A Report on

OFFLINE HANDWRITING RECOGNITION USING DEEP LEARNING

B.Tech. PBL Report

Submitted by

Chetan Malhotra 1413101098

Under supervision of

Mr. John A.

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY

GREATER NOIDA, UTTAR PRADESH, INDIA

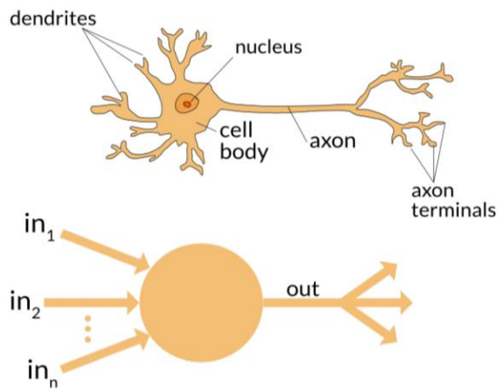
NOVEMBER, 2017

TABLE OF CONTENT

1. INTRODUCTION

1.1 Deep Learning

Before we delve into the details of Deep Learning, let's do a little review of what are neural networks and how they function. A neural network is a very powerful machine learning mechanism which basically mimics how a human brain learns. The brain receives the stimulus from the outside world, does the processing on the input, and then generates the output. As the task gets complicated multiple neurons form a complex network, passing information among themselves. Using an artificial neural network, we try to mimic a similar behaviour.



In this figure, we can clearly see the similarity between the neuron in our brain and an artificial neuron

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, partially supervised or unsupervised.

Deep learning architectures such as, deep belief networks deep neural networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation and bioinformatics where they produced results comparable to and in some cases superior to human experts.

Deep Learning in Artificial Neural Networks is about credit assignment across **many**(not just a few) subsequent computational stages or layers, in deep or recurrent NNs. The assumption underlying distributed representations is that observed data are generated by the interactions of layered factors. Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can provide different amounts of abstraction.

Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones.

Deep learning architectures are often constructed with a **greedy** layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features are useful for improving performance.

For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation.

Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabelled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks.

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modelling complex data with fewer units than a similarly performing shallow network.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back.

Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modelling. Long short-term memory is particularly effective for this use.

Convolutional deep neural networks(CNNs) are used in computer vision. CNNs also have been applied to acoustic modelling for automatic speech recognition (ASR)

1.1 Offline Handwriting Recognition

All the modern inventions in computer and communication technologies such as word processors, fax machines and e-mail are having their impact on handwriting. These in-variations have led to the fine-tuning and reinterpreting of the role of handwriting and handwritten messages. Despite these modern marvels, a pen together with a paper is much more convenient than a keyboard or a mouse. Computers that process handwritings will have to deal with many writing styles and languages, work with arbitrary user-defined alpha-bets, and understand any handwritten message by any writer. Several types of analysis, recognition, and interpretation can be associated with handwriting. Handwriting recognition is the task of transforming a language re-presented in its own spatial form of graphical marks into a symbolic representation. Handwriting interpretation is the task of determining the meaning of a body of handwriting, e.g., a handwritten address. Handwriting identification is the task of determining the author of a sample of handwriting from a set of writers. Identification and verification are processes that determine the special nature of the writing of a specific writer, while handwriting recognition and interpretation are processes whose objectives are to filter out the variations so as to determine the message. The task of reading handwriting is one involving specialized skills. A common complaint and excuse of people is that they couldn't read their own handwriting. So what chance does a computer have? Handwriting data is converted to digital form either by scanning the writing on paper or by writing with a special pen on an electronic surface. The two approaches are distinguished as offline and on-line handwriting(offline-handwriting recognition in my case), respectively. In the on-line case, the two-dimensional co-ordinates of successive points of the writing as a function of time are stored in order. In the off-line case, only the completed writing is available as an image. The recognition rates reported are much higher for the on-line case in comparison with the off-line case. Off-line systems are less accurate than on-line systems. However, thanks to deep learning they are now good enough that they have a significant economic impact for specialized domains such as interpreting hand-written postal addresses on envelopes and reading courtesy amounts on bank checks. The success of on-line systems made it attractive to consider developing, off-line systems that first estimate the trajectory of the writing from off- line data and then use on-line algorithms.

2. RECENT TRENDS

2.1 RECENT TRENDS IN DEEP LEARNING

As we know, Neural Networks have been here for a while now. To my knowledge, the ancient expression "Deep Learning" was introduced to the NN field by Aizenberg & Vandewalle. The field itself is much older though. The first Deep Learning systems of the feedforward multilayer perceptron type were created half a century ago (Ivakhnenko et al., 1965, 1967, 1968, 1971). The 1971 paper already described an adaptive deep network with 8 layers of neurons. (First general purpose recurrent Deep Learners were published much later, in 1991)

Recently the field has experienced a resurgence. Today, computing is almost a million times cheaper than it was in 1991. Today **graphics cards** or GPUs (mini-supercomputers for video games and high quality display graphics) are used to speed up learning on standard CPUs by a factor of up to 50. Thus, as of 2017 we don’t think twice before adding up more layers to our already deep neural network architecture.

2.2 RECENT TRENDS IN OFFLINE HANDWRITING RECOGNITION

Automatic and offline handwriting recognition is of academic and commercial interest. Current algorithms already excel at learning to recognize handwritten digits. Post offices use them to sort letters; banks use them to read personal checks. Some predict that in the near future billions of handheld devices such as cell phones will have handwriting recognition capabilities.

In recent decades new state-of-the-art results herald a renaissance of neural networks. Neither the fast and deep neural networks nor recurrent neural networks (also deep by nature) are limited to handwriting. They yield best known results on many visual and other pattern recognition tasks.

