Digital conversion model for hand-filled forms using optical character recognition (OCR)

Digital conversion model for hand-filled forms using optical character recognition (OCR)

J E M Adriano[1], K A S Calma[2], N T Lopez[3], J A Parado[4], L W Rabago[5], and J M Cabardo[6]

Asia Pacific College, 3 Humabon Place, Magallanes, Makati City, Philippines 1232

[1]jmadriano2@student.apc.edu.ph
[2]kscalma@student.apc.edu.ph
[3]ntlopez@student.apc.edu.ph
[4]japarado@student.apc.edu.ph
[5]lorenar@apc.edu.ph
[6]jayveec@apc.edu.ph

Abstract. The process of manual data entry used by several industries garners a high error rate. This is because the manual process relies too heavily on a human's capability to interpret handwritten forms. To reduce the high error rate of data entry, the researchers explored the different processes that comprise optical character recognition (OCR) and used it on a novel digital conversion model for hand-filled forms. The OCR process is made up of 4 major phases. The techniques for each stage are as follows: Sauvola binarization for image pre-processing; blob analysis for character segmentation; pre-trained Convolutional Neural Networks: GoogLeNet, AlexNet, and VGG16 for feature extraction, and Support Vector Machines (SVM), K-Nearest Neighbor (KNN), and Naïve Bayes for classification. The novel combination of Convolutional Neural Networks for feature extraction coupled with SVM for character classification showed promising results, going up to 98.62% in accuracy and 65.31% in F-Score.

1. Introduction

An educated human can look at a paper and read its contents as easy as it comes, but having a computer do the same is much more complicated than most people would think. One would first need to have a digital image of a document, process it to remove unnecessary information, get the computer to locate the characters and segment them so each individual letter can be identified. Only then can it get an output of a series of machine readable characters. This process is called Optical Character Recognition (OCR) which may be used to copy, search or edit digital text so the data can be utilized in various ways from something as simple as data storage, to more complicated applications like generation of reports or listing data-driven decisions.

As it stands in the modern age, the process of manual data entry is still widely used among industries such as healthcare, banking & finance, and real estate. These are industries whose integrity and profit rely on the accuracy of the data encoded to their databases from physical records. A study by Barchard [1] cracked down on the consequences of human data entry errors in the study they conducted in 2011. It's quite surprising how data entry errors, no matter the amount, largely affect the statistical results and conclusions of the higher management of companies. Furthermore, there is a concept known as 1-10-

100 developed by Labovitz et al. in 1992, quantifying the cost of errors in terms of units such as labor, money, time, etc. [2]. That is, a company spends 1 unit for preventing an error (Prevention Cost), 10 units for correcting an error (Correction Cost), and 100 units for working with data that were never corrected (Failure Cost). This means that the costs an organization must pay for data entry errors increase exponentially the later it is dealt with.

The average benchmark for data entry error rate is generally acknowledged to be 1% [3]. It may seem insignificant, but if 1 out of 100 documents from a total of thousands get an error, the cost they will accrue overtime will be heavy, especially for larger companies. The survey results on data entry errors in clinical databases showed that double-entry method ranged from 2.3% to 26.9% [4]. These errors were caused by mistakes in data entry and misinterpretation of the information from the original documents. Unfortunately, in the end, the ones who undertake these jobs are still human, who are naturally prone to error. This is especially pronounced when the data needed to be stored come from documents with handwritten elements because of the different ways letters can be written. Not to mention the different markings a writer can create, like a smudge or a correction, for example. To solve this, the multiple facets of optical character recognition is explored to be used in a Digital Conversion Model that is tailored for functioning on forms that contain handwritten content. This can potentially remove the human element which makes operations susceptible to errors, or at the very least, decrease data entry errors and increase efficiency. Additionally, the researchers noted that this is the only OCR methodology that utilizes pre-trained image CNNs for feature extraction instead of the usual global and local feature extraction techniques such as Local Binary Pattern (LBP), Features from Accelerated Segment Test (FAST), and HOG (Histogram of Gradients). The reason for this will be discussed later in the results and discussion portion of the paper.

