Handwritten Character Recognition of Tamil Font

Aditya Raj
Mayank Saharan
Supervised By -
Prof. Natesan Srinivasan

Department of Mathematics
IIT Guwahati

November 20, 2020

Introduction

Motivation

- Need to develop handwritten character recognition system for a rich and cultured language of Tamil, since not much work is done previously with considerable accuracy.

Introduction

Motivation

- Need to develop handwritten character recognition system for a rich and cultured language of Tamil, since not much work is done previously with considerable accuracy.
- We propose Deep and Machine Learning approaches for the same to achieve more accurate results.

Previous Work

- Duraiswamy and Shanthi(2010), achieved an accuracy of 82.04% using Support Vector machine(SVM) for only 34 characters.

Previous Work

- Duraiswamy and Shanthi(2010), achieved an accuracy of 82.04% using Support Vector machine(SVM) for only 34 characters.
- Bhattacharya and Ghosh(2007), achieved an accuracy of 89.66% using an unsupervised using clustering.

Previous Work

- Duraiswamy and Shanthi(2010), achieved an accuracy of 82.04% using Support Vector machine(SVM) for only 34 characters.
- Bhattacharya and Ghosh(2007), achieved an accuracy of 89.66% using an unsupervised using clustering.
- Vijayaraghavan and Sra(2015), achieved an accuracy of 94.4% using CNN with 35 classes.

Introduction

- We propose to use a CNN approach since it has proved to be the most effective model for developing OCR for any language.

Introduction

- We propose to use a CNN approach since it has proved to be the most effective model for developing OCR for any language.
- It gives much better accuracy compared to other unsupervised learning approaches(Described later).

Overview

- Optical Character Recognition

Overview

- Optical Character Recognition
- Artificial Neural Networks

Overview

- Optical Character Recognition
- Artificial Neural Networks
- Convolutional Neural Networks

Overview

- Optical Character Recognition
- Artificial Neural Networks
- Convolutional Neural Networks
- Image Processing and the Model Architecture

Overview

- Optical Character Recognition
- Artificial Neural Networks
- Convolutional Neural Networks
- Image Processing and the Model Architecture
- Experiments and Results

Overview

- Optical Character Recognition
- Artificial Neural Networks
- Convolutional Neural Networks
- Image Processing and the Model Architecture
- Experiments and Results
- Future Scope and Improvements

Introduction to Optical Character Recognition(OCR)

Different areas of Character Recognition

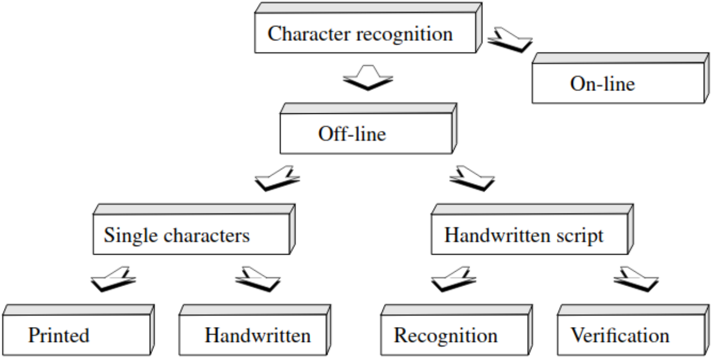


Figure: Different areas of Character Recognition.

OCR Methods

- **Online:** The online machine recognises the characters as they are inputed.

OCR Methods

- **Online:** The online machine recognises the characters as they are inputed.
- **Offline:** The offline machines recognises the characters after writing or printing is completed.

Components of an OCR system

- A setup of a typical OCR system

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner
  - Locate regions containing text by a segmentation process

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner
  - Locate regions containing text by a segmentation process
  - Extraction of each symbol once they are located

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner
  - Locate regions containing text by a segmentation process
  - Extraction of each symbol once they are located
  - Pre-processing of the extracted symbols , elimination of noise to provide for the feature extraction

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner
  - Locate regions containing text by a segmentation process
  - Extraction of each symbol once they are located
  - Pre-processing of the extracted symbols , elimination of noise to provide for the feature extraction
  - Feature extraction

Components of an OCR system

- A setup of a typical OCR system
  - Digitalise the analog document by an optical scanner
  - Locate regions containing text by a segmentation process
  - Extraction of each symbol once they are located
  - Pre-processing of the extracted symbols , elimination of noise to provide for the feature extraction
  - Feature extraction
  - Recognition of the characters

OCR system components

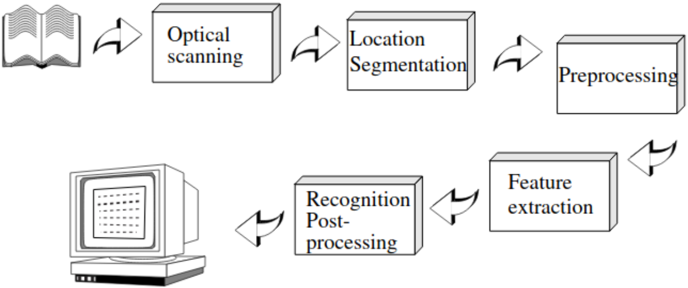


Figure: Typical OCR system components.

