Layout Error Correction using Deep Neural Networks

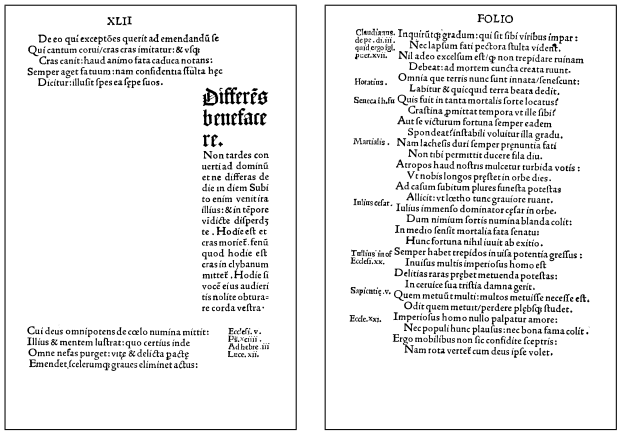
Srie Raam Mohan, Syed Saqib Bukhari, Prof. Andreas Dengel
German Research Center for Artificial Intelligence (DFKI), Germany
University of Kaiserslautern, Germany
{srieraammohan}@gmail.com{saqib.bukhari, andreas.dengel}@dfki.de

***Abstract*—Layout Analysis, mainly including binarization and page/line segmentation, is one of the most important performance determining steps of an OCR system for complex medieval historical document images, which contain noise, distortions and irregular layouts. In this paper, we present a novel layout error correction technique which include a VGG Net to classify non-textline and adversarial network approach to obtain the layout bounding mask. The presented layout error correction technique are applied to a collection of 15th century Latin documents, which achieved more than 75% accuracy for segmentation techniques.**

I. Introduction

Digital data has been growing exponentially over the recent past years. Over 90% of all the data that has been ever produced by the human has been done in the past two years. These data include but not limited to historical documents, printed books, handwritten documents and so on. Even though the popularity of PCs, Laptops and Tablets take over the creation of most of the documents digitally, the amount of information that has been held on paper has not been diminished. Information contained on paper is still considered as one of the most preferred media. Document Image Analysis and Recognition (DIAR) aims at extracting the information from non-digital media. The result of DIAR systems will be the digital media that is obtained from non-digital media to make it available for digital functions. The difficulty of DIAR lies in the fact that most of the non-digital media are error prone, noise contaminated and have nonuniform document structure. These difficulties make the systems complex to extract the information accurately. DIAR pipeline involves several tasks namely pre-processing, Layout Analysis, Character Segmentation and Recognition, Optical Character Recognition(OCR), Signature Verification etc.[3].

Text and non-text segmentation is an important layout analysis step, which may directly affect the performance of further layout processing tasks such as textline extraction, and/or character recognition. Text-line extraction is the backbone of a layout analysis system. Kumar et al. [7] have evaluated the performance of six algorithms for page segmentation on Nastaliq script: the x-y cut [8], the smearing [9], whitespace analysis [10], the constrained text-line finding [11], Docstrum [12], and the Voronoi-diagram based approach [13]. These algorithms work very well in segmenting documents in Latin script as shown in [14]. However, none of these algorithms were able to achieve an accuracy of more than 70% on their test data which had simple book layouts. More sophisticated approaches for text-line extraction have been presented in the domain of segmenting handwritten European document so far. However, the key problem addressed in these approaches is to handle local non-linearity of text-lines.



(a) (b)

Fig. 1. Sample Images.

Traditional approaches for layout analysis are Top-down and Bottom-up approaches. Connected component is a Bottom-up approach, where group of pixels are grouped into components based on their pixel connectivity. In the Top-down approach the entire document is scanned and further split into smaller entities. Traditional methods like histograms or Hough Transforms suffer difficulties with datasets having complex background or very high variation in the background. Neural Network techniques such as Multi-directional LSTM, RNN and CNN has been used to achieve better accuracy in Layout analysis [4], [5], [6].

