Abstract:

Convolutional Neural Networks(CNNs) have been confirmed as a powerful technique for classification of visual inputs like handwritten digits and faces recognition. Traditional convolutional layer's input feature maps are convolved with learnable kernel then combined for achieving better performance. The biggest drawback is that the combination of feature maps can lose features and do not apply well to large-scale neural networks. Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. Nowadays, convolutional neural networks have become a hot topic in machine learning community, showing significant gains over start-of-the-art machine learning methods used in various applications, like speech, image processing and sentence c1assification. Each application
requires higher accuracy performing closely to human by neural networks. More importantly, we need higher accuracy rate, even to one hundred percent, and faster training. To achieve these goals, some methods and tricks are proposed for CNNs. Deep big simple neural net without convolutional layers gets 0. 35% error rate . In order to avoid overfitting, elastic distortion was introduced to expand dataset and their simple network results in 0. 4% error rate. Multi-column deep neural networks average many networks' prediction and get 0. 23% error rate. Big networks with billions of parameters may easily overfit even with the largest datasets. compared the results of some of the most widely used Machine Learning Algorithms like SVM, KNN & RFC bandwidth. Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow. Using these, I was able to get the accuracy of 98.70% using CNN (Keras+Theano) as compared to 97.91% using SVM, 96.67% using KNN, 96.89% using RFC.

Table of Contents

List of Tables

List of Figures

Chapter 1:Introduction

1.1Introduction

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. A Convolutional Neural Network (CNN) is a type of feed- forward Artificial Neural Network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Convolutional Neural Networks consist of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product and optionally follows it with a non-linearity. The whole Convolutional Neural Network expresses a differentiable scorefunction that is further followed by a Softmax function.The data input into the Convolutional Neural Network is arranges in the form of its width, height and depth.The Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learning from existing artist styles etc.

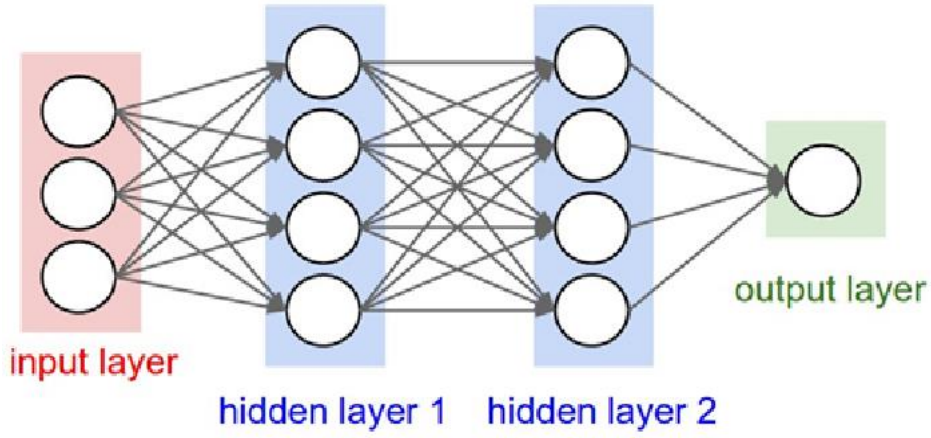


Fig 1.1 Convolutional Neural Network Basic Layout

1.2 Scope of the project

Nowdays, Convolutional Neural Networks (CNNs) have become an emergent topic in machine learning. It has lead its importance in various different applications such as music and speech recognition, image processing, and sentence classification to name a few. We need to achieve high accuracy rate close to human brain's neural network, to gain high accuracy as close to 100% as possible and minimization of the error rate. Various techniques have been proposed over the years to achieve these goals.