Neural Networks

Neural Networks: Biological Motivation

- Neural Networks are computational models that try to mimic the neural structure of the human brain.

Neural Networks: Biological Motivation

- Neural Networks are computational models that try to mimic the neural structure of the human brain.
- Biological Neurons communicate via electrical signals that are short-lived "spikes".

- Neural Networks are computational models that try to mimic the neural structure of the human brain.
- Biological Neurons communicate via electrical signals that are short-lived "spikes".
- Each neuron is connected to thousand of neurons and simultaneously receiving and sending multitude of these spikes.

Biological Neuron

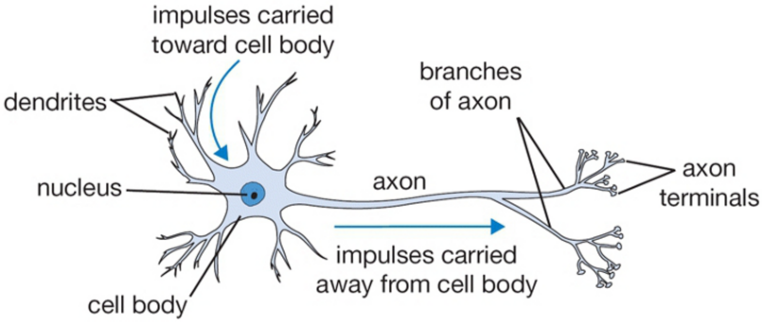


Figure: A Biological Neuron.

Artificial Neuron

- An artificial neuron assigns weights to all the inputs it receives and applies an activation function, f, to the weighted sum of the inputs along with some bias.

Artificial Neuron

- An artificial neuron assigns weights to all the inputs it receives and applies an activation function, f, to the weighted sum of the inputs along with some bias.

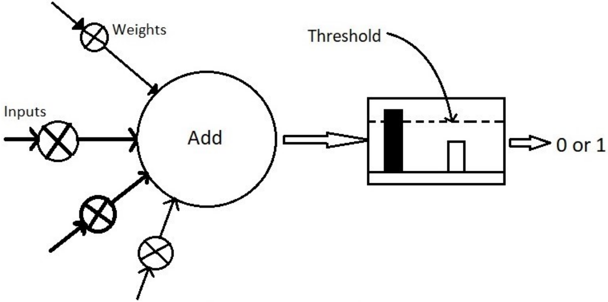


Figure: An Artificial Neuron.

Neural Network Architecture

- Network - system of artificial neurons.

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

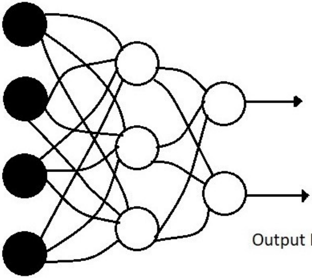


Figure: Two-layered basic model of Artificial Neural Network.

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

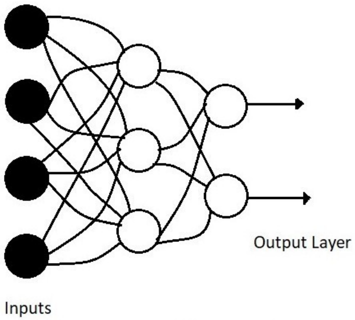


Figure: Two-layered basic model of Artificial Neural Network.

Layer is a block of nodes arranged vertically.

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

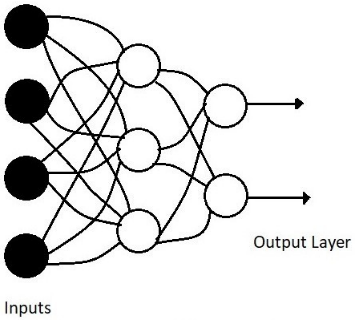


Figure: Two-layered basic model of Artificial Neural Network.

Layer is a block of nodes arranged vertically.

- Input Layer

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

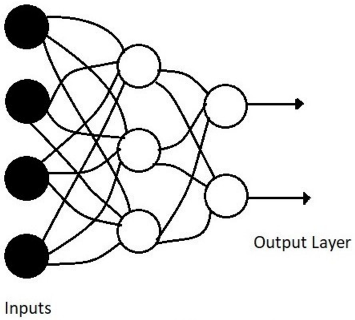


Figure: Two-layered basic model of Artificial Neural Network.

