# Custom OCR for Identity Documents:OCRXNet

**Kawal Arora[1], Ankur Singh Bist[2], Roshan Prakash[3],Saksham Chaurasia[4]**
Signy Advanced Technologies, India
Address : Level 39, One Canada Square, Canary Wharf, London E14 5AB
e-mail: kawal@signy.io[1], ankur@signy.io[2], roshan@signy.io[3], saksham@signy.io[4]

(APA style, Justify, Arial 10pt) Example:
To cite this document:
Arora, K., Bist, AS., Prakash, R., & Chaurasia, S. (2020). Custom OCR for Identity Documents:OCRXNet. Aptisi Transactions On Technopreneurship (ATT), 2(2), 111-119.

**DOI : https://doi.org/10.34306/att.v2i2.87**

**Author Notification**
08 June 2020
**Final Revised**
13 June 2020
**Published**
16 June 2020

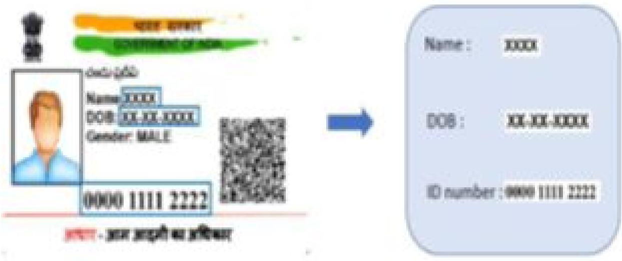
### *Abstract*

*Recent advancements in the area of Optical Character Recognition (OCR) using deep learning techniques made it possible to use for real world applications with good accuracy. In this paper we present a system named as OCRXNet. OCRXNetv1, OCRXNetv2 and OCRXNetv3 are proposed and compared on different identity documents. Image processing methods and various text detectors have been used to identify best fitted process for custom ocr of identity documents. We also introduced the end to end pipeline to implement OCR for various use cases.*

*Keywords: Text Detector, Tesseract, Yolo, CRAFT, Noise Removal.*

## 1. Introduction

OCR is one of the most important issues in the domain of computer vision. Different techniques have been proposed in past literature but with the advancement of deep learning techniques, now it's possible to use OCR capabilities for real world environment. If we want to use OCR for real world applications then accuracy is very important concern.



Picture 1: Basic OCR process

KYC is best example for this where we want to onboard verified user. In india, for kyc user has to provide passport, pan card etc. as identity document. Now there are different ways to validate user using identity documents. Manual process is time taking so organizations are moving towards AI and blockchain based methods to enhance user experience. Generally different companies are providing mobile app for on boarding process where user can upload identity document [1]. Within the app, deep learning based ocr technique is used to extract important information from document. Now the accuracy of text extraction is important because if there exist option for editing then it's not possible completely to validate the authenticity. To avoid this specific issue we developed ocr pipeline where we can extract and auto-fill the desired information from identity documents. We focused on three major techniques i.e. object detection, text detection and basic image processing to enhance the accuracy for text extraction from identity documents. Simplest way to achieve the better results includes image processing techniques like noise removal, Dilation/ erosion, rotation/Deskewing etc. Second approach used by us is object detection, by using the same we can locate the regions and then extract information from those regions. Third important method is text detection where we can design bounding boxes in input image where text is present and then extraction procedure will work.

An overview of the rest of the paper is as follows: in section 2 we present related work in this area; section 3 defines proposed model architecture used; section 4 defines results and discussions. Finally in section 5 we present conclusion and future work.

