



**HAL**  
open science

# Recognition of Handwritten Characters Using Google Fonts and Freeman Chain Codes

Alexiei Dingli, Mark Bugeja, Dylan Seychell, Simon Mercieca

► **To cite this version:**

Alexiei Dingli, Mark Bugeja, Dylan Seychell, Simon Mercieca. Recognition of Handwritten Characters Using Google Fonts and Freeman Chain Codes. 2nd International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE), Aug 2018, Hamburg, Germany. pp.65-78, 10.1007/978-3-319-99740-7\_5 . hal-02060035

**HAL Id: hal-02060035**

**<https://inria.hal.science/hal-02060035v1>**

Submitted on 7 Mar 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Recognition of Handwritten Characters Using Google Fonts and Freeman Chain Codes

Prof. Alexiei Dingli<sup>1</sup>, Mark Bugeja<sup>2</sup>, Dylan Seychell<sup>3</sup>, and Dr. Simon Mercieca<sup>4</sup>

University of Malta, Msida, Malta

**Abstract.** In this study, a unique dataset of a scanned seventeenth-century manuscript is presented which up to now has never been read or analysed. The aim of this research is to be able to transcribe this dataset into machine readable text. The approach used in this study is able to convert the document image without any prior knowledge of the text. In fact, the training set used in this study is a synthetic dataset built on the Google Fonts database. A feed forward Deep Neural Network is trained on a set of different features extracted from the Google Font character images. Well established features such as ratio of character width and height as well as pixel count and Freeman Chain Code is used, with the latter being normalised using Fast Fourier Normalisation that has yielded excellent results in other areas but never been used in Handwritten Character Recognition . In fact, the final results show that this particular Freeman Chain Code feature normalisation yielded the best results achieving an accuracy of 55.1% which is three times higher than the standard Freeman Chain Code normalisation method.

**Keywords:** Handwritten Character Recognition · Machine Learning · Deep Learning

## 1 Introduction

Handwritten Character Recognition (HCR) converts a set of segmented characters into digital format. Most HCR techniques make use of Machine Learning models as well as computer vision to accomplish this task. The approach used in this study is a deep learning approach where a unique data set, built using Google Fonts is initially created. A number of different variations of the different fonts were generated in order to further increase the dataset using augmentation techniques. This step is necessary as early studies have shown that Machine Learning models tend to over-fit particular features, such as placement of the character in the image, rotation and thickness of the generated character. The final documents are then processed to extract various features. The feature set uses a variety of features that are extracted from the training set and then passed as an input to a Deep Neural Network (DNN) model that uses a variation of hidden layers. This model in turn is able to use different variables extracted from the feature set to develop a probabilistic classifier that is able to classify various characters within the document images. The Google Fonts dataset was created in

order to make use of the variations found in different fonts to mimic variations in handwritten documents. Thus, we can create a more robust model that is able to transcribe documents written in different handwriting styles that were not used to train the model. Most models make use of some of pre-labelled documents as a training set. It is very time-consuming process to manually transcribe a subset of documents, and even more time consuming to transcribe the whole set. Furthermore, it is not always the case that there are sufficient character-images in hand that can be used to train a Deep Learning Model.

### 1.1 The Manuscript

The dataset referred to as the manuscript is a 512-page document written in the 17th century. It is an unofficial history of the Knights of Malta written by Salvatore Imbroli. There is no known digital analysis of the said manuscript. Analysing it from a digital point of view requires the overcoming of a number of hurdles related to the script, ink and paper used at the time. Problems arising from the manuscript include small variations in writing style and the quality of the scans themselves. In this paper we are not addressing the challenge of segmenting the documents and evaluating the character recognition on the segmented characters. That challenge is addressed in a dissertation researched by Dylan Seychell. The final evaluation results used are a set of manually segmented characters selected from 25 randomly selected document images. The manuscript was provided by the National Library of Malta, thanks to the collaboration established by dr. Simon Mercieca with Ms. Maroma Camilleri. The research and transcription of specimen pages from this manuscript was done by Dr. Simon Mercieca with the collaboration of students following the history course at the Faculty of Arts of the University of Malta.

