Optical Character Recognition using Deep Learning

MRS.SUNITHA A M.E Assistant professor

Department of Computer Science and Engineering

Mr. HERMAN JOSH A, B.E, Student of Computer Science Engineering

Mr. MIDHUN T M, B.E, Student of Computer science and Engineering

St. Joseph College of Engineering, Sriperumbudur, Chennai.

Abstract

In this project we propose an approach to recognize handwritten Tamil characters using a Deep learning with hidden layer. Also, an analysis was carried out to determine the number of hidden layer nodes to achieve high performance of back propagation network in the recognition of handwritten Tamil characters. The system was trained using several different forms of handwriting provided by both male and female participants of different age groups.

Character recognition is developed for various patterns of handwritten or optical characters to be recognized digitally. There are many Tamil literatures in undigitized form. Using deep learning the undigitized Tamil literatures can be converted into readable format. Many researches were carried on character recognition using deep learning for languages like Arabic, Devanagari, Telugu, etc… Due to the larger category set and confusion in similarities between handwritten characters Tamil character recognition is a challenge.

Key Terms: CNN, Deep learning, OCR system

Introduction

Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size and orientation angle of the characters. Among different branches of handwritten character recognition, it is easier to recognize English alphabets and numerals than

Tamil characters. Many researchers have also applied the excellent generalisation capabilities offered by CNNs to the recognition of characters. Many studies have used Fourier descriptors and Back Propagation Networks for classification tasks. Fourier descriptors were used in to recognise handwritten numerals. In Fourier descriptors and a Back Propagation Network were used to classify tools. The present work describes a system for offline recognition of Tamil script, a language widely spoken in South India. It is also one of the official languages in countries such as Singapore, Malaysia, and Sri Lanka apart from India. Recently, the Indian Government recognized it as a classical language. In this project, we propose a recognition system for handwritten Tamil characters.

Optical character recognition deals with an important issue called the character classification and this is more challenging because of the similarities between them. Character classification is something that identifies the characters in Tamil that are trained initially and it can identify the characters written by different users. Tamil language is most popular and its techniques grammar are formulated for the recognition of a Tamil script is one of the foremost among the 16 major national languages spoken reading and writing by the South Indian people. Character recognition can be online or offline. In On-line character recognition system, the representation of two-dimensional coordinates of successive points is done. It is automatic conversion of text into a digital form. Off-line character recognition is used to convert written text into letter codes. Since there are many hidden layers in deep neural network the parameters for the training is very huge. To prevent over fitting, we require a large set of examples. One such magnificence of deep mastering is the Convolution Neural Network (CNN) (1). Convolution Neural Network is a special type of neural network used effectively for image recognition and classification. By implementing these techniques, we recognize the handwritten Tamil characters. Tamil is one of the ancient languages spoken by many people

Literature Survey

The paper "**Online handwriting recognition for Tamil"** By G.V. Prakash in 2021, A system for online recognition of handwritten Tamil characters is presented. A handwritten character is constructed by executing a sequence of strokes. A structure- or shape-based representation of a stroke is used in which a stroke is represented as a string of shape features. Using this string representation, an unknown stroke is identified by comparing it with a database of strokes using a flexible string-matching procedure. A full character is recognized by identifying all the component strokes. Character termination, is determined using a finite state automaton. Development of similar systems for other Indian scripts is outlined.

The paper **"Comparison of elastic matching algorithms for online Tamil handwritten character recognition."** By S. Madhvanath in 2018, We present a comparison of elastic matching schemes for writer dependent on-line handwriting recognition of isolated Tamil characters. Three different features are considered namely, preprocessed x-y coordinates, quantized slope values, and dominant point coordinates. Seven schemes based on these three features and dynamic time warping distance measure are compared with respect to recognition accuracy, recognition speed, and number of training templates. Along with these results, possible grouping strategies and error analysis is also presented in brief.

