Text detection and classification of construction documents

Narges.Sajadfar[a], Sina.Abdollahnejad[a] , Ulrich.Hermann[b] , Yasser.Mohamed[a]

[a] Department of Civil and Environmental Engineering, University of Alberta, Canada
[b] PCL Industrial Management Inc., 5404 99th St., Edmonton, Alberta, Canada
E-mail: sajadfar@ualberta.ca, sabdolla@ualberta.ca, rhhermann@pcl.com, yaly@ualberta.ca

Abstract –
Large construction projects generate thousands of documents that require a careful management. The classification of documents is an important step in document management and control. Construction documents are generated in different formats, many of which are unstructured and contain drawings and images, which makes the task of document classification and control even more challenging. In this paper, a dataset of 5000 documents is used as a case study. Optical Character Recognition (OCR) bounding boxes are applied to extract text from the set of documents. In the next step, two classification methods are applied. One based on a predefined set of keywords and another based on deep learning long short-term memory (LSTM) network. The challenges of the proposed approaches are discussed in relation to OCR bounding box locations with different document layout and how to obtain a set of representative key words for each class. Initial results of the study are encouraging and show that OCR technique combined with text classification is a powerful method for construction documents' control and can reach an accuracy of 92%.

Keywords –
Construction document; Text detection; Classification; Data mining; Optical Character Recognition (OCR); Deep Learning

1 Introduction

Many construction documents are complex and include text, images and drawings, which represent a vital source of information and knowledge regarding a construction project.

Text in construction documents includes important information such as title and number, which can be used to manage and classify construction document. However, not all documents are digitized or machine readable. Therefore, we need to create an image file of any printed document for text analysis. In this case, text is presented as an image in the file. Then, we can use OCR software to convert the scanned text and image into a machine-readable document. Also, it is one of the most used extraction techniques for documents and images [1]. In this paper, the research of OCR in construction industry is reviewed. In the second step, text detection and classification of construction documents are described, then two classification methods are applied on the case study. First classification is based on a predefined set of keywords and the second one is based on deep learning long short-term memory (LSTM) network. Finally, we compared and analyzed the effectiveness of the two classification, and the research results are presented.

2 Related studies

The literature review shows that previous researchers used OCR in construction document for different purposes. Berkhahn et al., (2008) used OCR and Kohonen neural network in construction drawings, in order to extract information about the dimensions of construction parts and inscription texts. Their model checks out all the dimension line points and construction element points to extract the dimension number and text. However; the user needs to check the result and correct the errors [2]. Banerjee et al., (2016) used OCR system to hyperlink engineering drawing documents. They created a hyperlink based on the extracted information, as a result the engineers can quickly navigate between different files. They

achieved more than 94% accuracy on automatic hyperlinking [3]. Also, Banerjee et al., (2017) used OCR engine in (Architecture, Engineering & Construction) AEC industry drawing documents for detection of Elevation Datum (ED) name, and graphical shape of ED, also they used experimental analysis to validate the ED name. The result of their research shows they achieved overall accuracy of 95% for ED detection and accurate destination document text recognition [4]. Another research of Banerjee et al., (2017) used OCR engine for extraction of alphabetic code and text of reference document, for that purpose of creating automatic navigation among architectural and construction documents. Their result shows the OCR has more than 91% accuracy on character level recognition [5]. Seraogi et al., (2017) used OCR engine in AEC to find the correct orientation of the documents based on the information of extracted texts and graphical shapes. They used mixed text/image drawings as their case study and achieved more than 99% accuracy on automatic orientation [6]. Gupta et al., (2018) used OCR engine in AEC to extract the title of the documents. In their method, OCR engine is scanning the information table only to extract the title. Also, they used historical data to increase the accuracy of their model. However, the extracted title should be reviewed by user to achieve 100% accuracy [7].

3 Related software and applications

Several commercial software packages are using OCR technology to extract the document information. Procore is a construction project management software that is using OCR on pre-defined template to extract drawing number, drawing discipline and drawing title. Drawing block text should be on the bottom right of drawing with specific size and location, then the Procore can automatically pre-populate the fields [8]. Docparser is another software which is using OCR engine to extract the text from any document. The user must define the specific locations inside the document and rules to apply on the all documents. Then Docparser will train to find the location of each field. Finally, this software will extract the text from pre-defined location based on regular expressions and pattern recognition [9]. Microsoft Azure is a computer application which is using variety of technologies such as OCR engine for text analysis. It can extract the text from images and documents, which can be used for label recognition, key phrase extraction and enable searching. Microsoft Azure is also using computer vision algorithms for image classification. The user must provide the labeled images to train a custom vision algorithm and create a model, which can classify new images [10]. The biggest problem of existing software is that they are not suitable for documents with variety of templates and their accuracy will significantly decrease in such situations.