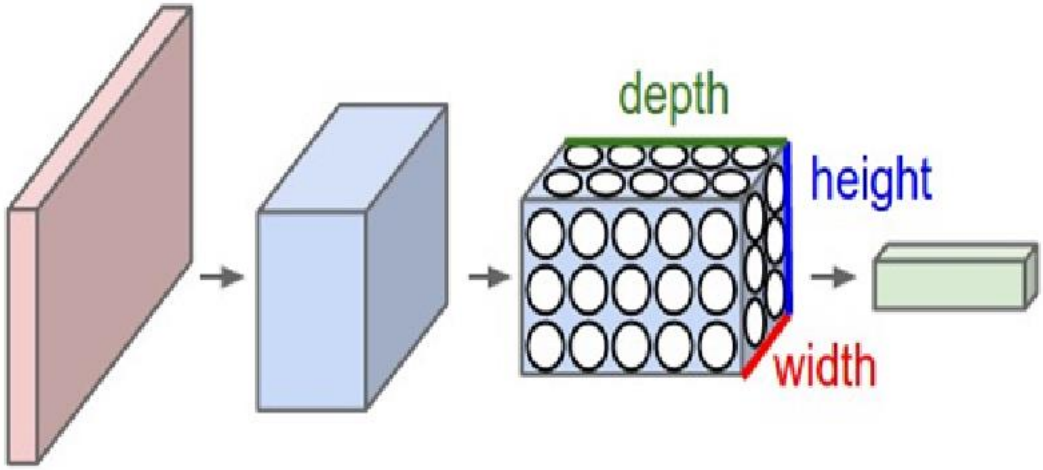


Fig 1.2 Arrangements of Neurons in CNN

Chapter 2: Literature Review

1. **Handwritten Digit Recognition Using Deep Learning**
   Anuj Dutt, Aashi Dutt

A lot of classification techniques using Machine Learning have been developed and used for this like K-Nearest Neighbours, SVM Classifier, Random Forest Classifier etc. but these methods although having the accuracy of 97% are not enough for the real world applications. One example of this is, if you send a letter with addressee name as "Anuj" and the system detects and recognizes it as "Tanuj" then it will not be delivered to "Anuj" but "Tanuj". Although eventually it may come to the right address but if the mail is important, this delay can cost a lot. In short, the accuracy in these applications is very critical but these techniques do not provide the required accuracy due to very little knowledge about the topology of a task. Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc.

2. **Handwritten Digit Recognition System Using No Combination of Feature Maps**
   Drashti Dobariya, Dhwani Prajapati

This paper introduced a replacement No combination of feature maps(NCFM) technique for abstraction of features. Every input feature map will generate an output feature map without combining, which lead to minimal data loss and achieve high precision rate. On analysis of results we can conclude that high performance can be achieved and that we can accomplish the acceptable precision rate on the dataset. The organization of the paper is as follows: Section II describes the outline of all the related previous research papers then follows introduction of NCFM technique to enhance the performance of the CNN. Section III then describes NCFM technique thoroughly and discusses the benefits. Section IV describes the architectures of the CNN with NCFM and CFM, that are used throughout section V. Section VI designs 3 experiments to verify the validity and analysis of the Ncfm technique.

3. **Accurate Handwritten Digits Recognition using Convolutional Neural Networks**
   Yan Yin, JunMin Wu, HuanXin Zheng University of Science and Technology of China

In this paper, we introduce a novel Ncfm (No combination of feature maps) technique to abstract a variety of features. Each input feature map can get a output feature map without combining, which can avoid information loss and get higher accuracy. Evaluation results show that performance can be improved and we achieve the state-of-the-art accuracy rate with 99.81 % on the MNIST datasets. Specifically we make the following contributions:
1. We propose CNNs with Ncfm technique to improve accuracy and speed up convergence. We reach the state-ofthe-art performance with 99.81% accuracy rate on the MNIST dataset.
2. We compare the Cfm(Combination of feature maps) technique with Ncfm technique and analyze the advantages. Firstly, Ncfm converges faster than Cfrn. Secondly, Ncfm performs better than Cfm with fewer convolutional filters.

Method	Accuracy	Description
Hand printed symbol recognition.[5]	97% overall.	Extract geometrical, topological and local measurements required to identify the character.
OCR for cursive handwriting. [6]	88.8% for exicon size 40,000.	To implement segmentation and recognition algorithms for cursive handwriting.
Recognition handwritten numerals based upon fuzzy model.[13-14]	95% for Hindi and 98.4% for English numerals overall.	The aim is to utilize the fuzzy technique to recognize handwritten numerals for Hindi and English numerals.
Combining decision multiple connectionist classifiers for Devanagari numeral recognition. [7]	89.6% overall.	To use a reliable and an efficient technique for classifying numerals.
Hill climbing algorithm for handwritten character recognition. [10]	93% for uppercase letters.	To implement hill climbing algorithm for selecting feature subset.
Optimization of feature selection for recognition of Arabic characters. [11]	88% for numbers and 70% for letters.	To apply a method of selecting the features in an optimized way.
Handwritten numeral recognition for six popular Indian scripts. [12]	99.56% for Devanagari, 98.99% for Bangla, 99.37% for Telugu, 98.40% for Oriya, 98.71% for Kannada and 98.51% for Tamil overall.	To find out the recognition rate for the six popular Indian scripts.

