# Racing Bib Number Recognition Using Deep Learning

*Completed Research*

**Erica Ivarsson**
Berlin School of Economics and Law
erica.e.ivarsson@gmail.com

**Roland M. Mueller**
Berlin School of Economics and Law
roland.mueller@hwr-berlin.de

## Abstract

This research investigates the use of deep convolutional neural networks for racing bib number recognition in sport images. As labelled images of racing bib numbers are scarce, this paper investigates the potential benefits of transfer learning, by training models partly and fully on the Street View House Numbers (SVHN) dataset. Several deep neural network architectures are studied. The best recognition results were obtained by a model that had been trained on a hybrid dataset of the SVHN images plus an additional 262,131 images of racing bib numbers, with a recall of 0.93, precision of 0.95, and F-measure of 0.94 on the RBNR Dataset and a recall of 0.98, precision of 0.98, and F-measure of 0.98 on the private dataset. Our results are much higher than those reported in related work and show that deep learning can effectively be used for racing bib number recognition. Also, the effectiveness of transfer learning for this problem is demonstrated.

**Keywords**

Racing Bib Number Recognition, Sports Analytics, Deep Learning, Convolutional Neural Networks, Multi-Digit Recognition, SVHN, Transfer Learning.

## Introduction

Every year, thousands of sport events take place across the globe. Most sport events, such as marathons, skiing, and cycling races hire professional photographers and provide a platform where the participants can access and purchase the resulting event photos. These platforms usually include a search functionality, where the participants can easily find their photos by entering their racing bib number (RBN)—the printed number which is usually pinned to the clothing—into a search field. For the search functionality to work, each image has to be tagged with the respective RBN. For a long time, the RBN tagging was done manually, but in recent years different techniques have emerged to perform this task automatically. These techniques stem from the vast research of text recognition in natural scene images. Although optical character recognition (OCR) existed for many years, it has proven to not be well suited for text and digits in natural scene images (Jaderberg et al. 2016; Ye and Doermann 2015). This has resulted in a large body of research on alternative methods for text and digit recognition in natural images (Chen and Yuille 2004; Epshtein et al. 2010; Jaderberg et al. 2016; Netzer et al. 2011; Wang et al. 2012).

The analysis of sports images with RBNs can be decomposed into two tasks: (1) RBN detection (also called RBN localization) that creates boundary boxes around detected RBNs, and (2) RBN recognition that translates the pictures in the boundary boxes into numbers. This paper focuses only on the latter task. When it comes to RBN detection and recognition there has been only a handful of papers published yet (these papers are described in detail in the related work section). Most of these papers focused on the detection of the RBNs, rather than the recognition. All prior papers that included the RBN recognition task applied OCR as the recognition engine at the end of the pipeline, despite the known fact that OCR is not well suited for natural scene images. As no previous research has performed RBN recognition using other methods than OCR, this study focuses on RBN recognition using convolutional neural networks (CNNs).

CNNs have proven successful in digit and text recognition on natural scene images in other related fields, such as images of house numbers and street signboards (Goodfellow et al. 2014; Jaderberg et al. 2016; Liao et al. 2017; Sermanet et al. 2012). It is therefore likely that using CNN for RBN recognition will yield good

results. Although Boonsim and Kanjaruek (2018) used a CNN for the localization of the racing bib numbers, until now, no one has used CNNs for the recognition of the RBNs. The main problem with training neural networks is that a substantial amount of labelled data is required, which might be the reason why this strategy has not yet been explored for RBN recognition. We approach this challenge with the following research question:

> *How well can neural networks be used for racing bib number recognition and to what extent can transfer learning be applied to help solve the data scarcity problem?*

By using a private and labelled dataset from FlashFrame.io, this paper demonstrates the success of RBN recognition using neural networks. Furthermore, this paper shows that RBN recognition can benefit from transfer learning (Weiss et al. 2016), by showcasing the performance of a network trained fully and partly on a completely different dataset, namely images of street view house numbers, by using the famous Street View House Numbers (SVHN) dataset (Netzer et al. 2011). All experiments are evaluated on the RBNR dataset created by Ben-Ami et al. (2012), as well as on a test set from the FlashFrame.io dataset for comparison; both are sport images with RBNs. The final results are compared with the previously published results on the RBNR dataset.

Our contributions to RBN recognition research are as follows:

- A comparison of various CNN architectures to perform RBN recognition.
- Showcase that RBN recognition can benefit from transfer learning, by demonstrating good recognition performance on the RBNR dataset from networks trained fully and partly on the SVHN dataset.
- Showcase that RBN recognition by CNN achieves better recognition results than previous methods have obtained on the RBNR dataset.

The remainder of this paper is structured as follows: The next section presents the related work in deep learning, digit recognition in natural scene images, and RBN recognition. We then describe the datasets, the network architectures, as well as the training and testing strategy. In the following section we discuss the results and assess the contribution by comparing it with prior work on RBN recognition. Finally, we conclude with a summary and suggestions for future research.