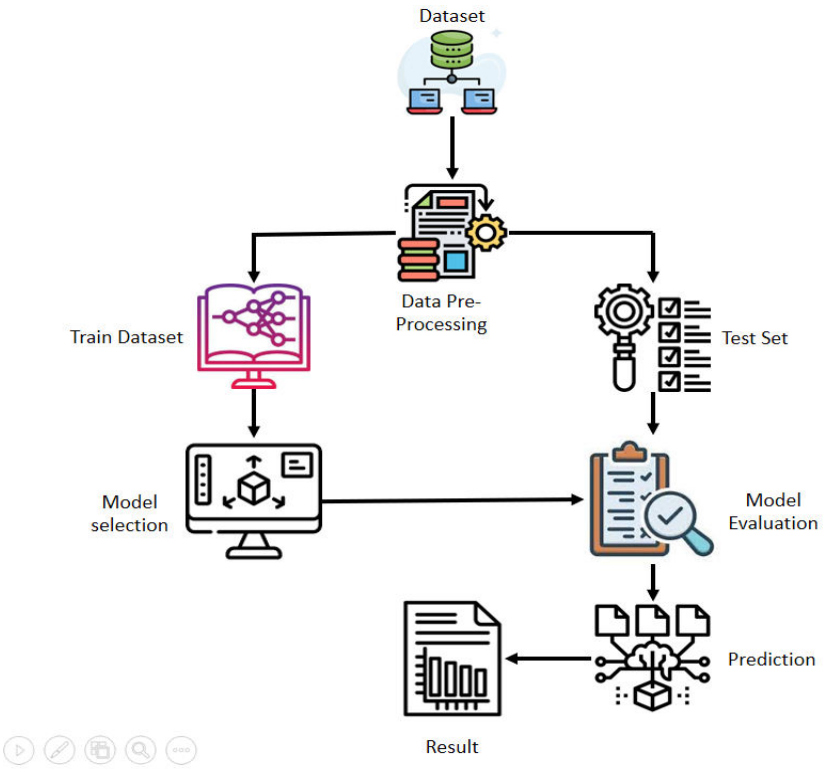
The paper" **A multi-level perception approach to reading cursive script**." By M.Božinović in 2020, Reading cursive script is the problem of transforming language from the iconic form of cursive human handwriting to one of ordinal representation. The problem involves elements of visual perception at one level of processing and those of language perception and understanding at a higher level. The problem is approached as one of multi-level perception in which a cursive script word image is transformed through a representational hierarchy. The levels are based on descriptions that use points, contours, features, letters, and words. Global control is hierarchical until an intermediate level after which it is heterarchical. A feature representation is generated bottom-up from the image using statistical dependencies between letters and features. Ratings for partially formed words are computed using a stack and a lexicon represented as a trie. Several heuristics for low- and intermediate-level processing for cursive script are introduced, including: reference-line finding using projection profile analysis, letter segmentation based on local lower contour minima and areas with low vertical profiles, simultaneous encoding of contours and their topological relationships, extracting features (e.g., middle loop, upper-zone stroke), and finding shape-oriented events. Two modes of learning are defined: initial training with user feedback and unsupervised adaptation to the writer. Experiments demonstrating the promise of the approach are described.

System Design

Designing an Optical Character Recognition (OCR) system using deep learning involves several steps, including data collection, preprocessing, training, and evaluation. Here's an outline of the system design:

1. Data Collection: The first step is to collect the training and testing data for the OCR system. The data should consist of a large number of images of printed and handwritten text. The data should be representative of the types of text that the OCR system will encounter in the real world.
2. Preprocessing: Before training the OCR system, the data needs to be preprocessed to make it suitable for deep learning. The preprocessing steps include resizing the images, converting them to grayscale or binary format, and normalization.
3. Training: The next step is to train the deep learning model using the preprocessed data. A popular architecture for OCR is the convolutional neural network (CNN). The CNN consists of several layers of convolutional and pooling operations, followed by fully connected layers. The training process involves optimizing the weights of the network using backpropagation and gradient descent.
4. Evaluation: After training, the model's performance needs to be evaluated on a separate test dataset. The evaluation metrics can include accuracy, precision, recall, and F1 score. The model can be further fine-tuned based on the evaluation results.
5. Deployment: The final step is to deploy the OCR system in production. The system should be optimized for speed and accuracy and should handle various types of text and formats. The system can be integrated into other applications, such as document management systems, to automate the text extraction process.