When the bubble of online handwriting recognition burst, the next mission to accomplish was offline handwriting. Many unsupervised and supervised learning techniques like SVM, etc were being used but Deep neural networks turned out to be producing the best results with max. error rate of only 0.35%. Handwriting in English language(roman script) was the first one to be recognized using Deep Neural Networks because of the huge database available .But recently, especially after 2010, if trends are to be followed, more and more training sets of different languages and scripts are being fed to learning algorithms. These languages include Mandarin/Chinese, Korean, Japanese, Urdu, Persian, Russian, etc. But as of 2017 no algorithm has been able to give as good results as that of English handwriting dataset, clearly due to shortage of training data.

3. PROJECT REQUIREMENT SPECIFICATIONS

3.1 Requirement Analysis

3.1.1 Requirements

Requirement for this project is based on the problem of not being able to use your handwriting as your Font. Today, we can use various algorithms to perform OCR on handwritten texts as well as printed text and recognize the text. But what to do with the recognized text? This is where my project kicks in. My app, MyFont uses the recognized characters and converts into a font just like Times New Roman, Comic Sans, etc.

3.1.2 Resources

The project development requires very few resources and can be done by a single person. It requires a good system to carry out test cases of the module, time for successfully building the app and testing it in real life scenario(different handwritings) will take about 2 to 3 months, and it requires the product to be tested physically with more and more handwriting samples to see how precise the app works in real life scenario and it will also give an idea if the project is reliable or not. This project as described requires less resources which decreases the total cost of development massively but will have a huge impact on the real life situation making it a very helpful and feasible project.

3.1.3 Proposed Work

Proposed work is to create an app(Android based for now) wiz MyFont.

MyFont is the ambitious idea and objective behind this project. MyFont is proposed to be an android(for now) based app that will provide you with the capabilities to use your handwriting as a font on your PC. This app is an application of offline handwriting using Deep neural nets. The app will allow the user to click a photo of the handwritten content which should contain all the alphabets, digits, symbols present in english language(0-9, A-Z, a-z, ?,/,!,@,#,$, etc). The photo will then be passed as an image to the Cloud. The pipeline including all the algorithms of character segmentation plus character recognition runs in the Cloud and each of the characters are recognized.

These characters are then mapped to the ASCII values the corresponding characters, so that it can be typed using keyword. You can download your font in .ttf format and use it with any word processing app like MS-Word, LibreOffice, etc

3.3 Project Objectives

The main objective of the project is to implement a system that uses your handwriting to produce font of the same and save the font for future purposes. You can store as many fonts of as many handwritings as possible and use it on any desktop by logging into your account and downloading whatever the font you want to use. An ideal account in your personal MyFont cloud storage looks like:

Serial No.	Font Name	Produced on	Operation
1	Chetan's Font	01/09/2017	Download/View
2	Chetan's Font-2	12/10/2017	Download/View
3	Raman's Font-4	03/12/2107	Download/View

4. MOTIVATION FACTOR

It is 2017 and domains like Machine Learning and Deep Learning have been seen to contribute to areas like Computer vision, Speech recognition, Text mining, etc. Same goes for offline handwriting recognition. There have been many algorithms and much research in areas of Online Handwriting recognition but it has been since 2009 that Offline Handwriting recognition has caught speed. Even in this domain, up till now document formats such as PDF, etc have been used to segment and recognize characters from. This is what lead to the idea of this project+app. The idea of recognition of what ever you write in the palm of your hand seems a little difficult but that is what drives and motivates me. Plus using that recognized characters as a font on your desktop or your laptop is pretty ambitious and enterprising in itself. Recent advances in the field of machine learning, deep learning, computer vision, etc and the need of these fields to produce better and better results with highly optimized results and minimized error rate is determining itself. As of now, there is no deep neural net architecture based app that provides us with the functionalities, MyFont proposes to delivers. Plus the accelerated hardware that we have today which makes thousands of parallel computations easier and faster makes you believe that Deep Neural Networks can really perform what they are admired for.

5. FEASIBILITY ANALYSIS

5.1 Technical Feasibility

Technically, it is very much possible as it requires servers and computers with high processing powers which are generally offered by almost every Cloud, a phone with normal camera capabilities and a desktop to use the font. After the app is developed, there will be hardly any maintenance cost for the app as it will all be a onetime investment and every camera enabled android device can run the app.

5.2 Economical Feasibility

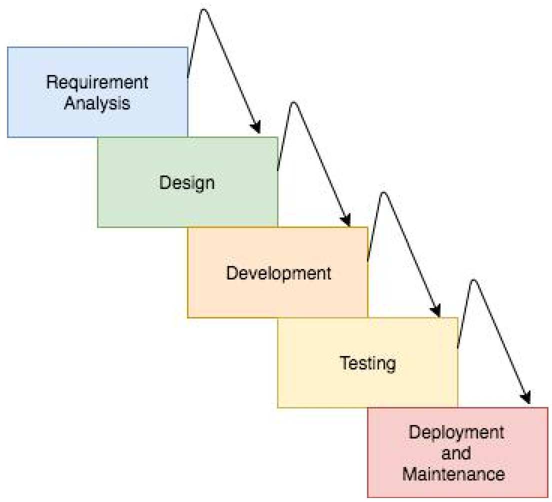
The cost of the total project from planning stage to deployment stage will be very less compared to its long-term benefits in the future to make lives more comfortable. The pricing of the app will be free of cost in the play store for every user. The training of neural networks in the cloud with the help of GPUs may cost a bit but that is Ok as I have $300 credits in Google Cloud and a student account in AWS which offers free credits for educational and research purposes.