## Related Works

### *Deep Learning*

Deep learning differs from traditional machine learning methods, which require careful human selection of features, by the fact that deep learning models can automatically discover representations needed for detection or classification (LeCun et al. 2015). Therefore, deep learning is a kind of representation-learning method that learns various representations from data at different levels of abstraction (LeCun et al. 2015). The architecture of a deep learning model consists of an artificial neural network, where the layers are deeper than simpler neural networks, characterized by the many hidden layers. CNNs are a special kind of neural networks, mainly focused on computer vision and image processing tasks. CNNs use convolution kernels in at least one of the layers to learn translational invariant features (Goodfellow et al. 2016, p. 274). When applying CNN to image data, the spatial information of the pixels is leveraged to yield classification results (Bishop 2006, p. 267). CNNs have shown promising performance in various computer vision tasks, such as handwritten digit recognition (Jarrett et al. 2009).

### *Digit Recognition in Natural Scene Images*

Recognizing the racing bibs numbers in sport images can be related to many other works in the field of digit recognition, such as recognition of street house numbers and vehicle license plates. Netzer et al. (2011) presented a method of unsupervised feature learning to read house numbers from street level photos. They introduced a new benchmark dataset, called the Street View House Numbers (SVHN) dataset, containing over 200.000 labeled house number images, cropped from street view images. Sermanet et al. (2012) proposed a CNN to solve the task of house number recognition, and achieved an accuracy of 94.85%, which is an improvement of 4.25 points compared to the previous best accuracy of 90.6 % from Netzer et al. (2011) on the SVHN dataset. Goodfellow et al. (2014) presented a unified method for digit recognition, by

integrating the localization, segmentation, and recognition steps using a single deep convolutional neural network that operates directly on the image pixels. They argued that transcription of street house numbers can be seen as a sequence recognition task. Instead of training a neural network to classify individual digits, they generated an output of a full sequence of digits present in the street house number. Using this approach, they evaluated their accuracy on two levels; (1) on a per-digit accuracy to compare with previous methods, and (2) accuracy of predicting the whole number, as a ratio over all complete numbers in the dataset, in which both the length of the sequence as well as each element in the sequence have to be predicted correctly, without any 'partial credits' for predicting an individual digit correctly (Goodfellow et al. 2014, p. 3). Their method achieved a per-digit accuracy of 97.84%, which improved upon the previous method from Sermanet et al. (2012) by almost 3 percentage points. For recognizing complete street numbers (the whole sequence), they achieved an accuracy of 96%. Their best architecture had eleven hidden layers, consisting of eight convolutional layers, one locally connected layer, and two densely connected layers just before the output layer.

### *Racing Bib Number Recognition*

Ben-Ami et al. (2012) were the first to raise the issue of automatic racing bib number recognition in academia and also created a public dataset named the RBNR dataset. They proposed an RBN detection method based on face recognition, torso estimation and stroke width transform (SWT) (Epshtein et al. 2010) to generate RBN candidates. To filter these, they used stroke width in proportion to the size of the runner's face, as well as the size of the RBN tag candidates in relation to the torso bounding box and face scale. They then segmented the RBN candidates and used a Tesseract OCR engine (Smith 2007) to recognize each digit individually. After their publication, several other papers have been published on the topic. Some authors focused only on RBN detection, while others proposed an end-to-end system.

Roy et al. (2015) used a combination of face and skin detection as a more robust method, followed by a text detection method to locate text candidate regions and then processed these before applying Tesseract OCR. Shivakumara et al. (2017) proposed another multi-modal technique, utilizing various methods for torso detection and text detection, including text extraction by a histogram of oriented gradients and classification by a support vector machine. They binarized the detected bib numbers and applied Tesseract OCR. Wrońska et al. (2017) used an edge-based and color-based method to detect and filter RBN candidates. The RBN candidates were then binarized and recognized through Tesseract OCR.

Boonsim (2018) focused solely on RBN detection and proposed an edge-based method followed by various image processing steps, combined with face detection and torso extension to verify the true position of the RBN tag. De Jesus and Borges (2018) also focused solely on RBN detection and proposed an improvement of the original SWT algorithm for better text detection, by reducing the number of rays produced by the SWT stage. Boonsim and Kanjaruek (2018) suggested a region convolutional neural network for RBN localization. They first used a model which had been trained on the CIFAR-10 dataset (Krizhevsky 2009) and then continued training it on images containing RBNs, in order to detect whether a bib number was present in the image. Their network proved successful at locating RBNs, but it also yielded a lot of false positives. They only focused on detection, i.e. they did not perform recognition on the identified RBNs.

## Artefact Design

### *Datasets*

This paper investigates the recognition performance on the RBNR dataset by training various deep neural networks on different datasets. The RBNR dataset (Ben-Ami et al. 2012) is a public dataset of 217 running event images with 290 RBNs in these images. The bounding boxes and labels are for the whole RBN rather than each individual digit. Three previous publications have reported their RBN recognition results on the RBNR dataset (Ben-Ami et al. 2012; Roy et al. 2015; Shivakumara et al. 2017). In our study, the RBNR dataset is cropped according to the RBN bounding boxes and used only for testing. Two base datasets (FF-RBN and SVHN) as well as four combinations of parts of these two base datasets (Combo 1, Combo 2, Combo 3, Combo 4) are used for the training (see Table 1).

