

Optical Character Recognition Using Convolutional Deep Learning Algorithm

S. Siva Nagendra, K. VenkataRamana

M. Tech Student, Assistant Professor
Department of Computer Science and Systems Engineering,
Andhra University, Visakhapatnam, India.

Abstract: Now a days, character recognition is most difficult task in computer vision. Optical Character Recognition (OCR) is technique in which a system can automatically extract the text from images or scanned documents into editable format. In this paper we proposed convolutional deepnet based algorithm for OCR. This proposed algorithm extracting the text from bounding boxes, which are generated from convolutional Artificial Neural Network(ANN). This algorithm was implemented for handled devices using server-based processing. The performance of proposed algorithm uses less space in the handled devices and recognition of text from the images takes less time.

IndexTerms – Deep learning, Convolutional network, bounding box, object detection .

I. INTRODUCTION

In the last two decades' collection of information and dissemination of the information over the electronic media has represented in many forms such as textual, graphical and scanned documents. While most of the graphical and scanned documents can be stored with a decipherable textual component. Optical Character Recognition (OCR) is the one of the most important process to identify typed, handwritten or printed textual characters from the scanned document. The text attained by the OCR process often suffers from low accuracy owing to irregularities in images, poor scans or simply the nature of arrangement of letters in a word. For example, reading "lwo" instead of "two", "ia" instead of "is", "m" instead of "rn", to name a few. These erroneous characters severely hamper the quality and readability of a converted document. Identifying and rectifying these erroneous characters in every OCR-processed document is complex task for the sheer volume of data. Deep learning techniques [1] are used automatic text to solve this kind of complex tasks. In this paper we adopted You Only Look Once(YOLO) based object detection approach [2] to achieve OCR.

The basic idea of YOLO approach is A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes in an input image and the object is detected from each bounding box as shown in Figure 1.

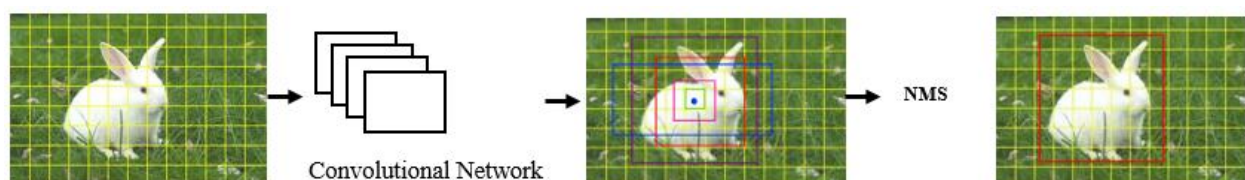


Figure 1 (a) Original Image Grid

(b) Bounding Boxes detection

(c) Précised Bounding Box

Image is embedded into our deepnet produce a grid. Each cell of grid generates bounding boxes. Centroid of bounding box ascertain the object and reframe this object into simple regression task straight from image pixels to bounding box coordinates and class probabilities. All existing OCR systems are incapable of recognize characters efficiently from image. In this paper we proposed an approach that recognize the characters in image using convolutional deepnet, which is inspired by YOLO network [8], but YOLO some objects like characters are not detected precisely because of lower number of featured maps. In our network

input of image of size 416 x 416, and replaced some internal convolutional layers for increasing the image resolution and instead of fully connected layer of YOLO network we inserted two 1024 convolutional layers as shown in Figure 2. These two convolutional layers extracts the exact features of object.

