2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops

Jersey Number Recognition with Semi-Supervised Spatial Transformer Network

Gen Li†1, Shikun Xu†1, Xiang Liu‡2, Lei Li† and Changhu Wang†
†Toutiao AI Lab, Beijing, China
‡Beijing Univ. of Posts & Telecoms, Beijing, China
{ligen.lab, xushikun, liuxiang.1995, lilei.02, wangchanghu}@bytedance.com

Abstract

It is still a challenging task to recognize the jersey number of players on the court in soccer match videos, as the jersey numbers are very small in the object detection task and annotated data are not easy to collect. Based on the object detection results of all the players on the court, a CNN model is first introduced to classify these numbers on the deteced players' images. To localize the jersey number more precisely without involving another digit detector and extra consumption, we then improve the former network to an end-to-end framework by fusing with the spatial transformer network (STN). To further improve the accuracy, we bring extra supervision to STN and upgrade the model to a semi-supervised multi-task learning system, by labeling a small portion of the number areas in the dataset by quadrangle. Extensive experiments illustrate the effectiveness of the proposed framework.

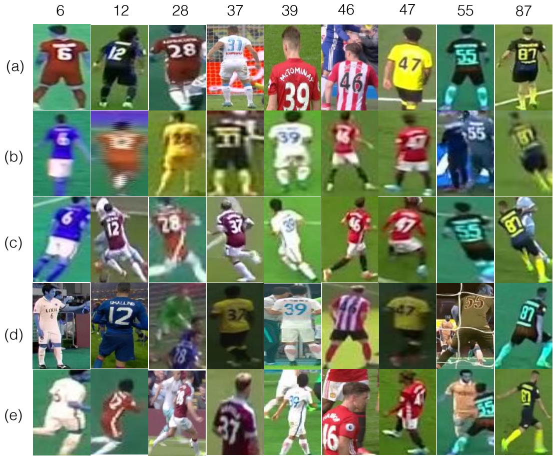

Figure 1. Illustration of our dataset. Images in a row indicate detected players in situations (a) to (e): normal, motion blur, pose tilt, light illumination, severe views. Images in a column indicate detected players of a same jersey number.

1. Introduction

Identifying players on the court is a highly desired component to automatically understand soccer match videos, which will boost new ways of story-telling on TV. Jersey number is one of the most effective characters about players that can be modeled in an easier way than other features such as face appearance, player's gait and team formation, which are either difficult to capture in a video or unstable to predict by means of camera's fast movement and low resolution of soccer matches. Although text/number recognition is a basic topic in computer vision tasks[3, 29, 25, 2, 4], jersey number recognition for real application is still a challenging problem due to camera perspective, motion blur, light illumination and soccer video resolution.

In the last decade, several works are proposed to solve jersey number recognition problem using classical methods[20, 22, 19], but they were based on manually designed features and the performance was far from satisfactory. Due to the tremendous development of Convolution Neural Networks (CNN) recently[11, 27, 12, 23], we are able to build a CNN model which is robust to different variations of data to fully explore the field which has not been paid attention to recently. A deep CNN based approach to the problem of automatically recognizing jersey numbers from soccer videos was firstly presented in [8], in which the histogram of oriented gradients (HOG) with a linear support vector machine (SVM) was used to detect players and a CNN to recognize jersey numbers from the previous results. However, the model is relatively shallow and only crops the top part of the player's images as detection prior, making their model sensitive to the detection results.

We intend to decouple this problem into a two-stage problem, making it much easier to solve in an end to end

[1] Equal contribution.
[2] This work was performed at Toutiao AI Lab.

2160-7516/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPRW.2018.00231

solution. In the first stage, the players on the court can be detected based on human detectors and then the jersey number on each player can be recognized in the second stage. Due to the improvement on modern object detectors[17, 6, 23, 10], we are able to detect the players on the court precisely. A Single Shot Detector (SSD) detector introduced in [17] is used to detect all the players on the court in our system in the first stage, then we establish a dataset for jersey number recognition with the detection results, some of which are shown in Figure 1. Inspired by recent works [11, 7], we devise a CNN model to recognize jersey numbers as our base model.