## 2. Related Work

Different open source OCR engines are available. OCRopy, OCRopus 3, Tesseract, Kraken, Ocropy2 and Calamari are openly available ocr engines. Further details can be taken from following links.
https://github.com/tmbdev/ocropy
http://kraken.re/
https://github.com/tmbdev/ocropy2
https://github.com/NVlabs/ocropus3
https://github.com/tesseract-ocr/tesseract
https://github.com/Calamari-OCR/calamari

Different companies are providing OCR service by giving their API that can be used in different platforms. Behavior of openly available ocr engines vary under different use case. As per our analysis on identity documents we found that tesseract is working well. There are different versions of tesseract, latest release is Tesseract 4.1.1.2. There are another openly available networks that can be used for OCR like https://github.com/da03/Attention-OCR. Use of networks like LSTM and its variants can be used widely to attain good results for customized requirements [2,3]. Authors in paper [4] developed deep learning architecture for OCR of Telugu language. Authors in paper[5] Convolutional neural network encoder approach for lexicon free ocr. Authors in paper [6] used inception-v3 model for ocr. Authors in paper [7] used LSTM networks text recognition in Devanagari. Authors in paper[8] used deep Convolutional neural network for identification of kannada characters. Authors in paper[9] used deep Autoencoder and Convolutional neural network for urdu character recognition. Authors in paper [10] used deep Convolutional neural network for ocr of Latin documents. Authors in paper [11] used gated recurrent Convolutional neural network for ocr. Authors in paper [12] used bi-directional LSTM for ocr. Authors in paper [13] used LSTM-RNN for ocr. Authors in paper [14] used MDLSTM for Pashto cursive script and works under different scale and rotation. Authors in paper [15] used hybrid Convolutional lstm for text recognition. Authors used BLSTM and MDLSTM for text recognition in Indian context.

## 3. Proposed Works

There are three different approaches used OCRXNetV1, OCRXNetV2 and OCRxNetV3. First approach involves use of image processing techniques with Tesseract i.e. OCRXNetV1.

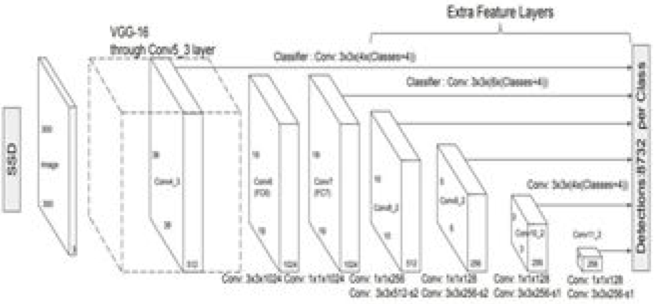
We used three image processing methods, Finding Contour (RoI) & Cropping, Adaptive Thresholding and Canny Edge Detection with gaussian Blur. There are various other image processing techniques available but for our use case, we found mentioned techniques are performing well. Contour is the process of connecting same intensity or color points. We used opencv for our task, for more details refer: https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html. Adaptive thresholding is the technique for identifying threshold of smaller regions that can further be utilized for character recognition. For more details refer: https://opencv-pythontutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html.CannyEdge Detection with gaussian Blur involves five steps.

1. Noise removal using gaussian filter
2. Intensity gradients estimation
3. Non-maximum suppression
4. Use of double threshold for identifying accurate edges
5. Track edges

In OCRXNetV2, for extracting information from different identity documents such as ADHAAR, PAN, Passport, etc., we broke down the problem into four different stages:

1. Collecting & creating a dataset of different IDs from different sources
2. Identifying regions of interest from the acquired images and making bounding boxes around the areas of text
3. Training our deep learning algorithm to identify such regions
4. Performing OCR on the identified region of interests

Different IDs such as adhaar, PAN, passport, etc is collected from different sources and a proper dataset is created. In order to create dataset for our deep learning algorithm we created bounding boxes around our areas of interests (text) using LabelImg. The dataset created in such a way that the name of annotation files being created is the same as the image files, and the annotation files are in a different folder as to the image files and in the same sequence arranged as the image files.



Picture 2: SSD architecture [16]

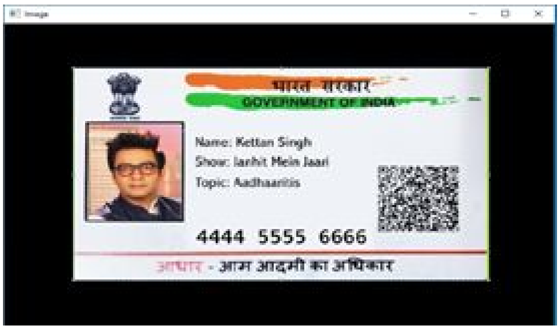
Annotation file contains multiple annotations' information. An annotation indicates where an object in the image is located along with its size like (x_top, y_top, x_bottom, y_bottom, width, height). Picture 2 shows the SSD architecture used for generating bounding box.