The FF-RBN dataset is a subset of a private dataset from FlashFrame.io, provided exclusively for this study, containing RBN images, labels, and bounding box information from various running events in the USA.

The FF-RBN dataset contains pseudo-labels, meaning that the images have been labelled through the algorithms of FlashFrame's software and not manually. FF-RBN has gone through extensive cleaning and manual checks. The bounding boxes and the labels are for the entire RBNs and not for each individual digit.

The SVHN dataset (Netzer et al. 2011) is a public dataset of cropped street view images of house numbers. The SVHN dataset will be used as the training set in many of the experiments in this paper, to evaluate the hypothesis that a model trained on house numbers should yield positive prediction results on RBNs, because of transfer learning (Weiss et al. 2016).

Finally, four combination datasets have been constructed, consisting of a mixture of the SVHN and FF-RBN datasets, to further explore the effects of transfer learning on RBN recognition by using both domain-specific and non-domain specific images. As labelled datasets for RBN recognition are scarce and expensive to create, this would help other practitioners in RBN recognition, by reducing the need of a large labelled RBN dataset. We want to find out how many additional images of RBNs are needed to obtain good RBN recognition performance. The datasets are: (1) *Combo 1* includes all SVHN images plus 10.000 images from FF-RBN, with a distribution of 2.000 images per RBN digit length of 1-5 digits, (2) *Combo 2* includes all SVHN images plus 15.000 FF-RBN images, of which 10.000 are images with 5 digits, and 5.000 are images of 1-4 digit length (1250 per digit length), (3) *Combo 3* consist of the entire SVHN dataset plus 50.000 images from FF-RBN, consisting of 10.000 images per digit length, and (4) *Combo 4* includes all SVHN images and all FF-RBN images to see if this composition yields maximum recognition results. The different datasets have been summarized in Table 1. A combination validation and test set have also been created.

| **Dataset** | **Image type** | **# Train images** | **# Validation images** | **# Test images** |
|---|---|---|---|---|
| RBNR *(only for testing)* | RBNs | NA | NA | 290 |
| FF-RBN | RBNs | 262,131 | 6,000 | 10,000 |
| SVHN | House numbers | 229,754 | 6,000 | 13,068 |
| Combo 1 (SVHN + FF-RBN) | Full SVHN + 10k FF-RBN | 239,754 | 6,500 | 14,068 |
| Combo 2 (SVHN + FF-RBN) | Full SVHN + 15k FF-RBN | 244,754 | 6,500 | 14,068 |
| Combo 3 (SVHN + FF-RBN) | Full SVHN + 50k FF-RBN | 279,754 | 6,500 | 14,068 |
| Combo 4 (SVHN + FF-RBN) | Full SVHN + Full FF-RBN | 491,885 | 6,500 | 14,068 |

**Table 1. Summary of datasets**

## *Network Architectures*

Due to the fact that the FF-RBN dataset has labels for the entire RBNs, and not per individual digit, a model that can learn the entire RBNs, i.e. multi-digit recognition, was a necessity in order to make use of this dataset for training. One architecture that can deal with this is the one proposed by Goodfellow et al. (2014). This is also the architecture that has obtained the best performance on the SVHN test dataset so far, making it the state-of-the-art architecture for multi-digit recognition in natural scene images. As this study investigates the recognition performance on the RBNR dataset of a network that has been trained solely and partly on the SVHN dataset, the network must perform well on the SVHN test data. Therefore, this study attempts to replicate the architecture by Goodfellow et al. (2014) with the objective of obtaining the same performance. Goodfellow et al. (2014) reported having experimented with multiple different architectures of different depth, thus the experiments in this study evolve less around the number of layers and units per layer but rather on the different parameters, such as dropout rate and the use of batch normalization. Information on these parameters is missing from their paper, so in order to obtain similar recognition results on the street number dataset, various versions of the main architecture had to be tested. See Table 2 below for the different variations of the architecture tested in this study.

