Soccer Jersey Number Recognition Using Convolutional Neural Networks

Sebastian Gerke
Fraunhofer HHI
Einsteinufer 37, 10587 Berlin, Germany
sebastian.gerke@hhi.fraunhofer.de

Karsten Müller
Fraunhofer HHI
Einsteinufer 37, 10587 Berlin, Germany
karsten.mueller@hhi.fraunhofer.de

Ralf Schäfer
Fraunhofer HHI
Einsteinufer 37, 10587, Germany
ralf.schaefer@hhi.fraunhofer.de

Abstract

In this paper, a deep convolutional neural network based approach to the problem of automatically recognizing jersey numbers from soccer videos is presented. It is meant as a tool for subsequent automatic player identification approaches that utilize jersey numbers together with knowledge about teams and the jersey numbers of their players. Two different jersey number vector encoding schemes are presented and compared to each other. The first treats every number as a separate class, while the second one treats each digit as a class. Additionally, the semi-automatic process for the annotation of a jersey number dataset consisting of 8281 jersey numbers is described. The best recognition rate of 0.83 was achieved by the proposed approach with data augmentation and without using dropout, compared to 0.4 for a more traditional histogram of oriented gradients (HOG) and support vector machine (SVM) based approach.

1. Introduction

Soccer is one of the most popular sports in the world. In recent years, interest in automatic soccer analysis tools grew significantly. Soccer analysis results can be used for new ways of storytelling on TV, for match preparations or for the generation of statistics. One of the fundamental analysis is the identification of players to associate actions and statistics to actual players. However, identifying players in broadcast soccer videos automatically (and even manually) is challenging. Especially for the overview camera it is difficult due to the low resolution per player, which makes face recognition impossible and jersey numbers are often hard to read, especially with standard definition resolutions. Only with the rise of widely available HD content in recent years, jersey number recognition became feasible.

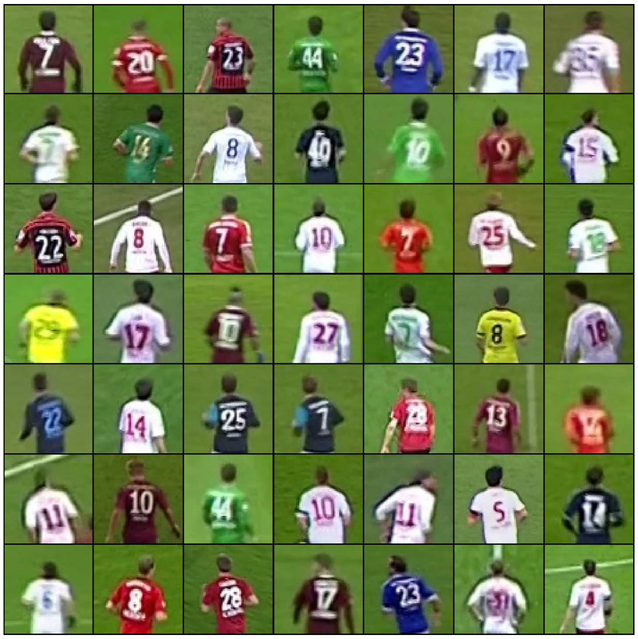

Figure 1. Examples of the player detector output that is used to create the dataset presented here. The upper half of the actual player bounding box is shown.

2. Related Work

Existing approaches for automatic player identification in broadcast soccer videos can be categorized in two groups: One performing face recognition on closeup shots (not overview shots) in variuos types of sports videos, while other approaches rely on jersey number recognition. For the latter group, no approach is known to operate on soccer overview shots. They either operate on other sports where the resolution per player is higher (e.g. in basketball [5, 9]), or they perform on closeup shots, where jersey numbers are better readable [1] and face recognition is feasible [2].

In [9] basketball players are detected using a deformable parts model (DPM), after which an exact localization of the jersey number is performed. Then, normalization, followed by thresholding and calculating the correlation between the digits and digit templates is applied. In [2], player identification is performed in overview shots by employing SIFT features for face recognition.

All approaches have a quite sophisticated, hand-engineered image processing pipeline in common. They often perform explicit localization of jersey numbers, followed by digit segmentation. In contrast to these approaches, a deep learning approach is proposed here. It does not rely on explicit localization of jersey number regions and no explicit segmentation is performed. Rather, a deep convolutional neural network is trained which handles the complete pixel-to-jersey number recognition process.