4 Methodology

4.1 Dataset preparation

10 different document types and 500 documents from each document type (in total 5000 documents) was selected for case study analysis. All documents were in PDF format, which should be converted to image format for the purpose of using OCR text detection. Adobe Acrobat Pro DC was used to convert PDFs to PNGs. Since the information we want to extract is usually on the first page of each document, only the first page of each document was selected for future text analysis.

4.2 Pre-processing steps for improving image quality

The case study includes different sizes of documents, with different qualities. The resolution of images should be at least 300 DPI for a better text detection result. The first step is to resize all the documents to A4 size (8.27*11.96 inch) which will affect the resolution of images. In the second step, *Matlab*'s image processing toolbox is used for improving the quality of images. It has different function and filters that can be applied to modify the images. *rgb2gray* filter is used to convert documents to grayscale images. *Imadjust* filter is applied to increases the contrast and brightness of the output image and

Median filter is used to remove the noise from the grayscale pictures [11]. Figure 1 shows an original image and the enhanced images using *Matlab*'s image processing toolbox.

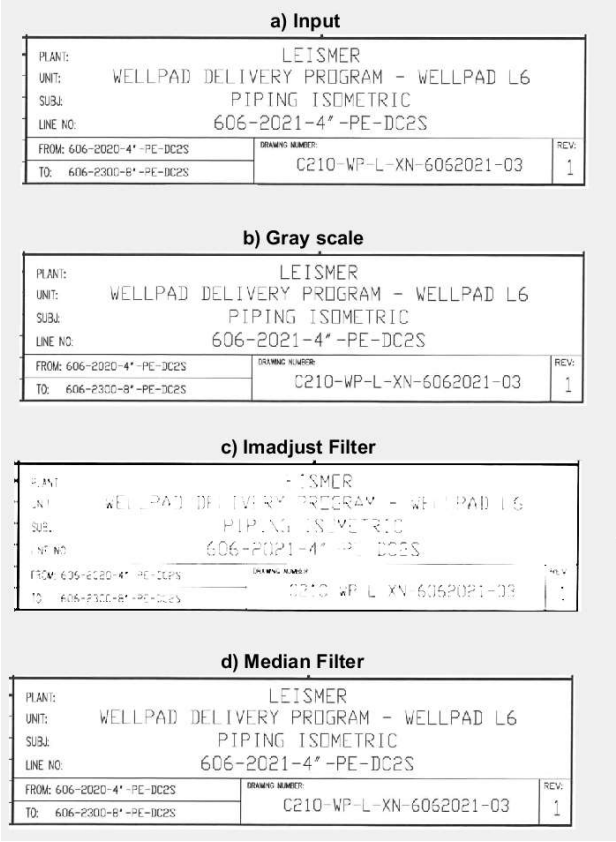


Figure 1: Original image and enhanced images using *Matlab* image processing

4.3 Layout analysis

Layout analysis such as page segmentation and region classification have an important role in text detection. Different regions such as texts, images, and tables should be identified in order to extract the text correctly. Layout analysis defines the possible location of text that needs to be extracted which can increase the OCR accuracy and extract more useful text from each document [12]. As figure 2 shows the dataset has different layouts which needs to classify to text and non-text segmentations.