5.3 Operational Feasibility

When the application is developed, it will help the users to save and use their handwritings as a font and store not only one, but multiple fonts which can be downloaded and used from any desktop by just logging into your account. This seems to innovate and spark many other ideas and uses that can be implemented using Offline Handwriting Recognition. Once the operations of recognition of characters written in Romans are fully functional, we can train the system to recognize other scripts like Hindi, Urdu, Russian, etc

6. WORKFLOW

6.1 Software Development Life Cycle



The classic Waterfall model is being used as the Software Development Life Cycle model. The waterfall model is a linear sequential (non-iterative) design approach for software development, in which progress flows in one direction downwards (like a waterfall) through the phases of Requirement analysis, Design, Development and coding, Testing, Deployment and Maintenance. I am currently in my Design phase and this report expresses the various UML diagrams that are used to define the project when it is in design phase. The previous report was regarding Requirement Analysis of the project.

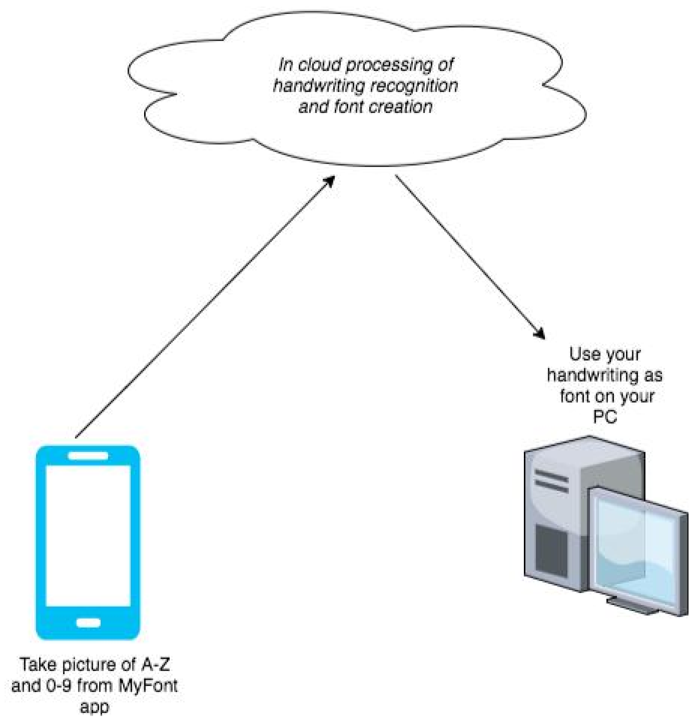
6.2 Design

UML is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems.

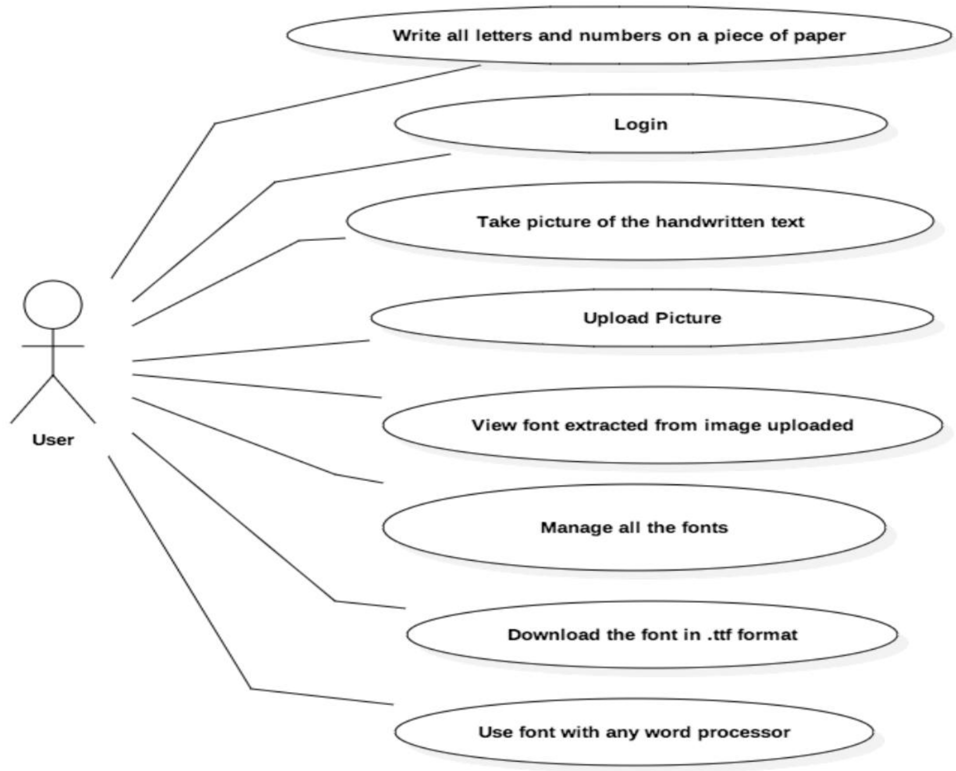
UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