3. Dataset Generation

Although there have been a few existing approaches for jersey number recognition, these approaches usually rely on video content where the image area of a single jersey number is relatively large, as they either originate from medium and close-up shots in soccer video broadcasts or from sports where the main camera has a narrower camera angle, e.g. in basketball broadcasts. Therefore, a ground truth dataset consisting of 10,000 cropped images (from 65 different soccer videos) containing soccer players and labelled by their jersey number (if visible) was created using manual and automatic labelling cooperatively. The workflow of the annotation process is depicted in figure 2.

3.1. Semi-automatic Labelling

First, an automatic player detection based on histogram of oriented gradients (HOG) [4] together with a linear support vector machine (SVM) was performed on 65 different soccer videos, similar to what is described in [6]. For each video, 100 random overview frames were selected and the player detector was applied in a high-precision setting in order to increase the probability for a true positive. This resulted in approx. 70,000 cropped images of players.

Then, a small subset of 2300 of these cropped player images was labelled if their jersey number is visible or not. By presenting this binary classification to the human annotators, this classification task is actually simpler and therefore faster than annotating whole numbers. A linear SVM classifier on HOG features was trained on this classification task. This should increase the number of cropped players presented to human annotators where a number is visible and readable. This classifier was applied to 70,000 cropped player images, of which the highest ranked 10,000 images were used for manual jersey number labelling. This step is crucial, as in most of the 70,000 images a number is not even visible, which would yield a very sparsely annotated dataset.

These 10,000 images are then manually labelled. Volunteers were asked to either assign the visible number (basically from 1 to 44, excluding a few numbers that are not present within the whole dataset), or they could indicate why it was not possible to assign a number. This could be one of *not visible*, *not readable*, *multiple players* and *box error*. *not visible* is supposed to be assigned to images where the number is not visible at all, while *not readable* is supposed to be assigned to images where the number is either only partly visible due to the player's pose or not readable, e.g. due to motion blur or illumination. *Multiple players* refers to images that contain more than one player and it is not obvious which player the annotation refers to. *Box error* is supposed to be assigned to images where a player is not correctly detected, being too small or too big. While the relatively fine-grained annotation of error cases might be usefull for future research, the error classes are currently not used.

After annotating each of the 10,000 images, a validation step was performed to reduce the number of false annotations. Therefore, all images of a jersey number are shown. False annotations are then easily visible and are corrected.

When analyzing the ratio of images that contain visible players both in the small initial subset of 2300 images and in set of 10,000 images that was selected by the aforementioned ranking, the ratio of images with visible numbers could be increased significantly. From the small subset, 1010 out of 2300 images have visible numbers (ratio 0.43), while about 8,000 out of 10,000 images (ratio 0.8) have visible (and readable) numbers on the pre-ranked dataset. That means by using this pre-processing step, the effort for obtaining 8,000 labelled samples was reduced by almost 50%.

For experimenting with automatic jersey number classifiers, the dataset described above is split into a training and a test corpus. It is split by video, i.e. all images from a video are either in the training or in the test set, in order to avoid unrealistic scenarios where classification relies on training samples from the same video. After splitting, the training corpus consists of 5759 images and the test corpus consists of 2520 images.

3.2. Dataset Properties

The number distribution is shown in figure 3. It shows that numbers are not equally distributed, but rather imbalanced. While there are e.g. 600 samples for number 10 (the most frequent number), there are only 7 samples for number 41. This could actually make training a classifier a challenging task. In comparison to a similar datasets, the Street View House Number dataset (SVHN)[10], where digits between

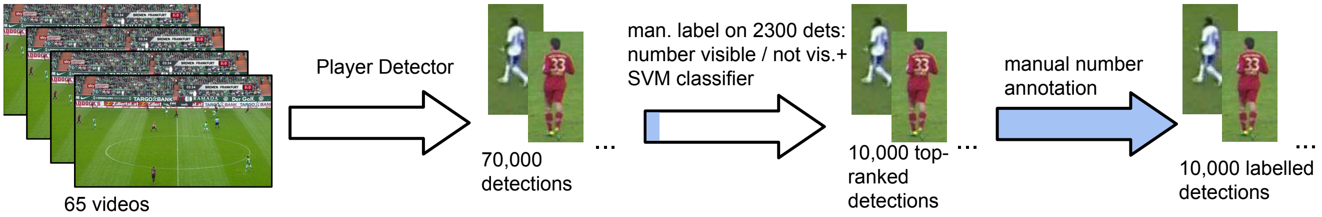
65 videos
Player Detector
70,000 detections
man. label on 2300 dets: number visible / not vis.+ SVM classifier
10,000 top-ranked detections
manual number annotation
10,000 labelled detections
...