| Model | Initial LR | Hidden Layers | Activation Function | Dropout | Batch normalization |
|---|---|---|---|---|---|
| Model 1 | 0.001 | 11 | 11 x ReLu | 0.15 on all hidden layers | Yes |
| Model 2 | 0.00001 | 11 | 11 x ReLu | 0.20 (Conv); 0,50 (rest) | Yes |
| Model 3 | 0.001 | 11 | 11 x ReLu | 0.25 on all hidden layers | Yes |
| Model 4 | 0.001 | 11 | 11 x ReLu | 0.50 on all hidden layers | Yes |
| Model 5 | 0.001 | 11 | 11 x ReLu | 0.10, 0.20, 0.30, 0.40 (Conv); 0.40 (rest) | No |
| Model 6 | 0.001 | 11 | 11 x ReLu | No Dropout | No |
| Model 7 | 0.001 | 11 | 1 x Maxout, 10 x ReLu | 0.20 (Conv), 0.40 (rest) | No |
| Model 8 | 0.001 | 11 | 1 x Maxout, 10 x ReLu | 0.25 (Conv), 0.40 (rest) | Yes |
| Model 9 | 0.001 | 11 | 1 x Maxout, 10 x ReLu | 0.25 on all hidden layers | Yes |
| Model 10 | 0.001 | 11 | 1 x Maxout, 10 x ReLu | 0.25 (Conv), 0.50 (rest) | Yes |
| Model 11 | 0.001 | 6 | 6 x ReLu | No Dropout | Yes |
| Model 12 | 0.001 | 6 | 6 x ReLu | No Dropout | Yes |

**Table 2. Model Parameters (LR = learning rate)**

The main architecture, as seen in Figure 1, contains eleven hidden layers, out of which eight are convolutional, one is a locally connected layer, and two are densely connected layers. The network accepts color images of 54x54 pixels. The architecture employs filters with size 48, 64, 128 and 160 for the first four layers, and 192 for all other locally connected layers. The fully connected layers contain 3072 units each. In some experiments all convolutional layers use ReLu as the activation function, while in other experiments the first convolutional layer contains an activation function called 'maxout unit', which takes the max of a set of inputs (Goodfellow et al. 2013). The latter version is the one that was reported most successful by Goodfellow et al. (2014). Each convolutional layer employs zero padding on the input to preserve representation size and each convolutional layer is followed by max pooling using a window size of 2 x 2 and 'same' padding. The stride alternates between 2 and 1 at each layer, which means that the spatial size of the representation is only reduced in half of the layers. The output consists of 6 SoftMax functions, where the first five are sequential digit classifiers with 11 classes (digits 0-9 plus '10' for 'no digit'), and the sixth output represents the predicted digit length in an image, with 6 classes (1-5 plus 'more than 5'). The applied loss function and optimizer are categorical cross entropy and Adam optimizer.

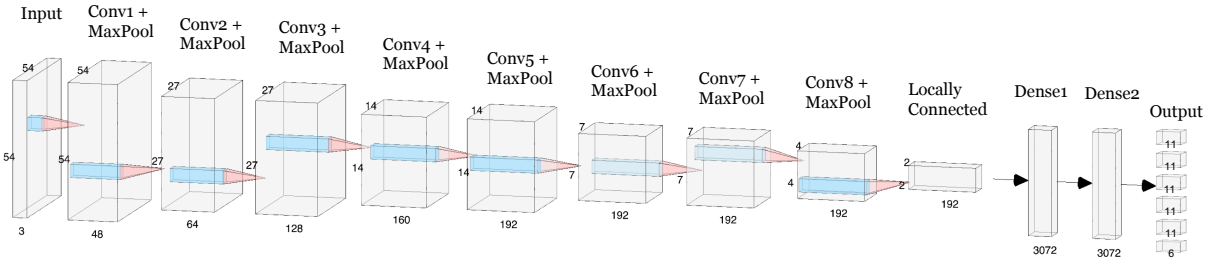


**Figure 1. Network Architecture (based on Goodfellow et al. (2014))**

Keras and TensorFlow were used for the implementation of the neural networks. The pre-defined bounding boxes were expanded by 8-30% (depending on the dataset), and then cropped according to the expanded bounding boxes, scaled and resized to 64x64 pixels. The SVHN images were expanded by 30%, but for RBN images 8% was used because otherwise the bounding boxes would contain noisy data such as other text and symbols usually present around the RBN. The labels were converted into a format of six-digit positions, where each number in the label would fill each position up to a total digit length of 5 digits, and the sixth digit position contained the total digit length. If a label is shorter than 5 digits, '10' is used to classify 'no digit' in the remaining positions. E.g. a label of "352" would be converted into '[3, 5, 2, 10, 10, 3]'. For

normalization, image mean subtraction (per color channel) was applied. The means were calculated based on all training images per dataset and subtracted from all images in the dataset.

## Experimental Results

The training proceeded in three steps. Firstly, 12 models (each with a slight variation of the main architecture of Goodfellow et al. (2014)) were trained on the SVHN Dataset. Secondly, the best performing architectures were trained from scratch on the FF-RBN Dataset. Finally, the best performing architectures were trained on the combination datasets. Real-time data augmentation was applied, by extracting a random patch of 54x54 pixels (from the 64x64 images) from each image, resulting in slight variations of each image per epoch. Initially, all models that used the SVHN train dataset were trained for 100 epochs, but all following experiments were trained for 50 epochs as it was deemed sufficient.