Fig. 2.1. Revolution in Neural Network

	RFC	KNN	SVM	CNN
Trained Classifier Accuracy	99.71%	97.88%	99.91%	99.98%
Accuracy on Test Images	96.89%	96.67%	97.91%	98.72%

Fig 2.2 Percent Accuracy of Each Classification Technique

Model	Test Error Rate
Random Forest Classifier	3.11%
K Nearest Neighbours	3.33%
Supervised Vector Machine	2.09%
Convolutional Neural Network	1.28%

Fig 2.3. Classifier Error Rate Comparison

Chapter 3 : Proposed System

3.1 Analysis of the proposed system

A. MNIST Dataset

The MNIST dataset, a subset of a larger set NIST, is a database of 70,000 handwritten digits, divided into 60,000 training examples and 10,000 testing samples. The images in the MNIST dataset are present in form of an array consisting of 28x28 values representing an image along with their labels. This is also the same in case of the testing images.

The data is stored in four files:

1. train-images-idx3-ubyte: training set images
2. train-labels-idx1-ubyte:training set labels
3. t10k-images-idx3-ubyte: test set images
4. t10k-labels-idx1-ubyte: test set labels

The Training Set Label file has the data in the following representation:

```
[offset] [type]          [value]          [description]
0000     32 bit integer  0x00000801(2049) magic number (MSB first)
0004     32 bit integer  60000            number of items
0008     unsigned byte   ??               label
0009     unsigned byte   ??               label
........
xxxx     unsigned byte   ??               label

The labels values are 0 to 9.
```

The Training Set Image file has the data in following representation:

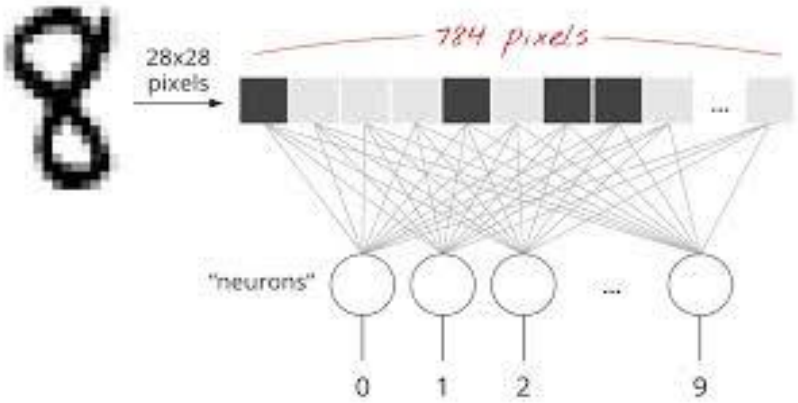
```
[offset] [type]          [value]          [description]
0000     32 bit integer  0x00000803(2051) magic number
0004     32 bit integer  60000            number of images
0008     32 bit integer  28               number of rows
0012     32 bit integer  28               number of columns
0016     unsigned byte   ??               pixel
0017     unsigned byte   ??               pixel
........
xxxx     unsigned byte   ??               pixel
```

B. MNIST Dataset Format Analysis

As you can see from above, the MNIST data is provided in a specific format. So, to be able to read the dataset it is first important to know that in what format the data is available to us. Both the Training and Testing images and labels have the first two columns consisting of the “Magic Number” and the number of items in the file.

The magic number has its first two bytes equal to zero. This magic number is read as MSB first and its format is as shown below:

2 Bytes	1 Byte	1 Byte
00	Data Type	Dimensions



Layers of Convolutional Neural Network

A CNN consists of a lot of layers. These layers when used repeatedly, lead to a formation of a Deep Neural Network. Three main types of layers used to build a CNN are:

1. **Input**: This layer holds the raw pixel values of image.

2. **Convolutional Layer**: This layer gets the results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5x5 window that slider over the input data and gets the pixel with the maximum intensity as the output.

3. **Rectified Linear Unit [ReLU] Layer**: This layer applies an element wise activation function on the image data. We know that a CNN uses back propagation. So in order to retain the same values of the pixels and not being changed by the back propagation, we apply the ReLU function.

4. **Pooling Layer**: This layer perform a down-sampling operation along the spatial dimensions (width, height), resulting in volume.