Figure 2. Workflow of the semi-automatic jersey number dataset annotation process. Blue arrows denote manual annotation steps.

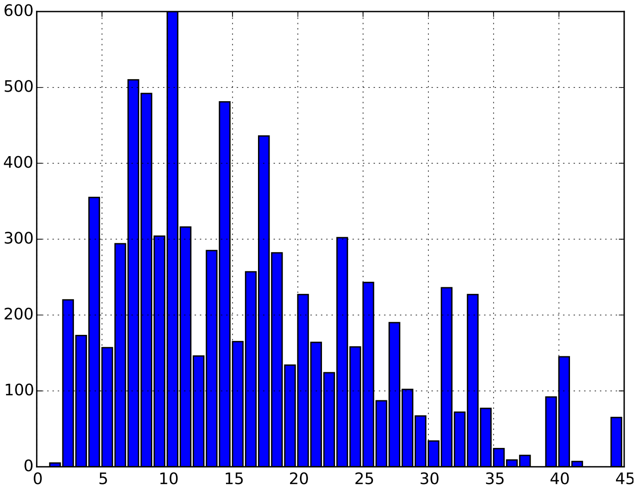

Figure 3. Jersey number distribution within the complete (training + test) dataset.

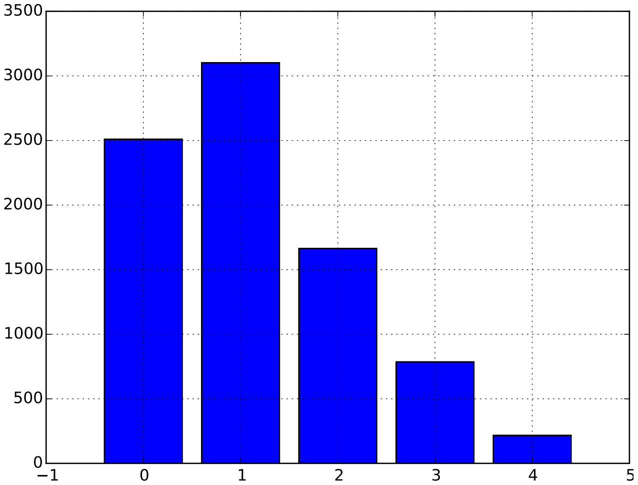

Figure 4. Distribution of first digit of jersey numbers within the complete (training + test) dataset.

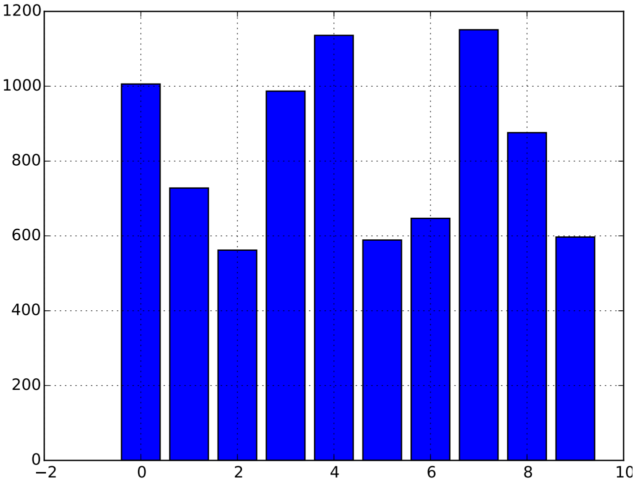

Figure 5. Distribution of second digit of jersey numbers within the complete (training + test) dataset.

0 and 9 are annotated, the ratio between the most frequent and the most rare label is much larger: It is 86 for the dataset presented here and 3 for the SVHN dataset.

