Document Analysis and Recognition

Shriyans Nidhish
Computer Science and Engineering
Galgotias University
Greater Noida, UP, India

Deepank Dhillon
Computer Science and Engineering
Galgotias University
Greater Noida, UP, India

Vaibhav Kamra
Computer Science and Engineering
Galgotias University
Greater Noida, UP, India

Shivani Rawat
Computer Science and Engineering
Galgotias University
Greater Noida, UP, India

***Abstract*— The cornerstone of this project is to create a DAR system that will convert its functionality into applications in the areas of research and commercial systems. The system will allow the user to convert printed text into an editable digital format, making recognition easier and more efficient. The system will rely on well-known methods, such as image processing, pattern recognition and artificial intelligence, to which will be added a unique technique, namely natural image processing. Effective use of the CNN algorithm would help reduce image blur while effectively recognizing text. The entire system will be a three-step process of creating a digital document from a hard copy, scanning the digital document, and recognizing templates in the document for correct, error-free results.**

Keywords - Document Analysis and Recognition (DAR), Convolution Neural Network (CNN), Artificial neural network (ANN), Hidden Markov Model (HMM), Image processing, Model.

I. INTRODUCTION

Document Analysis and Recognition (DAR) focuses on automatically extracting information from paper and converting it into a digital format. The expected output of the DAR system is a symbolic process that can be used by computers. Paper documents have always been the main source contributing to the progress of humanity and the trend has not changed yet; a considerable amount of information is always recorded, stored and distributed in paper format. Although current technology aims to move towards a paperless world, some studies show that the use of paper as a medium for information exchange continues to increase.

Moreover, there are still areas of application for which paper persists to be the preferred medium. The most used applications of DAR are the processing of office documents (such as invoices, bank documents, commercial letters and checks), thus reducing the falsification of documents. The current availability of low-cost, high-resolution scanning devices, powerful computers, and OCR packages is an added benefit and can solve simple document recognition tasks for many users. Recent research methods are expanding the use of DAR techniques, including the processing of old / historical documents in digital libraries, retrieval of information from "digital" documents, such as PDF and HTML, and natural image analysis. (Acquired with mobile phones and digital cameras) containing textual information [1].

The incorporation of several computer skills such as image processing, pattern recognition, natural language processing, artificial intelligence and database systems has led to the development of the DAR system. DAR applications are particularly well suited to incorporating machine learning techniques for two reasons: first, classification algorithms are used at several levels of processing, from image pre-processing to character classification; second, large sets of manually created data with associated images are available and can be used for automatic training of classifiers [1]. The DAR system will find application in commercial and research systems. In fact, in the coming decades, handwritten character recognition will be frequently used as a reference for document analysis and recognition.

II. PROPOSED SYSTEM

The above-mentioned system will find applications in the detection of document tampering, automatic sorting of mail and the creation of editable digital documents from hard copies. The system will be effective in converting paper copies into digital documents that will be editable to increase system functionality and improve the accuracy of the current system. In the future, with some updates, the system will allow the creation of new font styles if the detected font is not available in existing Google fonts. Image blur will be supported for increased accuracy and efficiency. The system will be effective in separating the different fonts and creating a field of these fonts in the database.

III. METHODS

A. Pre-Processing

The operation of pre-processing in document image analysis transform the input image into an improved image more appropriate for subsequent analysis. Image-to-image transformations in DAR belong to four main classes: filtering, geometric transformations (for example, bias detection), object boundary detection, and thinning.

B. Layout Analysis

1) Physical layout analysis

The physical layout analysis is designed to extract regions with uniform characteristics from the document. The segmentation algorithms used in image processing can be grouped into three main categories: pixel classification, edge-based segmentation, and region-based segmentation [1].

2) Logical layout analysis

Logical presentation analysis assigns meaning to the regions identified by the physical layout analysis. Examples of characteristics considered are the size of the blocks, their mutual position and information on the textual parts such as the predominant font, the size and the spacing of the characters [1].

C. Text Recognition

Text recognition is performed to convert an image containing printed or handwritten text into a format that is understandable by a computer (eg. ASCII or Unicode). Text reading techniques can be divided into two main categories (online and offline) depending on the input device used [1].

D. Software

The Kaggle library and TensorFlow are our main tool for executing all pre-processing, feature detection and learning algorithms. This package includes utilities for each computer vision and machine learning feature that we need for this project.

E. Data Set

All of the training and test data we use are handwritten letters produced with varying brush sizes and hardness values. The dataset includes grayscale 128x128 images with the following letters: _, _, _, _, _, _, and _ [3]. There are different images of each letter and a retained cross validation is used to separate the dataset into training / test sets and to compare training and generalization errors.