Layer is a block of nodes arranged vertically.

- Input Layer
- Hidden Layer

Neural Network Architecture

- Network - system of artificial neurons.
- It can be a single node or layer of nodes.

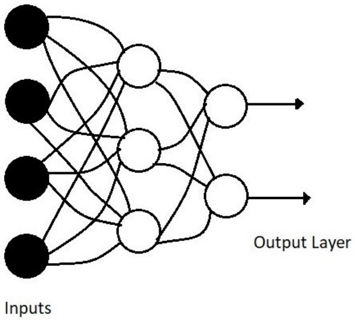


Figure: Two-layered basic model of Artificial Neural Network.

Layer is a block of nodes arranged vertically.

- Input Layer
- Hidden Layer
- Output Layer

Activation Function

- Activation function of a neuron defines the output of that node for a given set of inputs. This output is then used as input for the next node and so on.

Activation Function

- Activation function of a neuron defines the output of that node for a given set of inputs. This output is then used as input for the next node and so on.
- Some of the most common activation functions are:

Activation Function

- Activation function of a neuron defines the output of that node for a given set of inputs. This output is then used as input for the next node and so on.
- Some of the most common activation functions are:
  - ReLU Function:

$$f(x) = max(0, x)$$

Activation Function

- Activation function of a neuron defines the output of that node for a given set of inputs. This output is then used as input for the next node and so on.
- Some of the most common activation functions are:
  - ReLU Function:

$$f(x) = max(0, x)$$

  - Softmax Function:

$$\sigma(z)_i = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}}$$

Feed Forward Neural Networks

Feed Forward Neural Networks

- The information moves only in forward direction from the input nodes to the output nodes.

Feed Forward Neural Network

Feed Forward Neural Networks

- The information moves only in forward direction from the input nodes to the output nodes.
- No cycle is formed between the connections.

Feed Forward Neural Networks

- The information moves only in forward direction from the input nodes to the output nodes.
- No cycle is formed between the connections.
- The two types of Feed forward networks :-

Feed Forward Neural Networks

- The information moves only in forward direction from the input nodes to the output nodes.
- No cycle is formed between the connections.
- The two types of Feed forward networks :-

  1. **Single Layer Perceptron**

Feed Forward Neural Networks

- The information moves only in forward direction from the input nodes to the output nodes.
- No cycle is formed between the connections.
- The two types of Feed forward networks :-

  1. **Single Layer Perceptron**
  2. **Multi Layer Perceptron**

This model contains a single input neuron without any hidden layer.

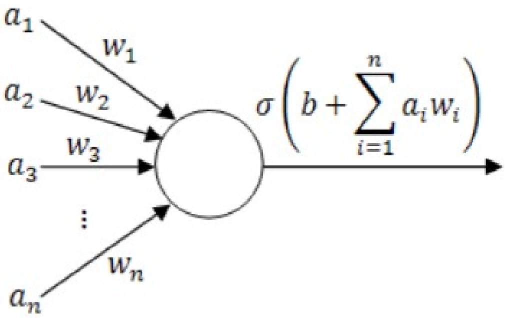


Figure: A Single-layer Perceptron.

Multiple-Layer Perceptron

This type of network contains multiple layers interconnected in a Feed-forward manner.

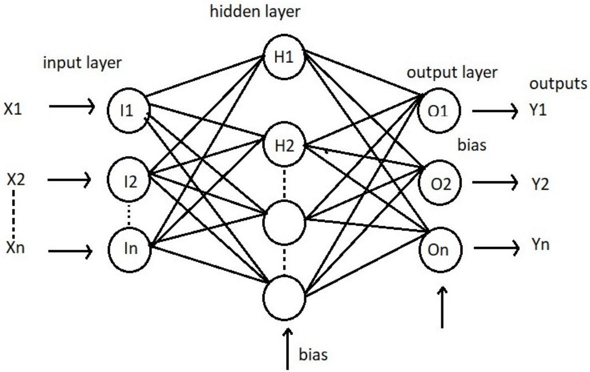


Figure: Multiple-Layer Perceptron

Types of Learnings:

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.
- Unsupervised Learning:This type of learning looks for previously undetected patterns in a data-set with no pre-existing labels.

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.
- Unsupervised Learning:This type of learning looks for previously undetected patterns in a data-set with no pre-existing labels.

Supervised Learning is further classified into two categories:

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.
- Unsupervised Learning:This type of learning looks for previously undetected patterns in a data-set with no pre-existing labels.

Supervised Learning is further classified into two categories:

- Classification

Types of Learnings

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.
- Unsupervised Learning:This type of learning looks for previously undetected patterns in a data-set with no pre-existing labels.

Supervised Learning is further classified into two categories:

- Classification
- Regression

Types of Learnings

Types of Learnings:

- Supervised Learning: The learning through pre-labelled inputs acting as targets.
- Unsupervised Learning:This type of learning looks for previously undetected patterns in a data-set with no pre-existing labels.