This paper addresses the problem of layout analysis in machine-printed 15th century Latin European Novel document images (Figure 1). Neural Networks has been widely practiced in the DIAR community to achieve great feats of accuracy to convert the documents digitally. A general introduction to the application of Neural Networks can be seen in lot of books and survey papers(eg., [1],[2]). We propose a new way of obtaining the layout boundary mask by using a two model architecture of neural networks trained separately. In section II of this paper we will discuss the two model architecture we have used for

obtaining the layout boundary mask. In Section III we will discuss the performance evaluation and the difficulties of this approach. In the final section we will discuss the results and discuss a short outlook on the improvements that can be made.

II. Methodology

The approach we used to obtain a layout boundary mask is achieved by three steps. The first step is to train an VGG model[15] with textline and non-textline data to classify them. The second step is to train a Pix2Pix model[16] with documents and the corresponding layout boundary mask. The third step is to pass on the data obtained from the text line segmentation method from anyOCR system [20] (which is based on OCRopus [17]) to the first model to classify it as textline or non-textline. If it is classified as non-textline it is then passed on to the second model to obtain the layout mask. The architecture and the dataset preparation is discussed briefly in the following sections.

A. Model I - VGG Net

CNN is used for classifying the textline and non-textline. The CNN model used is VGG Net[15]. VGGNet is trained with dataset obtained from the 15th century document to classify textline and non-textline data. The necessity of this training is to classify correctly the under-segmentation and over-segmentation errors present in a document. This reduces the overhead to obtain a layout boundary mask for the correctly obtained layout from the anyOCR system. The following section describes in detail how the VGG Net is trained and how the dataset is prepared for training.

B. Model I - Dataset Generation

For generating the data required for VGGNet we used the 15th century document annotated with the ground truth. Ground truth is annotated in such a way that each textline in the document is annotated with same pixel value in Palette mode. So the last unique pixel value in the document corresponds to the total number of textlines in the document. We used this data to crop the boundary box around each unique textline pixel data. Non-textline data is gathered with two approaches. First approach is to get the data by randomly cropping the document with little bit of added skew. Second approach is to compare the anyOCR obtained layout with the ground truth with the Mean Square Error(MSE) ratio and to extract all the negative samples which will include both under-segmentation and over-segmentation images. Both of these data are then used for training. A total of 1739 textline images and 967 non-textline images were used for training and validation.

C. Model I - Architecture

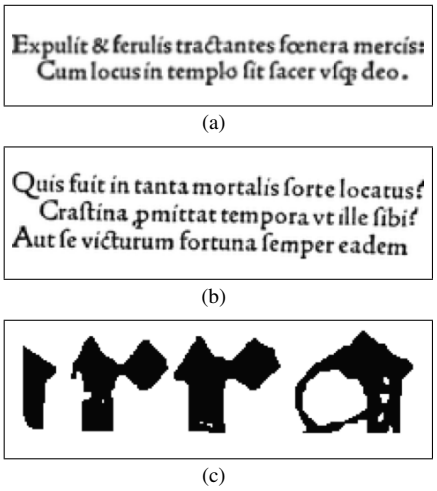
We used the tensorflow slim implementation of VGG19 model. The Network have an input layer of 224x224 image size followed by a convolution layer with 64 filters and kernel size 3x3. The next layer is a max-pooling layer with a 2x2 kernel. The output is then connected to another convolution layer containing 128 filters of 3x3 neurons followed by a 2x2 pooling layer. The next is a convolution layer with increased 256 filters followed by the third max pooling layer. It is then followed by two convolution and pooling layers with 512 maps and 3x3 kernel respectively. A fully connected layer of 4096 neurons is connected to the last max-pooling layer followed by a dropout layer. It is then connected to another fully connected layer with 4096 neurons with a dropout layer. Figure 2 describes the architecture used for training. In the table, Model E was used for training. The last Fully connected layer was replaced by two classes to represent textline and non-textline class. 1895 were used for training and 811 images were used for validation purposes.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 2. VGG Architecture.