Though the research deals with handwritten input, the type of characters to be recognized is limited to block (sample) characters only, as cursive (*sample*) writings require a different methodology from what is being proposed. Additionally, special characters (ñ, ~m, %) and punctuation marks such as periods (.) commas (,) exclamation points (!) etc. are not included. Finally, some character classes have been merged. These are characters that look similar regardless of case. All in all, there is a total of 47 classes of characters.

2. Related Works

ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) is an annual competition that evaluates different algorithms for large-scale object detection and image classification. Some notable entries and winners of the ILSVRC are pre-trained convolutional neural networks (CNN) such as AlexNet, VGG 16, and GoogLeNet. During the ILSVRC 2012 AlexNet was the winning entry. It contains 5 convolutional layers and 3 fully connected layers and makes use of Relu to add non-linearity which accelerates speed by 6 times while maintaining constant accuracy. It also makes use of dropout to deal with overfitting and reduces the size of network through overlap pooling [5]. During the ILSVRC 2014, there were two promising entries. One of these entries was the VGG-16, a pre-trained convolutional neural network that contains 16 weight layers. Its simple architecture uses a 3x3 convolutional layer stacked on top of each other and the depth increases by each layer. It reduces volume size by max pooling [6]. On the other hand, GoogLeNet is a pre-trained CNN developed by Google. It was the entry on ILSVRC 2014 that won that year's competition. It makes use of a new variant of CNN called Inception, which is used for classification; while R-CNN was used for detection. Its network was designed to increase the depth and width resulting to a 22-layer deep network while at the same time, keeping the computational budget constant [7]. Both VGG-16 andGoogLeNet gives a matrix output of [1000 x n] where n stands for the number of data entries.

All the pre-trained convolutional neural networks mentioned (AlexNet, VGG 16, and GoogLeNet) were trained using a subset of the ImageNet database. They were trained using more than a million images from ImageNet and can classify images into 1000 object categories. Furthermore, these pre-trained CNNs can be applied to the current study's (digital conversion model) feature extraction phase where the researchers intend to use convolutional neural networks.

The study done by Malon [8] successfully applied SVMs as their character classifier for mathematical symbols. Their objective was to address was the low success rate of conventional OCRs at classifying mathematical characters. A multiclass SVM classifier along with five different kernels were compared against a Naïve Bayes classifier reducing the overall error rate of InftyReader by 41%. Another study constructed an OCR methodology to be applied on Ottoman characters with SVM as the image classifier [9]. Three SVM kernels were used: linear, quadratic, and radial basis function (RBF). Of all the kernels the quadratic one was the most accurate giving a recognition rate of 87.32%. Lastly, a survey done by Thome [10] discusses the application of SVMs in the field of OCR. He focuses on Multiclass SVM which can be done in two ways: one-vs-one and one-vs-all. Additionally, Thome explains the advantage of SVMs over neural networks by stating the former's requirement of a smaller training data set requirement, greater ability to generalize, and the higher probability of generating good classifiers.

3. Proposed Method

Optical Character Recognition (OCR) is a complex combination of many processes that work subsequently to produce a machine-readable string of characters. Each major phase of OCR, consisting of Pre-processing, Character Segmentation, Feature Extraction, and Classification may utilize different techniques to fit different types of documents depending on their identifying characteristics. In fact, most studies today usually focus on improving only one stage without mentioning the others. **Figure 1** below shows the methods that the proponents have deemed feasible for the process.