Supervised Learning is further classified into two categories:

- Classification
- Regression

Convolutional Neural Networks(CNN) comes under supervised learning.

Convolutional Neural Networks

Introduction

- Convolutional Neural Networks (CNNs) are a complex variant of the Artificial Neural Networks.

Introduction

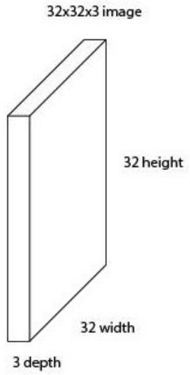
- Convolutional Neural Networks (CNNs) are a complex variant of the Artificial Neural Networks.
- Structure same as conventional Neural Networks.

Introduction

- Convolutional Neural Networks (CNNs) are a complex variant of the Artificial Neural Networks.
- Structure same as conventional Neural Networks.
- The most basic difference between the NNs and the CNNs is that the CNNs has volumes in place of vectors in each layer.

Introduction

- Convolutional Neural Networks (CNNs) are a complex variant of the Artificial Neural Networks.
- Structure same as conventional Neural Networks.
- The most basic difference between the NNs and the CNNs is that the CNNs has volumes in place of vectors in each layer.



Why is it better?

- CNN uses the concept of weight sharing
- There is a reduction in the number of parameters
- It doesn't suffer from *overfitting* and hence improved generalization
- It is much easier to implement large models in CNN

CNN Architecture

CNN model consists of three types of layers -

- Convolutional Layer

CNN Architecture

CNN model consists of three types of layers -

- Convolutional Layer
- Pooling Layer

CNN Architecture

CNN model consists of three types of layers -

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

CNN Architecture

CNN model consists of three types of layers -

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

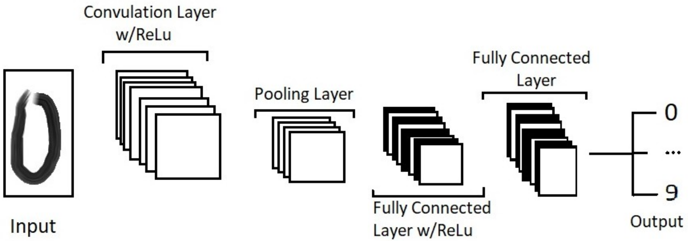


Figure: Layers of CNN.

Functioning of CNN

Functioning of CNN can be divided into 4 major parts :-

Functioning of CNN

Functioning of CNN can be divided into 4 major parts :-

- Input Layer holds pixel value of images.

Functioning of CNN

Functioning of CNN can be divided into 4 major parts :-

- Input Layer holds pixel value of images.
- The Convolution Layer gives output of each neuron in the layer.

Functioning of CNN

Functioning of CNN can be divided into 4 major parts :-

- Input Layer holds pixel value of images.
- The Convolution Layer gives output of each neuron in the layer.
- Pooling Layer further reduces the parameters.

Functioning of CNN

Functioning of CNN can be divided into 4 major parts :-

- Input Layer holds pixel value of images.
- The Convolution Layer gives output of each neuron in the layer.
- Pooling Layer further reduces the parameters.
- Fully Connected Layer performs actions that are performed in the general Neural Network model.

Convolutional Layer

- The Layer convolves each filter across the spatial dimensionality of the input. This produces a 2-dimensional activation map.

Convolutional Layer

- The Layer convolves each filter across the spatial dimensionality of the input. This produces a 2-dimensional activation map.
- These kernels form a field like structure called **receptor fields**.

Convolutional Layer

- The Layer convolves each filter across the spatial dimensionality of the input. This produces a 2-dimensional activation map.
- These kernels form a field like structure called **receptor fields**.

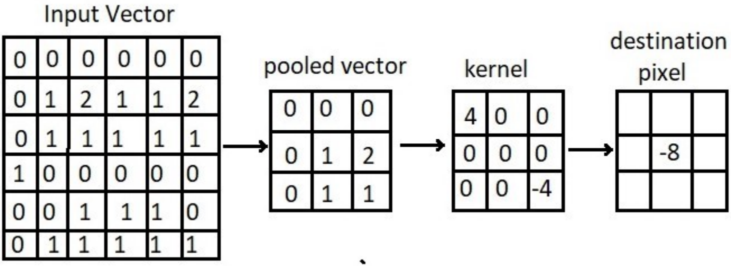


Figure: Representation of Convolutional Layer.