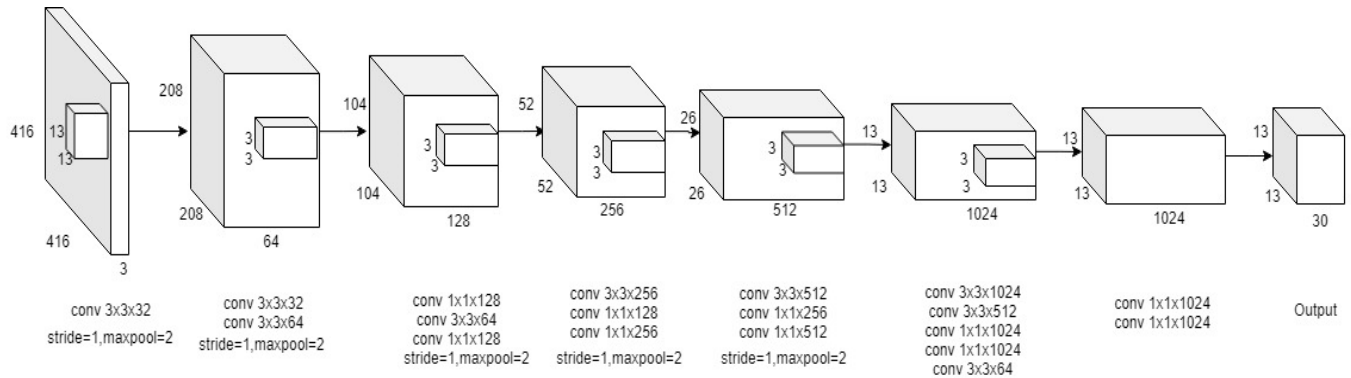


Figure 2. Architecture of our Proposed convolutional deepnet

II. RELATED WORK

Scanning the image pixel by pixel then extract data from them takes more time. Many researchers proposed many methods for text spotting. Such methods are SSD[4] (Single Shot multi box Detector), SSSTR[5], textbox++[3]. In The work of Jaderberg et al. [4] scene text segmentation was performed by a text proposals mechanism that was later refined by a CNN that regressed the correct position of bounding boxes. Afterwards, those bounding boxes were inputted to a CNN that classified them in terms of a predefined vocabulary. Gupta et al. [9] followed a similar strategy by first using a Fully Convolutional Regression Network for detection and the same classification network than Jaderberg for recognition. Liao et al. [3] used a modified version of the SSD [4] object detection architecture adapted to text and then a CRNN [6] for text recognition. However, breaking the problem into two separate and independent steps presented an important drawback since detection errors might significantly hinder the further recognition step. L. Gomez, et al. [5] read the image pixel by pixel by dividing problem into two parts as text detection and text recognition. This model is lagging at some situations especially in text detection. Liao M. et al. [3] proposed some new method multi scale oriented text recognition, for detecting text in horizontal and vertical screening format. This method is lagging at object occlusion and large character spacing. is failure at curved text detection because of its low level of quad lateral representation. The core problem of computer vision is object detection. Detecting the small objects in large space is more difficult task, for that YOLO[2][8], Faster RCNN[10] methods used. YOLO [2][8] struggles to generalize to objects in new or unusual aspect ratios or configurations.

III. DEEPNET BASED CHARACTER RECOGNITION FRAME WORK

In this framework consists two phases, first phase performs image acquisition and Location segmentation where as in second phase processing of OCR is accomplished. The architecture of our prosed system as shown in Figure 3. First we detect the input image contain blurriness or not. The detection of blurred images are as follows.