3.3. Comparison to other datasets

Table 1 gives an overview of key dataset characteristics of the presented dataset in comparison to similar computer vision datasets. It consists of the soccer jersey number dataset presented here (SJN), the MNIST database of handwritten digits dataset [8], the Street View house number dataset (SVHN) [10] and the traffic sign recognition dataset (TS) [12]. As can be seen, the dataset presented here consists of more classes than the MNIST or SVHN dataset, as jersey numbers are classified as whole numbers, not a sequence of digits. This means there are fewer positive samples per class than for these datasets, even if the dataset would be perfectly balanced. However, in section 4 we present an alternative coding scheme that separately models digits and yields a more even distribution of classes. Additionally, the resolution for other datasets is usually smaller, but the 64 × 128 resolution is the actual bounding box size of the whole player. For jersey number recognition, it is sufficient to only consider the upper half of the bounding box. Within the upper half of the bounding box, the precise location of the jersey number is not annotated. That makes this task harder than other datasets, where the number (or traffic sign) locations are annotated manually and therefore more precise. Similar to the SVHN and TS dataset, the dataset consists of RGB color images, whereas the MNIST dataset is a grey-scale dataset. However, the most significant difference is the actual size of the dataset. The presented SJN dataset is by far the smallest among those four, which could

Dataset	Classes	Resolution	Training	Test
MNIST [8]	10	$28 \times 28 \times 1$	60,000	10,000
SVHN [10]	10	$32 \times 32 \times 3$	73,257	26,032
TS [12]	43	$32 \times 32 \times 3$	39,209	12,630
SJN	36	$64 \times 128 \times 3$	5,760	2,521

Table 1. Comparison with other similar datasets. The image resolution and the number of channels, as well as training and test set sizes are given.

make approaches that rely on large datasets less promising. Also, given the smaller dataset size and larger problem size (number of classes), results on this presented datasets (in terms of accuracy) are expected to be not as good as reported results on the other datasets mentioned here.

4. Classification Problem

In this work, two different methods for (jersey) number recognition as a classification problem are evaluated. The first approach is to model all occuring jersey numbers as a separate class. In our case, this would mean a 40-class classification problem, as not all one- or two-digit numbers appear in the dataset. That means, that the classifier $c(x)$ assigns exactly one class (number) y to each input sample image x:

$$c(x) = y, \qquad y \in \{1, 2, 3, ..., 40\} \tag{1}$$

Alternatively, one could treat the problem as a two-label classification problem, with one label for each digit. One for the most significant digit of a one- or two-digit number, and one for the least signicant digit:

$$c(x) = (y_1, y_2), \quad y_1 \in \{10, 11, 12, 13, 14\}, \quad y_2 \in \{0, .., 9\} \tag{2}$$

where the continuous labels 10-14 stand for the first digit, i.e. 10 represents single digit numbers, 11 represents numbers whose first (most significant) digit is 1, etc. For a neural network, categorical labels are usually encoded by binary vectors whose dimensionality is equal to the number of different labels. That means that for the classification problem as described in equation 1, labels are converted to a 40-dimensional vector with exactly one dimension (that of the groundtruth label) set to one, all others element set to 0:

$$\mathbf{y}_{bin} = [0_0, \ldots, 0_{y-1}, 1_y, 0_{y+1} \ldots, 0_n]^T \tag{3}$$

The output of the neural network classifier then needs to be converted back to a class label $y_{predicted}$ by choosing the maximum element of the resulting vector $\mathbf{y}'$ (that contains real-valued entries):

$$y_{predicted} = \underset{i \in \{1,2,...40\}}{\operatorname{argmax}} \; y'_i \tag{4}$$

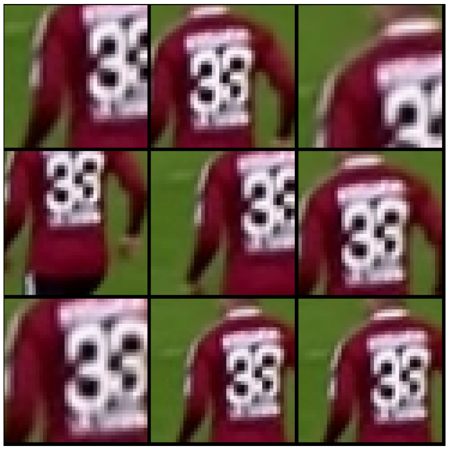

Figure 6. Training samples obtained by applying random scaling and cropping of the original sample.