D. Model I - Training & Results

Model is trained from scratch to learn the features for document related purposes. Training is performed for 20000 steps. Model result was then evaluated with the validation set. The model gave an accuracy of 94%. Trained model was then tested with non-textline images which is obtained from random cropping and MSE comparison from test document. Model gave very good accuracy for under-segmentation and incomplete images. Whereas, model found it difficult to classify side-note segmentation images where it is really difficult to tell with manual inspection. Figure 3 shows us a sample of images which the model correctly classified as non-textline whereas Figure 4 shows sample images which the model incorrectly classified as textline.

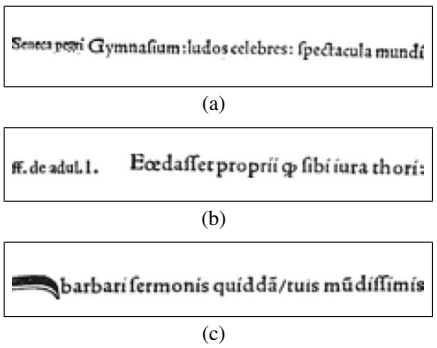


(a)

(b)

(c)

Fig. 3. VGG - Correctly classified as non-textline



(a)

(b)

(c)

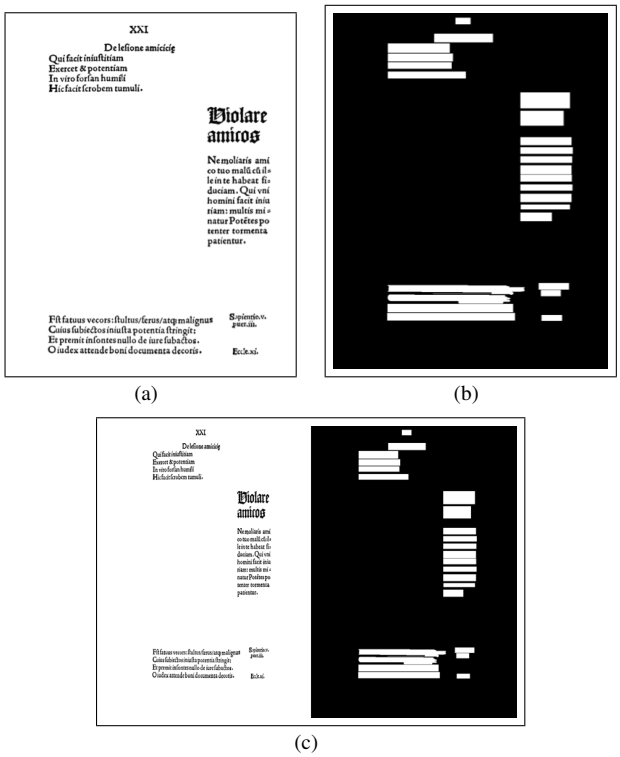
Fig. 4. VGG - Incorrectly classified as textline

E. Model II - Pix2Pix

The second model used is Pix2Pix[16]. Pix2pix is a variation of the adversarial networks where a model is trained for transforming one image to another image. In this sense, we have trained pix2pix to transform a document from full text document into a document with layout bounding mask for each textline in the image. The idea is that when it is trained for whole documents it will capture the features of line boundaries and it can able to distinguish them in the non-textline images we pass to the model.

F. Model II - Dataset Generation

15th century Latin Documents are used for the training the model. 50 documents were used for the dataset. All the documents were available with annotated ground truth. Documents were converted to bounding mask for the label. Mask boundary of the document is then combined with the original document to form the input image of the document. Sample dataset is shown in Figure 5. Figure 5 (a) shows the original document. Figure 5 (b) shows the mask boundary image obtained with the help of ground truth data. Figure 5 (c) shows the combined image that is fed to the model. The input size of the document is 2048x1024 as the original size of the document is resized to 1024x1024.