However, as shown in Figure1, recognizing the jersey number from detection results is still a challenging problem due to the camera's perspective movement, motion blur, pose tilt and many other barriers. In order to eliminate these adverse impacts, a natural way is to use an another model to locate the jersey number area and transform it into a normal observation view. However, localizing the jersey number is non-trivial due to the small size and low resolution, and the huge volume of training data is usually hard to gather in live soccer matches.

To improve the accuracy by localizing the jersey number more precisely, we focus on spatial transformation capability, as there are several prior works focusing on model transformations with the neural network or learning spatial invariant features[28, 14, 16]. To generalize the module to CNN but without a redundant neural network which will lead to extra consumption, we propose to fuse spatial transformer network (STN)[14] into our model that can be trained in an end-to-end way.

STN is a module that can be inserted into existing convolution architectures meanwhile spatially transform feature maps or even origin images without any extra supervision or model modification. STN module can be viewed as a form of attention mechanism to allocate the valid area of an image. These properties make it suitable to handle the image distortion in jersey number recognition with little extra consumption and the result of our experiments verified its effectiveness. We visualized the transformation result learned by STN as shown in Figure 3. It mainly focused on the upper half of player's bounding boxes, where the number area size is still relatively small. Thus there is still space to advance the localization performance.

In order to obtain a more precise localization of the jersey number area, we manually labeled some jersey number areas in a small part of detection results as extra supervise signal, making STN become a semi-supervised localizer and spatial transformer at the same time. The model becomes a multi-task system with our semi-supervision method for real coordinates. With only part of our data is labeled with jersey number localization, we highlight the performance of STN by a large margin and the result will be reported in the following parts of our paper.

Our contributions are mainly five folds :1) a better CNN network than [8] is proposed as our base model towards solving the jersey number recognition problem. 2) We use STN to enhance the performance of our network, bringing in the capability of localization and spatial transformation. 3) We add extra localization supervision on the STN module, training our proposed system end-to-end in a semi-supervised way. 4) We compared with current best system on jersey number recognition and demonstrate the effectiveness of the proposed methods. 5) We will make our dataset public to the research community.

Our paper is organized as follows. Section 2 discusses recent works related to jersey number recognition, STN and number recognition in the natural scene. Section 3 talks about our model design and implementation details. Experiments and results are given in Section 4. In the end, we will conclude our framework in Section 5.

2. Related Work

In this section we introduce recent works about jersey number recognition, followed by a short overview about STN.

2.1. Jersey Number Recognition

Early works about jersey number recognition mainly use hand engineered features[1, 20, 22, 19]. [1] introduces three kinds of human manipulate features for number recognition on basketball players' jerseys. [22, 21] uses image color features to extract number or text in sports videos followed by an OCR system. [24] detects player's numbers in the HSV color space and leverages number contours to yield the result. [30] identifies jersey number on JUV color space and designs some rules to filter outliers. [18] uses DPM to detect players in basketball videos, and then adopts gradient contrast with an OCR system to detect and recognize jersey numbers. All above works adopt manually crafted features and could be only applied to limited circumstances due to the lack of generalized features.

[8] is the first work to use CNN to recognize jersey number and the performance is significantly improved compared to earlier works. It uses HOG features together with a linear SVM to detect players and classify the players' bounding-box image with a six-layer convolution neural network. However, without considering any treatment about the image distortion or blur, they get a relatively better result compared with HOG method due to the power of CNN feature.

2.2. Spatial Transformation

The approach[14] explicitly solved spatial transformation problems using Neural Networks, which is easy to implement and embed into an existing system. It parameterized the transformation into a differentiable neural network module, processing cropping, translation, rotation, scale jittering and skew on the input feature map. However, the accuracy of the transformation is inferior due to the lack of supervision. After using part of the data with extra quadrangle labeling, we obtained a powerful number recognizer that could successfully localize the number area and perform a solid transformation, which eliminates poor-quality images in the soccer game live videos.