In OCRXNetV3, we used text detector CRAFT i.e. character region awareness for text detection. Purpose of this text is to select regions where text is present and then perform extraction using Tesseract. This approach is better than OCRXNetV2 because for SSD, we have to prepare large dataset for training.

## 4. Experimental Analysis

p-ISSN: 2655-8807
e-ISSN: 2656-8888

To implement OCRXNetV1, we prepared dataset of identity documents. For present work we are focusing on indian identity documents (Passport, Pan card and Aadhar card). Here we are taking example of aadhar card. Picture 3 is the input image that needs to be processed so for transforming it into proper input format we are using image processing techniques.

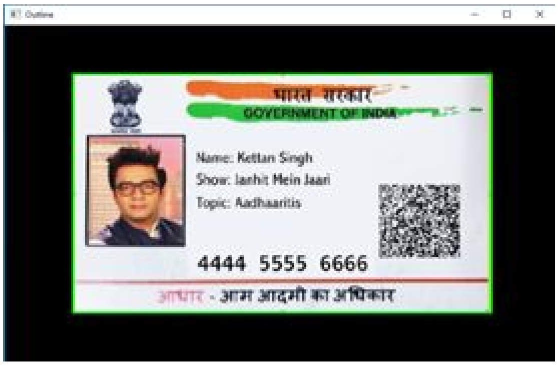


भारत सरकार
GOVERNMENT OF INDIA
Name: Kettan Singh
Show: Janhit Mein Jaari
Topic: Aadhaaritis
4444 5555 6666
आधार - आम आदमी का अधिकार

Picture 3: Input Aadhar card Image

As mentioned in proposed work section initially we are finding contour to extract aadhar card region after that adaptive thresholding or Canny Edge Detection with gaussian Blur can be used. Picture 4 shows cropped region and Picture 5 and Picture 6 shows the output of image pre-processing.

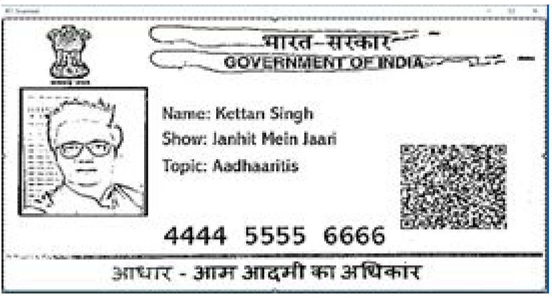
***Finding Contour (RoI) & Cropping***



भारत सरकार
GOVERNMENT OF INDIA
Name: Kettan Singh
Show: Janhit Mein Jaari
Topic: Aadhaaritis
4444 5555 6666
आधार - आम आदमी का अधिकार

Picture 4: Input image of identity document (Aadhar Card)

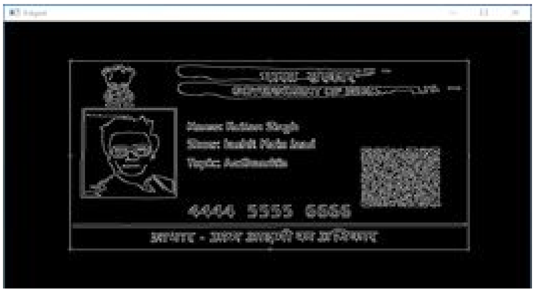
**Adaptive Thresholding**



भारत-सरकार
GOVERNMENT OF INDIA
Name: Kettan Singh
Show: Janhit Mein Jaari
Topic: Aadhaaritis
4444 5555 6666
आधार - आम आदमी का अधिकार