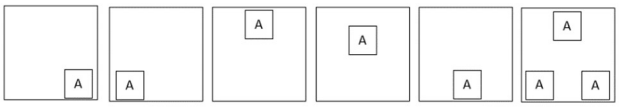


Figure 2: Possible locations of text in dataset

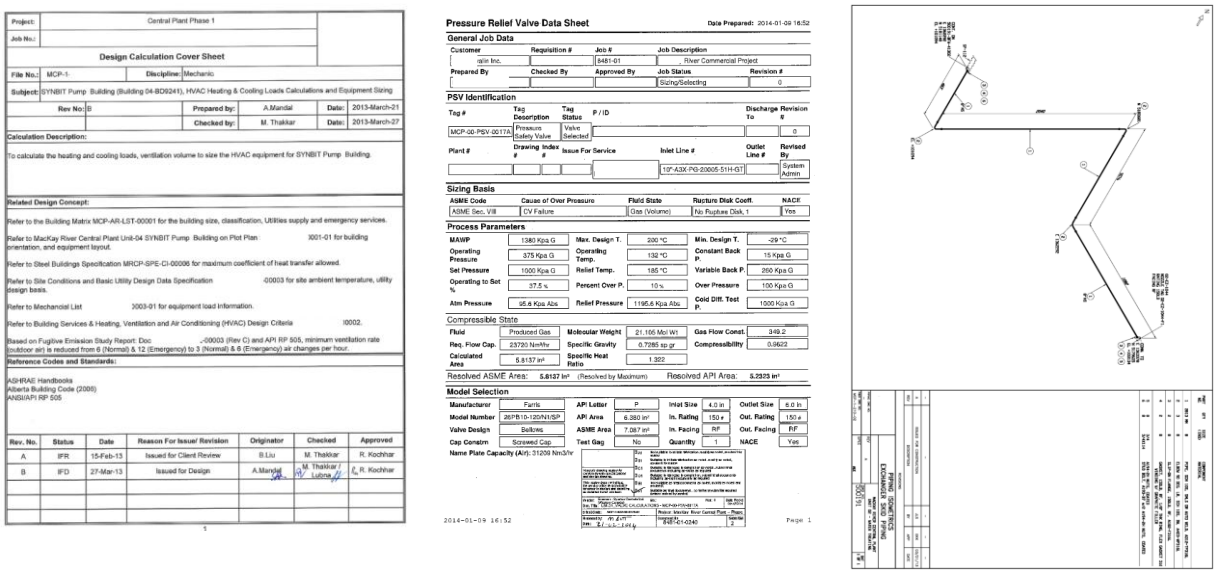
For the case study, connected component analysis was used to classify text and non-text segmentations. Non-text segmentation involves table, image, and lines. In the next step all the text segmentations will be processed through *Matlab*'s OCR in Computer Vision System Toolbox™. Figure 3 illustrates the layout analysis and text extraction in next section.

4.4 Text extraction from Construction Document

Different OCR engines were tested for text detection. The best result for case study was achieved via *Matlab* OCR in Computer Vision System Toolbox™. As figure 3 shows, the following steps were applied [13]:

- Resize the image if needed
- Binarize image (creates a binary image from image)
- Connected component analysis
- Detect text regions using MSER algorithm
- Remove non-text regions
- Provide region of interest (ROI) around the text
- Expand bounding boxes around words
- Dilate image to make letters thicker
- Apply OCR to find as much text as possible in no specific order, even if they are embedded in images.
- Store the detected text in text file format

Matlab OCR can provide the word confidence and character confidence for extracted result. To have a better accuracy, we only accept more than 80% confidence level.