3. Proposed Methods

In this section, we first introduce our baseline model with a deep CNN that can classify jersey number from 0 to 99, without training a 100-class classifier by the elaborate design. Next, we will talk about how to embed an STN module into our baseline model, which brings in the ability of spatial transformation without extra data information. Then the extra supervision is added, upgrading the STN to a semi-supervised module for further performance improvement.

3.1. Base Model: A CNN for Number Recognition

Due to the low resolution of the live soccer matches, we propose a baseline model instead of using off-of-shelf models like ResNet or VGGNet[26] to recognize jersey numbers from the dataset above-mentioned, which contains several convolution blocks. Each block consists of a convolution layer, a batchnorm layer[13] and an activation layer. Skip connections are added between two blocks that have the same size just like the way in [11]. Comparing with the model introduced in [8], our model is deeper and take usage of the modern designs on CNN architecture such as batch normalization and skip connection. The implementation details are depicted in Section 4.2.

We implement two methods to classify numbers using extracted CNN features. The first approach is to use two separate classifiers directly using the feature yielded from the same CNN. Each classifier is responsible to recognize one digit, making the classifier predict 10 probability each time, i.e., number 0 to 9 represents all numbers may occur and number 10 represents not exist. The final prediction is as follows, almost like the method used in [8]:

$$\begin{aligned} &Num_{predict} = \{Y_{predict1}, Y_{predict2}\} \\ &Y_{predict1} = argmax\{0, 1, ..., 10\} \\ &Y_{predict2} = argmax\{0, 1, ..., 10\} \end{aligned} \tag{1}$$

Where $Num_{predict}$ indicates the final result which is combined with the two number classifier outputs $Y_{predict1}$ and $Y_{predict2}$. The disadvantage of such design is that each classifier is trained to detect and predict the value of the particular number that it is responsible for. However, sometimes it might be difficult for such a model to separate two numbers clearly, because there is no explicit separator in jersey number and camera perspective variation may make it more severe.

Another design is firstly proposed in [9]. It uses two classifiers to recognize numbers just like the first approach, but it adds an extra classifier to estimate the length of the number sequence, which enables the classifier to solve some of the false positive cases. One particular case is that the number needs to be detected is 1 and our first approach of Equation 1 outputs 11, means both classifiers have a valid result. After we adding a sequence length predictor, even if both classifiers have their outputs, we will only take one value of them to make the right prediction if the length is one. The method can be shown as follows:

$$\begin{aligned} &Num_{predict} = \{Y_{predict1}, Y_{predict2} | Length_{predict}\} \\ &Length_{predict} = argmax\{0, 1, 2\} \\ &Y_{predict1} = argmax\{0, 1, ..., 9\} \\ &Y_{predict2} = argmax\{0, 1, ..., 9\} \end{aligned} \tag{2}$$

$Length_{predict}$ means the length of numbers that is predicted. Matching the fact that there are no three-figure cases of jersey number in our dataset, the values of length prediction is between zero to two. The number value classifier is set to predict values from zero to nine, because we use the length of sequence value to control the final predict. If there are some cases that we must choose one value of the two classifiers, we will pick the first predicted number by default. The disadvantage of this approach is that its performance completely relies on the prediction of length classifier. The predicted results might get worse if the length predictor is not well trained.

Although there are still many other solutions for the number recognition problem, we won't lucubrate into this anymore, which is beyond our discussion.

3.2. Model with Spatial Transformers

The drawbacks of the base model mentioned above are as follows:1) It could not localize the area precisely which is actually useful for the result. Although the architecture of pooling in CNN can reduce the impact of spatial translation, it is limited by the depth and kernel settings of the neural network, making CNN not robust to recognize small objects. 2) CNN sometimes fails to classify objects with non-rigid deformations, such as scaling jittering, cropping and rotations itself. Researchers often employ data augmentation to solve these issues partially. We intend to bring the capability of attention and transformation on the key area of origin image by using STN proposed in [14]. It is designed as a module that can be incorporated into CNN without any other bells and whistles, making it very suitable for our demand because of its efficiency and effectiveness.

