chosen Sweaty net model 1 over the the other two models because it performs significantly better than the other two. The performance of the model with the presence of the additional convolutional LSTM layer is better than the model without it in tracking the balls missed out by the latter. Table I illustrates the same.

Model	RC	FDR	IOU	Accuracy	Precision
SweatyNet-1	96.6	3.4	93.3	96.2	96.6
SweatyNet-2	96.4	6.9	90.0	94.4	93.1
SweatyNet-3	96.5	8.0	90.3	94.4	93.3
SweatyNet-1 and ConvLSTM	98.9	1.1	98.4	98.7	98.64

Table 1. Metrics Calculated to illustrate the model performance

6 Conclusion

From the experiments it has been shown that the SweatyNet-1 model is robust enough in dealing with spatial data for object detection. Adding the Conv-LSTM to it has shown significant improvement in performance while dealing with spatiotemporal data.

References

[1] Fabian Schnekenburger et al. “Detection and Localization of Features on a Soccer Field with Feedforward Fully Convolutional Neural Networks (FCNN) for the Adult-Size Humanoid Robot Sweaty”. In: Jan. 2017.

[2] William Lotter, Gabriel Kreiman, and David D. Cox. “Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning”. In: *CoRR* abs/1605.08104 (2016). arXiv: 1605.08104. URL: http://arxiv.org/abs/1605.08104.

[3] Daniel Speck et al. “Ball Localization for Robocup Soccer Using Convolutional Neural Networks”. In: Nov. 2017, pp. 19–30. ISBN: 978-3-319-68791-9. DOI: 10.1007/978-3-319-68792-6_2.

[4] Grzegorz Ficht et al. “NimbRo-OP2X: Adult-Sized Open-Source 3D Printed Humanoid Robot”. In: *18th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2018, Beijing, China, November 6-9, 2018.* 2018, pp. 1–9. DOI: 10.1109/HUMANOIDS.2018.8625038. URL: https://doi.org/10.1109/HUMANOIDS.2018.8625038.

[5] Xingjian Shi et al. “Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model”. In: *CoRR* abs/1706.03458 (1991).

[6] Nicolas Ballas et al. “Delving Deeper into Convolutional Networks for Learning Video Representations”. In: *CoRR* abs/1511.06432 (2015). arXiv: 1511.06432. URL: http://arxiv.org/abs/1511.06432.

[7] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.