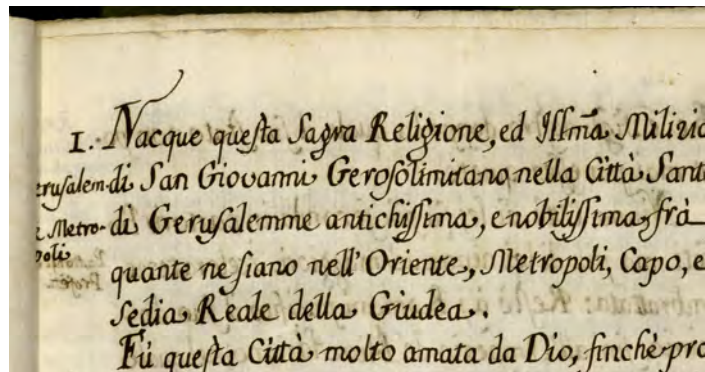


Fig. 1. Partial document image taken from the Manuscript

## 2 Background

### 2.1 Introduction

Artificial Neural Networks have been successfully used in a number of HCR models. In particular, supervised learning models have yielded excellent results. In the following section a number of models used in HCR systems are evaluated including Feed Forward Networks, Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN).

### 2.2 Feature based models

One of the most important components of Machine Learning is the feature set used for training. Depending on the number of features and the quality of the features, the models accuracy is increased. Moreover, the larger the dataset and more varied the dataset, the more generalized the model becomes. Some of the features used in handwritten recognition include the following

**Diagonal feature extraction scheme:** Each individual character image is resized to 90x60 pixels it is then further divided into 10x10 pixel bins. Features are extracted from the pixels for each bin by going through each diagonal for all available bins. A set of 54 features are extracted using this scheme [14].

**Contour feature scheme:** Character images are converted to a binary image. The contour of the image is extracted. Algorithms such as contour analysis can be used for this step. Each contour point is now a feature. For a given character there might be multiple contour points. When using such a scheme it is not always the case that for each image the same number of contour points are extracted. Thus, the feature set is normalized the gain the same set of features for training using Artificial Neural Networks. In addition, direction of pixels can also be used as part of the contour feature scheme feature set [9].

**Freeman Chain Codes scheme:** Freeman chain code algorithms go through the edge of a character image and extract the direction each pixel takes with respect to the contour of the shape of the character image. This scheme has yielded excellent results with accuracy going over 90%. As in the contour feature scheme the extracted freeman chain codes need to be normalised [13].

**Curvlet Transform based feature scheme:** The authors of this [18] noted the importance of handwritten text orientation written by the writer. Based on the needle shaped elements from the edge along the curves of the text. The extracted elements contain the directional sensitivity found in smooth contours. Character images are resized to a standard width and height. The Curvlet feature coefficients is extracted from each document character image. Apart from the above features there are a number of features that are considered standard in feature extraction. Including, character image width, height, centroid of the character

as well as black pixel count.

### 2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a class of feed forward networks. These networks have been applied and have achieved excellent results in areas of image recognition. CNN use the concept of convolution in order to create an architecture using several layers of convolution and nonlinear activation functions [17] [10]. One of the most early convolutional network was pioneered by Yann LeCun in 1988 [12]. The resulting architecture called LeNet was used for character recognition for zip codes and numerical digits. What makes CNN attractive is the fact that no features apart from the images are used when training the network. In most of the other networks explored the quality and accuracy of the neural net is depended on the features extracted. The more information can be gained from the features the better the accuracy as well as the generalisation of the model. Whilst this is desirable most of the time we do not know what features work well or what we want to extract. CNN accept images as inputs and extract features through leveled steps of convolution. Each layer in the network performs a simple computational function and the result is fed to the subsequent layer with the final result fed to the classifier. The computational process done by each layer is achieved using back-propagation. This process specifies that for each variable, the difference in the classification loss with respect to that parameter is computed and the parameters are updated with the goal of minimizing the loss function. Simard et al developed a CNN for handwritten digit recognition [17]. Their approach maximizes on the potential of this network by augmenting the dataset. Dataset augmentation is the process of adding distortion to the dataset to improve the generalization obtained by the neural net. The MNIST dataset was used for training and testing. The dataset was split into 60000 images for training and a further 10000 for testing. The overall architecture described makes use of two subsequent convolutional layers with a kernel size of 5x5 and two fully connected layers. The first fully connected layer uses 100 hidden units for training and the final classification layer is made up of 10 nodes representing the 10 classes used to classify digits. The authors also describe than the first fully connected layer was varied in size according to the number classification used. In the case of handwritten Japanese characters the number of hidden nodes was changed two 400 units for optimal results. By using the distorted dataset, they managed to obtain an overall error of 0.4% which was considered as state of the art at the time. Deformed training set is also used in the CNN proposed by Ciresan et al [3]. The architecture of the model uses an input layer of images size 29x29, where the original images are resized from 128x128 to 20x20 size image and centred over a 29x29 blank image. A convolutional layer with a 4x4 kernel followed by a max pooling layer with kernel size of 2x2 connected to a 9x9 kernel convolution layer and connected to a final max pooling layer with kernel size of 3x3. The final fully connected layer made up of 150 nodes and a final classification layer with a varying number of nodes