a) Text document b) Table document c) Drawing document

Figure 1. Document examples of the dataset

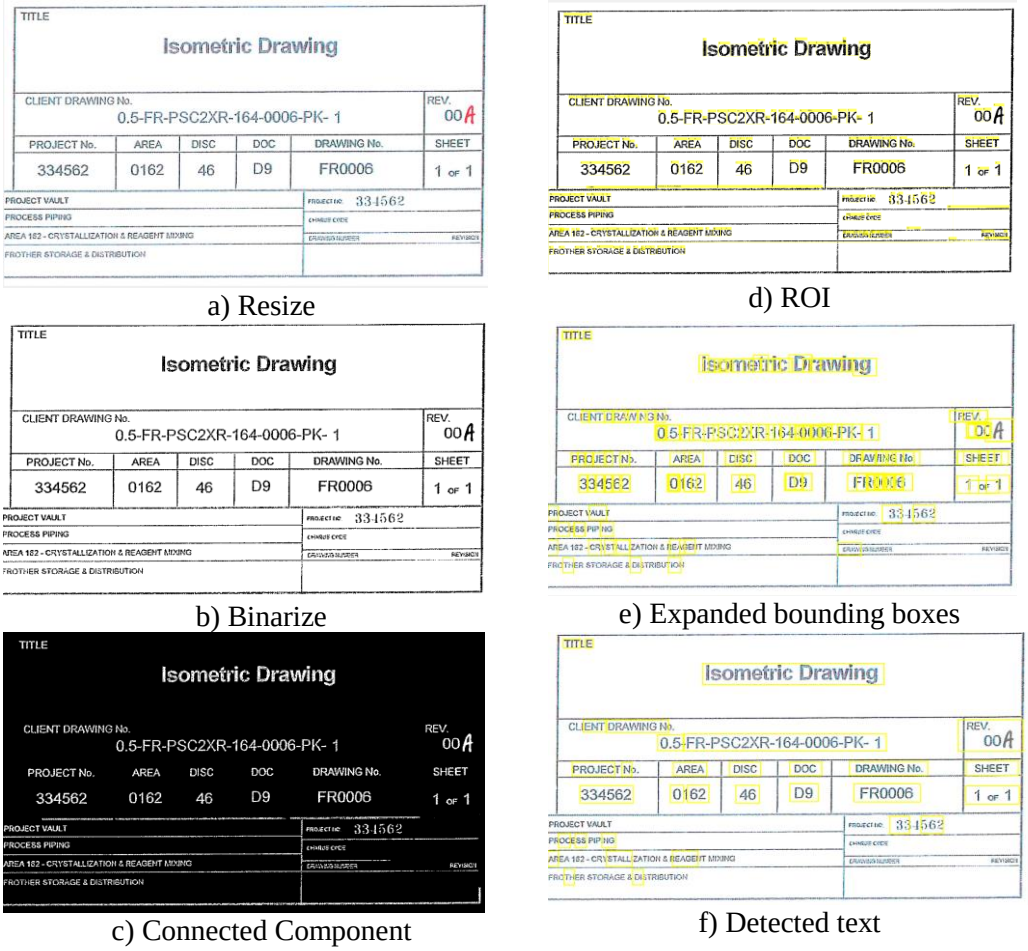


Figure 4. Text extraction steps

4.5 Classification based on predefined set of keywords

In order to process text categorization, we need to provide a list of keywords before classification. The keywords can either be determined by experts in the specific domain of the application or based on historical documents. In case study experiment, we program a *Matlab* tool box to find the keywords in the text files and then assign the appropriate classes to the documents. Based on our case study experiment, we analyzed 5000 documents for text categorization from 3 different construction projects. The accuracy of classification is tested based on 100 documents for each class and it was between 73% to 85% for different classes. However, in this method each document can be assigned to more than one class since each document can contain the keywords of more than one class. The other challenge of this method is reliability of accuracy. The accuracy of classification for a new project that we do not have any samples is questionable. To increase the reliability of the pre-defined keywords, text mining was applied via *Matlab*'s Text Analytics Toolbox™. The purpose of this method is analyzing the most repeated words in each document type, and then unique words were added to the keywords, which can be another option instead of pre-defined keywords for any new project. In this step, the accuracy of classification has increased between 85% to 92%.

4.6 Classification based on deep learning long short-term memory (LSTM) network

Recurrent neural network (RNN) is a class of artificial neural network which is developed during the 1980s [14]. RNN is a repeating model of neural network with different loops to connect the previous information to the present task. When there is a gap between relevant information and output, the RNN is unable to connect the information. A long short-term memory (LSTM) network is a type of RNN which is designed to fill the gap and avoid the long-term dependency problem. LSTM network will take three decisions about the information: decide about the useless information which should be removed, decide about the new information which should pass to the next layer, and decide about the output of each layer [15]. Recently, LSTM is increasingly used to classify text data. Text data is naturally sequential, and LSTM can learn sequences from the training data [16]. Several researchers reported the high accuracy of their text classification result base on LSTM, such as [17], [18], [19]. We used *Matlab*'s Deep Learning Toolbox™ to test LSTM network with a word embedding layer on our dataset. The dataset was imported to the toolbox in .CSV format and it contains two columns: The first one is label of document types and the second column contains the text of each document. In the next step, each word converted to numeric sequences to be use as an input in LSTM network. The LSTM model was defined by hidden layers, and word embedding layer. Table 1. shows the architecture of LSTM network used in our case study. The training model partitioning is: 70% training, 15% validations, and 15% test observations. The maximum number of epochs is 10 and initial learn rate is 0.01. The validation accuracy of classification is between 75% to 83%.