In order to reduce the variations between the images, we implement two different pre-processing techniques. In the first method, we calculate the centroid (_x; _y) of the image via moments. These are given by _x = M10 M00 and _y = M01 M00, where Mpq = X128 x = 1 X128 y = 1 xp _ yq _ ImageIntensity (x; y).

F. Feature Detection

We compare two different types of feature vectors for our learning models. The first is to treat the grayscale image as a matrix of pixel values and then reformat that matrix into a single-line vector. This vector thus codes each of the 128 values of 128 pixels of the image. The second type of feature detection method we use is the Harris operator to detect the corners of an image [3]. We use an abstract kaggle implementation that assigns a "confidence level" to each of the detected corners, then returns a vector of the coordinates of the highest k scores. In our research, we compare performance when using values of 5 angles, 7 angles and raw pixels.

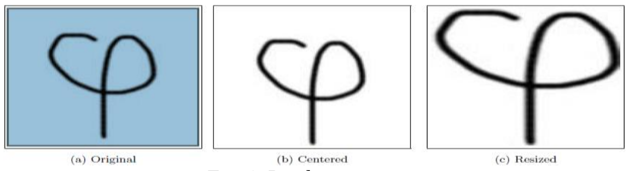


Fig. 1. Pixel comparison.

IV. Models

We used two different learning approaches in our project: Artificial neural network, Hidden Markov Model. Here we outline each of this approach.

A. Artificial Neural Network

The idea of ANN is based on the belief that the work of the human brain in establishing the right connections can be imitated by using silicon and wires as living neurons and dendrites. ANNs are composed of multiple nodes mimicking the biological neurons of the human brain. Neurons are inter-connected by links where they interact with each other. Every node can take input data and perform simple operations on the same. The result of these operations is transmitted to other neurons. The output on each node is called its activation or node value.

The simple ANN illustration is as follows –

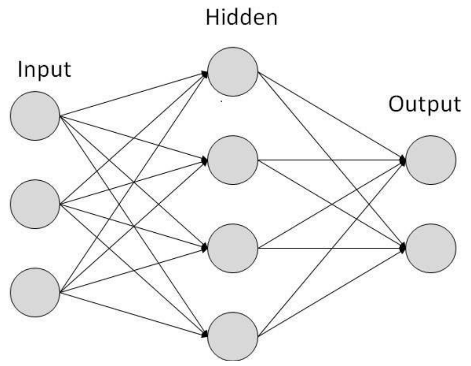


Fig. 2. ANN States

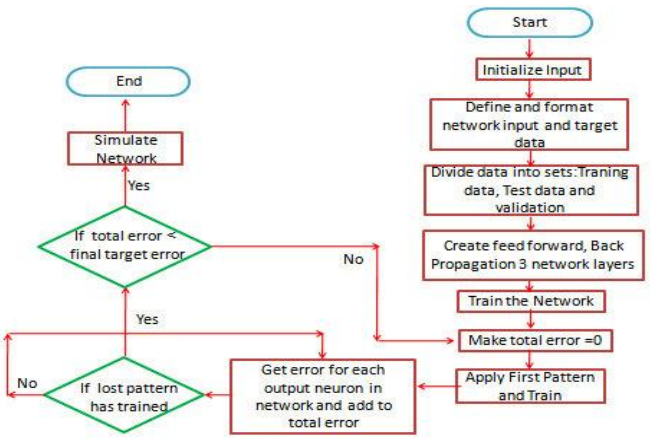

Start
Initialize Input
Define and format network input and target data
Divide data into sets:Traning data, Test data and validation
Create feed forward, Back Propagation 3 network layers
Train the Network
Make total error =0
Apply First Pattern and Train
Get error for each output neuron in network and add to total error
If lost pattern has trained
No
Yes
If total error < final target error
No
Yes
Simulate Network
End

Fig. 3. ANN Flow Chart

Few important code used in our application of above algorithm is illustrated below:

```
def build(width, height, depth, weightsPath=None):

# initialize the model
model = Sequential()
# first layer CONV => RELU => POOL
model.add(Convolution2D(32, (3, 3), input_shape =
(width, height, depth)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
# second layer CONV => RELU => POOL
```

```
model.add(Convolution2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
# third layer of CONV => RELU => POOL
model.add(Convolution2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
# set of FC => RELU layers
model.add(Flatten())
# number of neurons in FC layer = 128
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
# as number of classes is 36
model.add(Dense(36))
model.add(Activation('softmax'))
# if weightsPath is specified load the weights
if weightsPath is not None:
print('weights loaded')
model.load_weights(weightsPath)
# return model
return model
```

B. *Hidden Markov Model*

A hidden Markov model is a doubly stochastic process, with an underlying unobservable process (hence the hidden word), but can be observed by another stochastic process that produces the sequence of observations [2]. The hidden process consists of a set of states connected to each other by transitions with probabilities, while the observed process consists of a set of outputs or observations, each of which can be issued by each state in accordance with a function of output probability density (PDF) [5-7]. Depending on the nature of this PDF function, several types of HMM can be distinguished.