### *Models trained on the SVHN dataset*

In total, 12 models were trained on the SVHN dataset, of which two models did not converge at all, even after 100 epochs (probably because these two did not include batch normalization). For the remaining models, the accuracy rates on the SVHN test dataset varied between 75.5% - 95.2%. The best model (Model 1) achieved 95.2% accuracy on the SVHN test dataset, not far from the results from Goodfellow et al. (2014). This was however not the best performing model on the RBNR dataset, as that was Model 3, which applied a dropout rate of 25% on all hidden layers rather than 15%. Model 1 achieved 73.10% accuracy on the RBNR dataset while Model 3 obtained 74.8%, as seen in Table 3. Model 3 also performed better on the FF-RBN test dataset. All models performed better on the FF-RBN test data than the RBNR dataset, possibly implying that the RBNR dataset contain more complicated images that are more difficult to predict.

| **Model** | **Trained on** | **SVHN Accuracy** (%) | **FF-RBN Accuracy** (%) | **RBNR Accuracy** (%) |
|---|---|---|---|---|
| Model 1 | SVHN | 95.2 | 78.8 | 73.1 |
| Model 3 | SVHN | 95.1 | 81.4 | 74.8 |
| Model 8 | SVHN | 94.2 | 77.3 | 70.0 |

**Table 3. Performance of the top 3 models trained on the SVHN dataset**

All models trained only on the SVHN dataset performed particularly poor on RBNR images with five digits (as seen in Table 4 for Model 1, 3 and 8). Images of five digits had only 3-12% accuracy, compared to 93-100% accuracy on images with fewer digits. A probable reason for this is that the SVHN train dataset contains only 0.05% images with 5 digits. If the performance on images of five digits would have been better, then the total accuracy on the RBNR dataset would have been much higher, because ¼ of the images in the RBNR dataset contain 5 digits. Excluding images of five digits from the accuracy calculation yields a total weighted accuracy of 95.8% on the RBNR dataset for Model 3. This shows that images of house numbers and racing bibs are very similar and transfer learning is applicable for RBNR. However, one has to be careful about which dataset is used and that there is not such a class imbalance for images of certain digit length.

| **Digits in image** | **RBNR images** | **RBNR Accuracy** (%) | | |
|---|---|---|---|---|
| | | **Model 1** | **Model 3** | **Model 8** |
| 1 | 0 | NA | NA | NA |
| 2 | 3 | 100.0 | 100.0 | 100.0 |
| 3 | 46 | 95.7 | 97.8 | 89.1 |
| 4 | 168 | 95.8 | 95.2 | 93.5 |
| 5 | 73 | 5.5 | 12.3 | 2.7 |
| **Total** | **290** | **73.1** | **74.8** | **70.0** |

**Table 4. RBNR Accuracy per digit length of the top 3 models trained on the SVHN dataset**

### *Models Trained on the FF-RBN Dataset and the Combination Datasets*

In the previous section, Model 3 had the best performance on the RBNR Dataset, followed by Model 1 and 8. The next step was to train different neural networks on the FF-RBN Dataset based on these three models. These two models were also used for training on the combination datasets (Combo 1 - Combo 4). The results from these experiments are presented in Table 5.

| **Model** | **Dropout** | **Trained on** | **FF-RBN Accuracy** (%) | **Combo Accuracy** (%) | **RBNR Accuracy** (%) |
|---|---|---|---|---|---|
| Model 1 | 0.15 on all hidden layers | FF-RBN | 98.0 | 83.5 | 89.0 |
| Model 3 | 0.25 on all hidden layers | FF-RBN | 97.3 | NA | 84.8 |
| Model 8 | 0.25 (conv); 0.40 (rest) | FF-RBN | 97.5 | NA | 86.9 |
| Model 1 | 0.15 on all hidden layers | Combo 1 | 93.9 | 94.6 | 84.5 |
| | | Combo 2 | 96.0 | 94.9 | 89.0 |
| | | Combo 3 | 97.0 | 95.0 | 89.3 |
| | | Combo 4 | 97.4 | 94.9 | 91.7 |
| Model 3 | 0.25 on all hidden layers | Combo 1 | 94.9 | 94.9 | 85.2 |
| | | Combo 2 | 96.0 | 94.8 | 88.6 |
| | | Combo 3 | 97.1 | 94.9 | 91.7 |
| | | Combo 4 | 97.7 | 95.2 | 93.4 |
| Model 8 | 0.25 (conv); 0.40 (rest) | Combo 1 | 96.2 | 94.2 | 76.2 |
| | | Combo 2 | 95.1 | 93.8 | 85.5 |
| | | Combo 3 | 85.2 | 93.2 | 86.6 |
| | | Combo 4 | 97.4 | 94.2 | 90.3 |

**Table 5. Performance of the top 3 models trained on the FF-RBN & the Combo datasets**

The models trained on the FF-RBN dataset performed much better on the RBNR dataset than those trained on SVHN, with an improvement of over 10 percentage points. Model 8 trained on FF-RBN performed better on both the RBNR and the FF-RBN datasets than Model 3. For the models trained on the combination datasets, Model 3 trained on the Combo 4 dataset had the best recognition results for the RBNR dataset, with 93.4% accuracy. This dataset contains the whole of the SVHN and FF-RBN dataset. It had the best results on the RBNR dataset across all models that were trained in this experiment. Most of the models trained on the combination datasets resulted in better recognition performance on the RBNR dataset than the ones trained solely on FF-RBN. This might be because the SVHN dataset contains clearer images (less distortions than the FF-RBN dataset), causing the model to generalize better.