Picture 5: Adaptive Thresholding

**Canny Edge Detection with gaussian Blur**



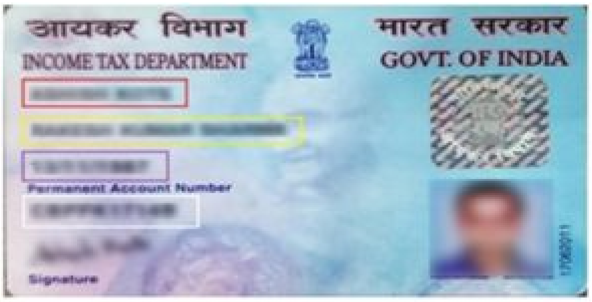
Picture 6: Canny Edge Detection with Gaussian Blur

After pre-processing we are using tesseract to extract text. On the top of that we have code structure to map the extracted details.

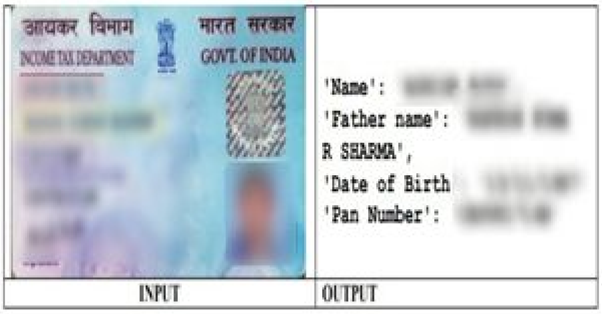
There various algorithms for object detection like R-CNN, SPP, Fast R-CNN, Faster R-CNN, Feature Pyramid networks, RetinaNet (Focal loss), Yolo Framework — Yolo1, Yolo2, Yolo3, SSD etc. After testing different variants of object detection we found that yolov3 and SSD are appropriate for current use case. Three scale detection the important feature of Yolov3 by down sampling at 32, 16 and 8 respectively. We selected SSD for our use case. SSD requires input image with ground truth. The outcome is bounding boxes on specified positions like name, pan number etc.

**SSD object detection**

To implement OCRXNetV2, we used object detection technique i.e. SSD. For using it, we have to annotate the samples and based on annotation training will processed. SSD was found accurate for our use case. Picture 7 shows output of SSD algorithm and Picture 8 is the extracted details using tesseract.



Picture 7: Output of SSD



```
'Name': [redacted]
'Father name': [redacted]
R SHARMA',
'Date of Birth [redacted]
'Pan Number': [redacted]
```

Picture 8: Text extraction

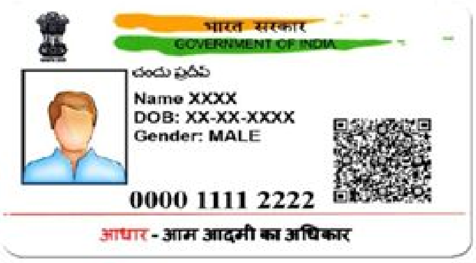
Text detection algorithms are getting matured by the use of deep learning. There are various challenges in the process of text detection which includes image noise, viewing angles, blurring effects, Lighting conditions, resolution, non-paper objects, non-planar objects unknown layout etc. Literature of text detector algorithms is growing; we took

p-ISSN: 2655-8807
e-ISSN: 2656-8888

various algorithms for analysis. Finally we selected east text detector and CRAFT text detector for pipeline of OCR for identity documents.

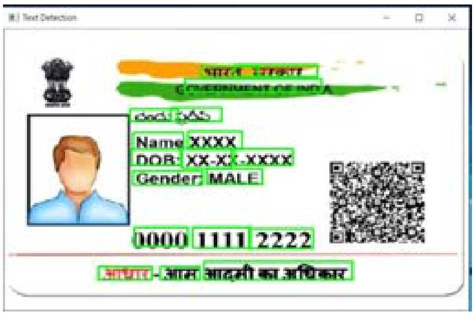
**Text Detector**

East text detector [17] is efficient and accurate scene text detector. Picture 9 and Picture 11 is the input for text detector algorithm and the output can be seen in Picture 10 and Picture 12.



