Barcode Segmentation using Fully Convolutional Neural Networks

Ricardo Santos Pereira
Mestrado Integrado em Engenharia Informática e Computação
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
up201503716@fe.up.pt

***Abstract*—This paper describes the implementation of a machine learning model applied to the problem of segmentation of barcodes in static images and presents the results for several variations of the solution.**

I. Introduction

The problem of efficiently detecting 1D barcodes in images is not recent, and several solutions have been devised over time, mostly relying on image filtering and segmentation algorithms, which display some issues when presented with situations where the primary factors that allow identification and subsequent selection of the relevant region [in an image] deviate from the limits established in preconfigured parameters. Namely, usual approaches make use of Hough transforms and apply other various image processing methods [1] [2] in an attempt to extract the barcodes solely by its characteristics. While relatively successful, these operations are computationally expensive and may fail to detect barcodes in situations of varying degrees of visibility of the barcodes. We intend to present a solution that is robust against these issues, based on machine learning methods, which have proven time and again their capability in similar problems, such as in [3], [4] and [5]. In section II we present the proposed solution, as well as the steps to build it and obtain results from it. In section III, we present the results achieved after several testing sessions, afterwards extracting insight from these. Finally, in section IV, we briefly discuss the developed work and pose our afterthoughts on the general problem and the devised solution.

II. Proposal

The solution proposed by this paper utilizes a fully convolutional neural network in order to achieve the segmentation of barcode regions, by producing a mask that highlights the region belonging to the barcode, or possibly several regions for multiple barcodes that may be visible in an image. The goal is that subsequent processing of the image should allow to easily separate and isolate these regions and perform orientation corrections in each of them, through image masking and affine transformations. The resulting regions should then be ready for application of a simpler model or algorithm in order to extract the information (decode) from each barcode.

A. Data preparation

The training was done using the Muenster BarcodeDB [6] and the ArTe-Lab 1D Medium Barcode [7] datasets, using 595 images from the former and 365 images from the latter, which were joined and comprised the base dataset, totalling 960 annotated images. The annotations consist of binary images, where the values in the regions with barcodes in the corresponding image have the value ”one” and the remaining regions have the value ”zero”, which were used as the ground truth images to train the model. The images and corresponding annotations had varied sizes, and were normalized to the dimensions of 256×256 before further processing.

In each session, we submitted the base dataset through a process of data augmentation, comprising several transformations of the base images:

- Image translation, bounded to 25% of the image's size on each axis;
- Image rotation;
- Image scaling, bounded to within 25% of the image's original size;
- Shear transformation;
- Channel value shifting;

These were selected randomly, with possibility of zero to multiple transformations being applied on randomly sampled images of the base dataset, which were then added to an augmented dataset. This dataset comprised both the base dataset and the augmented images. The augmentation process only terminated once the augmented dataset reached a predefined number of images. Finally, the augmented dataset was permutated several times, the number of which was proportional to its size. The resulting dataset was the one effectively used for the training of the model.

B. Model architecture

The model that we built is a variation of a fully convolutional architecture, more specifically an architecture commonly known as U-Net [5], comprising convolution layers followed by transposed convolution (”deconvolution”) layers. The convolution layers used kernels of size 4×4, strides of 2 pixels and filters ranging from 64 to 512 on the lowermost layers. The activation function used in these layers was the Leaky Rectified Linear Unit (Leaky ReLU) [8]. Additional batch