Convolutional Layer

Output can be optimized by tuning these hyper-parameters in Convolutional Layer :-

- Depth

Convolutional Layer

Output can be optimized by tuning these hyper-parameters in Convolutional Layer :-

- Depth
- Stride

Convolutional Layer

Output can be optimized by tuning these hyper-parameters in Convolutional Layer :-

- Depth
- Stride
- Zero-padding

Output can be optimized by tuning these hyper-parameters in Convolutional Layer :-

- Depth
- Stride
- Zero-padding

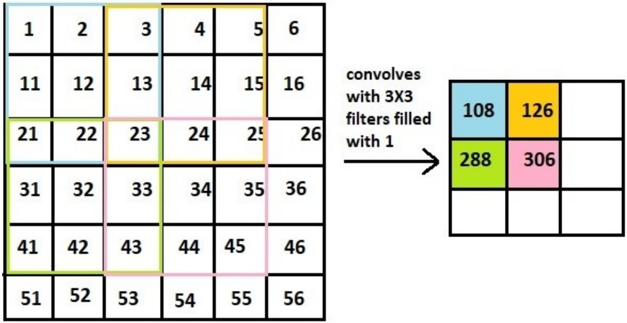


Figure: When stride is set to 2.

- The exact location of a feature becomes less important once it has been detected.

Pooling Layer

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.

Pooling Layer

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.
- This method is called down-sampling.

Pooling Layer

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.
- This method is called down-sampling.

Types of Downsampling:-

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.
- This method is called down-sampling.

Types of Downsampling:-

- Max Pooling

Pooling Layer

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.
- This method is called down-sampling.

Types of Downsampling:-

- Max Pooling
- Average Pooling

Pooling Layer

- The exact location of a feature becomes less important once it has been detected.
- A window is selected and inputs are passed through the pooling layer.
- This method is called down-sampling.

Types of Downsampling:-

- Max Pooling
- Average Pooling
- Sum Pooling

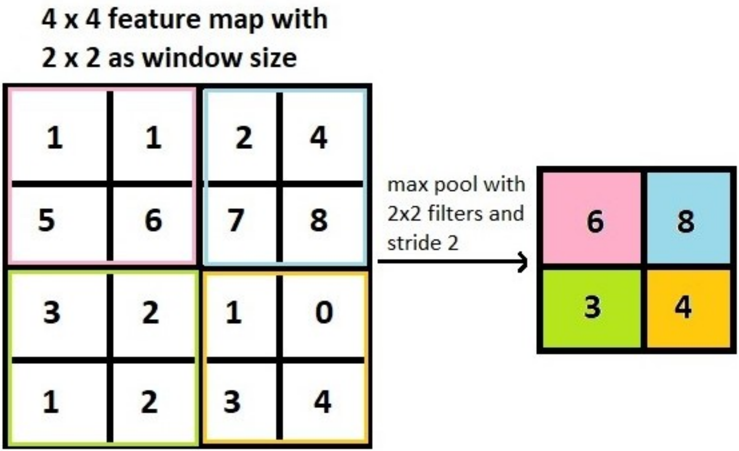


Figure: Max Pooling

Fully Connected Layer

- It is similar to conventional Neural Network models.

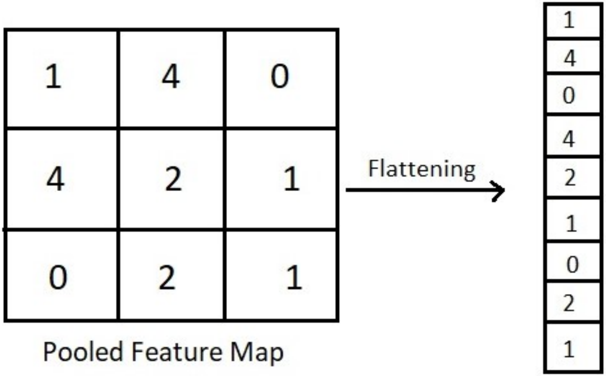
Fully Connected Layer

- It is similar to conventional Neural Network models.
- Output of first two layers(Convolution and Pooling) is fed into the Fully Connected Layer.

Fully Connected Layer

- It is similar to conventional Neural Network models.
- Output of first two layers(Convolution and Pooling) is fed into the Fully Connected Layer.
- Before it, flattening of the input matrix needs to be performed.

Flattening



- Gradient Descent is used to reduce the cost function by estimating the cost over the whole input and update the parameters.

Fully Connnected Layer

- Gradient Descent is used to reduce the cost function by estimating the cost over the whole input and update the parameters.
- This updation is done after every epoch.

Fully Connnected Layer

- Gradient Descent is used to reduce the cost function by estimating the cost over the whole input and update the parameters.
- This updation is done after every epoch.
- It has recently been replaced by Stochastic Gradient Descent.

Fully Connnected Layer

- Gradient Descent is used to reduce the cost function by estimating the cost over the whole input and update the parameters.
- This updation is done after every epoch.
- It has recently been replaced by Stochastic Gradient Descent.
- ReLu is used as activation function between these layers.