For the second case, the binary vector consists of two non-zero elements for groundtruth labels. One for each digit, with numbers smaller than ten having an imaginary 0 as their first digit.

$$\mathbf{y}_{bin} = [0_0, \ldots, 0_{y_2-1}, 1_{y_2}, 0_{y_2+1} \ldots, 0, 1_{y_1}, 0, \ldots, 0_n]^T \tag{5}$$

Converting network predictions back to numbers is then a combination of the maximum element of the first 10 elements of $\mathbf{y}'$ and the maximum of the subsequent 5 elements:

$$y_{predicted} = (\underset{i \in \{0,1,2,...9\}}{\operatorname{argmax}} \; y'_i, \; \underset{j \in \{10,11,...,14\}}{\operatorname{argmax}} \; y'_j) \tag{6}$$

Both approaches have their advantages and disadvantages: As can be seen in figure 3, treating all numbers as separate classes imposes a very imbalanced dataset. Given the dataset it is even conceptually impossible to recognize two-digit numbers that do not occur, i.e. all numbers > 45. When applying two separate classification problems, it would be possible to model jersey numbers that have not been seen until 49, i.e. where for each number, each digit has been seen in all places (first and second digit of the number) in the training set. However, it might be difficult for an algorithm to separate the first and second digit of the number when no explicit localization or segmentation has been performed. Additional factors such as slight perspective changes might make separating the digits even more difficult. Therefore, it might be more appropriate to model numbers holistically.

5. Data augmentation

As the soccer jersey number dataset is quite small, data augmentation is expected to play a key role for good recognition results. Here, we apply data augmentation to increase the number of training samples from 5,760 samples to approx. 56,000 training samples. As the jersey numbers are

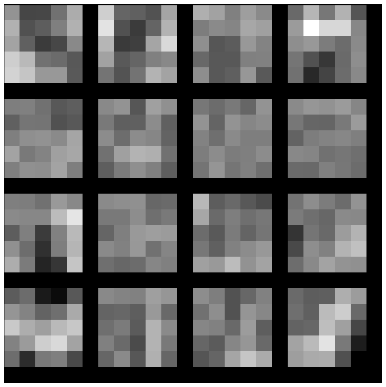

Figure 7. Output of the first convolutional layer for a sample image.

not centered within in a certain region of the image, a classifier is supposed to be tranlation invariant. In order to improve this invariance, multiple variants of an existing training sample are generated, each cropping a different 40×40 patch from the upper half of a bounding box (64×128) shifted within a certain range. Additionally, as the size of the actual region of the jersey number within the player's bounding box is not known, differently scaled samples (the scale factor is randomly choosen between 0.9 and 1.1) are generated for augmentation, as shown in the example in figure 6.

As described later, runs that operate on color and grayscale images are tested. For the grayscale runs, additional data augmentation by inverting all training samples was performed, yielding a training dataset of approx. 108,000 samples.

6. Deep Convolutional Neural Network

As a baseline, a HOG based radial basis function (RBF) kernel SVM classifier was used, similar to [10]. However in [10], a linear SVM was used. HOG features are calculated only for the upper half of player bounding boxes to reduce the influence of irrelevant image parts. On these features, an RBF kernel based SVM is trained. Using this baseline, an accuracy of 0.404 was obtained.

Additionally, a convolutional neural network was trained to recognize numbers. The Keras [3] Python library for deep neural networks was used throughout the following experiments. Its architecture is inspired by models for generic image classification (similar to a model for the CIFAR-10 [7] dataset) and recognizing house numbers in street view images (using the street view house number dataset). The base architecture consists of three convolutional layers, each followed by a max-pooling layer and a rectified linear unit (ReLU). Then, there are three fully connected hidden layers with optional dropout [11] layers and finally a softmax loss layer follows. The network architecture consists of three convolutional layers and three subsequent fully connected layers. It has been trained and tested and is described in detail in section 7. Without any further data augmentation and parameter tuning, the accuracy obtained was approx. 0.60, which is already better than the more classical HOG+SVM based approach.