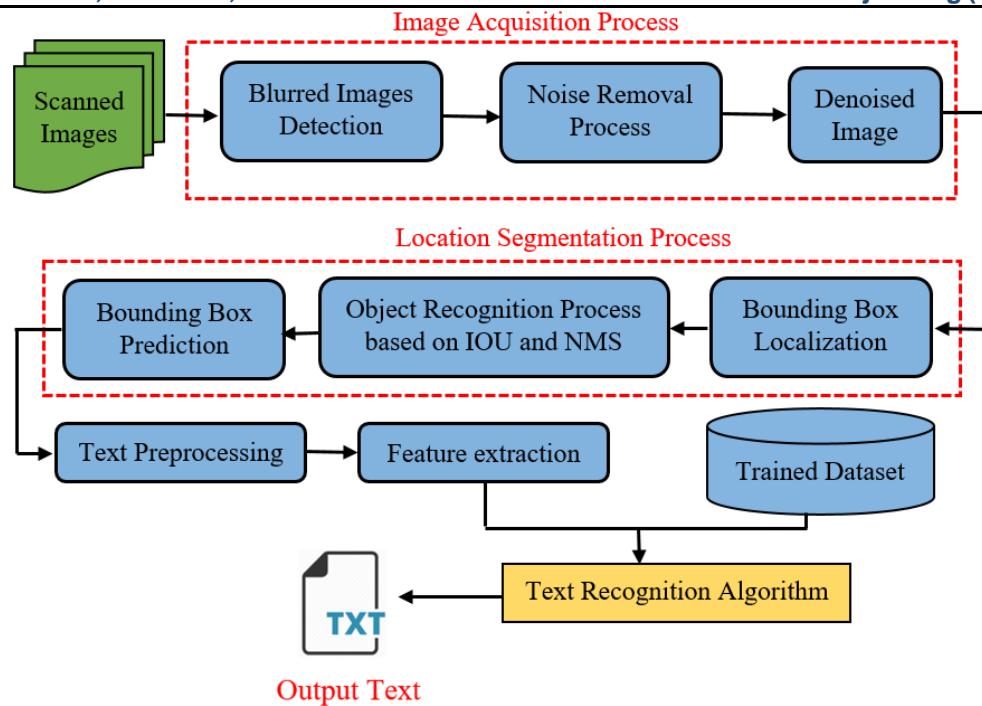


Figure 3. Design of OCR system

Image Blur Detection: Generally, sharp images have high frequency components and blurred images have mostly low-frequency components. The distribution of frequency components in an images are computed by using Gaussian blur filters, it uses the Gaussian function. If the distribution component is low amount of high frequencies, then the image is considering as blur image, otherwise sharp image. The Gaussian function of two dimensional image is $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$ here, x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. The values produced from this distribution are used to construct convolution matrix, which is applied to the original image. The generation of each new pixel value is set to a weighted average of that pixel's neighborhood. The original pixel value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries. After identify the blur in an image the noise has to be removed by using non-local means de-noise technique. In this technique, replace the color pixel value with an average of color pixel values of nearby pixels. The variance law in probability theory ensures that if nine pixels are averaged, the noise standard deviation of the average is divided by three. In pixel wise implementation each pixel value is restored as an average of the most resembling pixels, where this resemblance is computed in the color image. So for each pixel, each channel value is the result of the average of the same pixels and resulting de-noised image has formed.

Bounding Box localization: In the localization, features of an object detected by convolutional feature extractor and based on these features forming the coordinates of the bounding boxes. Predicting offsets instead of coordinates simplifies the problem and makes it easier for the system to learn.

The working of convolutional feature extractor (CFE) based on Artificial Neural Network (ANN) as shown in the Figure 2. In CFE contains total 23 layers in which 18 are convolution layers and 5 are pooling layers. Convolution layers' extraction the features of object and pooling layer resizing image, for acquiring the high resolution some fully connected layers or pooling layers are eliminated. Once the features are

extracted, initialize the bounding boxes based on the different aspect ratios of an image. The initial bounding boxes are composed by the parameters, top left corner and bottom right corner of an object as follow.

$$\begin{aligned} x_1 &= (gb.x - gb.w/2) * w \\ y_1 &= (gb.y - gb.h/2) * h \\ x_2 &= (gb.x + gb.w/2) * w \\ y_2 &= (gb.y + gb.h/2) * h \end{aligned}$$

Here $gb.h$, $gb.w$ are ground truth box height and width values respectively and w , h are original image width and height respectively. In our work we consider five different scaling values of image. After bounding box localization, to generate all possible bounding boxes for each object in an image. The localization loss measures the errors in the predicted boundary box locations and sizes. We only count the box responsible for detecting the object.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

Where $\mathbb{1}_{ij}^{obj} = 1$ if the j^{th} boundary box in cell i is responsible for detecting the object, otherwise 0. λ_{coord} increase the weight for the loss in the boundary box coordinates. Once the location of the bounding box predicted, the object content in bounding box can be computed by the confidence loss.