Softmax Function

- The purpose of activation function is to introduce non-linearity in our CNN model .

Softmax Function

- The purpose of activation function is to introduce non-linearity in our CNN model .
- Softmax function is special because it can handle multiple classes at once.

Softmax Function

- The purpose of activation function is to introduce non-linearity in our CNN model .
- Softmax function is special because it can handle multiple classes at once.
- Softmax converts them into values between 0 and 1,to interpret them as probabilities.

Softmax Function

- The purpose of activation function is to introduce non-linearity in our CNN model .
- Softmax function is special because it can handle multiple classes at once.
- Softmax converts them into values between 0 and 1,to interpret them as probabilities.
- Since Softmax is a continuously differentiable function, it's possible to calculate the derivative of the loss function with respect to every weight in the network.

Softmax Function

- The purpose of activation function is to introduce non-linearity in our CNN model .
- Softmax function is special because it can handle multiple classes at once.
- Softmax converts them into values between 0 and 1,to interpret them as probabilities.
- Since Softmax is a continuously differentiable function, it's possible to calculate the derivative of the loss function with respect to every weight in the network.

$$\sigma(z)_i = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}}$$

Adam Optimization

- Adaptive Moment Estimation is an optimization algorithm.

Adam Optimization

- Adaptive Moment Estimation is an optimization algorithm.
- It is computationally efficient.

Adam Optimization

- Adaptive Moment Estimation is an optimization algorithm.
- It is computationally efficient.
- Updation of parameter is completely invariant to gradient re-scaling.

Adam Optimization

- Adaptive Moment Estimation is an optimization algorithm.
- It is computationally efficient.
- Updation of parameter is completely invariant to gradient re-scaling.
- Disadvantage is that it increases cost of the algorithm.

Adam Optimization

- Adaptive Moment Estimation is an optimization algorithm.
- It is computationally efficient.
- Updation of parameter is completely invariant to gradient re-scaling.
- Disadvantage is that it increases cost of the algorithm.

$$w(t) = w(t-1) - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Image Pre-processing and Model Architecture

Data-set

- We used HP Labs data-set which is available online.

Data-set

- We used HP Labs data-set which is available online.
- It contains 156 characters of Tamil script.

Data-set

- We used HP Labs data-set which is available online.
- It contains 156 characters of Tamil script.
- Around 480 samples of each of the characters.

Data-set

- We used HP Labs data-set which is available online.
- It contains 156 characters of Tamil script.
- Around 480 samples of each of the characters.
- Data is available in raw form, so we need to process the images.

Tamil vowels and consonants

Tamil vowels and consonants , with combinations of vowels and consonants.

Vowels	Consonants	Consonant + vowels	Compound Characters
அ	க்	க்+அ	க
ஆ	ங்	க்+ஆ	கா
இ	ச்	க்+இ	கி
ஈ	ஞ்	க்+ஈ	கீ
உ	ட்	க்+உ	கு
ஊ	ண்	க்+ஊ	கூ
எ	த்	க்+எ	கெ
ஏ	ந்	க்+ஏ	கே
ஐ	ப்	க்+ஐ	கை
ஒ	ம்	க்+ஒ	கொ
ஓ	ய்	க்+ஓ	கோ
ஔ	ர்	க்+ஔ	கௌ
	ல்		
	வ்		
	ழ்		
	ள்		
	ற்		
	ன்		

Figure: Tamil script characters

Characters in the Data-set

The 156 characters are encoded as labels numbering from '0-155' in the same sequence in which they are encoded

அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ			
0	1	2	3	4	5	6	7	8	9	10	11			
க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய	ர	ல	வ	ழ
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
ள	ற	ன	ஸ	ஷ	ஜ	ஹ	க்ஷ	கி	ஙி	சி	ஞி	டி	ணி	தி
27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
நி	பி	மி	யி	ரி	லி	வி	ழி	ளி	றி	னி	ஸி	ஷி	ஜி	ஹி
42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
க்ஷி	கீ	ஙீ	சீ	ஞீ	டீ	ணீ	தீ	நீ	பீ	மீ	யீ	ரீ	லீ	வீ
57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
ழீ	ளீ	றீ	னீ	ஸீ	ஷீ	ஜீ	ஹீ	க்ஷீ	கு	ஙு	சு	ஞு	டு	ணு
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
து	நு	பு	மு	யு	ரு	லு	வு	ழு	ளு	று	னு	கூ	ஙூ	சூ
87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
ஞூ	டூ	ணூ	தூ	நூ	பூ	மூ	யூ	ரூ	லூ	வூ	ழூ	ளூ	றூ	னூ
102	103	104	105	106	107	108	109	110	111	112	113	114	115	116
ா	ெ	ே	ை	ஸ்ரீ	ஸு	ஷு	ஜு	ஹு	க்ஷு	ஸூ	ஷூ	ஜூ	ஹூ	க்ஷூ
117	118	119	120	121	122	123	124	125	126	127	128	129	130	131
க்	ங்	ச்	ஞ்	ட்	ண்	த்	ந்	ப்	ம்	ய்	ர்	ல்	வ்	ழ்
132	133	134	135	136	137	138	139	140	141	142	143	144	145	146
ள்	ற்	ன்	ஸ்	ஷ்	ஜ்	ஹ்	க்ஷ்	ஃ						
147	148	149	150	151	152	153	154	155						