5. **Fully Connected Layer**: This layers is used to compute the score classes i.e which class has the maximum score corresponding to the input digits.

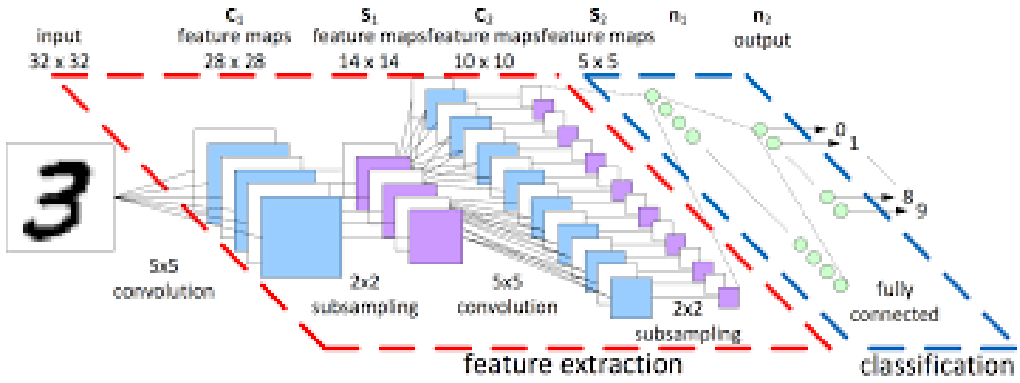


Fig 3.1 Feature Extraction For CNN

3.2 Design of the proposed System

Phases of the Convolutional Neural Network

The CNN for Handwritten Digit Recognition works in three main phases.

Phase1 - Input MNIST Data1: The first phase is to input the MNIST data. The MNIST data is provided as 784-d array of pixels. So firstly we convert it to grayscale images using 28x28 matrix of pixels.

Phase2 – Building Network Architecture: In the second phase, we define the models to be used to build a convolutional neural network. Here, we use the *Sequential* class from *Keras* to build the network. In this network, we have three layer sets of layers "*CONV =>ReLU=> POOL".*

a)First Convolution Layer: In the first layer, we take 20 convolutional filters that go as a sliding window of size 5x5 over all the images of 28x28 matrix size and try to get the pixels with most intensity value.

b)ReLU Function: We know that convolution is a method that uses *Back Propagation.* So using the ReLU function as the activation function just after the convolutional layer reduces the likelihood of the vanishing gradient and avoids sparsity. This way we don‟t lose the important data and even get rid of redundant data like a lot of 0‟s in the pixels.

c) Pooling Layer: The pooling layer gets the data from the ReLU function and down-samples the steps in the 3D tensor. In short it pools all the pixels obtained from previous layers and again forms a new image matrix of a smaller size. These images are again input into the second set of layers i.e. "CONV =>ReLU=> POOL" and this process goes on till we get to a smallest set of pixels from which we can classify the digit.

Phase 3 –Fully Connected Layer: The fully connected layer is used to connect each of the previous layers to the next layers. This layer consists of 500 neurons. Finally, we apply a Softmax Classifier that returns a list of probabilities for each of the 10 class labels. The class label with the largest probability is chosen as the final classification from the network and shown in the output. This output received is used to make the confusion matrix for the model. In this we can add more number of layers but adding more layers might affect the accuracy of the system. Since, it uses multiple layers, so it's called a Deep Learning system.