Sequence input layer	1 dimension
Word Embedding Layer	100 dimensions and 87106 unique words
LSTM layer	180 hidden units
Fully connected layer	Number of classes, 10 layers
Softmax layer	softmax
Classification Output layer	Number of outputs, 10 classes
Loss function	Cross entropy

Table1. LSTM architecture

5 Results and conclusion

Document text detection and classification is a challenging task especially for construction documents which includes text, images and tables. The result of case study on construction documents confirmed that OCR is a feasible tool for text detection and the result of that can be used for classification of construction documents. We conducted experiments on large-scale datasets consisting of 5000 construction documents. The first classification was based on predefined set of keywords and most repeated words in each document. The accuracy of this classification was between 85% to 92%. The second classification

was based on LSTM network and the accuracy was between 75% to 83%. The evaluation demonstrated that predefined set of keywords has more accurate result. however, the layout of documents and texts should be analyzed in advance which is time consuming and expensive. LSTM has a better accuracy on new sets of data, and it does not require data analysis. The result of classification can improve if we can increase the accuracy of OCR text detection.

The paper shows that deep learning algorithms such as LSTM has potential benefit to be use in construction documents classification. However, other deep learning algorithms need to be tested. In this paper, we consider the experiments as preliminary result and more data and algorithm should be test in future research.

6 Acknowledgments

The authors would like gratefully to acknowledge the PCL company for providing case study material for this project and their support.

References:

[1] Zhang, J., Cheng, R., Wang, K., & Zhao, H. (2013, September). Research on the text detection and extraction from complex images. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on* (pp. 708-713). IEEE.

[2] Berkhahn, V., & Tilleke, S. (2008). Merging neural networks and topological models to re-engineer construction drawings. *Advances in Engineering Software*, *39*(10), 812-820.

[3] Banerjee, P., Choudhary, S., Das, S., Majumdar, H., Roy, R., & Chaudhuri, B. B. (2016, April). Automatic Hyperlinking of Engineering Drawing Documents. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on* (pp. 102-107). IEEE.

[4] Banerjee, P., Das, S., Seraogi, B., Majumdar, H., Mukkamala, S., Roy, R., & Chaudhuri, B. B. (2017, November). Automatic Elevation Datum Detection and Hyperlinking of Architecture, Engineering & Construction Documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (Vol. 2, pp. 37-38). IEEE.

[5] Banerjee, P., Choudhary, S., Das, S., Majumder, H., Mukkamala, S., Roy, R., & Chaudhuri, B. B. (2017, November). A System for Creating Automatic Navigation among Architectural and Construction Documents. In *Document Analysis and Recognition*

[6] Seraogi, B., Das, S., Banerjee, P., Majumdar, H., Mukkamala, S., Roy, R., & Chaudhuri, B. B. (2017, November). Automatic Orientation Correction of AEC Drawing Documents. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (Vol. 2, pp. 9-10). IEEE.

[7] Gupta, S., MuNherjee, J., Bhattacharya, D., Majumder, H., Roy, R., & Chaudhuri, B. B. An Efficient Approach for Designing Deep Learning Network on Title Block Extraction for Architecture, Engineering & Construction Documents. In Document Analysis Systems (DAS), 2018: 13th IAPR International Workshop on Document Analysis Systems (pp. 5-6) VIENNA, 5.

[8] Which fields can Procore automatically populate when uploading drawings? On-line: https://support.procore.com/faq/which-fields-can-procore-automatically-populate-when-uploading-drawings, Accessed: 25/01/2019.

[9] Extract Data From PDF: How to Convert PDF Files into Structured Data. On-line: https://docparser.com/blog/extract-data-from-pdf/, Accessed: 25/01/2019.

[10] What is Computer Vision? On-line: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home, Accessed: 25/01/2019.

[11] Inglot, J. (2012). Advanced image processing with MATLAB.

[12] Antonacopoulos, A., Clausner, C., Papadopoulos, C., & Pletschacher, S. (2015, August). ICDAR2015 competition on recognition of documents with complex layouts-RDCL2015. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on* (pp. 1151-1155). IEEE.

[13] Computer Vision System Toolbox. On-line: https://www.mathworks.com/help/vision/index.html, Accessed: 25/01/2019.

[14] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533.
[15] Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.
[16] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
[17] Nowak, J., Taspinar, A., & Scherer, R. (2017, June). Lstm recurrent neural networks for short text and sentiment classification. In International Conference on Artificial Intelligence and Soft Computing (pp. 553-562). Springer, Cham.
[18] Rosander, O., & Jim, A. (2018). Email Classification with Machine Learning and Word Embeddings for Improved Customer Support.
[19] Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1422-1432).