Confidence loss: If an object is detected in the box, the confidence loss (measuring the objectness of the box) is:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(C_i - \hat{C}_i)^2]$$

Where \hat{C}_i is the box confidence score of the box j in cell i .

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

If the object is not detected in the bounding box, the confidence loss is

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

Where $\mathbb{1}_{ij}^{noobj}$ is the complement of $\mathbb{1}_{ij}^{obj}$

\hat{C}_i is the box confidence score of the box j in cell i

λ_{noobj} weights down the loss when detecting background.

Now, classification loss can be calculated such that predicted object belongs to which class. It can be determined by squared error of conditional class probabilities of each class.

Classification loss: If an object is detected, the classification loss at each cell is the squared error of the class conditional probabilities for each class:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

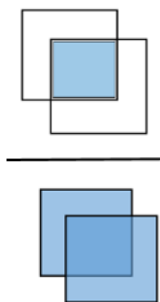
Where $\mathbb{1}_i^{obj} = 1$ if an object appears in cell i , otherwise 0, $\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i .

Total Loss= Localization loss + confidence loss + Classification loss

However, the prediction of correct bounding box of an object based on the two metrics Intersection Over Union (IOU) and Non- Maximum Suppression (NMS).

Bounding Box Prediction: Now the output will be grid of size of an image $S \times S$ and each cell in grid produces B bounding boxes with C confidence score for N number of classes. The tensor should be size of $S \times S \times (B \times 5 + N)$.

By IOU value, we eliminate all bounding boxes which are doesn't have object. Intersection over Union is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. i.e, how accurate the predicted bounding box is overlapped with the ground truth bounding box. This can be determined by

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Now we get bounding boxes contains the objects. To find exact bounding box which contains the object (two or more bounding boxes contains the same object with 25%, 50%, 75%). these two lowest bounding boxes are pruned by non-maximum suppression (NMS). The probability of predicted bounding box falls below nms value that box is discarded. And returns the output as x, y, w, h, c and probability of each class.

Text Recognition process: Finally, we obtained bounding box coordinates and cropping each box from given image. Each box is segmented by foreground and background to remove logo like symbols in the background. Now the normalization process transform text into pronounceable form, size normalization is used to adjust the character size to certain standard. The character in each box of segmented image divided into set of zones and every zones are scaled and sent into the feature extractor. In the extraction phase, statistical features rely on the computation of curvature, slope, end-points, axes ratio, and the length variations of strokes [7]. We use the Robert's operator, in which extracted 68 gradient features from a normalized image for the classification. Now train the SVM classifier with predefined label set contains 8 different size and rotated instances of each of 26 lower case, 26 upper case, 10 digits and all ASCII alpha numerical symbols. If the Curvature, width and height of character in label set is equals to the character recognized from the bounding box segment, the recognized character is opted out from the segment. All recognized characters are grouped up then output will be printed.

IV Algorithm

The algorithm runs in three steps. During the first step initializing the threshold values for the given input image and validating the given image (ImgeGrid) has Blur or not using the GaussianFilter as shown from line 1 to line 5 in algorithm. If the image is blurred image, then remove the noise by using NonLocalMeanDenoise function show in line 6 to 8 in algorithm. In the second step identifying the