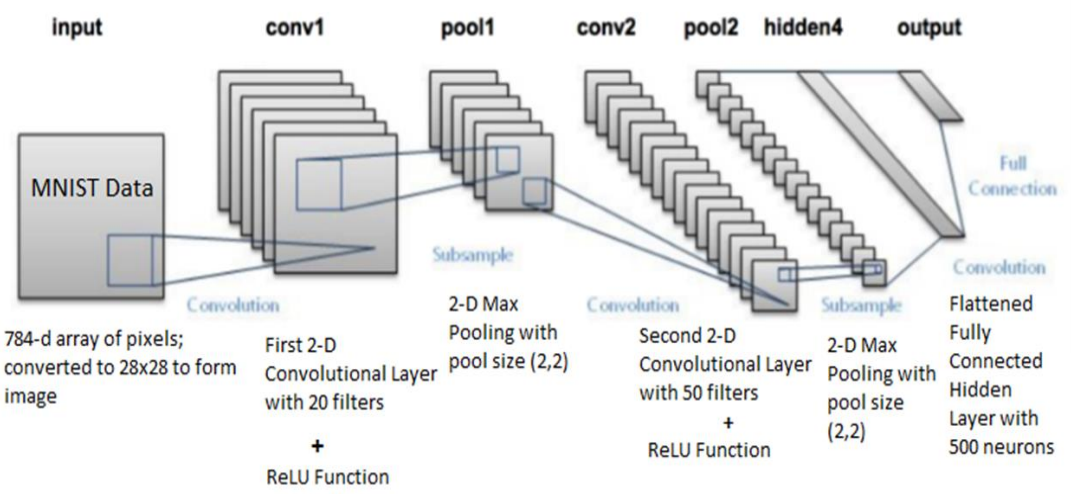


Fig 3.2 Phases of Convolutional Neural Network

Flow Chart

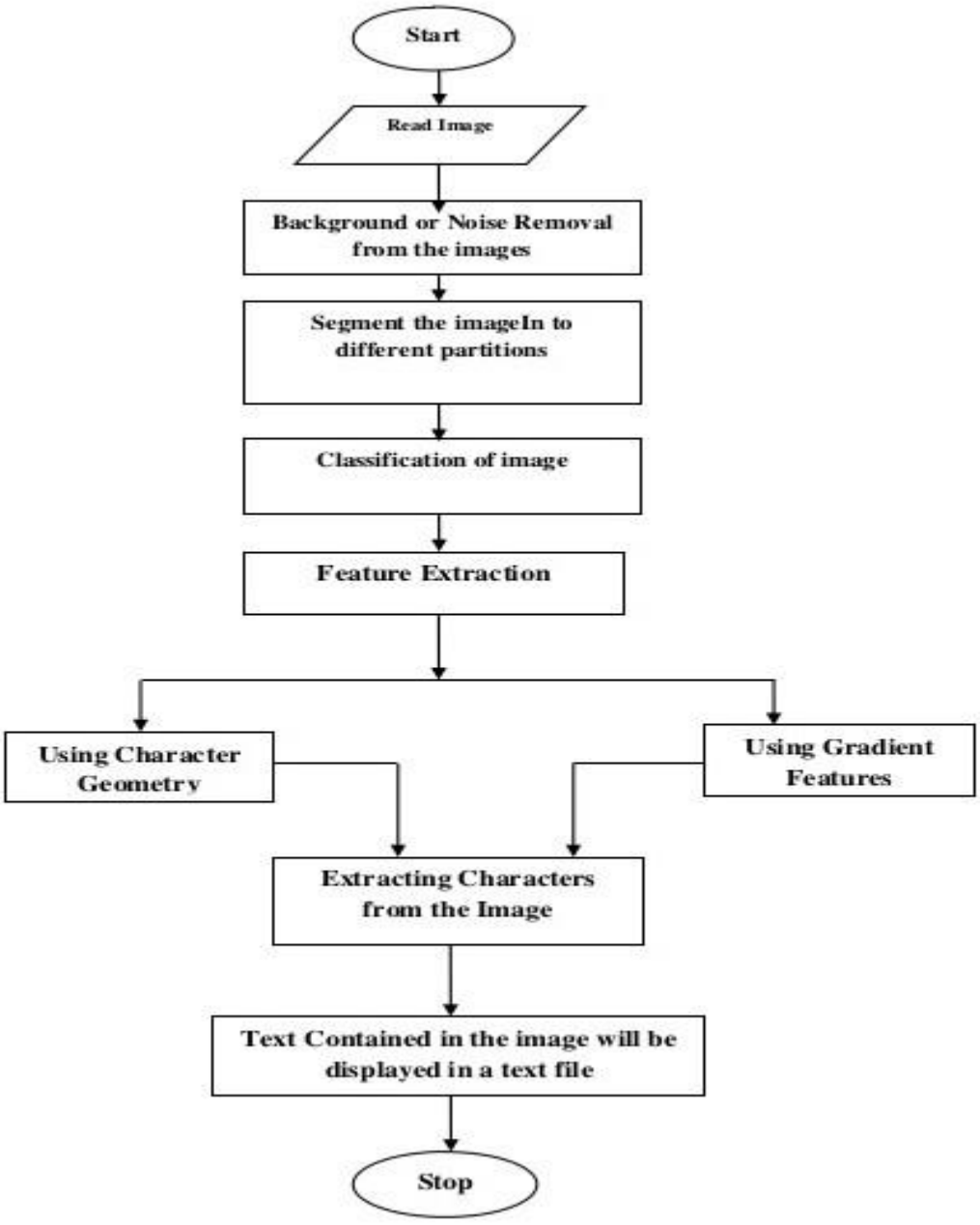


Fig 3.2 Flowchart

3.3 UML Diagrams

1.Use Case Digram

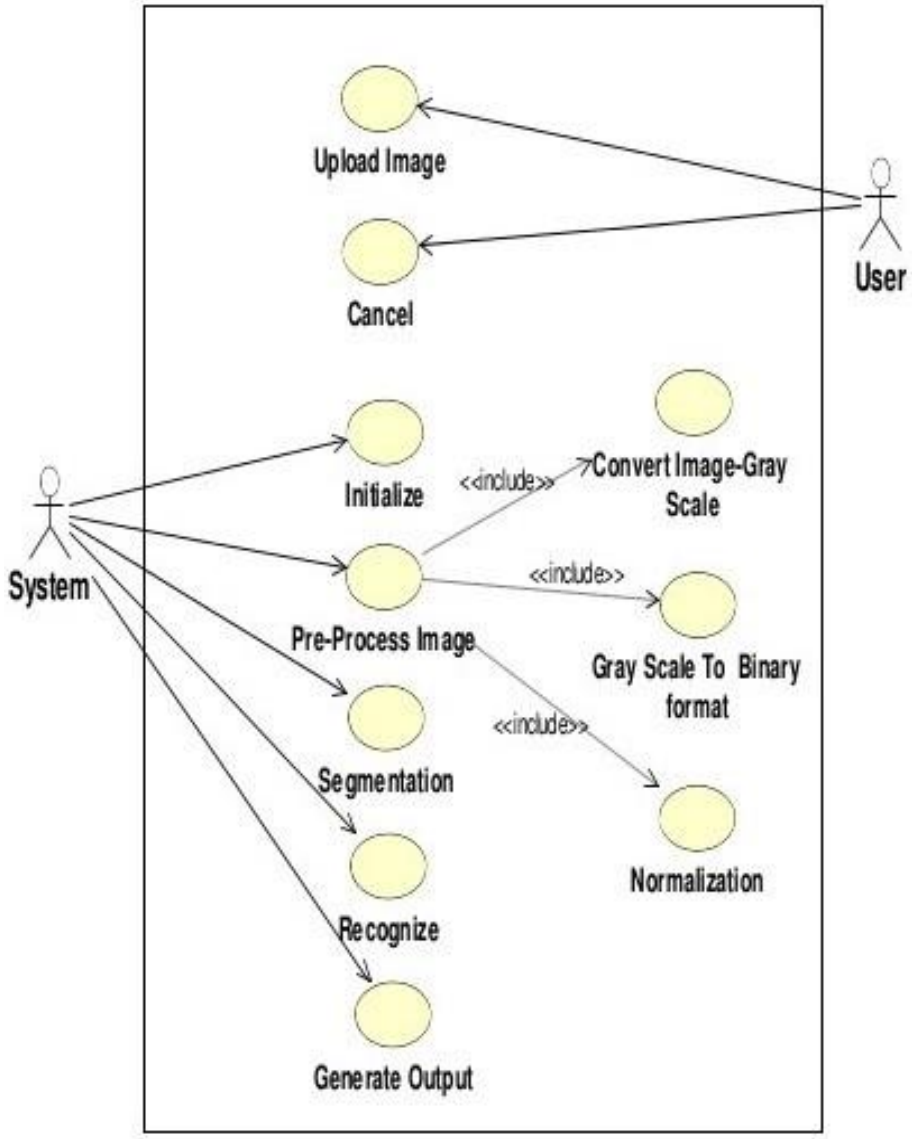


Fig 3.3.1 Use Case Diagram

User Module

User Case	Description
Actor	**User**
Precondition	Input image should be available.
Main Scenario	User uploads image.
Extension Scenario	If the image is not compatible. Not possible to upload file.
Post Condition	Image successfully uploaded.