The simple HMM flowchart illustration is as follows−

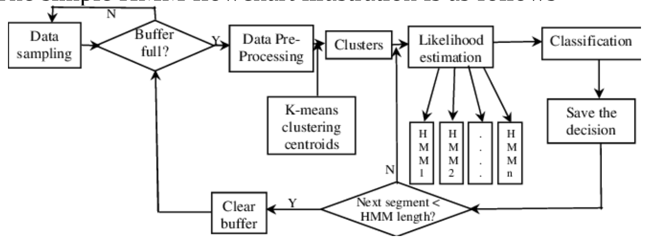


Fig. 4. HMM Flow Chart

Few important code used in our application of above algorithm for training of the data set is illustrated below :

```
#initialize model
model = Net.build(width = img_width, height = img_height,
depth = no_of_channels)
print('building done')
# Compile model
rms = optimizers.RMSprop(lr=0.001, rho=0.9, epsilon=None,
decay=0.0)
print('optimizing done')

model.compile(loss='categorical_crossentropy',
          optimizer=rms,
          metrics=['accuracy'])
print('compiling')
# this is the augmentation configuration used for training
# horizontal_flip = False, as we need to retain Characters
train_datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rescale=1. / 255,
    shear_range=0.1,
    zoom_range=0.1,
    rotation_range=5,
    width_shift_range=0.05,
    height_shift_range=0.05,
    horizontal_flip=False)
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

V. COMPARISON

Table. 1. Existing and proposed system comparison

Previous Systems	**Our System**
The current system is based on structural pattern.	Our system is based on neural pattern recognition.
The error percentage was 6.10	The error percentage is reduced to 1.0 to 2.3
The older system divides the paragraph into words.	Our system divides the paragraph into characters which increases the accuracy of the system.
The previous system does not allow creation of new scanned fonts.	Our system provides the facility to create new fonts.
This feature is not present in the existing system.	Our system provides image de warping which flattens the page image and helps in recognizing text easily.
This facility is not available in present system.	The facility of classifying documents after recognizing characters is present.

VI. RESULT

We found that the best performance came from a combination of centering or resizing, raw pixel value functions, and a carrier vector machine model. The figure below shows our results when using a linear kernel. Subsequently, we achieved slightly better results with a Kaggle kernel of degree 3, generating a generalization error rate of about 0.8% when centering images and using raw pixel values. Other types of nuclei, such as the radial basic function and the higher order polynomials, have significantly degraded due to over-adjustment, as shown by a 0% error in learning and an error in error generalization greater than 0.8%.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

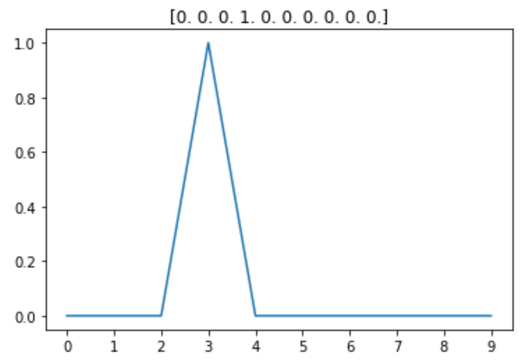
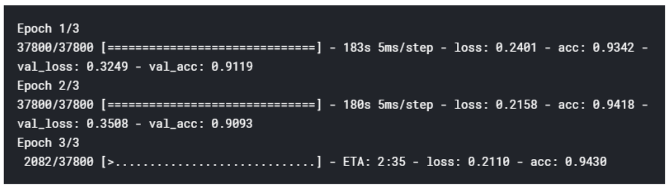


Fig. 5. Error percentage final readings