STN as shown in Figure 2 can be split into three parts: Localization, Sampling Grid, and Image Sampling. First,

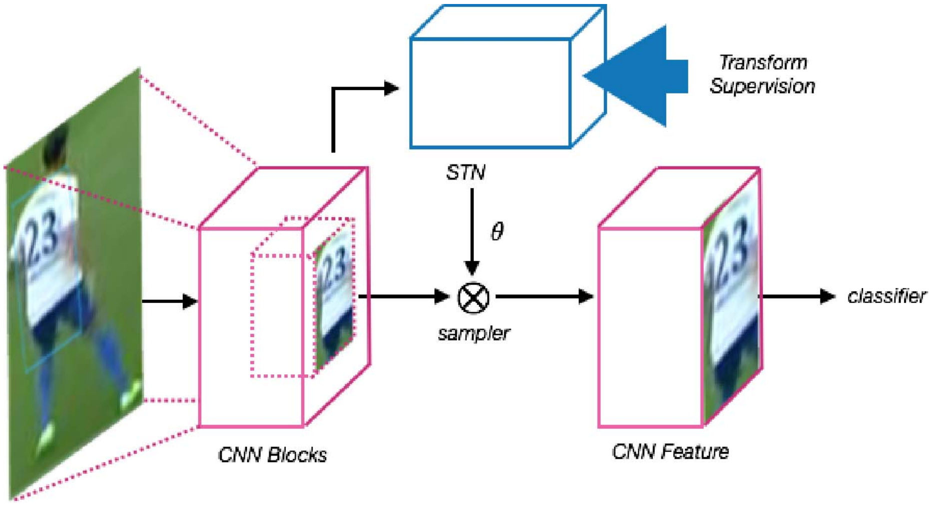

Figure 2. The basic architecture of our proposed method. It contains several CNN blocks for feature extraction and an STN is inserted after CNN blocks to produce the transformed features for the final classification. We can also add localization loss to enhance SxTN's performance.

the localization network will inference the parameters of spatial transformations, which is often arranged as a matrix which could be affine or perspective transformation regard of the demand. Second, in order to warp from the input feature map, we use the transformation matrix to produce a sample grid on the input feature map. Third, the input layers and sampling grid are taken to generate the output feature map, using an image sampling kernel which defines the interpolation method.

The localization network takes the input feature map $I \in \mathrm{R}^{\mathrm{CxHxW}}$. C is the number of channels, H and W are the hight and width of the input layer. The transformation parameters θ are the network outputs. The parameters of affine transformation that we apply can be learned and arranged as matrix shown as follows:

$$f_{affine} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{bmatrix} \tag{3}$$

The localization network is a three-layer CNN with skip connection like [11]. We have tried other architectures like fully connected or complete convolution neural network, but experiments show that adding residual connections like ResNet makes the system converges faster and more stable.

After the parameters required by the transformation are calculated, we perform warping of the input feature as follows:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{4}$$

We normalize the width and height of the input feature map and the output feature map by making $-1 \leq x_i^s, y_i^s \leq 1$, $-1 \leq x_i^t, y_i^t \leq 1$, where x_i^s, x_i^t are the x coordinates of source layer and target layer, so are y coordinates. A_θ is the transformation matrix which is affine transformation in our implementation. Then we can attend to the area of the input by learning parameters given by the localization network, and such grid sample can be produced by all the coordinates of (x_i^s, y_i^s).

After sampling grid and producing the corresponding coordinates from the input feature map, we need to sample values from the input feature map due to the missing alignment caused by the transformation. We use bilinearly interpolated values from the nearest neighbors and the output values can be given as follows:

$$\begin{aligned} O_{i,j}^c = \sum_n^H \sum_m^W I_{nm}^c max(0, 1- \mid x_i^s - m \mid) \\ max(0, 1- \mid y_i^s - n \mid) \end{aligned} \tag{5}$$

where I_{nm}^c is the value at location (m, n) in channel c of the input and $O_{i,j}^c$ is the output value for pixel i at location (x_i^t, y_i^t) in channel c. So each channel is transformed in an identical manner. The whole architecture of the network is shown in Figure 2.