Figure: HP Labs Tamil Data-set

Training and Testing of model

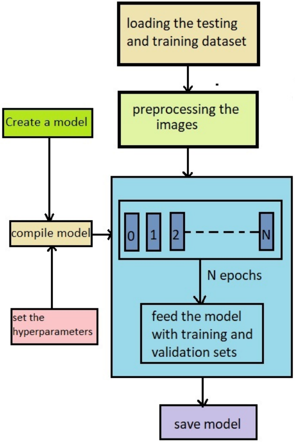


Figure: Proposed Model

Image Pre-processing

- Image Resizing (Bilinear Interpolation)

Image Pre-processing

- Image Resizing (Bilinear Interpolation)
- Image Smoothing

Image Pre-processing

- Image Resizing (Bilinear Interpolation)
- Image Smoothing
- Colour Inversion

Image Pre-processing

- Image Resizing (Bilinear Interpolation)
- Image Smoothing
- Colour Inversion

Bilinear Interpolation

$$g(n_1, n_2) = A_0 + A_1 n_1 + A_2 n_2 + A_3 n_1 n_2$$

Model Architecture

- CNN model for our HTCR system consists of 10 layers -
    - 4 Convolutional Layer

Model Architecture

- CNN model for our HTCR system consists of 10 layers -
  - 4 Convolutional Layer
  - 4 Max Pooling Layer

Model Architecture

- CNN model for our HTCR system consists of 10 layers -
    - 4 Convolutional Layer
    - 4 Max Pooling Layer
    - 2 Fully Connected Layer

Model Architecture

- CNN model for our HTCR system consists of 10 layers -
  - 4 Convolutional Layer
  - 4 Max Pooling Layer
  - 2 Fully Connected Layer

We perform Batch normalization after every Convolution Layer.

Model Architecture

- ReLu activation function is used in each Convolutional Layer .

Model Architecture

- ReLu activation function is used in each Convolutional Layer .
- Input Layer is composed of sample images of 64×64 .

Model Architecture

- ReLu activation function is used in each Convolutional Layer .
- Input Layer is composed of sample images of 64×64 .
- The Output Layer is composed of a **softmax** classifier with 156 labels.

Model Architecture

- ReLu activation function is used in each Convolutional Layer .
- Input Layer is composed of sample images of 64×64 .
- The Output Layer is composed of a **softmax** classifier with 156 labels.
- First two Convolution Layers have filter set to 32 and next two have filter set to 64.

Model Architecture

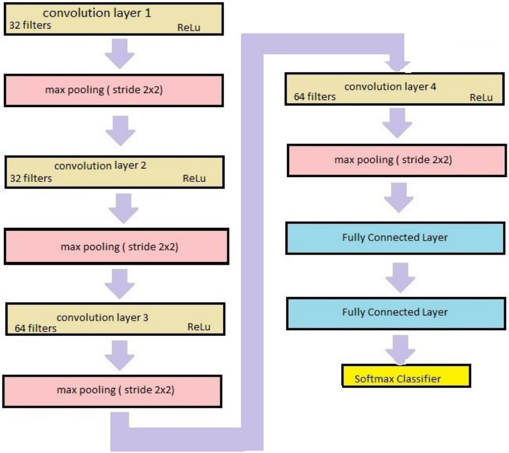


Figure: Model Architecture

Hyper-parameters

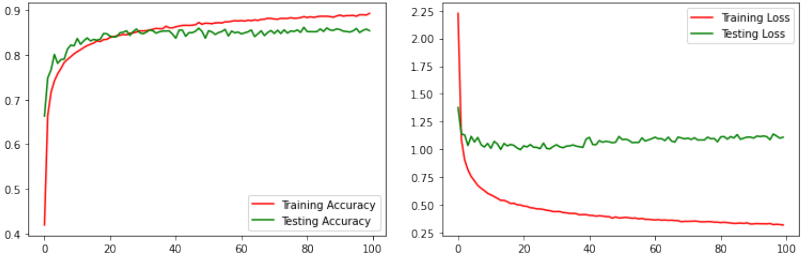
Hyper-parameters	Values
Batch Size	64
Initialization	Xavier
Optimizer	Adam
Epochs	100
Activation function	ReLu
Learning rate	0.001