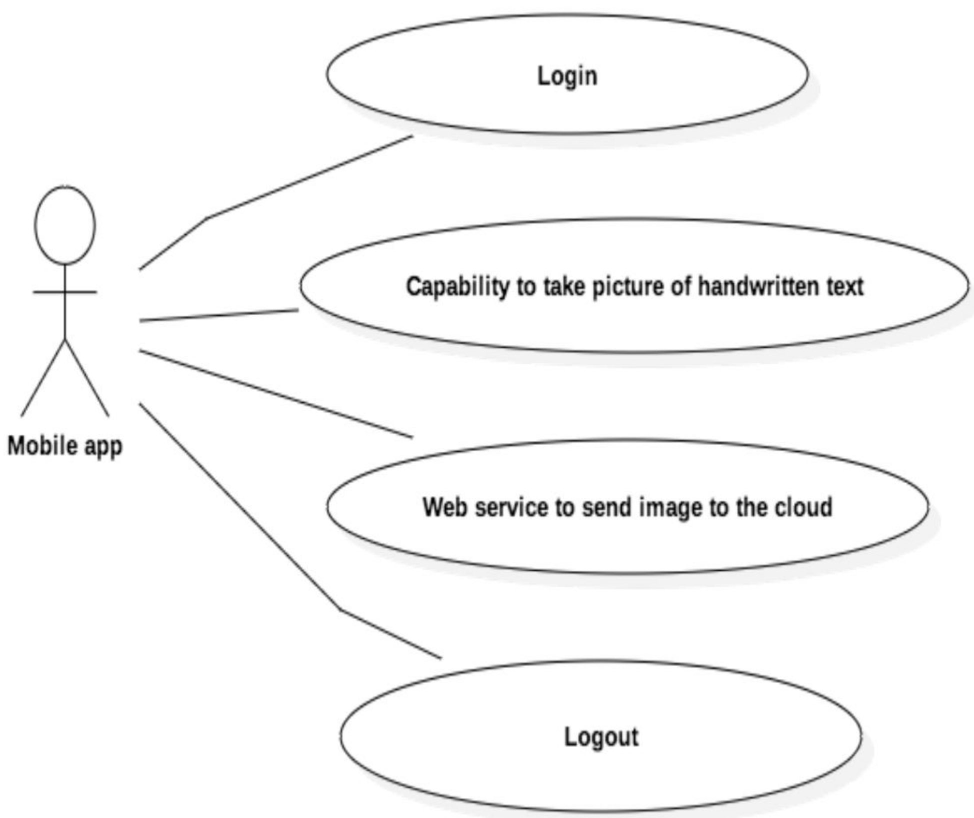
Various UML diagrams like Use-Case diagram, Activity Diagram, etc are used to define the design phase of a project and express the whole functionalities and working of the system through diagrams.