भारत सरकार
GOVERNMENT OF INDIA
ఆంధ్ర ప్రదేశ్
Name XXXX
DOB: XX-XX-XXXX
Gender: MALE
0000 1111 2222
आधार - आम आदमी का अधिकार

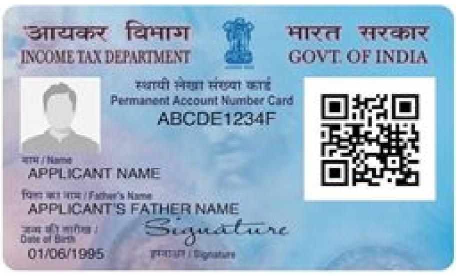
Picture 9: Input Aadhar image



Text Detection

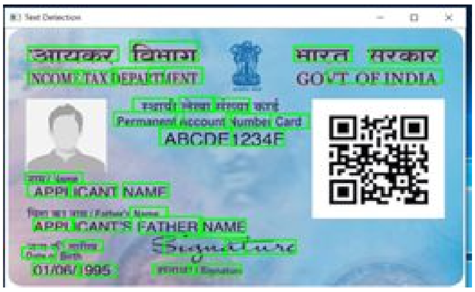
भारत सरकार
GOVERNMENT OF INDIA
ఆంధ్ర ప్రదేశ్
Name XXXX
DOB: XX-XX-XXXX
Gender: MALE
0000 1111 2222
आधार - आम आदमी का अधिकार

Picture 10: Text detector output



आयकर विभाग
INCOME TAX DEPARTMENT
भारत सरकार
GOVT. OF INDIA
स्थायी लेखा संख्या कार्ड
Permanent Account Number Card
ABCDE1234F
नाम / Name
APPLICANT NAME
पिता का नाम / Father's Name
APPLICANT'S FATHER NAME
जन्म की तारीख / Date of Birth
01/06/1995
Signature
हस्ताक्षर / Signature

Picture 11: Text detector input

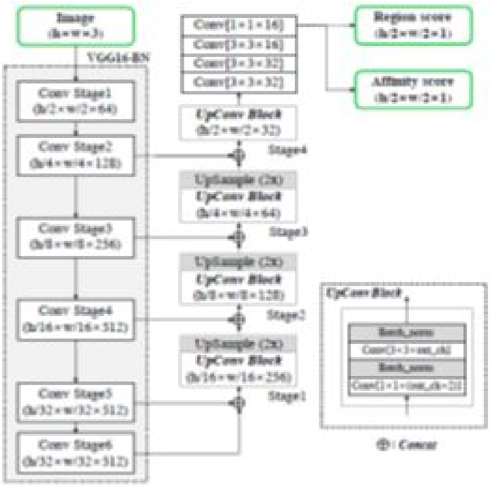


Text Detection

आयकर विभाग
INCOME TAX DEPARTMENT
भारत सरकार
GOVT. OF INDIA
स्थायी लेखा संख्या कार्ड
Permanent Account Number Card
ABCDE1234F
नाम / Name
APPLICANT NAME
पिता का नाम / Father's Name
APPLICANT'S FATHER NAME
जन्म की तारीख / Date of Birth
01/06/1995
Signature
हस्ताक्षर / Signature

Picture 12: Text detector output

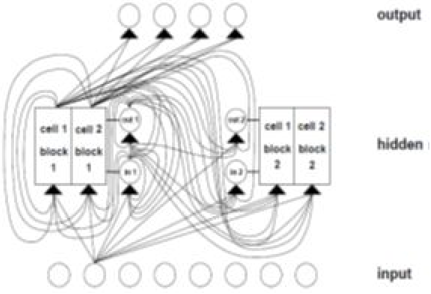
Results of East text detector are not good in many cases so we moved towards more stable algorithm that is CRAFT text detector. Picture 13 shows the detailed architecture of CRAFT algorithm. After testing it on various input set we found it suitable for current use case.



Picture 13: CRAFT text detector [18]

**Tesseract**

Tesseract was originally developed at Hewlett-Packard Laboratories Bristol and at Hewlett-Packard Co, Greeley Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++izing in 1998 [19]. In 2005 Tesseract was open sourced by HP. Since 2006 it is developed by Google. For more details please refer: https://github.com/tesseract-ocr/tesseract. Tesseract latest form is available with LSTM. Basic network for these task are recurrent neural network but there is issue of vanishing gradient and exploding gradient. To solve this issue LSTM are used. Basic intuition can be taken from following Picture .