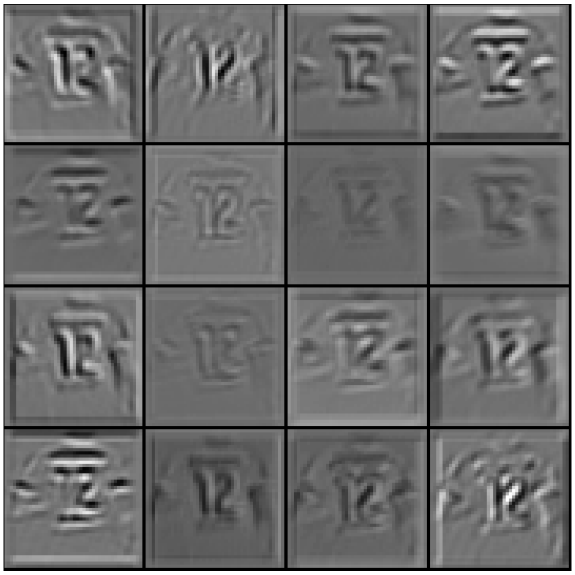

Figure 8. Output of the first convolutional layer for a sample image.

The detailed network architecture is as follows: Three convolutional layers (with $16 \times 5 \times 5$ / $30 \times 7 \times 7$ / $50 \times 3 \times 3$ parameters), each with rectified linear units (ReLU) as their activation function, followed by a max-pooling layer. Then, three fully connected layers with ReLU activation follow. Table 2 gives the details of the network architecture which holds for all runs. Only data augmentation, dropout parameters and color space vary between runs. The convolutional stride is always set to one pixel, while pooling size and stride is two pixels for the first convolutional layer and three pixels for the remaining convolutional layers. In comparison to the network architecture in [11] for the SVHN dataset, they used more filter channels ((96, 128, 256) instead of (16, 30, 50) used here) for the convolutional layers. The two fully connected layers in their work each have 2048 units, while in this work, only 34 units are used. The reason for reducing the number of units is mainly the lack of a large dataset. The SVHN dataset is two orders of magnitude larger (as an extended training corpus of the SVHN dataset was used) than the jersey number dataset used here.

Figure 9 shows sample classification results using the best-performing recognizer (ConvNet grey aug inv.) for different categories, namely 2, 3, 4, 6, 8, 10, 13, 15, 16, 21, 20 and 25. Figure 7 depicts the 16 learned convolution filters in the first layer. It shows that mainly edge filters have been learned, with some filters . Figure 8 shows the 16 responses

Stage	1	2	3	4	5	6
Layer type	conv + max	conv + max	conv + max	full	full	full (output)
# channels	16	30	50	34	34	45/15
Filter size	5×5	7×7	3×3	-	-	-
Conv. Strides	1×1	1×1	1×1	-	-	-
Pooling Size	2×2	3×3	3×3	-	-	-
Pooling Str.	2×2	3×3	3×3	-	-	-
Spatial input Size	40×40	20×20	6×6	2×2	-	-

Table 2. Deep convolutional network architecture.

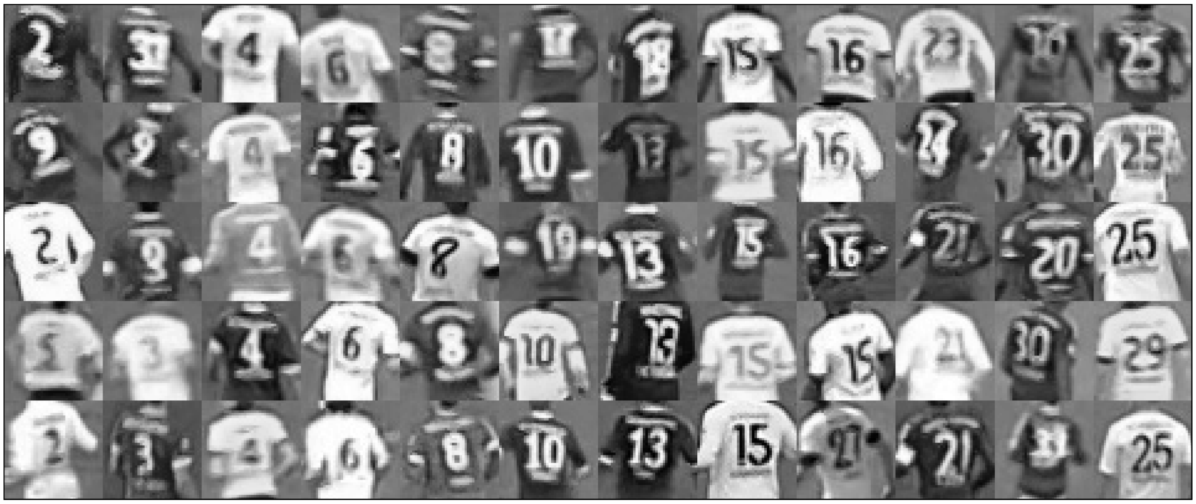

Figure 9. Sample classification results using the best configuration. Each column shows random results for the classes 2, 3, 4, 6, 8, 10, 13, 15, 16, 21, 20 and 25.