bounding boxes from given image grid using the metrics IOU and NMS that shown in line 9 to line 16. In the third steps of this algorithm extract text from bounding boxes using SVM classification show in function SVMTextRec shown in line 17.

```

1  Begin
2      GThreshold = 0.7
3      IOUThreshold = 0.75
4      NMSValue = 0.3
5      Blur ← GaussianFilter(ImageGrid)
6      If (Blur > Threshold)
7          ImgeGrid ← NonLocalMeanDenoise(ImgeGrid)
8      End if
9      GtBox = [x, y, w, h]
10     OutBox[] =  $\Phi$ 
11     BoundingBox ← GenerateBBoxes(ImgeGrid)
12     For each BoundingBox in ImgeGrid do
13         PBox ← ObjRec(BoundingBox)
14         BoxIOU = PBox/GtBox
15         If ((BoxIOU > IOUThreshold) && (NMS(PBox) > NMSValue))
16             OutBox ← PBox
17             SVMTextRec (OutBox)
18     End for
19     Return Text
20 End

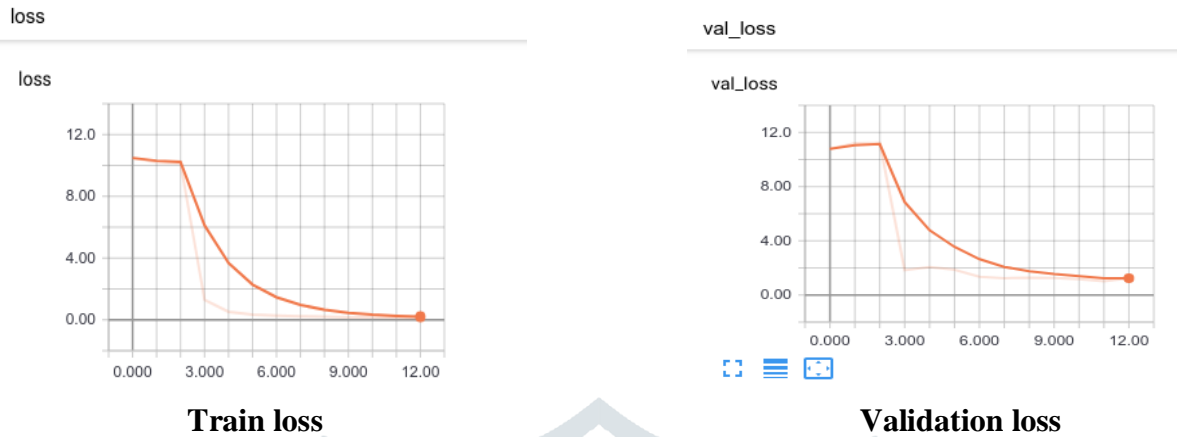
```

IV. RESULTS AND DISCUSSION

The experiment result shows different standard bench mark for text based retrievals. These images contain both color and grayscale images. First categorize the type of ID card (aadhar, pan card, voter_id, passport, driving license and other_id) by custom data dataset of size 15,000 images. Then we generated synthetic dataset of size 1000 images, post to that all objects in the image are annotated manually with five classes. Annotated image with ground truth bounding box coordinates are sent into the network for training.

The experiments were conducted on TitanX i7 processor with NVidia GeForce GTX 1060TX .GPU. For efficient text retrieval purpose we trained the deep net for 70 hours. After training we saved all good weights. In testing phase, first we recognize the objects in document by passing an ID image to the neural network. Once objects are recognized these objects with predicted bounding box coordinates are sent into combination of SVM classifier and our own label set it will recognize the text in object, then finally recognized text will printed.

While training we measured the train loss and validation loss for how exactly the model is recognizing the objects. Our model lowest train and validation losses as 0.06 and 0.08 respectively. Below images shows the loss curve.