Experiment

- *ImageDataGenerator* function is used to import input images to the environment
- In the first experiment, we use batch size of 32
- Image dataset is the same but with the dimensions 32×32

No of epochs	100
Batch Size	32
Total parameters	93,244
Trainable Parameters	92,476
Non-trainable Parameters	768
Training Loss	0.3174
Training Accuracy	0.8934
Testing Loss	1.1097
Testing Accuracy	0.8546

Results of 1^{st} experiment

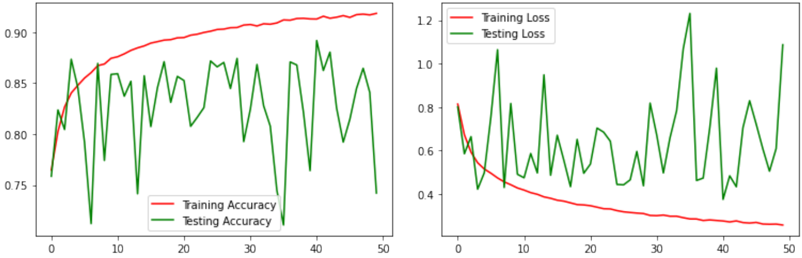


2^{nd} Experiment Results

- In the 2^{nd} experiment, we use batch size of 64
- Image dataset is the same but with the dimensions 64×64
- Number of epochs was set to 50 only because of computational constraints

No of epochs	50
Batch Size	64
Total parameters	158,780
Trainable Parameters	158,012
Non-trainable Parameters	768
Training Loss	0.2569
Training Accuracy	0.9285
Testing Loss	1.0871
Testing Accuracy	0.7421

Results


0.90
0.85
0.80
0.75
0 10 20 30 40 50
Training Accuracy
Testing Accuracy
1.2
1.0
0.8
0.6
0.4
0 10 20 30 40 50
Training Loss
Testing Loss

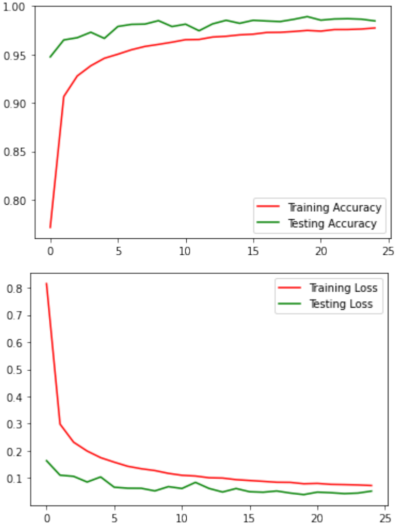
Same model on Devnagri Script (3^{rd} experiment)

- We have a vast set of data for Devnagri available online
- Number of labels are changed to 56
- Total 2000 samples, split into 1700 images for training set and 300 for validation set
- Number of epochs is set to 100 here as well.

Results

No of epochs	100
Batch Size	32
Total parameters	86,094
Trainable Parameters	85,326
Non-trainable Parameters	768
Training Loss	0.0724
Training Accuracy	0.9774
Testing Loss	0.0520
Testing Accuracy	0.9847

Results



1.00
0.95
0.90
0.85
0.80
Training Accuracy
Testing Accuracy
0
5
10
15
20
25
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Training Loss
Testing Loss
0
5
10
15
20
25

Comparison of Results

Comparing the results of all three experiments:-

Parameters	1^{st} exp	2^{nd} exp	3^{rd}exp
Batch Size	32	64	32
Training Loss	0.3174	0.2569	0.0724
Training Accuracy	0.8934	0.9285	0.9774
Testing Loss	1.1097	1.0871	0.0520
Testing Accuracy	0.8546	0.7421	0.9847

Future Scope & Further improvements

- Increase the number of samples available per characters.
- Improved pre-processing of the sample images from the data-set
- Development of automatic feature extraction methods and combine them with the proposed model.
- Not considering all characters of Tamil font because of lack of Data-set, we propose to apply the model to all the characters of Tamil script.

References

O'Shea Keiron, Ryan Nash et al, 2015
Introduction to convolutional neural networks
Research Gate', 15th December, 2015.

Kevin Gurney[1997]
An introduction to neural networks
CRC press,1997

BR Kavitha and C Srimathi[2019]
Benchmarking on offline handwritten tamil character recognition using convolutional neural networks.
Journal of King Saud University-Computer and Information Sciences,20190

Sakshi Indolia, Anil Kumar Goswami, SP Mishra, and Pooja Asopa[2018]
Conceptual understanding of convolutional neural network-a deep learning approach.
Procedia computer science,132:679-688,2017

HP Labs Tamil-Character Data-set
http://lipitk.sourceforge.net/datasets/tamilchardata.htm

The End