depending on the dataset used. 62 classes for the NIST SD dataset and 10 for the MNIST dataset. The dataset is distorted at the beginning of every epoch iteration using an elastic deformation with a variable value of  $\alpha = 6, 36$  as well as a vertical and horizontal scaling of range from 0 to 15% and distorted by 15%. An error of 11.8% is reported when classifying uppercase and lowercase letters. The presented architecture is able to classify around 10,000 characters per second. The above implementations trained CNN to classify English handwritten text. The method proposed by Rahman et al is applied to handwritten Bangla characters [15]. The dataset used contains 20000 handwritten characters with 400 character images representing each character of the Bangla alphabet. The input player of the network accepts grey scale image with dimension of 28x28 pixels. The architecture of the network used includes 2 convolutional layers using a 5x5 kernel, 2 max pooling layers using a 2x2 kernel a fully connected layer with 192 nodes and the final output layer made up of 50 nodes representing all the letters in the Bangla alphabet. The dataset was split into 17,500 character images for training and 2500 character images for testing. The overall accuracy of the system obtained was that of 85.36% accuracy. The authors noted that although the results were promising they did not compare to results achieved by other machine learning algorithms [15].

#### 2.4 Recurrent Neural Networks

Unlike feed forward networks, RNN do not form a forward connected cycle of nodes. In fact, using this architecture the network uses the parameters learned from the node as a new recurring input. Thus, being able learn how to classify character images from unsegmented handwritten text. This is due to a property in RNN's where the output is depended on the previous output. This memory like property records information on what has been computed beforehand and predicts the next output. In most Neural Network models input  $x$  passed to the network corresponds to some label  $y$  in a one to one relationship. Using RNN this behaviour changes [5]. In fact, an input could be a sequence of data which corresponds to either one output or another sequence of outputs with varying size. Ultimately the output is depended on all the history of the inputs that had been fed to the network beforehand. In most handwritten text recognition systems segmentation and transcription are two completely different components. In this paper, the authors propose a system that combines segmentation and transcription using RNN [6]. The system is built up of 7 layers. The input layer accepts images as input. The images might not necessarily be character images rather whole words or even sentences. These images are split into small zones and converted to a one-dimensional vector. The input layer is connected to a Multidimensional Long Short Term Memory layer (LSTM). A standard LSTM is made up of three distinct gates. The input gate, forget gate and the output gate that is connected to one RNN. In Multidimensional LSTM, the connections have been extended to  $n$  recurrent connections for each of the nodes previous states. The resultant output is converted back to the size of the original zone and fed to a feed forward network which uses a tanh activation function. This

process is repeated another two times up to which the final Multidimensional LSTM layer converts the output to a one-dimensional vector and transcribed to the Connectionist Temporal Classification layer. The latter output layer is specifically designed classification layer used in RNN where it transforms the output into sequence labeling and does not require pre-segmentation of labels or post processing to convert to transcription values. The training data used were a set of 1518 images with 120 distinct characters. The system achieved an accuracy of 96.75% Jameel et al, made some observations on what input can be fed to the network [8]. Instead of feeding images a set of features were extracted and used as input. Jameel et al argue that although curves, lines and intersections are intuitive features to extract the sequence in which they appear is also very important. Thus, shadow features were extracted by computing a sequence of values that depict what happens when scanning a character image and at what time curves and other features appear in the image. The sequence was then used as a feature set to input into the RNN. A back propagation neural network was used that is, a fully connected RNN. The training set was made up of 877 character images including uppercase and lower case letters. It took from 10,000,000 to 15,000,000 steps to train the network and achieve 91.4% accuracy [8].