6.2.1 Over-view of the system



6.2.2 Use-Case Diagram





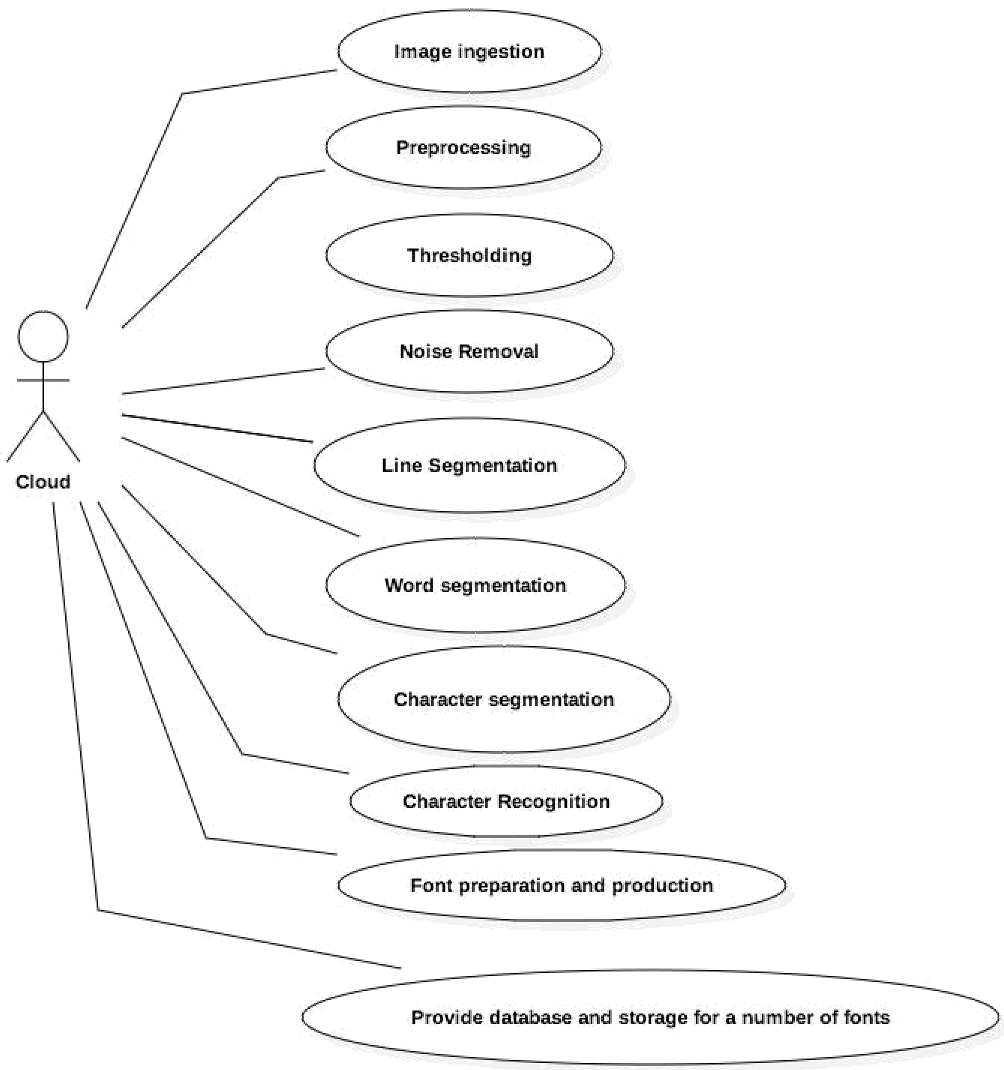
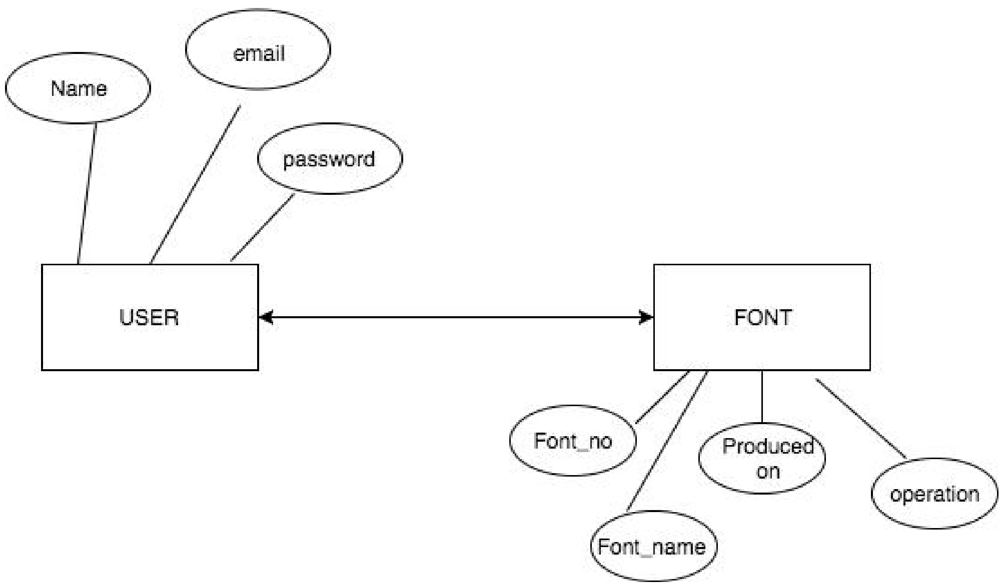
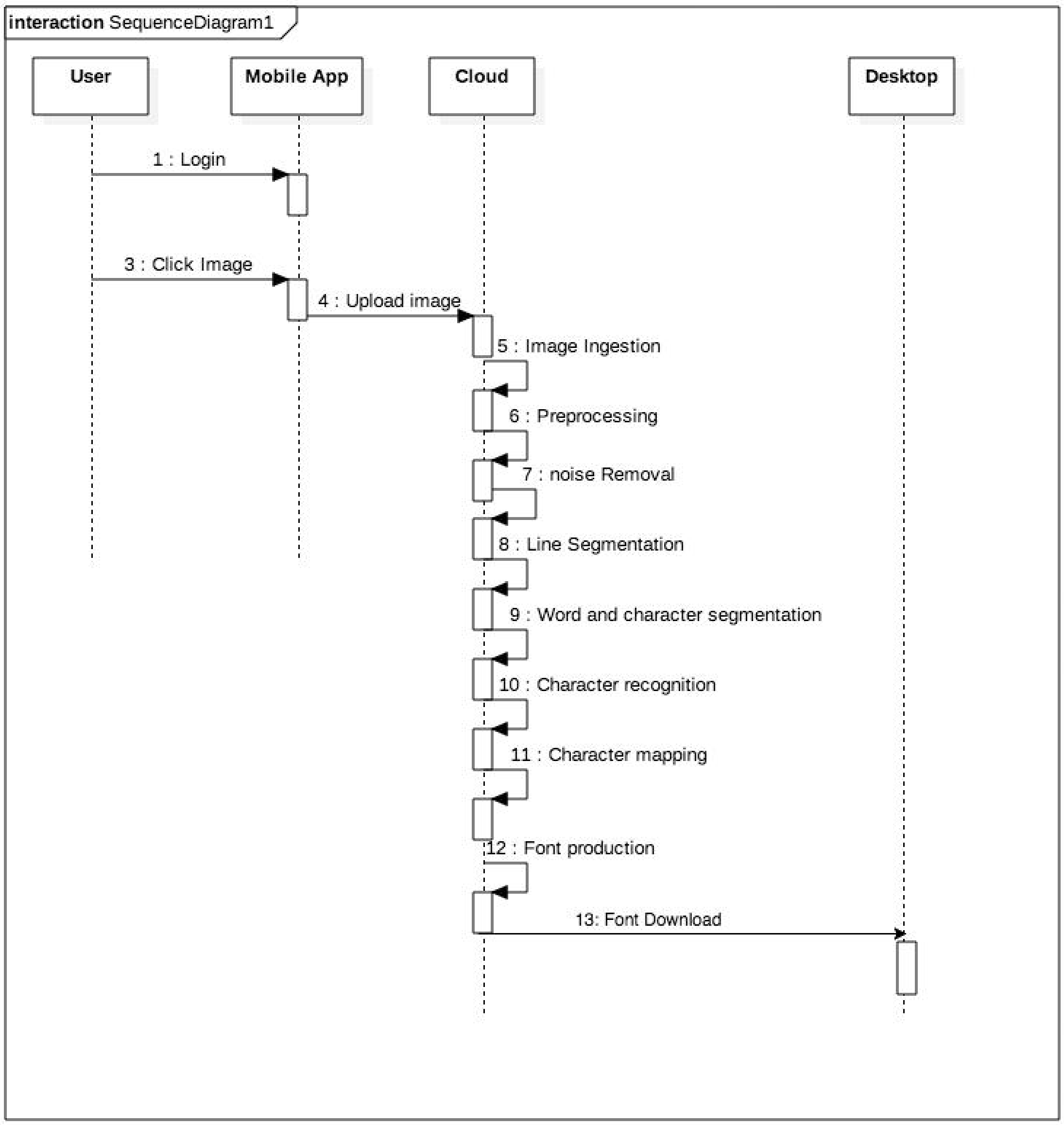