3.3. Semi-Supervised Spatial Transformer

The performance of the system is boosted after adding the spatial transformer module. However, we found that there was still room to improve after analyzing the transformation results learned by the localization network. As

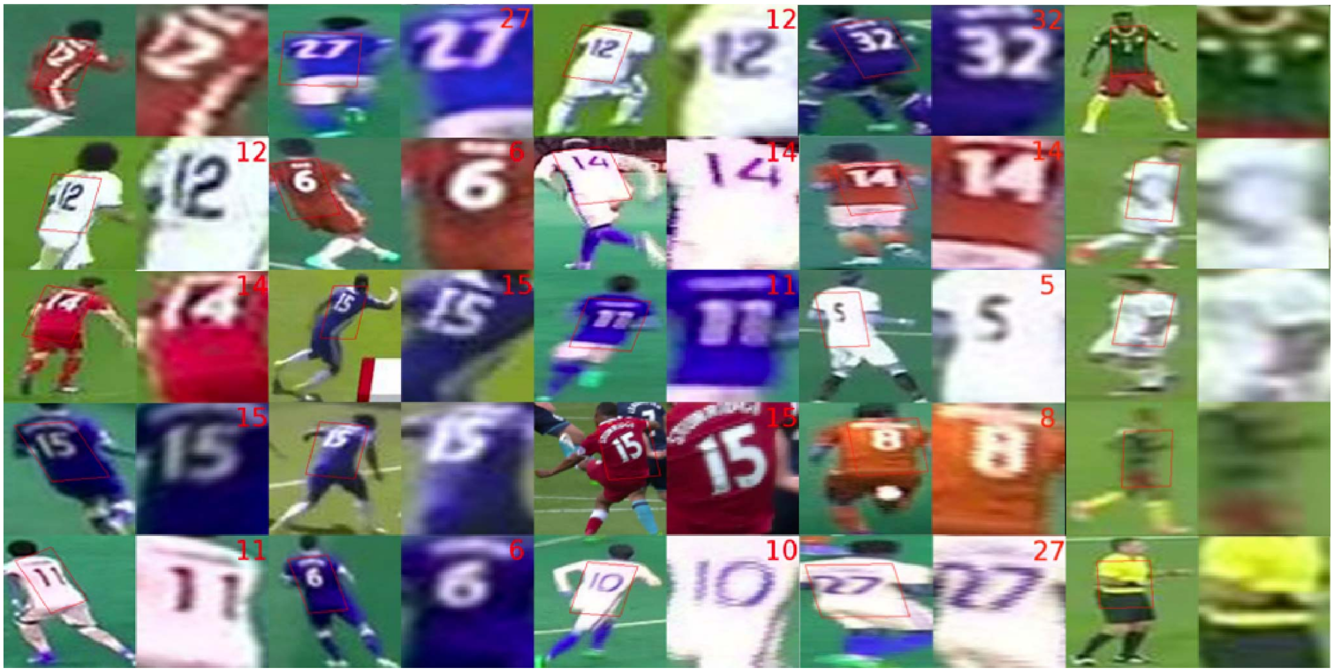

Figure 3. Examples of results using spatial transformer. Odd column contains origin player images, while even column contains the spatial transformed image with recognized results.

shown in Figure 4, the localization module can be attended on the area that contains numbers, but mostly it crops the upper half of the player's body. The result of the transformation is still defective. The original STN embedded in our model is only supervised by the classification label of the whole player's image that is already detected and try to minimize the overall cost function during the training process.