Run	Accuracy
HOG	0.40
ConvNet	0.61
ConvNet Dropout	0.71
ConvNet grey Dropout	0.72
ConvNet inv Dropout	0.76
ConvNet inv grey Dropout	0.72
ConvNet augmented grey digit-wise	0.62
ConvNet augmented	0.68
ConvNet augmented Dropout	0.71
ConvNet augmented grey	0.73
ConvNet augmented grey inv.	0.82
ConvNet augmented grey inv. Dropout	**0.83**

Table 3. Results for different approaches and settings for jersey number recognition.

for the sample image for these filters.

7. Experimental Results

In table 3, all results in terms of accuracy are given. There, *ConvNet* denotes the baseline neural network run, while *HOG* denotes the run consisting of HOG features together with a support vector machine (SVM). If the run descriptions contain the *grey* keyword, training and testing is performed on greyscale images rather than RGB color images in the standard case. *augmented* stands for spatial data augmentation as described earlier in section 5. *Inv.* stands for data augmentation by inverting images and *Dropout* for those networks with dropout layers after each fully connected layer.

During this optimization, dropout parameters were chosen carefully. When adding higher (around 0.5) dropout ratios to all fully connected layers, the obtained accuracy was below the case when moderately dropout ratios (around 0.2) were used. Also adding dropout to the first fully connected layer gave better results than adding dropout to all layers. It is assumed that the loss of information by dropping many activations in the network leads to sub-optimal results. However, overfitting was reduced and the train and test loss did not diverge, which they did when not using dropout at all.

Data augmentation by applying spatial transformations (scaling and translation) as well as applying color (or greyscale) inversion result in an increased accuracy of up to 0.83. More experiments are necessary to check if additional data augmentation is necessary to further improve performance.

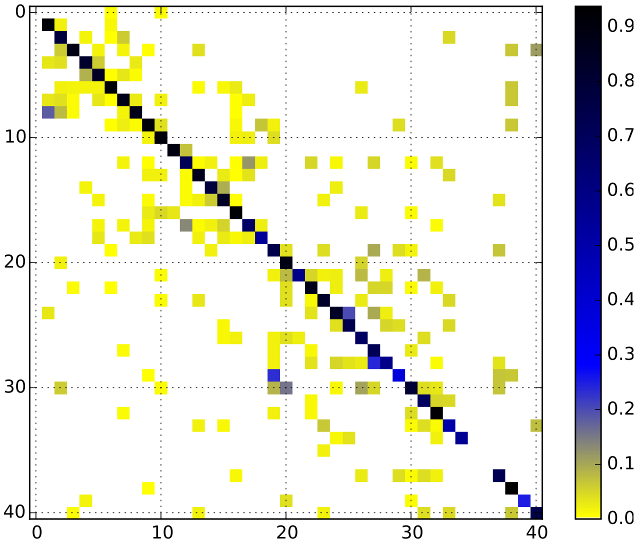

Figure 10. Confusion matrix of the best configuration. Misclassifications mostly appear where the predicted shares at least one digit with the groundtruth label.

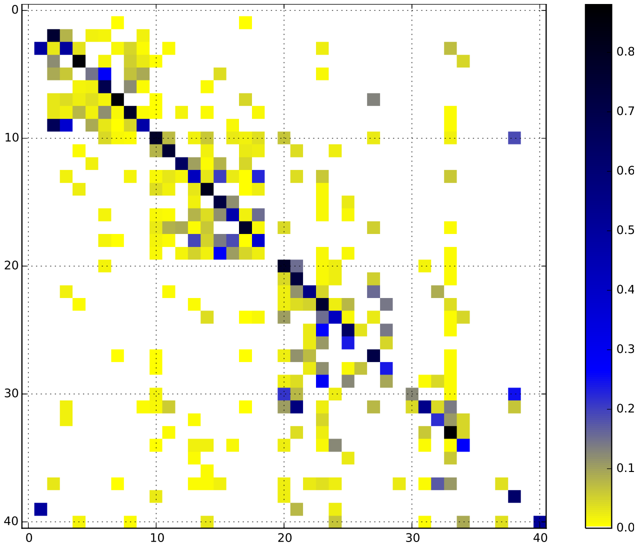

Figure 11. Confusion matrix of digit-wise classifier. Did not improve misclassifications from wrong digit order in comparison to the *one class per jersey number* configuration.