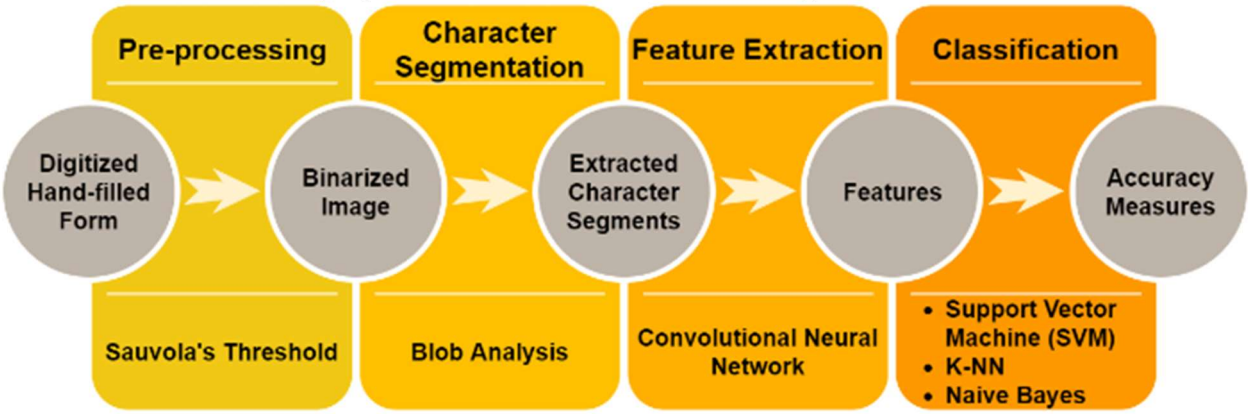

Figure 1. The Digital Conversion Model

In pre-processing, the image obtained through digitization is grayscaled. It changes the value of each pixel to represent only the intensity information such that its value ranges from 0-255. It is then binarized; that is, each pixel is classified into either 0 (black) or 1 (white), depending on the value of each pixel as compared to the threshold value. Without image binarization, it would be very hard for any device to properly identify characters because of the various noise that usually surround the relevant text area. In this case, Sauvola's Threshold, with its nature as a local thresholding technique, has been chosen as the most suitable for binarizing handwritten forms. Since it is exceptional in separating the foreground and the background even in noise cluttered areas of the image, which is apparent in some forms possessing color shading, background patterns, and unintended markings from the writer. The formula for Sauvola's threshold is,

$$T_S = m * \left(1 - k * \left(1 - \frac{s}{R}\right)\right) \quad (1)$$

Where m is the median and s is the standard deviation of the area [11].

The resulting binarized image is then ready for character segmentation phase where it will be segmented into areas. The objective of this phase is to determine the segments of the image that contains individual characters from a line of text. Blob Analysis, the technique to be used, computes the statistics

for connected regions within the binary image to identify blobs (region or area of connected pixels in an image) based on the values 0 or 1 [12]. The output of this phase is a collection of images containing a single character which will be used in the next phase.

During the feature extraction phase, text segments are analyzed for differentiating features. These differentiating features are extracted from the matrices of segmented characters. They are selected to uniquely identify a given text segment [13]. These features can then be used for machine learning training and predicting purposes. Both training data and test data are transformed into feature vectors prior to training the machine and having it make predictions. For this phase, Convolutional Neural Networks (CNN) were used; CNN is a deep learning algorithm or a type of machine learning algorithm that learns to classify images, video, text, or sound. Its architecture is trainable and is composed of multiple layers or stages. Each stage's input and output are a feature map of a set of arrays [14]. Furthermore, CNN incorporates constraints and deformation invariance through local receptive fields, shared weights and spatial subsampling [15]. An attractive quality of CNN is that its capacity can be controlled through depth and breadth variations. It contains fewer connections and parameters as compared to standard feedforward neural networks [16]. These characteristics make CNN less complicating to train than other neural networks thus it is opted to be the algorithm used for the digital conversion model's feature extraction phase.

4. Experiment and Results

In this section, results will be discussed according to the data and tools used and the optical character recognition pipeline. The data used for OCR applications for this research came from the National Institute of Standards and Technology Special Database 19 [17]. This dataset already provides a form to apply the character extraction techniques as well as individual character images to train the classifiers. As for the training and testing of classifiers, the researchers used MATLAB, a proprietary, numerical, computing environment and language that has a variety of "apps" that provide a graphical user interface for different functions like machine learning algorithms, algorithm optimization, and image processing, among others. GNU Octave, an open-source implementation of MATLAB whose syntax and formatting are largely compatible to MATLAB, was also used to some extent.

First, images are binarized using Sauvola's thresholding technique. The picture is reduced to grayscale to eliminate colors, background noise, and significantly lessen the impact of uneven lighting or shading. Hand-filled forms from the National Institute of Standards and Technology Special Database 19 were used.