Fig 3.3.2 User Module

Pre-processing Module

User Case	Description
Actor	**System**
Precondition	Uploaded input image
Main Scenario	Pre-processing is carried out by converting the image from RGB format to binary format.
Post Condition	Extract characters Before segmentation

Fig 3.3.3 Pre processing Module

2. Activity Diagram

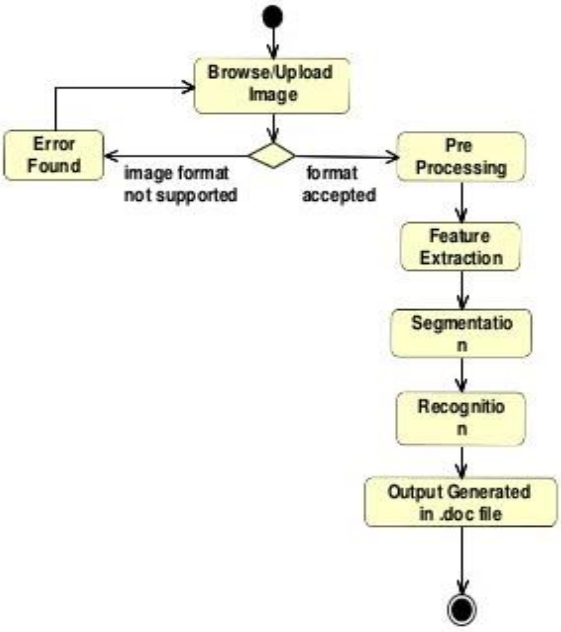


Fig 3.3.4 Activity Diagram

3. Sequence Diagram

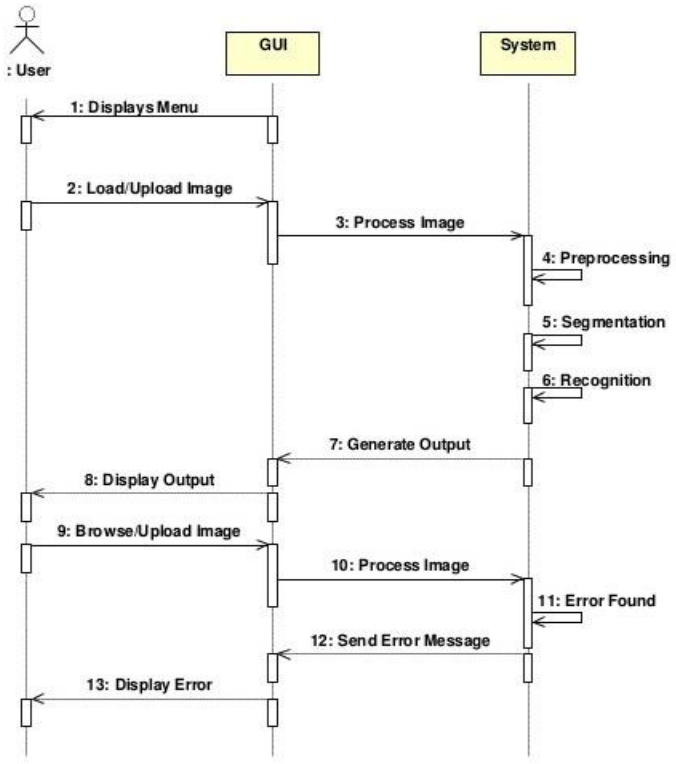


Fig 3.3.5 Sequence Diagram

4. System Architecture

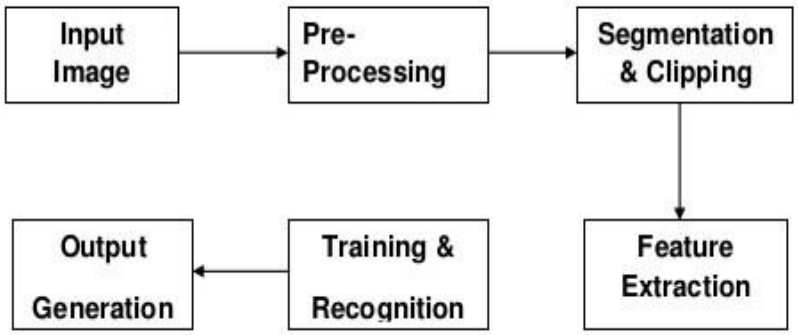


Fig 3.3.6 System Architecture

Chapter 4 : Hardware and Software Requirements

4.1 KERAS

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research. Keras is compatible with: Python 2.7-3.6.
Use Keras if you need a deep learning library that:
1. Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
2. Supports both convolutional networks and recurrent networks, as well as combinations of the two.
3. Runs seamlessly on CPU and GPU.