Overall, designing an OCR system using deep learning requires expertise in image processing, deep learning, and software engineering. The system design should be iterative and based on continuous feedback from the evaluation and deployment phases.



IMPLEMENTATION

```
from flask import Flask, render_template, request
from base64 import b64decode
from io import BytesIO
from PIL import Image, ImageOps, ImageFilter
import numpy as np
from scipy import ndimage
import torch
import torch.nn as nn
import torch.nn.functional as F
from torchvision import transforms
```

```
import random
import inference
classes = ['அ', 'ஆ', 'ஓ', 'ஙூ', 'சூ', 'ஜா', 'டூ', 'ணா', 'தா', 'நா',
'பூ', 'மூ', 'யூ', 'ஃ', 'ரூ', 'லா', 'வூ', 'ழு', 'ளூ', 'றா', 'னா',
'ா', 'ெ', 'ே', 'க', 'ை', 'ஸ்ரீ', 'ஸு', 'ஷு', 'ஜு', 'ஹு',
'கூு', 'ஸூ', 'ஷூ', 'ஜூ', 'ங', 'ஹூ', 'கூூ', 'க்', 'ங்', 'ச்',
'ஞ்', 'ட்', 'ண்', 'த்', 'ந்', 'ச', 'ப்', 'ம்', 'ய்', 'ர்', 'ல்', 'வ்', 'ழ்',
'ள்', 'ற்', 'ன்', 'ஞ', 'ஸ்', 'ஷ்', 'ஜ்', 'ஹ்', 'கூ்', 'ஔ', 'ட',
'ண', 'த', 'ந', 'இ', 'ப', 'ம', 'ய', 'ர', 'ல', 'வ', 'ழ', 'ள', 'ற',
'ன', 'ஈ', 'ஸ', 'ஷ', 'ஜ', 'ஹ', 'கூ', 'கி', 'ஙி', 'சி', 'ஞி', 'டி',
'உ', 'ணி', 'தி', 'நி', 'பி', 'மி', 'யி', 'ரி', 'லி', 'வி', 'ழி', 'ஊ',
'ளி', 'றி', 'னி', 'ஸி', 'ஷி', 'ஜி', 'ஹி', 'கூி', 'கீ', 'ஙீ', 'எ',
'சீ', 'ஞீ', 'டீ', 'ணீ', 'தீ', 'நீ', 'பீ', 'மீ', 'யீ', 'ரீ', 'ஏ', 'லீ', 'வீ',
'ழீ', 'ளீ', 'றீ', 'னீ', 'ஸீ', 'ஷீ', 'ஜீ', 'ஹீ', 'ஐ', 'கூீ', 'கு', 'ஙு',
'சு', 'ஞு', 'டு', 'ணு', 'து', 'நு', 'பு', 'ஒ', 'மு', 'யு', 'ரு', 'லு',
'வு', 'ழு', 'ளு', 'று', 'னு', 'கூ']
app = Flask(__name__)
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()

        self.conv1 = nn.Conv2d(1, 16, 3, padding=1)
        self.bn1 = nn.BatchNorm2d(16)
        self.conv2 = nn.Conv2d(16, 16, 3, padding=1)
        self.bn2 = nn.BatchNorm2d(16)
        self.pool1 = nn.MaxPool2d(2, 2)
        self.conv3 = nn.Conv2d(16, 32, 3, padding=1)
        self.bn3 = nn.BatchNorm2d(32)
        self.conv4 = nn.Conv2d(32, 32, 3, padding=1)
        self.bn4 = nn.BatchNorm2d(32)
        self.conv5 = nn.Conv2d(32, 64, 3, padding=1)
        self.bn5 = nn.BatchNorm2d(64)
        self.conv6 = nn.Conv2d(64, 64, 3, padding=1)
        self.bn6 = nn.BatchNorm2d(64)

        self.fc1 = nn.Linear(64 * 8 * 8, 1024)
        self.bn7 = nn.BatchNorm1d(1024)
        self.fc2 = nn.Linear(1024, 512)
        self.bn8 = nn.BatchNorm1d(512)
        self.fc3 = nn.Linear(512, 156)
    def forward(self, x):

        x = F.relu(self.bn1(self.conv1(x)))
```

```
                x = self.pool1(F.relu(self.bn2(self.conv2(x))))

                x = F.relu(self.bn3(self.conv3(x)))
                x = self.pool1(F.relu(self.bn4(self.conv4(x))))

                x = F.relu(self.bn5(self.conv5(x)))
                x = self.pool1(F.relu(self.bn6(self.conv6(x))))

                x = x.view(-1, 64 * 8 * 8)
                x = F.relu(self.bn7(self.fc1(x)))
                x = F.relu(self.bn8(self.fc2(x)))
                x = F.softmax(self.fc3(x), dim=1)
                return x
net = Net()
net.load_state_dict(torch.load("tamil_net.pt",
map_location=torch.device('cpu')))
net.eval()
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    string_data = request.get_data().decode('utf-8')
    prediction = inference.get_prediction(string_data, net)
    return prediction
@app.route('/suggest', methods=['GET', 'POST'])
def suggest():
    suggestion = random.choice(classes)
    return suggestion
```