The models trained on the combination datasets show an interesting trend: the more FF-RBN images are added to the SVHN dataset, the better the recognition result. In fact, looking closer at the results of Model 3 across all experiments, we can conclude that when predicting RBNs using a model trained only on the SVHN dataset, an accuracy of around 75% can be expected; while training on the SVHN plus 10,000 RBN images yield around 85% accuracy; SVHN plus 15.000 RBN images yield almost 89%; SVHN plus 50,000 RBN images yield almost 92% and finally SVHN plus 260,000+ RBN images yield almost 94% accuracy on RBNs. One can notice the diminishing returns, and the jump from 50K images to 262K images is rather large. If the RBN images are hand-labelled, one might consider 91.7% accuracy to be sufficient, requiring only 50,000 additional images of RBNs in addition to the SVHN dataset.

Figure 2 shows the training and validation loss (left) and mean accuracy (right) for Model 3 trained on Combo 4. As one can see, the validation loss and accuracy curves become rather flat after about 20 epochs. None of the models showed any tendencies to overfitting. As the experiments were set to a fixed number of epochs, early stopping was not enabled, a common regularization technique (Goodfellow et al. 2016). Nevertheless, it was regarded as not necessary, as the validation loss either kept decreasing or became flat for all the experiments (i.e. it did not form a U-shaped curve). It is also worth mentioning again that dropout was applied to a majority of the models, which typically also helps to prevent overfitting.

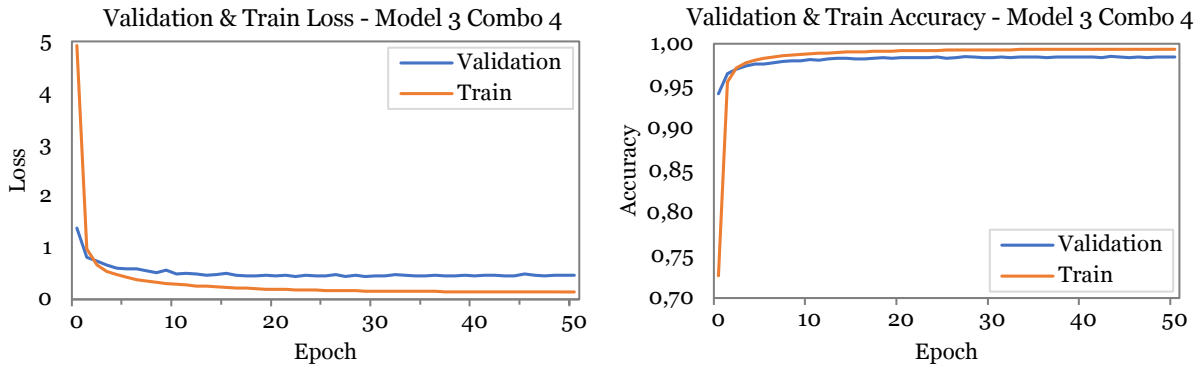


**Figure 2. Validation & Train loss/accuracy for Model 3 trained on Combo 4**

## *Comparison to Previous Results*

The previous section showed the results based on recognition accuracies. It is valuable to also report the results using the same measures as previously used for the RBNR Dataset. Ben-Ami et al. (2012) proposed to calculate precision, recall, and F-measure based on the whole bib number, giving no credits to partially correct RBNs. Table 6 and 7 show the results for precision, recall, and F-score calculated in the same manner as proposed by Ben-Ami et al. (2012). More specifically, Table 6 shows results on both the RBNR dataset as well as the FF-RBN test dataset for the top 3 models trained on the FF-RBN dataset and the four combination datasets. As one can see, Model 3 trained on Combo 4 (i.e. all SVHN images and all FF-RBN images) had the best recognition on the RBNR dataset. Out of the 290 total RBNs, 271 were recognized entirely correctly, 15 were recognized partly correctly and 4 RBNs had completely incorrect predictions.