While for some configurations, utilizing full RGB color information seem to yield slightly better results than a similar network operating on greyscale images, we think that using greyscale has some advantages in terms of expected generalizability. Whenever using color information would yield better results, this might be due to some correlation between jersey colors and jersey numbers. For example, some rarely occuring jersey number might appear only in a single team. While this would help if all teams are known at training time, this correlation does not help when applying jersey number recognition to new unknown data. Using dropout for regularization did not always improve results, when the other network parameters remain constant. It was not tested if increasing the networks capacity by adding more layers or adding connections would benefit from dropout.

Modelling jersey number recognition as two separate classification problems (the *digit-wise* run) for the first and second digit did not work as good as the holistic approach. The best approach on augmented grayscale images performed worse (accuracy of 0.62) than most holistic approaches.

Interestingly, although the dataset is quite small, the accuracy reached by using deep convolutional networks outperforms that of the more traditional HOG+SVM approach by a large margin (0.83 vs 0.40). This at first sight seems to be counter-intuitive, as the promise of deep learning approaches is actually to make use of larger datasets.

For a closer analysis, confusion matrices are used, which contain correctly classified entries at the main diagonal, while wrongly classified entries occur at other positions. When looking at the confusion matrices for both the best holistic and the best digit-wise networks in figure 10 and 11, it is apparent that mainly classes that share one digit are confused. These are all confusions that are in the diagonal decimal blocks (adjacent to the true positive diagonal, i.e. where the first digit is recognized correctly, but the second one is misclassififed. The lines parallel to the diagonal – shifted by ten classes - represent misclassifications where the last digit was correctly identified but the first one was not.

In contrast to the previous assumption, modelling the two digits separately did not circumvent these misclassifications. Rather, the classification results as a whole became worse and the same misclassification errors were noticeable, apparently even more noticeable than in the holistic case.

8. Conclusion

In this paper, a dataset consisting of 8521 annotated soccer player images is presented, together with convolutional neural network based approach for jersey number recognition. The problem of jersey number recognition, which consists of one- or two-digit numbers for all known team sports, was posed as two different classification problems. One holistic approach of one class per number and one digit-wise approach that models each digit at each position within a number separately. By conducting experimental evaluations, it was shown that the holistic approach performed better throughout the experiments. Another interesting finding was that deep learning approaches yield quite good results even with smaller datasets like the one presented here. By utilizing data augmentation, the training set size can be increased significantly. Applying dropout for regularization improved results especially for those runs where no data augmentation was performed.

In the future, it would be interesting to analyze more network architectures, especially if applying dropout would allow for deeper network architectures. Another promising direction could be the use of spatial transformer networks as well as more data augmentation techniques. For example, additional rotation or perspective distortion could improve invariance to slightly different player poses.

References

[1] E. Andrade, E. Khan, J. Woods, and M. Ghanbari. Player identification in interactive sport scenes using region space analysis prior information and number recognition. In *International Conference on Visual Information Engineering (VIE 2003). Ideas, Applications, Experience*, pages 57–60. IEE, 2003.

[2] L. Ballan, M. Bertini, A. D. Bimbo, and W. Nunziati. Soccer Players Identification Based on Visual Local Features. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 258 – 265, Amsterdam, The Netherlands, 2007. ACM.

[3] F. Chollet. Keras: Theano-based deep learning library. `https://github.com/fchollet/keras`, 2015.

[4] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 886–893. IEEE, 2005.

[5] D. Delannay, N. Danhier, and C. De Vleeschouwer. Detection and recognition of sports(wo)men from multiple views. In *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–7. IEEE, Aug. 2009.

[6] S. Gerke, S. Singh, A. Linnemann, and P. Ndjiki-Nya. Unsupervised color classifier training for soccer player detection. In *Visual Communications and Image Processing (VCIP).*, 2013.

[7] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[8] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998.

[9] 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 - ICIMCS '13*, page 113, New York, New York, USA, 2013. ACM Press.

[10] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, number 2, page 5. Granada, Spain, 2011.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958, 2014.

[12] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460, 2011.