(a) (b)

(c)

Fig. 5. Pix2Pix - Sample dataset

G. Model II - Architecture

Affine-layer implementation of pix2pix is used for training. pix2pix is a conditional generative adversarial network(cGAN) [19] which is trained to map from input image to output image. The architecture of the pix2pix can be broken down into "Generator" and "Discriminator". Generative Adversarial Networks, attempts to train an "Generator" network which by the way is an encoder-decoder and simultaneously trains a "Discriminator" network by challenging it to improve on the classification accuracy. The Generator network takes an random input with noise and is trained to generate a sample data similar to that of given dataset. This generated data along with the real data is then fed into the Discriminator which is trained to predict whether it is real or generated image. Better training of GAN network determines the output quality of the generated image. Figure 6 gives us an overview the GAN architecture.

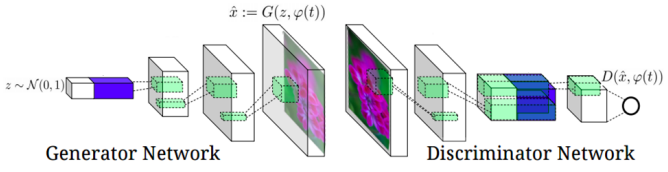


Fig. 6. General Architecture of Generative Adversarial Networks [18] .

In the function mentioned in Figure 7, ***V(D,G)*** is the information that the data from real input images goes through

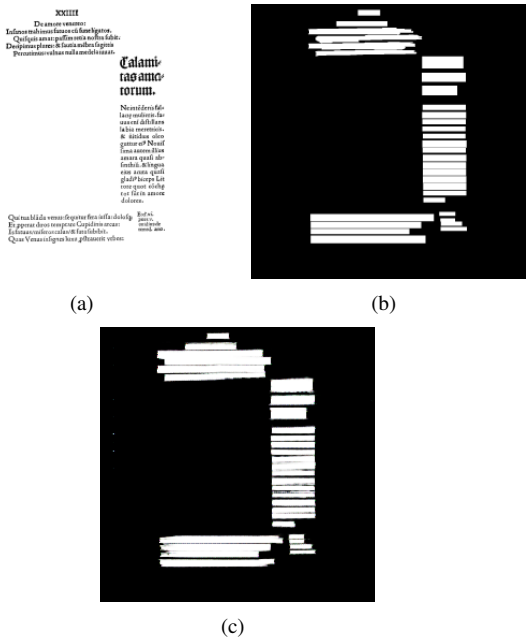
the discriminator. $P_{data}(x)$ denotes the probability distribution of the real data and $P(z)$ denotes the probability distribution of the generator. X mentions the sample from the $P_{data}(x)$ and Z mentions the sample from $P(z)$. $G(z)$ is the generator network and $D(x)$ is the discriminator network. Both $D(x)$ and $G(z)$ play the minimax game with the value function $V(D,G)$. Discriminator trains to maximize the function V whereas the Generator's task is the exact opposite where it tries to minimize the fucntion V. This model is trained with the latin document so that it learns to generate image with the boundary mask. 45 documents were used for the training purpose and the remaining 5 documents were used for testing.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Fig. 7. Mathematical representation of Generative Adversarial Networks [19]
.

H. Model II - Training & Results

Model is trained from scratch for full scale transformation of documents. Training is performed for 400 epochs. Trained model was then tested with Latin documents from the test set. Model gave good accuracy for line boundary upon manual inspection. The drawback is the layout of the obtained masks were not smooth but visible enough to figure out the segmentation. Figure 8 (a) shows us a sample test document passed on the model to obtain the layout boundary mask. Figure 8 (b) shows us the ground truth data for the layout boundary mask obtained programatically by masking the line area. Figure 8 (c) shows us the output of the trained model with the predicted layout boundary mask.