| **Model** | **Trained on** | **FF-RBN (10.000 images)** | | | **RBNR (290 images)** | | |
|---|---|---|---|---|---|---|---|
| | | **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** |
| Model 1 | FF-RBN | 0.98 | 0.98 | 0.98 | 0.90 | 0.89 | 0.90 |
| | Combo 1 | 0.94 | 0.94 | 0.94 | 0.86 | 0.84 | 0.85 |
| | Combo 2 | 0.96 | 0.96 | 0.96 | 0.91 | 0.90 | 0.90 |
| | Combo 3 | 0.97 | 0.97 | 0.97 | 0.91 | 0.89 | 0.90 |
| | Combo 4 | 0.97 | 0.97 | 0.97 | 0.93 | 0.92 | 0.93 |
| Model 3 | FF-RBN | 0.97 | 0.97 | 0.97 | 0.87 | 0.85 | 0.86 |
| | Combo 1 | 0.95 | 0.95 | 0.95 | 0.87 | 0.86 | 0.86 |
| | Combo 2 | 0.96 | 0.96 | 0.96 | 0.90 | 0.89 | 0.89 |
| | Combo 3 | 0.97 | 0.97 | 0.97 | 0.93 | 0.92 | 0.93 |
| | Combo 4 | 0.98 | 0.98 | 0.98 | 0.95 | 0.93 | 0.94 |
| Model 8 | FF-RBN | 0.97 | 0.97 | 0.97 | 0.89 | 0.87 | 0.88 |
| | Combo 1 | 0.85 | 0.85 | 0.85 | 0.78 | 0.76 | 0.77 |
| | Combo 2 | 0.95 | 0.95 | 0.95 | 0.87 | 0.86 | 0.86 |
| | Combo 3 | 0.96 | 0.96 | 0.96 | 0.88 | 0.87 | 0.88 |
| | Combo 4 | 0.97 | 0.97 | 0.97 | 0.92 | 0.90 | 0.91 |

**Table 6. Precision, Recall and F-score on the FF-RBN and RBNR datasets**

The best model has been presented in Table 7 along with the results of previous publications on the RBNR dataset. As seen in Table 7, the proposed method outperforms all prior RBN recognition techniques applied on the RBNR dataset. All prior approaches for RBN recognition applied OCR for the recognition step. The results show that a deep CNN can yield much better recognition results than OCR on RBN images. The reason why the recognition results for Ben-Ami et al. (2012) are taken from Shivakumara et al. (2017) is because the reported recognition results in the paper of Ben-Ami et al. most certainly include the detection step (which would make the results less comparable).

| Method | RBNR (290 images) | | |
|---|---|---|---|
| | **Precision** | **Recall** | **F-score** |
| *Proposed Method (Model 3, Combo 4)* | *0.95* | *0.93* | *0.94* |
| Ben-Ami et al. (2012) (as reported by Shivakumara et al. 2017, p. 490) | 0.39 | 0.69 | 0.50 |
| Shivakumara et al. (2017, p. 490) | 0.41 | 0.72 | 0.52 |
| Roy et al. (2015, p. 492) | 0.77 | 0.55 | 0.64 |

**Table 7. Comparison to previous RBN recognition results on the RBNR Dataset**

## Conclusion, Limitations, and Further Research

The research question for this study was "How well can neural networks be used for racing bib number recognition and to what extent can transfer learning be applied to help solve the data scarcity problem?" The best model achieved 93.4% accuracy on the RBNR dataset, with precision, recall and f-score of 0.95, 0.93 and 0.94. This outperforms the previously best F-score of 0.64 from Roy et al. (2015) and proves that neural networks can indeed be used successfully for RBN recognition. The best model was trained on all SVHN plus all FF-RBN images. The second-best model, with precision, recall, and f-score of 0.93, 0.92 and 0.93, was trained on the combination dataset of all SVHN images and only 50,000 FF-RBN images (Combo 3). Just like the best performing model, the second best model outperformed all models that had been trained solely on SVHN or FF-RBN images. This shows that transfer learning helps to improve the RBN recognition results and proves that a very large labelled RBN dataset is not required to get satisfying results.

In this paper, we only looked at one way to solve RBN recognition: by deep convolutional neural networks based on the works of Goodfellow et al. (2014). However, there are other ways neural networks could be used to solve this problem. Future research could investigate the performance of a different neural network type and architecture, such as using a convolutional recurrent neural network (CRNN) (Shi et al. 2017), a fully-convolutional regression network (FCRN) (Gupta et al. 2016) or a LSTM-based network (He et al. 2016). Future research could also investigate the use of dictionary-based classification (i.e. a list of possible bib numbers) (Jaderberg et al. 2016; Wang et al. 2012). The idea of training on different datasets could also be investigated further, for example by using a synthetic dataset (Gupta et al. 2016). Finally, future research should try to connect RBN recognition with RBN detection (localizing the bibs) by one neural network, in order to create an end-to-end neural-networks-based RBN detection and recognition engine.

## REFERENCES

Ben-Ami, I., Basha, T., and Avidan, S. 2012. *Racing Bib Number Recognition*, presented at the British Machine Vision Conference 2012, 19.1-19.10. (https://doi.org/10.5244/C.26.19).

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*, Springer Science+Business Media, LLC: Springer.

Boonsim, N. 2018. "Racing Bib Number Localization on Complex Backgrounds," *WSEAS Transactions on Systems and Control* (13), pp. 226–231.

Boonsim, N., and Kanjaruek, S. 2018. *Racing Bib Number Localization Based on Region Convolutional Neural Networks*, presented at the The 8th International Workshop on Computer Science and Engineering (WCSE 2018).

Chen, X., and Yuille, A. L. 2004. *Detecting and Reading Text in Natural Scenes*, in (Vol. 2), presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), IEEE.