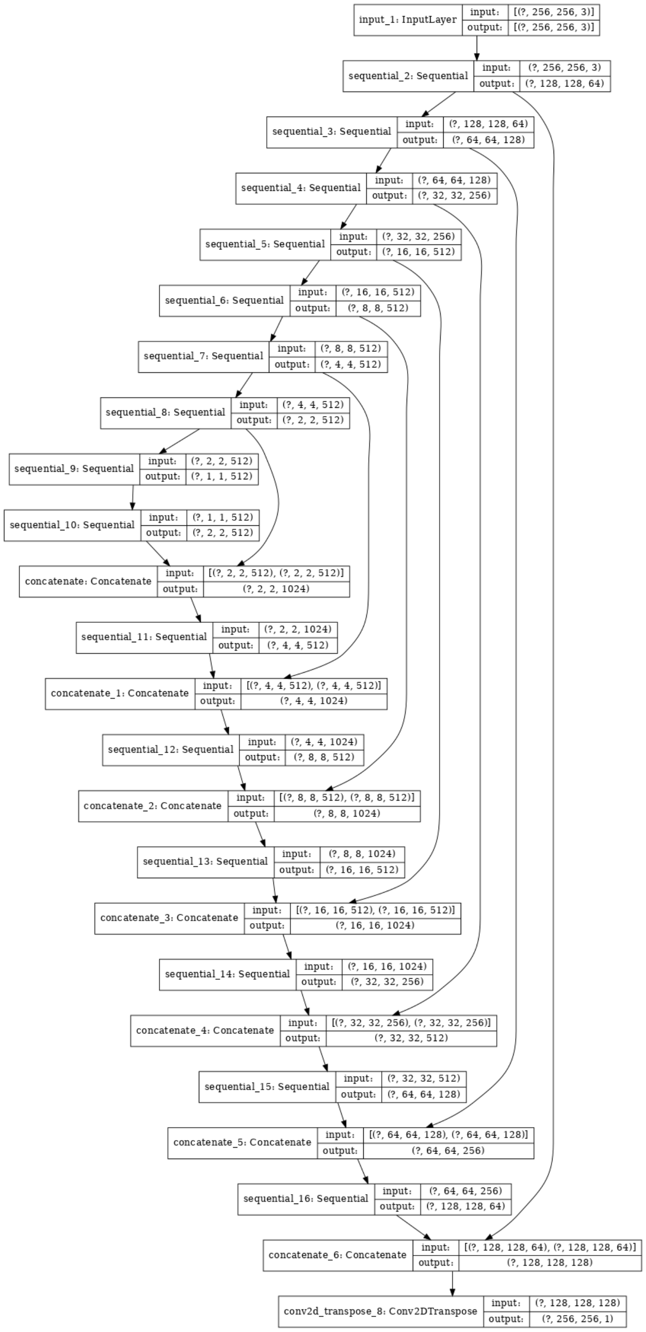

Fig. 1. The architecture of the network.

normalization was applied in the first of these layers. The following transposed convolution layers had similar characteristics to the convolution layers, with exception of the activation function, which was the Rectified Linear Unit (ReLU) [9]. The first three of these layers also had a dropout factor of 0.5. The filters of all the layers were initialized with values sampled from a normal distribution of mean equal to 0.0 and standard deviation equal to 0.02. The output of each convolution layer was fed to the input of a corresponding transposed convolution layer, such that the first convolution layer connected to the last transposed convolution layer. The network required input of size $256 \times 256 \times 3$ (three channels). The output layer had size of $256 \times 256 \times 1$ and used a sigmoid activation function. The architecture can be seen in figure 1.

C. *Loss functions*

The model was trained using different loss functions, in an attempt to see which one produced better results. Because this problem could essentially be construed as a problem of classifying whether each pixel of the image belonged to a barcode (class 1) or not (class 2), one of the functions used was (binary) cross entropy. The other functions used were the mean squared error and mean absolute error.

III. Results

As explained in the previous section, the model was trained with several different loss functions. Additionally, we also trained each one of those instances for different epoch durations. In each run, the dataset consisted of 2500 images after augmentation, with a split ratio of 0.9 between train and test data, and a batch size of 50 images. We used a stochastic gradient descent method called Adam [10] for optimization with a fixed learning rate of 0.001. For each of the loss functions, we trained the model for a duration of 20, 40, and 60 epochs.

The figures 2, 3 and 4 show the values of train sessions for different loss functions and number of epochs. For all figures, the The horizontal axis represents the number of the batch across all epochs for that session and the vertical axis represents the loss value; the blue lines correspond to the values for the train data and the orange lines correspond to the values for the test data. The values in the vertical axis are represented on a logarithmic scale.

Fig. 2. Results obtained with binary cross entropy as the loss function.

After training, the models were evaluated for accuracy. Unfortunately, due to the nature of the problem, this evaluation was made with a rather imprecise and naive method: applying the trained models to a small set of test images and see which one produced the results that ”looked the best”, that is, which one seemingly approximated the barcode region more exactly when compared side-by-side with the ground truth images.

Fig. 3. Results obtained with mean squared error as the loss function.

Fig. 4. Results obtained with mean absolute error as the loss function.

Despite this, analysis of the values of the loss functions registered during training allows us to extrapolate some information regarding the methods used: as seen in the graphs, the model converges to a low value irrespective of method used, which indicates learning capability. Furthermore, the model also keeps some generalization capabilities, which we can infer by the observation that the difference of values between the training and the testing data is relatively low.

IV. Conclusion

After careful observation of the obtained results, our final assessment is that the utilization of machine learning methods is a viable and potentially more efficient solution for this class of problems comparatively to more traditional approaches. As future work, we would like to couple the application of this trained model with a separate method for the decoding of the barcodes and present that as a possible solution for systems where this functionality is required.

References

[1] Muñiz, Rubén & Junco, Luis & Otero, Adolfo. (1999). A robust software barcode reader using the Hough transform. 313-319. 10.1109/ICIIS.1999.810282.

[2] Namane, Abderrahmane & Arezki, Madjid. (2017). Fast Real Time 1D Barcode Detection From Webcam Images Using the Bars Detection Method.

[3] Hansen, Daniel & Nasrollahi, Kamal & Rasmusen, Christoffer & Moeslund, Thomas. (2017). Real-Time Barcode Detection and Classification using Deep Learning. 321-327. 10.5220/0006508203210327.

[4] Long, Jonathan & Shelhamer, Evan & Darrell, Trevor. (2014). Fully Convolutional Networks for Semantic Segmentation. Arxiv. 79.

[5] Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.

[6] Wachenfeld, S. & S. Terlunen, S. & Jiang, X. (2010). Robust 1-D Barcode Recognition on Camera Phones and Mobile Product Information Display. In Mobile Multimedia Processing Fundamentals, Methods, and Applications, LNCS, vol. 5960, pp. 53-69.

[7] Zamberletti, Alessandro & Gallo, Ignazio & Albertini, Simone. (2013). Robust Angle Invariant 1D Barcode Detection.

[8] Maas, Andrew L. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models.

[9] Hahnloser, R. & Sarpeshkar, R. & Mahowald, M. A. & Douglas, R. J. & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit.

[10] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.