The literature evaluated in this study established that a set of handcrafted features might give excellent results in the area of HCR. The adopted architecture used is a simple feed forward network which uses a substantial number of neurons and hidden layers. The features established from the literature used include Geometrical properties and Freeman Chain Codes.

### 3 Methodology

Some of the most successful techniques use Artificial Neural Networks (ANN) combined with a set of handcrafted features to recognise handwritten character images. Our approach differs in two ways. Instead of a shallow artificial neural network a deep neural network is used. The authors in [4] argue that the more hidden layers are used in a neural network the more accuracy can be gained. Consequently, a more varied dataset yields better generalisation. In the following section a detailed explanation of the training set used is presented as well as justification on the number of steps used for varying the dataset.

#### 3.1 Training Set

The aim of this study is to build a system that is able to automatically analyse handwritten text from the manuscript without any prior knowledge of the text. This implies that the classification model used needs to be robust enough to convert any character image into the corresponding ASCII value. This by no means is a trivial task. Many of the approaches undertaken by the authors in [14], [9], [13], [18] use the MNIST dataset to test and train the classifier but none of these approaches take into consideration approaches which test on a different dataset. Thus, it was decided to develop a new dataset computed from a number

of fonts. The font database used is the google font database found in google Google Fonts database . A dataset creation step generates a character image for every character and font found in the font database. In order to maximise generalisation a series of distortion steps are added to the creation stage this include rotation, scaling and moving the centre point of the character inside the image frame. Once a font style is loaded a random font size is selected. The selected range of font size varied this step is used to cater for characters with different widths and heights. This is an important step as variations in handwritten text might include variations in size of the same character. Thus, for example the letter a written by the same author might not always have the same dimension. The next step varies the position of the letter with respect to the character image. It is highly unlikely that the extracted characters in the character segmentation module are centred directly at the centre of the image. In fact, due to the scan line approach most of the time the character position varies in the character image window space. This step was therefore introduced so that the classification model does not overfit the classification label with respect to the character position and start classifying character images according to this unwanted feature. In an ideal scenario, the model trains on the shape of the character image. Due to the steps used in the feature extraction phase a number of pre-processing techniques are applied to the dataset. The images were binarised using the fast 2-D Otsu thresholding algorithm [20]. This technique converts the character image into a black and white image using a 2 dimensional histogram projected on the diagonal of the image. The image is then thinned with the Zhang Suen thinning algorithm [19] converting the stroke of the character to a pixel of depth. Thus, retaining only the skeleton shape of each character. These steps are used to optimise on the FCC extraction process. This step produced a total of 110,584 64x64 binarised character images.

### 3.2 Feature extraction

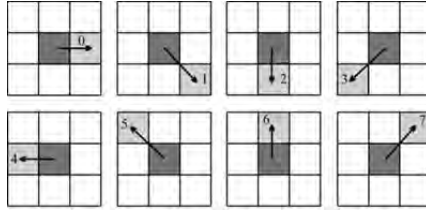
The second most important step in the classification module is the feature extraction. The overall accuracy of the classifier depends on the quality of the features extracted from the character images as well as the variety training set used. The following set of features compiled from the literature are used given the high accuracy reported by the authors using these features in [14], [9]. Furthermore, a number of experiments were undertaken using these features to find the best model to classify the handwritten characters. The experiments include variations in the feature schemes used as well as variation in the amount of training data and number of steps taken for the classifier to reach optimal performance. In the following section a description of the features and their corresponding extraction procedure is presented.

### 3.3 Freeman Chain Code

As mentioned throughout the paper FCC is an algorithm generally used to encode shapes in data compression. The algorithm lends itself well for edge



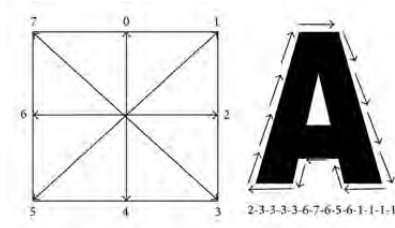
detection. Another property of FCC is that the resulting encoding is a chain of values corresponding to the change in direction at the edge of the pixel. This is a desirable training feature as irrespective of the character position the direction that the edge contour of a character shape has it will always remain the same. Couple with descriptive features such as the width and height of an image and the number of black pixels should lead to interesting results. In order to extract the FCC the character image is scanned pixel by pixel. Starting from the top left pixel moving towards the left direction. Once the initial transition is found, that is, finding the first black to white or white to black pixel change, then the start position is of the shape is noted. The transition value depends on whether the image background is black or white whilst the character has the opposite colour value.