We attempt to add some additional signals to help STN optimize better in order to enhance the final performance of our recognizer, thus making the whole learning technique becomes a multi-task system. Jersey numbers are usually painted on the back of the uniform and can be labeled by a quadrangle to improve our supervised method based on STN, which can be seen as an extra localization loss. However, labeling quadrangle needs a lot of human labor and we cannot afford two much, so we only manually labeled a small part of the cropped player images. After data labeling, we need a mechanism to jointly train on the full classification data and detection-specific information data. So the total loss function is proposed as follows:

$$L = \frac{1}{N}(L_{cls} + \alpha \mathbf{1}_A L_{loc}) \tag{6}$$

Where N is the batch number during training, α is a weighted term set to be 1 by cross validation. The classification loss is the softmax loss and the localization loss is smooth ℓ_1 similar to [23], which is shown as follows:

$$L_{loc}(p,g) = \sum_{c\in(x,y)} \sum_{m\in(p1,p2,p3,p4)} smooth_{L1}(p_m^c - g_m^c) \tag{7}$$

where p indicates the coordinates that STN predicted, g indicates the coordinates of ground truth, c means we need compute the loss on both x, y coordinate axis and m indicates the four points that depict a quadrangle. All the coordinates are normalized between [-1, 1], making the localization loss smoothly.

During training we mix all the data from both classification and localization data. When we get data with extra labels for localization, we can back-propagate the full loss, otherwise we will only back-propagate the cross entropy loss through the softmax layer when it sees a classification data, which makes our whole model as a semi-supervised spatial transformer network (SSTN).

4. Experiments

In this section, we introduce our dataset, training settings and experiment results, respectively.

4.1. Dataset

Soccer is one of the most popular sport in the world. In order to eliminate the information of the players and jerseys appearance influence, we gather 164 soccer match videos spread from The Premier League to AFC U-19 Championship. Most of the videos are recorded in at least 1080x720 resolution, about 10% of them are in a higher resolution of 1920x1080.

We use a detector based on SSD[17] to detect players on the court of the videos we mentioned above. Due to the players' pose variation, there are no jersey number on most of the detected results. So we manually label 215036 detected results with their ground truth numbers and get a dataset of cropped players, about 90% of which are negative samples. In order to supervise STN with localization signals, we make 12746 samples in the dataset labeled with quadrangle to fit the number area as much as possible. We name our dataset as SJN-210K and will make it available to the community.

4.2. CNN Network

We implement our base model on a widely used modern framework MXNet[5]. Our CNN is composed of seven convolution blocks, each block has a 3x3 convolution layer followed by a batch normalization layer[13] and relu activation[15]. Max-pooling layers are used after the first, third, fifth and the last block. Same as ResNet architecture, we add shortcut connection between the two block with the same size. At the end of the network, convolution features are flattened and connected with a fully connected layer to use softmax as our classifier. There are two separate softmax classifiers if we predict two numbers at a time. An extra softmax classifier is added when it's needed to predict the length of jersey numbers.

4.3. CNN with Spatial Transformer

As explained in Section 3.2, we apply an STN module into the above-mentioned CNN model. The CNN model's last convolution layer has an output with 16*16 resolution, which is flattened into a 256-dim vector and becomes input for the following 256 hidden fully connected layer. The last output fully connected layer's hidden dim depends on the transformation we use, either 6-dimension for affine transformation or 9-dimension for perspective transformation. The output of STN is used for sampling grid and bilinear interpolation from the last convolution layer.

4.4. Semi-Supervised Spatial Transformer

The origin STN implemented in [14] is a module which is trained without supervision of the task, which often converges slowly and yields inferior result. We add extra ℓ_1 loss same as mentioned in [23] to supervise STN's results to make its performance more reasonable. Because both the input and output coordinates are normalized into [-1,1], we can compute the output feature map's coordinates after it is transformed. The differences between the ground truth and coordinates that are transformed can be computed with an ℓ_1 loss, which can be used to supervise the whole model. Due to the limitation of our dataset, we only back-propagate the ℓ_1 loss when we get the quadrangle label.