Epshtein, B., Ofek, E., and Wexler, Y. 2010. *Detecting Text in Natural Scenes with Stroke Width Transform*, presented at the 2010 IEEE Conference, IEEE, pp. 2963–2970.

Goodfellow, I. J., Bengio, Y., and Courville, A. 2016. *Deep Learning*, Cambridge: MIT Press.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. 2014. "Multi-Digit Number Recognition from Street View Imagery Using Deep Convolutional Neural Networks," *International Conference on Learning Representations (ICLR)*.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. 2013. *Maxout Networks*, in (Vol. 28), presented at the Proceedings of the -th International Conference on International Conference on Machine Learning (ICML'13), pp. 1319–1327.

Gupta, A., Vedaldi, A., and Zisserman, A. 2016. *Synthetic Data for Text Localisation in Natural Images*, presented at the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324. (https://doi.org/10.1109/CVPR.2016.254).

He, P., Huang, W., Qiao, Y., Loy, C. C., and Tang, X. 2016. *Reading Scene Text in Deep Convolutional Sequences*, presented at the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, USA.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. 2016. "Reading Text in the Wild with Convolutional Neural Networks," *Int J Comput Vis (IJCV)* (116), pp. 1–20. (https://doi.org/10.1007/s11263-015-0823-z).

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. 2009. "What Is the Best Multi-Stage Architecture for Object Recognition?," in *2009 IEEE 12th International Conference on Computer Vision*, , October 29, pp. 2146–2153. (https://doi.org/10.1109/ICCV.2009.5459469).

de Jesus, W. M., and Borges, D. L. 2018. *An Improved Stroke Width Transform to Detect Race Bib Numbers*, in (Vol. 10880), J. Martínez-Trinidad, J. Carrasco-Ochoa, J. Olvera-López, and S. Sarkar (eds.), presented at the Pattern Recognition. MCPR 2018, Springer, Cham. (https://doi.org/10.1007/978-3-319-92198-3_27).

Krizhevsky, A. 2009. "Learning Multiple Layers of Features from Tiny Images," Technical Report, University of Toronto.

LeCun, Y., Bengio, Y., and Hinton, G. 2015. "Deep Learning," *Nature* (521), pp. 436–444. (https://doi.org/10.1038/nature14539).

Liao, M., Shi, B., Bai, X., Wang, X., and Liu, W. 2017. *TextBoxes: A Fast Text Detector with a Single Deep Neural Network*, presented at the Thirty-First AAAI Conference on Artificial Intelligence, AAAI, pp. 4161–4167.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. 2011. *Reading Digits in Natural Images with Unsupervised Feature Learning*, in (Vol. 2), presented at the NIPS workshop on deep learning and unsupervised feature learning.

Roy, S., Shivakumara, P., Mondal, P., Raghavendra, R., Pal, U., and Lu, T. 2015. *A New Multi-Modal Technique for Bib Number/Text Detection in Natural Images*, in (Vol. 9314), presented at the Advances in Multimedia Information Processing - PCM 2015, Springer, Cham, pp. 483–494. (https://doi.org/10.1007/978-3-319-24075-6_47).

Sermanet, P., Chintala, S., and LeCun, Y. 2012. *Convolutional Neural Networks Applied to House Numbers Digit Classification*, presented at the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, pp. 3288–3291.

Shi, B., Bai, X., and Yao, C. 2017. "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (39:11), pp. 2298–2304. (https://doi.org/10.1109/tpami.2016.2646371).

Shivakumara, P., Raghavendra, R., Qin, L., Raja, K. B., Lu, T., and Pal, U. 2017. "A New Multi-Modal Approach to Bib Number/Text Detection and Recognition in Marathon Images," *Pattern Recognition* (61), (Y. S. Ho, J. Sang, Y. Ro, J. Kim, and F. Wu, eds.), pp. 479–491. (https://doi.org/10.1016/j.patcog.2016.08.021).

Smith, R. 2007. *An Overview of the Tesseract OCR Engine*, in (Vol. 2), presented at the ICDAR 2007, IEEE.

Wang, T., Wu, D. J., Coates, A., and Ng, A. Y. 2012. *End-to-End Text Recognition with Convolutional Neural Networks*, presented at the 2012 21st International Conference, IEEE, pp. 3304–3308.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. D. 2016. "A Survey of Transfer Learning," *Journal of Big Data* (3:9), pp. 2–40. (https://doi.org/10.1186/s40537-016-0043-6).

Wrońska, A., Sarnacki, K., and Saeed, K. 2017. *Athlete Number Detection on the Basis of Their Face Images*, presented at the 2017 International Conference on Biometrics and Kansei Engineering (ICBAKE), IEEE, pp. 84–89. (https://doi.org/10.1109/ICBAKE.2017.8090642).

Ye, Q., and Doermann, D. 2015. "Text Detection and Recognition in Imagery: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (37:7), pp. 1480–1500. (https://doi.org/10.1109/TPAMI.2014.2366765).