(a) (b)

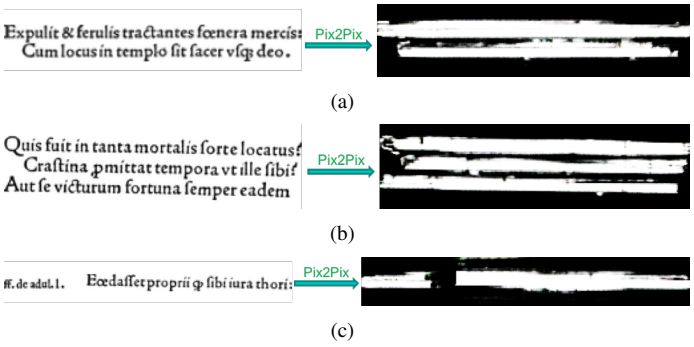
(c)

Fig. 8. pix2pix Model - Sample Output

III. Performance Evaluation

Both the trained models are used in combination to correct the layout error obtained from anyOCR system [20]. The text line segmentation method from the anyOCR system performs the algorithmic line segmentation and gives the output. The output document will have a unique pixel value for every single line it classified. These documents are error prone to have both under-segmentation and over-segmentation regions. This document is then cropped into lines by the unique coded pixel value. The cropped lines are classified with the first VGG model as a textline image or a non-textline image. The cropped images which are classified as non-textline from the VGG Model is then passed on to the pix2pix model to obtain the layout boundary mask. Figure 9 and 10 shows sample of images which are classified as non-textline by the VGG model.

Though, the output of the pix2pix was little bit fuzzy, the model achieves good results on manual inspection.



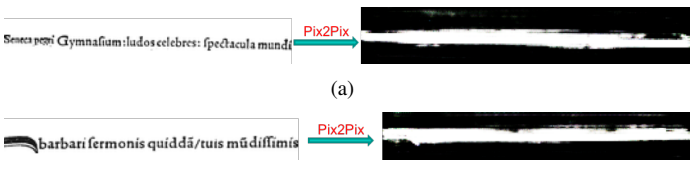
(a)

(b)

(c)

Fig. 9. Pix2Pix - Layout Boundary Mask output for under-segmentation images

The output of the pix2pix model was not sharp due to the dissolving of the boundaries to nearby pixel regions. On manual inspection, the line segmentation achieved good results. Notably for under-segmentation images, the model gave the layout boundary masks in a distinguishable way. Figure 9 shows us a sample of images which the model gives us an good layout boundary masks for under-segmented images. The model does not bode well for some under-segmented images as the line boundary was very close which is quite tricky for human inspection too. Figure 10 shows sample images which the model considered it as an textline. We can see that the boundary is very difficult to classify in these type of under-segmented images.



(a)

(b)

Fig. 10. Pix2Pix - Layout Boundary Mask output for under-segmentation images

Accuracy of the system is then measured by pixel-to-pixel comparison of the ground truth and the pix2pix model. The dissolving effect of the boundary image pixels was smoothened by setting a threshold where pixels more than threshold were set to white and pixels less than the threshold value were defaulted to black pixels. Ground truth for the cropped images are obtained manually by masking out the line boundary from the ground truth data of latin documents. Evaluation of the segmentation accuracy is done by adopting the F-measure metric which combines precision and recall values into a single representative value. Precision and Recall are estimated according to the below equations "(1)" ad "(2)"

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where *True-Positive(TP)*, *False-Positive(FP)* and *False-Negative(FN)* for the line segmentation are mentioned below:

- TP:line mask pixel value classified as line mask
- FP:line mask pixel value classified as non-line mask
- FP:non-line mask pixel value classified as line mask

F-measure is then calculated with the precision and recall values. Eq. "(3)" describes the calculation of F-Measure with β= 1.

$$F - Measure = \frac{(1 + \beta^2) * Precision * Recall}{\beta^2 * Recall + Precision} \tag{3}$$

The F-Meausure is calculated for the non-textline images in the test data. Test data contained both under-segmented and over-segmented images. Table I shows the accuracy achieved for both set of images. As we can see from the Table I, model gives good accuracy of more than 75% for under-segmented images as the line boundary mask is visible clearly. The over-segmented image test score was 60% because of the less distance between the lines and incomplete images. An overall accuracy of 71% is achieved for a total 40 images that were used for testing.