**Fig. 2.** 8-Connectivity Direction Matrix used in FCC

From the start position the method looks at the neighbour pixels and searches for pixel value of the same colour as the start position pixel. The search is done clockwise starting from the perpendicular (top centre in matrix) position. In order for the adjacent pixel to be accepted as a candidate for the chain code it needs to be classified as a border pixel. This step prevents the algorithm from going through the area of the shape rather than the edge. The algorithm repeats this process until it creates a closed loop of the shape and returns back to the start position pixel. Although in our approach we use a clockwise moving direction the algorithm would still work if the directions are captured in an anticlockwise manner. The important rule is that only one direction is take to capture all features within the dataset to keep consistency [16].

The final output of this system is two separate data objects. One data object contains the chain code and the directions of the edge pixels whilst the other object contains the set of points traversed at the border. It is important to note that at this stage depending on the character shape and size, the dimensions of the chain set and the border position set varies. In most machine learning approaches as well as in the model used for this study the model will accept a fixed set size of features. Given this, normalisation techniques are applied to the data sets so that irrespective of the output size of the dataset the dimensions are always consistent. Furthermore, 8-connectivity matrix is used for feature extraction. This is due to the fact that more shape variations can be captured using this method.



**Fig. 3.** Result of Freeman Chain Code

### 3.4 Histogram Normalization

The chain code normalisation process is fairly simple. An array of 8 features is first created. For every value in the chain code set for the chain value that corresponds to a position in the array then the array value is incremented by 1. Effectively a histogram of values for the directions in the chain code is created.

### 3.5 Elliptic Fourier Feature Normalization

This normalisation technique is used to normalise the edge points of the character shape. Using elliptic Fourier features to describe a shape has been successfully applied to closed shapes in different research fields. The authors in [11] use four different co-efficient that represent each harmonic used to identify the closed shape. For a closed shape of  $k$  elements  $n$  harmonics are used. These co-efficients represent the major and minor projections of the x-axis and the y-axis. When this method is applied to the edge contour points a normalized set of 37 features is computed. Apart from the normalization advantage this method also produces a feature set that is invariant to any rotation, dilation or translation of the shape. Thus, irrespective of the shape position, stroke and rotation the same feature value is extracted each time. This property of elliptic Fourier feature normalisation is extremely desirable in the case of handwritten character recognition given that as outlined before variations in the testing set rotation or scale might vary the accuracy of the result. Such an implementation has been used and tested in different fields such as [2]. Up to the submission of this study no research has been submitted that use this normalisation for handwritten character recognition or compares the effects of using such procedure with respect to the histogram normalisation technique.

### 3.6 Geometrical Properties

The final set of features extracted from the character image describe the geometrical property of a character. Including the ratio of the width and height of a character image. The centroid of the character with respect to the image window as well as a count numbering the number of black pixels found in the character image. The latter is an important feature as through the thinning and dilation

normalisation process each character has the same stroke thickness. Thus, according to the shape and size of character an estimation of the number of black pixels can be computed for different character images falling under the same class.

### 3.7 Network Architecture

The adopted machine learning model is a simple feed forward neural network. This model has shown in different implementation of handwritten text recognition to yield excellent results. Our approach aims to increase the accuracy of the system by using a deep neural network. There is no formal definition for describing a neural network as a deep neural network. The authors in [4] attempt to classify a neural network as a deep network depending on the number of hidden layers. The overall accuracy achieved by the deep neural network is depended on the number of neurons available in the hidden layer as well as the number of hidden layers. The authors in [7] argue that by increasing the number of neurons in the hidden layer higher accuracy is obtained. On the other hand, Increasing the number of neurons in the hidden layer also increase the probability that over-fitting occurs.

## 4 Evaluation and Results

In this section, an evaluation of the character recognition module is presented.

### 4.1 DNN Models

The DNN Classification models are split into two main models. Each model is trained on a different feature set. The features include:

1. Freeman Chain Code normalized using Histograms referred to as FCC-HIST.
2. Character Contours normalised using Elliptic Fourier Analysis referred to as FCC-EFT.

All of the feature sets also contain structural data that includes width and height of the characters and number of black pixels found in the character image. The datasets are split 35000 character images and 100000 character Images. The testing set is 10% of the training set as in Table 1.