After pre-processing is segmentation where character segments are determined from the image and line of texts using blob analysis. From the binary image having (0) as black and (1) as white, blob analysis detects objects by bounding (0) black connected pixels. The MATLAB function *vision.BlobAnalysis()* is used for segmentation. One of the outputs is the region of interest (x, y, width, height) of each blob detected where x and y are the coordinates of where the blob is in the image. To find the characters in an image, the program or function for detecting is modified. Blob analysis is not created specifically for character segmentation, it bounds any (0) black pixels that it could recognize no matter how big, long, or small it is. The proponents limited the aspect ratio, and area of the blob to suit the size of a handwritten character. Originally, there are no constraints indicated which lead to the blob analysis bounding anything that it detects. This leads to redundant miscalculations of extracted segments. From observations, a handwritten character in the form is not lower than an area of 20 pixels. The area constraint is needed so most small characters like the period, and comma is excluded in the extracts. Constraining the aspect ratio between 0.2 and 2.0 of the blobs, which is the size of an individual character further increased the segmentation accuracy and shows a segmented image without constraints.

The researchers also used segmentation to collect testing data for classification to ensure that it truly tests the extracted segments from blob analysis and not from the pre-segmented individual characters NIST data set. From 11 randomly selected forms from the NIST 19 data set, there were 10,319 extracted segments. 1,742 of the extracted images were excluded due to being outside of the research's scope or being unidentified segments. This leaves 8,577 segments for the classification phase.

The feature extraction process for both test and training sets follow the exact same procedure. The data set images are stored in a folder which contain subfolders that correspond to their character class. These images are then turned into a MATLAB image datastore class object. After labelling the data within the image datastore, it is resized (only within the datastore) into whatever dimensions the CNN requires it to be. The images are then transformed into feature vectors of variable size depending on the network being utilized. **Table 1** below shows the feature vector dimensions for the CNNs.

Table 1. Feature vector dimensions by CNN

CNN	Training Data	Test Data
AlexNet	65800 x 4096	8057 x 4096
GoogLeNet	65800 x 1000	8057 x 1000
VGG-16	65800 x 1000	8057 x 1000

Lastly, classification utilizes the output of the feature extraction as well as the data set labels as inputs for training and testing the classification models. The testing covers the determination of the success metrics: precision, recall, accuracy, and F-score. **Table 2** below shows the final tabulated results of the research.

Table 2. Consolidated test results

Convolutional Neural Network	Algorithm		Metrics (%)				Training Time (sec)	
			Precision	Recall	Accuracy	F-Score	Machine Training	Feature Extraction
AlexNet	SVM	1v1 Linear	**71.17**	**68.39**	**98.62**	**65.31**	1286	**221**
		1vAll Linear	66.25	63.07	98.35	59.99	5860	
	K-NN		69.19	54.60	98.08	56.83	9	
	Naïve Bayes		51.92	35.86	97.43	37.86	82	
GoogLeNet	SVM	1v1 Linear	59.20	54.24	98.07	51.79	237	686
		1vAll Linear	59.97	45.08	97.66	48.32	**12500**	
	K-NN		51.72	43.86	97.56	43.15	3	
	Naïve Bayes		40.03	24.47	96.87	25.02	18	
VGG - 16	SVM	1v1 Linear	64.24	58.17	98.22	58.17	356	**1980**
		1vAll Linear	61.87	51.63	97.94	53.80	11637	
	K-NN		44.18	36.19	97.42	35.56	**2**	
	Naïve Bayes		**29.72**	**21.44**	**96.84**	**20.77**	18	

5. Conclusion

To remove the high error rate produced by manual entry process of hand-filled forms, the proponents of the study explored the use of a digital conversion model using Optical Character Recognition technology which is tailored for functioning on forms containing handwritten content. This digital conversion model for hand-filled forms was designed using a unique combination of major OCR phase techniques – Sauvola's threshold (pre-processing), Blob Analysis (character segmentation), Pre-trained Convolutional Neural Networks (feature extraction), and Support Vector Machine (classification). AlexNet, GoogLeNet, and VGG-16 were the pre-trained CNNs used for feature extraction. The results of characters classified using the 1v1 Linear and 1vAll Linear Approach of SVM were also compared with the results of characters classified using K-Nearest Neighbor (K-NN) and Naïve Bayes Classifier. Furthermore, the precision, recall, accuracy and f-score of each combination of pre-trained CNN and classifier were determined. The best performing pipeline is feature extraction via AlexNet and a one-vs-one coding design, linear kernel support vector machine with an F score of 66.31%. Meanwhile, feature extraction via VGG-16 and classification via Naïve Bayes garnered the lowest accuracy, precision, recall

and F-score. Overall, using convolutional neural network as a means of feature extraction turned out to give very positive results which could be a solid foothold for eliminating manual data entry.