Performance of our Object recognition is measured by mean average precision (mAP) which is average of all precisions of correctly recognized objects in image. Our model produced mAP of 60.8.

$$\text{Mean Average Precision} = \frac{1}{n} \left(\frac{TP}{TP+FP} \right)$$

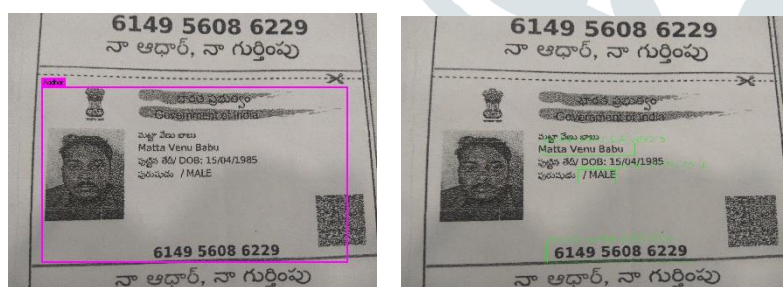
Performance of text recognition model is measured by text accuracy of

$$\text{Text accuracy} = \frac{\text{number of correctly recognized characters}}{\text{Total number of classes}}$$

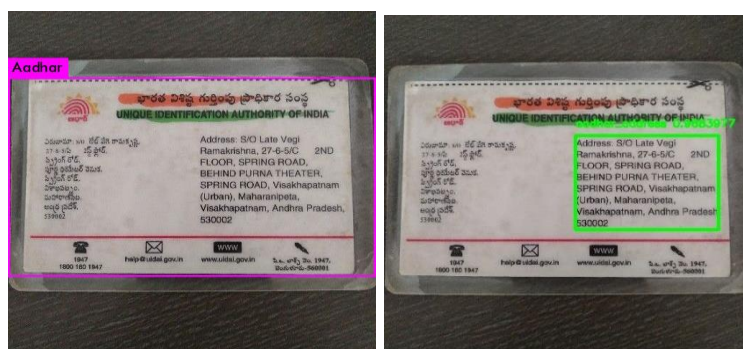
Accuracy of our text recognizer is 96%.

Tested results of aadhar cards are as shown below

Aadhar_front and Aadhar_back



```
aadhar_name: Matta Venu Babu
aadhar_gender: | Male
aadhar_dob: 15/04/1985
aadhar_number: 6149 5608 6229
aadhar_address:
```



```
aadhar_name:
aadhar_gender:
aadhar_dob:
aadhar_number:
aadhar_address: Address: S/O Late Vegi
Ramakishna, 27-6-5/C 2ND
FLOOR, SPRING ROAD,
BEHIND PURNA THEATER,
SPRING ROAD, Visakhapatnam
(Urban), Maharanipeta,
Visakhapatnam, Andhra Pradesh,
530002,
```

REFERENCES

- [1] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998. Gradient-based learning applied to document recognition. In Proc of IEEE Conference Vol. 86, pp. 2278–2324.
- [2] Redmon, J., Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger. In Proc of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA. pp. 6517 - 6525.
- [3] Liao, M., Shi, B., Bai, X. 2018. Textboxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing, Vol. 27(8), pp. 3676 – 3690.
- [4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. 2016. SSD: Single shot multibox detector. In Proc. of the European Conference on Computer Vision, Lecture Notes in Computer Science book series LNCS, Springer, Vol. 9905, pp. 21-37.
- [5] L. Gomez, A. Maa, M. Rusinol and D. Karatzas. 2018. SSSTR: Single Shot Scene Text Retrieval. In Proc. of European Conference on Computer Vision (ECCV), LNCS, Springer, Vol. 11218, pp. 728 – 744.
- [6] B. Shi, X. Bai, and C. Yao. 2017. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, In IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 39(11), pp. 2298–2304.
- [7] Ray Smith. 2007, An Overview of the Tesseract OCR Engine, In Proc. 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Parana, Brazil.
- [8] Redmon, J., Santhosh, D., Ross Girshick, Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 779 – 788.
- [9] Gupta, A., Vedaldi, A., Zisserman, A. 2016. Synthetic data for text localisation in natural images. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315 - 2324.
- [10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proc. Of the IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 39(6), pp. 1137 - 1149.