TABLE I
PERFORMANCE OF LINE BOUNDARY MASK BY CALCULATING F-MEASURE

Accuracy for under-segmented and over-segmented images	71.087 %
Accuracy for over-segmented images	60.043 %
Accuracy for under-segmented images	77.329 %

IV. CONCLUSION

In this paper, we have presented a Neural Network technique for improving the Layout Error Correction in the OCR pipeline. The technique was applied on 15th century Latin Documents. The Technique involves using a combination of VGG Net and pix2pix architecture to obtain a layout boundary mask for the document. It can be used as an additional step to improve the accuracy of the layout analysis in the DIAR pipeline. The model was evaluated on 50 European document images, which are composed of variety of layouts as shown in Figure 1. We have achieved good accuracy for under-segmented image which can be used for improving the information extraction. pix2pix architecture has variety of applications regarding the transformation of one image to another. Localization and Sharpening of the boundary mask will help us achieve better accuracy as a future improvement. Future researches in this area will be fruitful to achieve even better results in the DIAR community.

REFERENCES

[1] Ganis, M. D., Charles L. Wilson, and James L. Blue. "Neural network-based systems for handprint OCR applications." IEEE Transactions on Image Processing 7.8 (1998): 1097-1112.

[2] Burr, David J. "Experiments on neural net recognition of spoken and written text." IEEE Transactions on Acoustics, Speech, and Signal Processing 36.7 (1988): 1162-1168.

[3] Marinai, Simone, Marco Gori, and Giovanni Soda. "Artificial neural networks for document analysis and recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 27.1 (2005): 23-35.

[4] Marinai, Simone, Marco Gori, and Giovanni Soda. "Artificial neural networks for document analysis and recognition." IEEE Transactions on pattern analysis and machine intelligence 27.1 (2005): 23-35.

[5] Garcia, C., and X. Apostolidis. "Text detection and segmentation in complex color images." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 4. IEEE, 2000.

[6] Delakis, Manolis, and Christophe Garcia. "text Detection with Convolutional Neural Networks." VISAPP (2). 2008.

[7] Kumar, K. Sesh, Sukesh Kumar, and C. Jawahar. "On segmentation of documents in complex scripts." Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on. Vol. 2. IEEE, 2007.

[8] Nagy, George, Sharad Seth, and Mahesh Viswanathan. "A prototype document image analysis system for technical journals." Computer 25.7 (1992): 10-22.

[9] Wong, Kwan Y., Richard G. Casey, and Friedrich M. Wahl. "Document analysis system." IBM journal of research and development 26.6 (1982): 647-656.

[10] Baird, Henry S. "Background structure in document images." International Journal of Pattern Recognition and Artificial Intelligence 8.05 (1994): 1013-1030.

[11] Breuel, Thomas M. "Two geometric algorithms for layout analysis." International workshop on document analysis systems. Springer, Berlin, Heidelberg, 2002.

[12] O'Gorman, Lawrence. "The document spectrum for page layout analysis." IEEE Transactions on Pattern Analysis and Machine Intelligence 15.11 (1993): 1162-1173.

[13] Kise, Koichi, Akinori Sato, and Motoi Iwata. "Segmentation of page images using the area Voronoi diagram." Computer Vision and Image Understanding 70.3 (1998): 370-382.

[14] Shafait, Faisal, Daniel Keysers, and Thomas Breuel. "Performance evaluation and benchmarking of six-page segmentation algorithms." IEEE Transactions on Pattern Analysis and Machine Intelligence 30.6 (2008): 941-954.

[15] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[16] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." arXiv preprint arXiv:1611.07004 (2016).

[17] Breuel, Thomas M. "The OCRopus open source OCR system." DRR 6815 (2008): 68150.

[18] Adit Deshpande. (2016, September 30). Deep Learning Research Review Week 1: Generative Adversarial Nets [Blog post]. Retrieved from https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-1-Generative-Adversarial-Nets

[19] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

[20] Syed Saqib Bukhari, Ahmad Kadi, Mohammad Ayman Jouneh and Andreas Dengel, ””anyOCR: An Open-Source OCR System for Historical Archives.” The 14th IAPR International Conference on Document Analysis and Recognition (ICDAR2017), Kyoto, Japan, 2017.