SNAPSHOTS

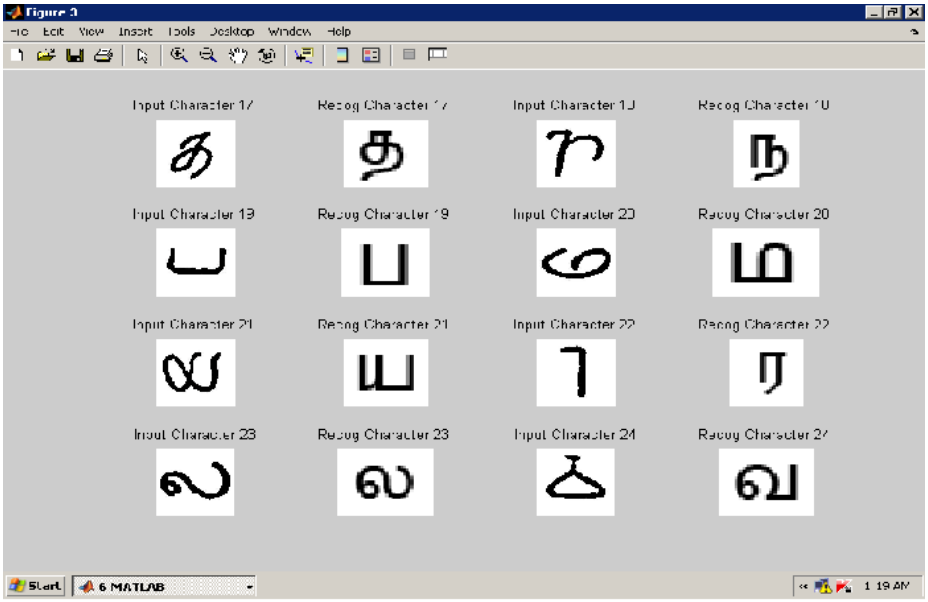
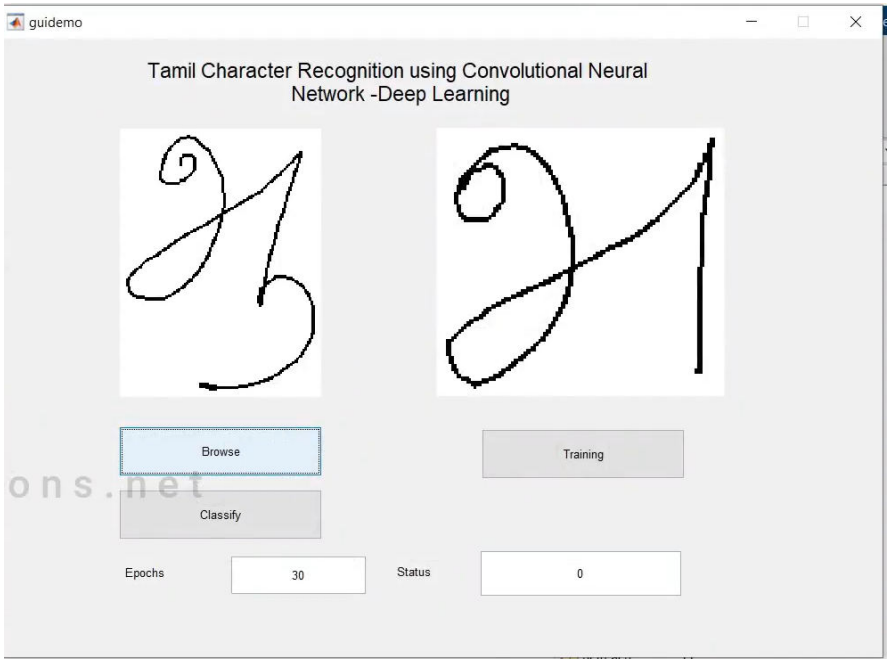