Image ingestion
Preprocessing
Thresholding
Noise Removal
Line Segmentation
Word segmentation
Character segmentation
Character Recognition
Font preparation and production
Provide database and storage for a number of fonts
Cloud

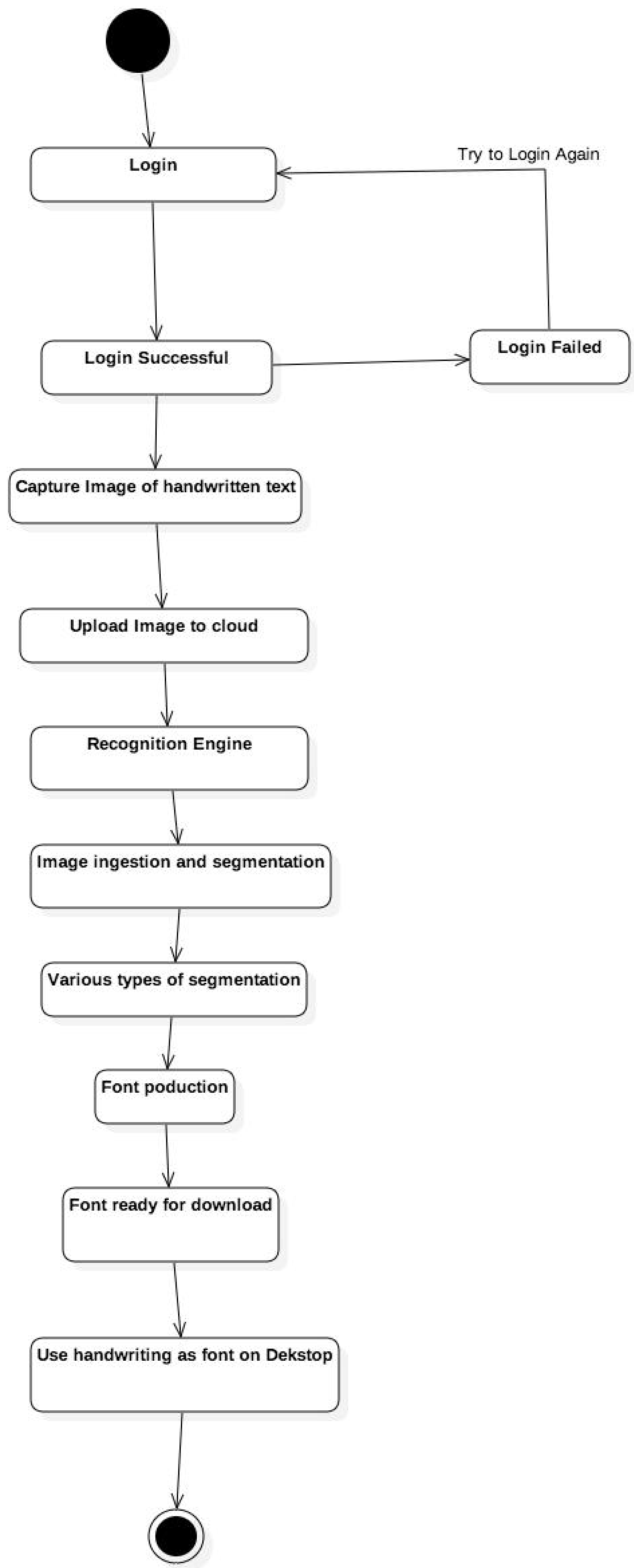
6.2.3 E-R Diagram



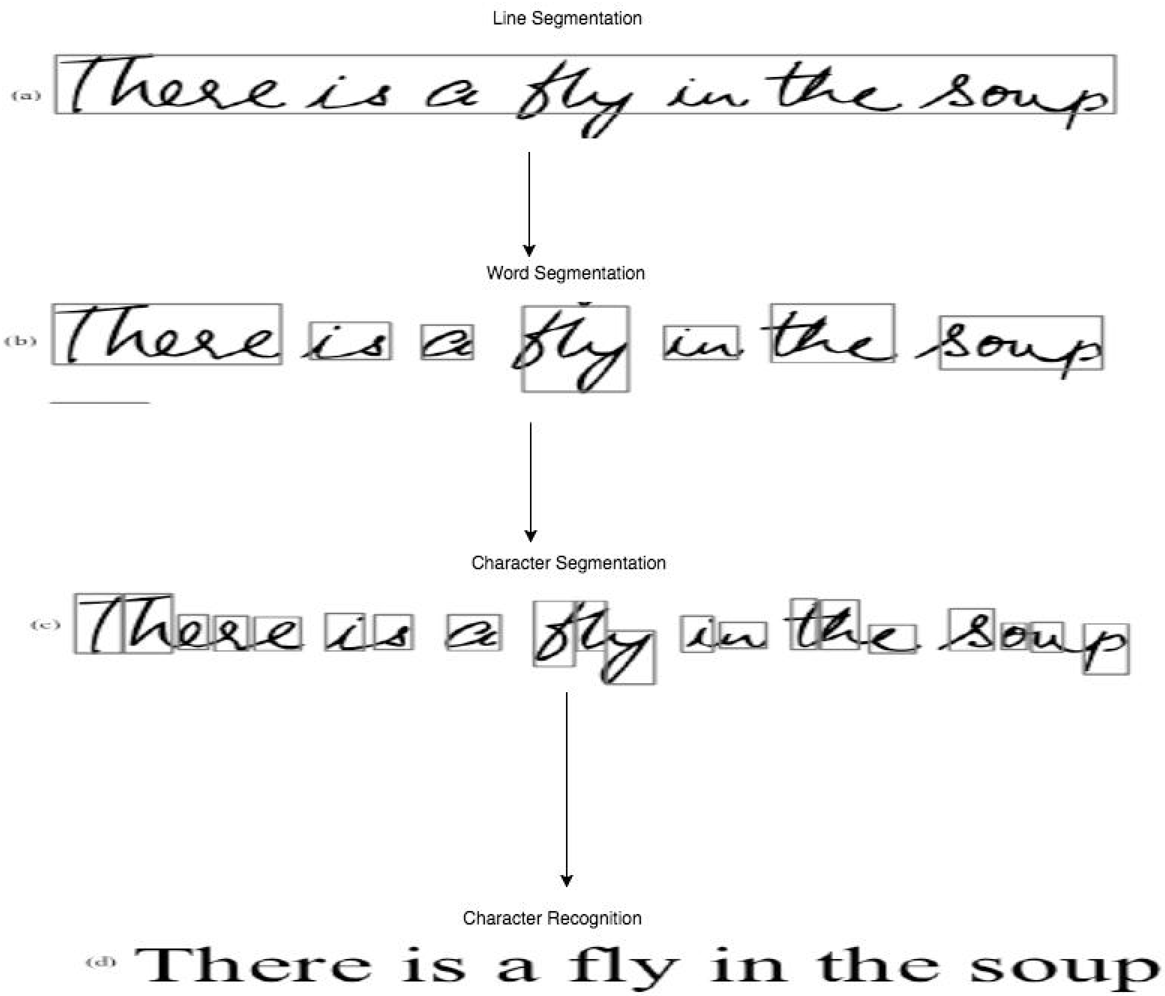
6.2.3 Sequence Diagram



6.2.4 Activity Diagram



6.2.5 Working of Recognition Engine

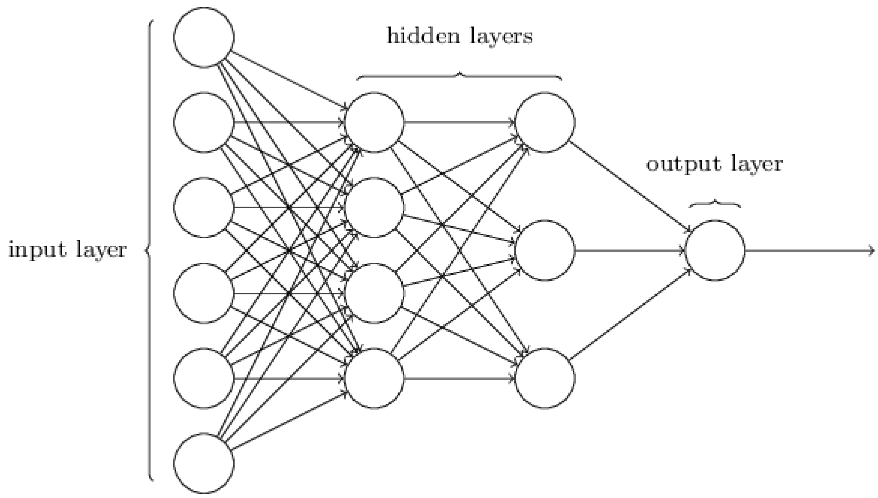


Img. Recognition Engine[1]