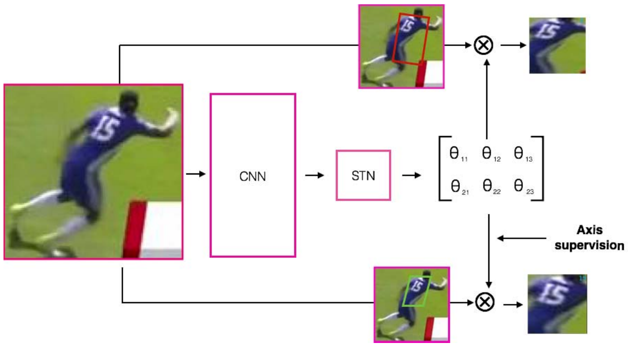
CNN
STN
θ_{11} θ_{12} θ_{13}
θ_{21} θ_{22} θ_{23}
Axis supervision

Figure 4. Comparison between semi-supervised STN and unsupervised STN. With axis supervision, the spatial transformer localizes better.

4.5. Implementation Details

We comprehensively evaluate our method comparing with the baseline model proposed in [8], which achieves the best performance on jersey number recognition recently.

Without access to their dataset, we reproduce their models based on our self-build dataset and use the same training settings as follows. We crop a bounding box on every player image and resize it to 200x200 as our training samples. If we rescale the size of the origin layer image to [0,1], the left coordinate of jersey number's bounding box lies between [0.1, 0.2] relative to origin player image, and right, top, bottom coordinates lie between [0.8, 0.9], [0.1, 0.2], [0.8, 0.9]. All the models are trained with RGB images and no other data augmentation methods are implemented. We use the metric of precision, recall and F1 for the positive samples because 90% images are negative as they do not include jersey numbers. All networks are trained by starting with a learning rate of 1e-5 with SGD.

4.6. Experiment Results

We conduct experiments on the introduced dataset to demonstrate the effectiveness of the proposed models and improvements.

To achieve better results, we try to find the best base model. As mentioned before, we add number length supervision as [9] do. But in our dataset, the number length supervision does not work, as jersey number lengths are always the same, either 0 or 2 in most cases. Beside, we do some trials on data augmentation as mentioned in [8]. As shown in Table 2, adding crops to image produces 3% drop on F1 score. Because in our dataset, numbers are not always in the center of image, random cropping can make the numbers incomplete. Nevertheless, cropping will remove the

Method	Number level Acc	Digit level Acc	Precision	Recall	F1
Gerke [8]	0.714	0.820	0.315	0.282	0.298
Gerke [8] + crop	0.719	0.828	0.324	0.292	0.307
Base model	0.853	0.909	0.677	0.645	0.660
Base model + STN	0.861	0.913	0.699	0.670	0.684
Base model + STN + Supervision	0.867	0.918	0.714	0.680	0.696

Table 1. Experiments on SJN-210k dataset

head part of a player, which is useful to judge the player's heading. So we produce another comparing set which only cut the bottom of image randomly. Results show that it is better than cropping both, but overall performs nearly the same as the base model.

Method	Precision	Recall	F1
CNN+SC+Length	0.646	0.642	0.644
CNN+SC	0.677	0.645	0.660
CNN+SC+Crop	0.656	0.615	0.635
CNN+SC+Crop(bottom)	0.671	0.647	0.659

Table 2. Variations on base model.

Then we compare our method with Gerke's method [8], results are shown in Table 1. As Gerke's dataset is not available on the Internet, comparisons cannot be made on their dataset. Alternatively, we implement their method by following the details in their paper. As their input images are 40x40, we also randomly crop the images when training their model (0.1 to 0.2 of width margin, 0.1 to 0.3 of top margin), data augmentation including gray image inverse and dropout is added.

Still, our model achieves better performance, with 30% increase in both precision and recall. This is because deeper convolution layers and skip connections ensure better feature extraction and more stable and flexible back-propagation.