6. Recommendations

The researchers encountered technological limitations during the implementation of the methodology. One of the problems encountered was the limitation on RAM. Some MATLAB variables are so large, they fully occupy the entire RAM space, even before any script was ran. As such, the team has determined a minimum of 16GB and recommends 32GB of RAM should future researchers attempt to emulate the procedure. Stronger hardware equates to increased capability for more training data as shown by direct relation of the training data with accuracy. There are many more CNNs that can be used other than the three mentioned in this paper. For example, Resnet50, Resnet101 and Squeezenet. Importing ready-made ones or creating one's own is also possible and should be explored. In addition, one can also use other SVM kernels aside from linear, such as Gaussian and RBF. If these limitations are surpassed, a wide avenue of possibilities may be explored by future researchers.

References

[1] Barchard K A and Pace L A 2011 Preventing human error: The impact of data entry methods on data accuracy and statistical results *Computers in Human Behavior* no **27** pp 1834-9
[2] Labovitz G, Chang Y S and Rosansky V 1992 Making quality work : a leadership guide for the results-driven manager *Chichester: Essex Junction* (VT : Omneo)
[3] Rongala A 2015 Top 6 Manual Data Entry Challenges Companies Face *Invensis Technologies 27 April 2015* Online Available: https://www.invensis.net/blog/data-processing/top-6-manual-data-entry-challenges-companies-face/ Accessed 13 July 2018
[4] Goldberg S I, Niemierko A and Turchin A 2008 Analysis of data errors in clinical research databases *AMIA Annu Symp Proc.* vol **v.15** no **3** pp 242–6
[5] Gao H 2017 A walk-through of AlexNet *Madium 7 August 2017* Online Available: https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637 Accessed 16 August 2018
[6] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition *ICLR*
[7] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanchoucke V and Rabinovich A 2014 *Going Deeper with Convolutions* Online Available: https://arxiv.org/pdf/1409.4842.pdf Accessed 16 August 2018
[8] Malon C, Uchida S and Suzuki M 2008 Mathematical symbol recognition using support vector machines *Pattern Recognition Letters* no **29** pp 1326-32
[9] Kilic N, Gorgel P, Ucan O and Kala A 2008 Multifront Ottoman character recognition using support vector machine *IEEE Xplore* pp 328-33
[10] Thome A C G 2012 Advances in character recognition *SVM Classifiers - Concepts and Applications to Character Recognition* (IntechOpen) pp 25-50
[11] Kalaiselvi T, Nagaraja P and Indhu V 2017 A comparative study on thresholding techniques for gray image binarization *Int. J. of Advanced Research in Computer Science* vol **8** no **7**
[12] Sookman S 2006 Blob analysis and edge detection in the real world *Evaluation Engineering 1 August 2006* Online Available: https://www.evaluationengineering.com/blob-analysis-and-edge-detection-in-the-real-world Accessed 16 August 2018
[13] Verma R and Ali D J 2012 A-survey of feature extraction and classification techniques in OCR systems *Int. J. of Computer Applications & Information Technology* vol **1** no **3**
[14] LeCun Y, Kavukcuoglu K and Farabet C 2010 Convolutional networks and applications in vision *IEEE Int. Symp. on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems* pp 253-6
[15] Lawrence S, Giles L, Ah C T and Back A D 1997 Face recognition: a convolutional neural-network approach *IEEE Transactions on Neural Networks* vol **8** no **1**
[16] Krizhevsky A, Sutskever I and Hinton G E ImageNet classification with deep convolutional

neural networks *Proc. of the 25th Int. Conf. on Neural Information Processing Systems* vol 1 pp 1097-05
[17] Grother P 2016 *NIST Special Database 19 September 2016* Online Available: http://doi.org/10.18434/T4H01C Accessed 30 July 2018