6.3.1 Core Implementation

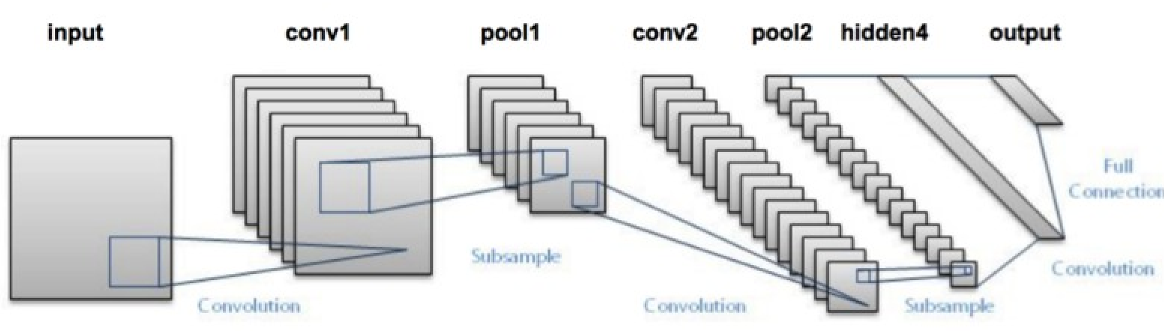
For this evaluation, I have implemented 2 models for handwritten digit recognition. Offline Handwritten digit recognition is an important module and subset of my recognition system and one of the classic problems that can be easily soled using Deep Learning but requires GPU for faster computing. The two models I have deployed are written using Keras and Jupyter Notebooks. The Two models are:

1. Multi-Layer Perceptron Feed Forward Neural Network:



A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

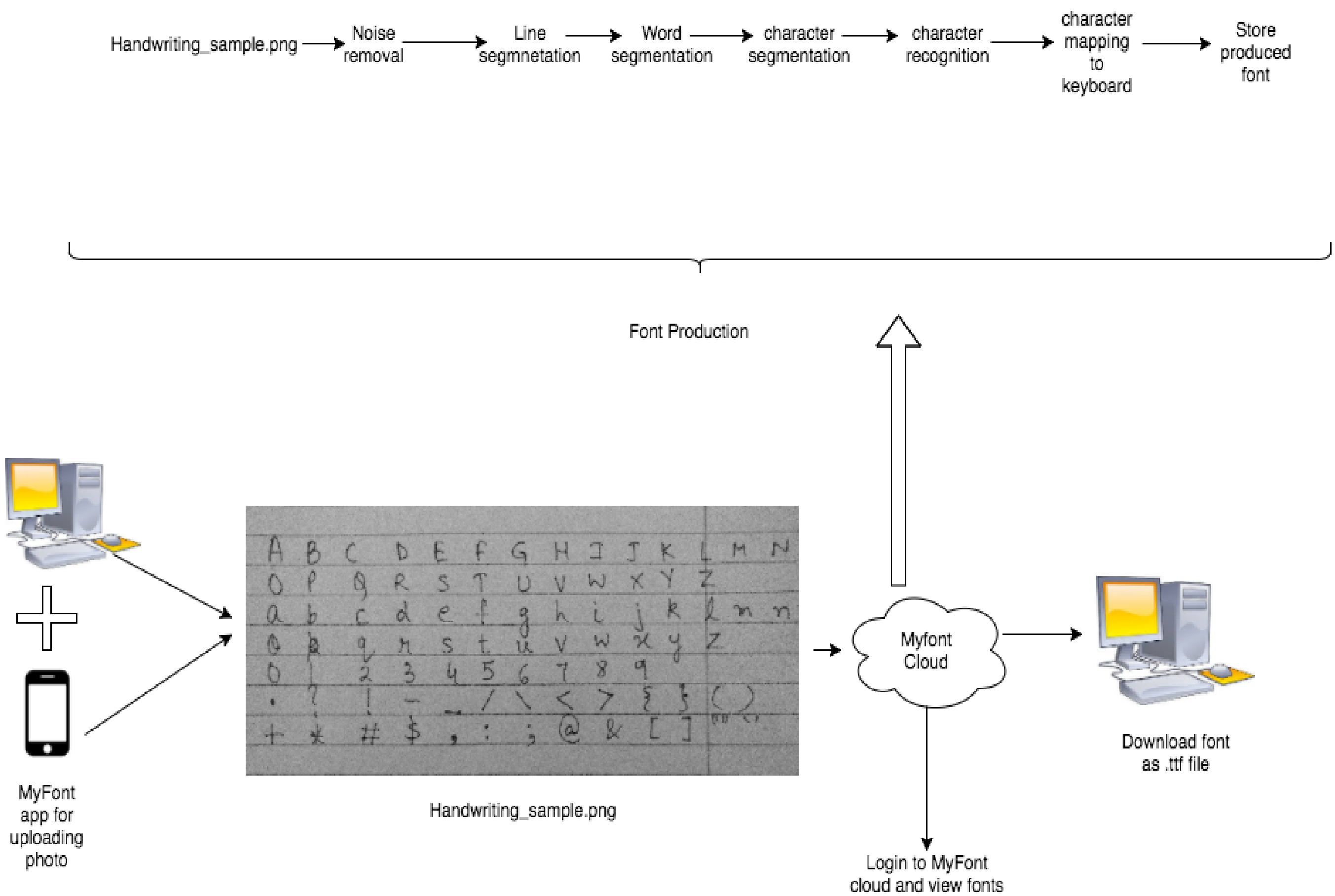
2. Convolutional Neural Networks:



Convolutional Neural Networks (CNN's) are multi-layer neural networks (sometimes up to 17 or more layers) that assume the input data to be images. By making this requirement, CNN's can drastically reduce the number of parameters that need to be tuned. Therefore, CNN's can efficiently handle the high dimensionality of raw images.

6.3.2 Synchronization of Design and Implementation

Though, as of now this report only consists of implementation of a handwritten digit recognition system, the whole design of the system and process of the system looks like:



7. System Requirements

7.1 Hardware

→ A MacBook Air with 8G RAM and Intel i5 processor for neural network training and development.

→ A Dell Vostro-1540 with Intel i3 and 4GB RAM for misc. development.

→ An android based smartphone with Camera. Moto X Play for testing purposes.

7.2 Software

7.2.1 Design Tools

→ StarUML

→ Draw.io

7.2.2 Development Tools

→ MyFont App: Since the app is android based, basic android modules and classes will be implemented in java with XML or UI will be used.

→ Cloud Platform: Google cloud engine or Floydhub with minimum of 4-core CPUs, GPUs will be added if needed since they can cost upto 300$/month.

→ Version control system: Git and GitHub

→ Continuous Integration using Travis-CI

→ **Deep Learning architecture**: All the main work including deep neural nets, training of algorithm, etc that can be stated as the brain of the app is implemented using:

1. Python: Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990.

2. Tensorflow: TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.

3. Keras: Keras is an open source neural network library written in Python. It is capable of running on top MXNet, Deeplearning4j, Tensorflow, CNTK or Theano.(Tensorflow in my case)

4. Scipy: SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering.

5. Scikit-learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

8. REFERENCES

1. Shashank Mathur, Vaibhav Aggarwal, Himanshu Joshi, Anil Ahlawat. “Offline Handwriting Recognition Using Genetic Algorithm” Sixth International Conference On Information Research And Applications – I.Tech 2008, Varna, Bulgaria, June-July 2008.

2. P. Shankar Rao, J. Aditya {Dept of CSE, Andhra University}. “Handwriting Recognition – Offline Approach ”

3. Liu, Xia, and Zhixin Shi. "A format-driven handwritten word recognition system."2013 12th International Conference on Document Analysis and Recognition. Vol. 2. IEEE Computer Society, 2003.

4. Http://People.Idsia.Ch/~Juergen/Handwriting.Html

5. Http://People.Idsia.Ch/~Juergen/Deeplearning.Html

6. https://machinelearningmastery.com

7. https://keras.io/

8. http://yann.lecun.com/exdb/mnist/