On the basis of our base CNN model, a spatial transformer network is inserted, where the localization network is a two-layer fully connected neural networks with 256 units in it. Then a grid sampler is added to the last convolution layer in the CNN model previously mentioned, generating an output feature map using bilinear interpolation. And the interpolated feature map is used to predict the outputs. As we can see, adding STN increases both precision and recall by 2% percent, and severely tilt images could be recognized. Examples in Figure 3 show that by adding the STN layer, jersey number of tilt player could be localized and reconstructed well. Even in negative samples (jersey number do not appear) as shown in the last column, STN could help to focus on the center body of a player. So our experiments show that STN works well both as a localizer and a spatial transformer.

But still in some cases, we find that spatial transformer can only localized a rough position of the jersey number. So adding a bounding box supervision would help the spatial transformer to perform better. We add a parallel branch after STN's output to get the transformed coordinates of the input feature and compare it with the ground truth numbers' coordinates, generating a ℓ_1 loss to supervise STN's training process. Even with only 6% of the data labeled, we still get a 0.8% improvement after training. We also try to visualize the difference in Figure 4, where axis supervision obviously help the network to localize better. With more and more bound box labels of the data, better results could be expected.

5. Conclusions

In this paper, we target at the problem of recognizing soccer players jersey number from live match videos. We separate this issue into a two-stage problem due to the limitation of jersey number resolution and resource consumption, and try to dig a better performance on the recognition phase. After detecting players on the court, we present a novel CNN architecture for jersey number recognition on players' bounding box images, outperforming the current best performance introduced in [8]. In order to enhance the recognition performance, we use STN to localize and transform the numbers on the bounding boxes without adding another stage for number detection. Part of data is labeled with quadrangle to supervise STN, boosting the final performance of the above-mentioned model.

6. Future Work

Due to the uninterrupted improvements of detection and recognition in computer vision, recognizing jersey numbers get renewed interest recently, as the biggest obstacle of perceiving every player on the court become how to inference the number of players with their back oriented to the camera. To further explore and help boost research of the jersey number task, we will maintain our dataset and collect more truthful images after we make our dataset publicly available. Due to the expensive manual force, we will apply unsupervised learning method to better the information of huge volume of unannotated data.

References

[1] G. Àlvarez Criach. Comparison of methods for the number recognition of sport players. 2010.

[2] C. Bartz, H. Yang, and C. Meinel. See: Towards semi-supervised end-to-end scene text recognition. *arXiv preprint arXiv:1712.05404*, 2017.

[3] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 785–792, 2013.

[4] M. Bušta, L. Neumann, and J. Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2223–2231, 2017.

[5] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[6] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017.

[8] S. Gerke, K. Muller, and R. Schafer. Soccer jersey number recognition using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 17–24, 2015.

[9] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 991–999. IEEE, 2015.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, 2016.

[18] C.-W. Lu, C.-Y. Lin, C.-Y. Hsu, M.-F. Weng, L.-W. Kang, and H.-Y. M. Liao. Identification and tracking of players in sport videos. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, pages 113–116. ACM, 2013.

[19] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1704–1716, 2013.

[20] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little. Identifying players in broadcast sports videos using conditional random fields. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3249–3256, 2011.

[21] S. Messelodi and C. M. Modena. Scene text recognition and tracking to identify athletes in sport videos. *Multimedia tools and applications*, 63(2):521–545, 2013.

[22] J. Poignant, L. Besacier, G. Quénot, and F. Thollard. From text detection in videos to person identification. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 854–859, 2012.

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2017.

[24] M. Šarić, H. Dujmić, V. Papić, and N. Rožić. Player number localization and recognition in soccer video using hsv color space and internal contours. In *The International Conference on Signal and Image Processing (ICSIP 2008)*, 2008.

[25] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016.

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

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[28] T. Tieleman. *Optimizing neural networks that generate images*. PhD thesis, University of Toronto (Canada), 2014.

[29] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4049, 2014.

[30] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao. Jersey number detection in sports video for athlete identification. In *Visual Communications and Image Processing 2005*, volume 5960, page 59604P. International Society for Optics and Photonics, 2005.