```
Epoch 1/3
37800/37800 [==============================] - 183s 5ms/step - loss: 0.2401 - acc: 0.9342 -
val_loss: 0.3249 - val_acc: 0.9119
Epoch 2/3
37800/37800 [==============================] - 180s 5ms/step - loss: 0.2158 - acc: 0.9418 -
val_loss: 0.3508 - val_acc: 0.9093
Epoch 3/3
 2082/37800 [>.............................] - ETA: 2:35 - loss: 0.2110 - acc: 0.9430
```

Fig. 6. final accuracy reading

VII. CONCLUSION AND FUTURE WORK

Due to lack of time, only the recognition of English and Hindi characters in manuscript and print is implemented. Hindi character recognition is partially implemented without being optimized. In addition, the current segmentation technique can also detect characters with clear boundaries. The algorithm and classification technique used recognizes characters regardless of the quality of the background or patterns. If two consecutive characters touch each other, they will not be treated as one character. In addition, when testing images collected at random, those that are written very large and fine also perform well in the classification. After scanning and resizing, the image lost a lot of pixels during the reduction process. Therefore, to update the developed system and resolve some of the persisting issues, the following tasks are recommended:

- Optimize image processing to maintain the original image characteristics as much as possible, regardless of the size, thickness, and angle of the character.
- Look for better segmentation techniques to enable successful separation of touching characters. (You may want to consider setting a fixed width when searching for individual letters)
- Introduction of character recognition for different languages, including French, Spanish, etc.
- Continue to use the extended version of algorithms and database creation techniques that will allow us to recognize characters. Depending on the bold text, the table will be created in the database with the name of the columns in bold and the insertion of the text mentioned entries.
- Introduce a differentiation based on the type of recognized character that can be used for sorting physical documents on a particular basis.
- Font creation if the recognized font does not match the mentioned dataset fonts.

REFERENCES

[1] Introduction to Document Analysis and Recognition Simone Marinai University of Florence Dipartimento di Sistemi e Informatica (DSI) Via S. Marta, 3, I-50139, Firenze, Italy-marinai@dsi.unifi.it

[2] Implementation of Character Recognition using Hidden Markov Model Karishma Tyagi, Vedant Rastogi Department of Computer Science & Engineering, IET, Alwar, Rajsthan-273010, U. P., INDIA

[3] On the Recognition of Handwritten Math Equations Quan Nguyen, Maximillian Wang, Le Cheng Fan December 12, 2014

[4] Application of Neural Network In Handwriting Recognition Shaohan Xu, Qi Wu, and Siyuan Zhang Stanford University 353 Serra Mall Stanford, CA 94305 USA

[5] Ko Y, aidya NHV. Location-aided routing (LAR) in mobile ad hoc networks. Proc. The ACM/IEEE International Conference on Mobile Computing and Networking, 1998. 66{75}.

[6] Ma XL, Sun MT, Zhao G, et al. An efficient path pruning algorithm for geographical routing in wireless networks. IEEE Trans. Vehicuar Technology, 2008, 57(4): 2474{2488}.

[7] Kim Y J, Govindan R, Karp B, et al. Geographic routing made practical. Proc. the 2nd Symposium on Networked Systems Design and Implementation, 2005. 217{230}.

[8] Wang, J., Jean, J.: Segmentation of merged characters by neural networks and shortest path. Pattern Recognition 27(5) (1994) 649–658

[9] You, D., Kim, G.: An approach for locating segmentation points of handwritten digit strings using a neural network. In: Int'l Conference on Document Analysis and Recognition. (2003) 142–146

[10] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11) (1998) 2278–2324

[11] Suzuki, M., Uchida, S., Nomura, A.: A ground-truthed mathematical character and symbol image database. In: Int'l Conference on Document Analysis and Recognition. (2005) 675–679

[12] Marti, U., Bunke, H.: A full english sentence database for off-line handwriting recognition. In: Int'l Conference on Document Analysis and Recognition. (1999) 705–708

[13] Liu, Cheng-Lin, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques.". Pattern Recognition. 37.2 (2004): 265-79. Web

[14] Ciresan, Dan, Ueli Meier, and Jurgen Schmidhuber. "Multi-column Deep Learning Neural Networks.

[15] Image Classication.". IEEE Conference on Computer Vision and Pattern Recognition. (2012): 3642-649 IEEE. Web.

[16] J.Pradeep, E.Srinivasan, and S.Himavathi, Diagonal Based Feature Extraction For Handwritten Character Recognition System Using Neural Network

[17] O. Matan, J. Bromley, C. J. Burges, J. S. Denker, L. D. Jackel, Y. LeCun, E. P. Pednault, W. D. Satterfield, C. E. Stenard, T. J. Thompson, Reading Handwritten Digits: A Zip Code Recognition System

[18] Marinai, S., Gori, M., Soda, G.: Artificial neural networks for document analysis and recognition. IEEE Transactions on PAMI 27(1) (2005) 23–35

[19] Ha, T., Bunke, H.: Image processing methods for document image analysis. In: Handbook of character recognition and document image analysis. World Scientific (1997) 1–47

[20] Watanabe, T., Luo, Q., Sugie, N.: Structure recognition methods for various types of documents. MVA 6(6) (1993) 163–176.