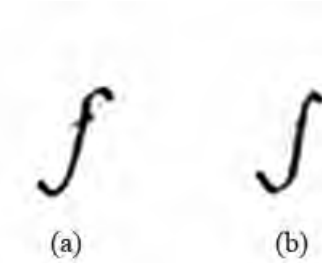
The models are further divided into two architectures. In order to choose the architecture and number of neurons to use an empirical process of elimination was used. A number of models were evaluated using different configurations and trained for 10,000 steps. The models that achieved the most promising results were then chosen for further training. The final two architectures used included a shallow model with 2 layers and a deep model with 6 layers. The first architecture is a shallow model containing two hidden layers with 64 and 32 hidden neurons respectively for each hidden layer. The second architecture is a deep architecture

consisting of 6 layers with 1024, 512, 256, 128, 64, 52 hidden neurons. Finally, each model is trained for three iterations of 100,000 steps and the accuracy is calculated from the average accuracy obtained on the same configuration model. In the following sections a presentation of the results obtained for these models is presented in Table 1 as well as a discussion on the results obtained.

**Table 1.** Results obtained when evaluating the models

Model Used	Accuracy (%) with training set of 35,000	Accuracy (%) with training set of 110,000
<b>FCC_HIST 2 Layers</b>	5.2	9.4
<b>FCC_HIST 6 Layers</b>	12.6	17.8
<b>FCC_EFT 2 Layers</b>	25.3	37.7
<b>FCC_EFT 6 Layers</b>	42.8	55.1

The overall accuracy of the system when classifying the handwritten characters segmented from the Manuscript is quite low. The best model was able to properly classify only 55.1% on the characters. A number of factors have contributed to this accuracy value. On inspection of the data although the characters were processed in the same manner as the testing set slight variations in the characters resulted in broken features or inconclusive feature extraction values. These include for example a chain code with only 3 values which is impossible as this would mean that the edge pixels of the character are made up of three directions only. In fact, when analysing the data, the characters which had a complete closing loop obtained better classification results with respect to the other characters with an increase in accuracy of 13%. Furthermore, characters such as the letter s was completely misclassified. This is mostly due to the way the letters were written in the manuscript. Figure 4 shows the letter s in various variations written in the manuscript.



**Fig. 4.** Comparison of letter 'f' (a) and 's' (b) found in the Manuscript

Even in today's writing style the character could easily be confused with an f. Another reason which contributes to the low accuracy obtained in the final testing phase of this study is that the model might have over trained on the testing set. More distortion and more character images would effectively result in better classifications. Moreover, other feature selection techniques need to be added such as zoning [14] and curvelet scheme [9].

Other approaches that might yield better results using different neural networks such as CNN and RNN. The latter were not chosen due to the requirement when used with handwritten text to have an initial labeled testing set. CNN were used at the start of the experimentation phase during the course of the research. The reason that the implementation was rejected is that although the training and testing results obtained were substantially better than the final models when tested on the actual Manuscript the classification accuracy was a bit better than random distribution.

## 5 Conclusion

The primary objective of this study is the creation of a database of characters that can be used as a training set for the Manuscript. The database was created and distortion effects were applied to the images to try and improve accuracy of the final text recognition component. Although the aim is met there is a lot more that needs to be done to consider such an objective complete. When developing the database, a lot of issues such as fonts that were too fancy to be used for training as well as problems with characters not having a shape that can be mapped along the edge as a complete loop resulted in a number of issues with the final accuracy of the system. A lot of manual intervention was needed to clean out the database. The final result was a dataset of 110,254 character images. Tests involving different number of machine learning models need to be computed in order to completely measure how effective the database is for handwritten text classification and this includes evaluation on other datasets such as the MNIST dataset. A large part of this study was spent researching on different implementations of machine learning and deep learning models for HCR. There is a lot of research in the area and the state of the art results are impressive to say the least. Most of the research is done on datasets that have been already identified and labeled such as the MNIST dataset and builds on a lot of models that have yielded excellent results, but, have not been implemented upon real world data. The approach chosen was built on some of the models discussed in the background [9] [13]. These models yielded excellent results using FCC feature extraction schemes coupled with contour analysis schemes and normalisation techniques using histograms. Our approach on the other hand uses elliptic Fourier features to normalise the contour features of a character. The latter normalisation technique is used in other fields but as of the time of the writing for this dissertation never featured in handwritten text recognition modules. The model adopting this scheme yielded the best results in combination with a deep learning architecture comprised of 6 hidden layers and 2036 hid-