Guiding principles
User friendliness : Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

Modularity : A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as little restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models.

Easy extensibility : New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.

Work with Python : No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

4.2 THEANO

Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.
Theano features:

1. Tight integration with NumPy – Use numpy.ndarray in Theano-compiled functions.
2. Transparent use of a GPU – Perform data-intensive computations much faster than on a CPU.

3. Efficient symbolic differentiation – Theano does your derivatives for functions with one or many inputs.
4. Speed and stability optimizations – Get the right answer for log(1+x) even when x is really tiny.
5. Dynamic C code generation – Evaluate expressions faster.
   extensive unit-testing and self-verification – Detect and diagnose many types of errors

4.3 PYTHON 3.6

1.New library modules:

- typing: PEP 484 – Type Hints.
- zipapp: PEP 441 Improving Python ZIP Application Support.

2.New built-in features:

- bytes % args, bytearray % args: PEP 461 – Adding % formatting to bytes and bytearray.
- New bytes.hex(), bytearray.hex() and memoryview.hex() methods. (Contributed by Arnon Yaari in bpo-9951.)
- memoryview now supports tuple indexing (including multi-dimensional). (Contributed by Antoine Pitrou in bpo-23632.)
- Generators have a new gi_yieldfrom attribute, which returns the object being iterated by yield from expressions. (Contributed by Benno Leslie and Yury Selivanov in bpo-24450.)
- A new RecursionError exception is now raised when maximum recursion depth is reached. (Contributed by Georg Brandl in bpo-19235.)

3. Significant improvements in the standard library:

- collections.OrderedDict is now implemented in C, which makes it 4 to 100 times faster.
- The ssl module gained support for Memory BIO, which decouples SSL protocol handling from network IO.
- The new os.scandir() function provides a better and significantly faster way of directory traversal.
- functools.lru_cache() has been mostly reimplemented in C, yielding much better performance.
- The new subprocess.run() function provides a streamlined way to run subprocesses.
- The traceback module has been significantly enhanced for improved performance and developer convenience.

4.Security improvements:

- SSLv3 is now disabled throughout the standard library. It can still be enabled by instantiating a ssl.SSLContext manually. (See bpo-22638 for more details; this change was backported to CPython 3.4 and 2.7.)
- HTTP cookie parsing is now stricter, in order to protect against potential injection attacks. (Contributed by Antoine Pitrou in bpo-22796.)

4.4 DEEP LEARNING

- Deep learning (also known as deep structured learning or hierarchical 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
- Some representations are loosely based on interpretation of information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.
- Research attempts to create efficient systems to learn these representations from large-scale, unlabelled data sets.
- Deep learning architectures such as deep neural networks, deep belief 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.

5.References

- http://ufldl.stanford.edu/wiki/index.php/Using_the_MNIST_Dataset
- https://keras.io/
- http://deeplearning.net/software/theano/
- Kai Ding, Zhibiniu, ianwen Jin, Xinghua Zhu, A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition, *International Conference on Wavelet Analysis and Pattern Recognition*, pp. 1182-1186, Beijing, China, 2007.
- Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, ”A Review on the Various Techniques used for Optical Character Recognition”, *International Journal of Engineering Research and Applications*,2(1), pp. 659-662, 2012.
- Bhatia Neetu, Optical Character Recognition Techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(5), pp. 1219-1223, 2014.
- Iana M. origo and VenuGovindaraju, Offline Arabic Handwriting Recognition: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5), pp. 712724, 2006.
- K. Gaurav and Bhatia P. K., Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition, *2 nd International Conference on Emerging Trends in Engineering & Management, ICETEM*, pp. 14-22, 2013.

Timeline Chart of the Project

TIMELINE CHART FOR SEMESTER VII																	
MONTH	JULY				AUGUST					SEPTEMBER				OCTOBER			
WEEK NO.	W1	W2	W3	W4	W1	W2	W3	W4	W5	W1	W2	W3	W4	W1	W2	W3	W4
WORK TASKS																	
1.PROBLEM DEFINITION																	
Search for topics																	
Identify the goal of the project & 1st Presentation																	
2.PREPARATION																	
Gathering information on the basic types of convolution neural Systems																	
Study of related available Articles																	
Search for hardware and software requirements																	
Analysis of the format of MNIST dataset																	
3.PLANNING																	
Analysis of the layers of the CNN																	
Design phases of the project																	
Methodology followed																	