Picture 14: Basic net of LSTM [19]

## 5. Conclusion and Future Work

In this paper we proposed OCRXNetV1, OCRXNetV2 and OCRxNetV3 which can effectively perform the task of character recognition for identity documents. Use of image processing techniques, objection detection algorithms (SSD) and text detection procedures (CRAFT) produced a pipeline for real world application. In future, we will create datasets by collecting more samples of identity documents in different conditions for training different models in pipeline. Deep learning architectures are evolving with very fast pace, that will be helpful for designing robust system. In future, we will develop deep network for our use case. To extend proposed ocr versions for different use cases is our second priority. Current work will be very useful for industrial or academic purpose.

**Acknowledgement**

This project is fully funded by Signy Advanced Technologies, Level 39 One Canada Square, Canary Wharf, London E14 5AB.

**References**

[1] https://signy.io/#/, Last visited: 29 January, 2020.
[2] Brzeski, Adam, et al. "Evaluating performance and accuracy improvements for attention-OCR." IFIP International Conference on Computer Information Systems and Industrial Management. Springer, Cham, 2019.
[3] Saluja, Rohit, et al. "OCR On-the-Go: Robust End-to-end Systems for Reading License Plates & Street Signs." 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). 2019.
[4] Achanta, Rakesh, and Trevor Hastie. "Telugu OCR framework using deep learning." arXiv preprint arXiv:1509.05962 (2015).
[5] Namysl, Marcin, and Iuliu Konya. "Efficient, lexicon-free OCR using deep learning." arXiv preprint arXiv:1906.01969 (2019).
[6] Wei, Tan Chiang, U. U. Sheikh, and Ab Al-Hadi Ab Rahman. "Improved optical character recognition with deep neural network." 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA). IEEE, 2018.
[7] Kundaikar, Teja, and Jyoti D. Pawar. "Multi-font Devanagari Text Recognition Using LSTM Neural Networks." First International Conference on Sustainable Technologies for Computational Intelligence. Springer, Singapore, 2020.
[8] Chandrakala, H. T., and G. Thippeswamy. "Deep Convolutional Neural Networks for Recognition of Historical Handwritten Kannada Characters." Frontiers in Intelligent Computing: Theory and Applications. Springer, Singapore, 2020. 69-77.
[9] Ali, Hazrat, et al. "Pioneer dataset and automatic recognition of Urdu handwritten characters using a deep autoencoder and convolutional neural network." SN Applied Sciences 2.2 (2020): 152.
[10] Springmann, Uwe, et al. "OCR of historical printings of Latin texts: problems, prospects, progress." Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage. 2014.
[11] Wang, Jianfeng, and Xiaolin Hu. "Gated recurrent convolution neural network for ocr." Advances in Neural Information Processing Systems. 2017.
[12] Ahmed, Saad Bin, et al. "Deep learning based isolated Arabic scene character recognition." 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR). IEEE, 2017.
[13] Naseer, Asma, and Kashif Zafar. "Meta features-based scale invariant OCR decision making using LSTM-RNN." Computational and Mathematical Organization Theory 25.2 (2019): 165-183.
[14] Maalej, Rania, and Monji Kherallah. "Improving MDLSTM for offline Arabic handwriting recognition using dropout at different positions." International conference on artificial neural networks. Springer, Cham, 2016.
[15] Breuel, Thomas M. "High performance text recognition using a hybrid convolutional-lstm implementation." 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 1. IEEE, 2017.
[16] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.
[17] Zhou, Xinyu, et al. "EAST: an efficient and accurate scene text detector." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017.
[18] Baek, Youngmin, et al. "Character region awareness for text detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.
[19] Kettunen, Kimmo, and Mika Koistinen. "Open Source Tesseract in Re-OCR of Finnish Fraktur from 19th and Early 20th Century Newspapers and Journals-Collected Notes on Quality Improvement." DHN. 2019.