den units. The best model obtained 55.1% accuracy after 100,000 epochs on the training set. A change in the configuration of the network might have yielded better results. The results obtained by the 2-layer Deep Neural Network was slightly less accurate that might prove the theory presented by Ba et al [1] that suggest that shallow networks are able to achieve the result as deep networks. The issue mostly lies with the current implementations used for shallow networks. Thus, with other classification models the model results might have yielded an improved accuracy. On the other hand, Goodfellow et al [4] provide empirical results that prove that the deeper a models architecture the better accuracy is yielded over time. More experiments on the configuration of the network might yield better results. Another approach that can be researched on is using RNN. This approach was rejected in the dissertation as it required a number of labeled documents matching the manuscript to build a supervised model. On the other hand, a similar approach undertaken in the study using the Google Font database can be used. Instead of a number of characters a document is created using text found in Italian literature. A number of different fonts are applied on the text and distortion applied on the text lines to create a more natural handwritten text effect. The model is then trained on these set of already labeled documents and evaluated on the system. The overall aim of creating a system that goes through the motions of converting a set of unseen handwritten character images to an ASCII representation has been met the results show that using FTT increases recognition accuracy by 37.3%.

## References

1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in neural information processing systems. pp. 2654–2662 (2014)
2. Ballaro, B., Reas, P., Tegolo, D.: Elliptical fourier descriptors for shape retrieval in biological images. In: Int. Conf. on Electronics, Control & Signal Processing. SG (2002)
3. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. pp. 1135–1139. IEEE (2011)
4. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
5. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: International Conference on Machine Learning. pp. 1764–1772 (2014)
6. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21, pp. 545–552. Curran Associates, Inc. (2009), <http://papers.nips.cc/paper/3449-offline-handwriting-recognition-with-multidimensional-recurrent-neural-networks.pdf>
7. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural networks 4(2), 251–257 (1991)
8. Jameel, A.: Experiments with various recurrent neural network architectures for handwritten character recognition. In: Tools with Artificial Intelligence, 1994. Proceedings., Sixth International Conference on. pp. 548–554. IEEE (1994)

9. Kimura, F., Shridhar, M.: Handwritten numerical recognition based on multiple algorithms. *Pattern recognition* **24**(10), 969–983 (1991)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp. 1097–1105 (2012)
11. Kuhl, F.P., Giardina, C.R.: Elliptic fourier features of a closed contour. *Computer graphics and image processing* **18**(3), 236–258 (1982)
12. LeCun, Y., Touresky, D., Hinton, G., Sejnowski, T.: A theoretical framework for back-propagation. In: *Proceedings of the 1988 connectionist models summer school*. pp. 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann (1988)
13. Nasien, D., Omar, F.S., Azmi, A.N., Yulianti, D.: Freeman chain code route length optimization using meta-heuristic techniques for handwritten character recognition (2015)
14. Pradeep, J., Srinivasan, E., Himavathi, S.: Diagonal feature extraction based handwritten character system using neural network. *International Journal of Computer Applications* (0975–8887) **8**(9), 17–22 (2010)
15. Rahman, M.M., Akhand, M., Islam, S., Shill, P.C., Rahman, M.H.: Bangla handwritten character recognition using convolutional neural network. *International Journal of Image, Graphics and Signal Processing* **7**(8), 42 (2015)
16. Shahab, W., Al-Otum, H., Al-Ghoul, F.: A modified 2d chain code algorithm for object segmentation and contour tracing. *Int. Arab J. Inf. Technol.* **6**(3), 250–257 (2009)
17. Simard, P.Y., Steinkraus, D., Platt, J.C., et al.: Best practices for convolutional neural networks applied to visual document analysis. In: *ICDAR*. vol. 3, pp. 958–962 (2003)
18. Singh, B., Mittal, A., Ansari, M., Ghosh, D.: Handwritten devanagari word recognition: A curvelet transform based approach. *International Journal on Computer Science and Engineering* **3**(4), 1658–1665 (2011)
19. Zhang, T., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* **27**(3), 236–239 (1984)
20. Zhu, N., Wang, G., Yang, G., Dai, W.: A fast 2d otsu thresholding algorithm based on improved histogram. In: *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*. pp. 1–5. IEEE (2009)