Fig 3 Output-2



CONCLUSION

We have presented a system for recognizing handwritten Tamil characters. Experimental results shows that good recognition of accuracy. The results of structure analysis shows that if the number of hidden nodes increases the number of epochs taken to recognize the handwritten character is also increases. The methods described here for Tamil handwritten character recognition can be extended for other Indian scripts by including few other pre-processing activities like line segmentation and character segmentation. Our future work aims to improve classifier to achieve still good recognition rate and also to perform experimental works for larger data set.

FUTURE ENHANCEMENTS

In recent times, with the increase of convolutional Neural Network (CNN), deep learning. This project is to observe the variation of accuracies of CNN to classify handwritten Tamil character using various characters of hidden layers and epochs and to make the comparison between the accuracies. This performance evaluation of CNN, we performed our experiment.

REFERENCES:

[1] K. H. Aparna, Vidhya Subramanian, M. Kasirajan, G. Vijay Prakash, V.S. Chakravarthy and Sriganesh Madhvanath, "Online Handwriting Recognition for Tamil", Proceedings of the 9th IEEE International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004.

[2] R. M. Bozinovic and S. N. Srihari, "Off-line cursive script word recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence., vol. 11, no. 1, pp. 68-83, Jan. 1989.

[3] Y.Y. Chung, and M.T. Wong, "Handwritten Character Recognition by Fourier Descriptors and Neural Network", Proceedings of IEEE TENCON – Speech and Image Technologies for Computing and Telecommunications, 1997.

[4] M. Hanmandlu, K.R.M. Mohan and H. Kumar, "Neural-based handwritten character recognition", Proceedings of 5th IEEE International Conference on Document Analysis and Recogniton, ICDAR'99.

[5] S. Hewavitharana and H.C. Fernando, "A Two Stage Classification Approach to Tamil Handwriting Recognition", Proc. The Tamil Internet 2002 Conference, California, USA, pp.118-124, 2002.

[6] Hongbong Kim and Kwanghee Nam "Object Recognition of One-DOF Tools by a Back-Propagation Neural Net", IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 484-487, March 1995.

[7] Niranjan Joshi, G. Sita, A.G. Ramakrishnan, and Sriganesh Madhvanath, "Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition", Proceedings of the 9th IEEE International Workshop on Frontiers in Handwriting Recognition, IWFHR-9 2004.

[8] N. Shanthi and K. Duraiswamy, "Preprocessing Algorithms for the Recognition of Tamil Handwritten Characters", 3rd International CALIBER – 2005, Cochin, pp.77-82, Feb. 2005.

[9] R.M. Suresh, S. Arumugam and L. Ganesan, "Fuzzy Approach to Recognize Handwritten Tamil characters", Published by IEEE, ICCIMA'99, New Delhi, pp.459-464, 1999.

AUTHOR 1

Mrs. A. Sunitha M.E., is an Assistant Professor in the Department of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. She has completed her M.E, CSE under Anna University Affiliation College in the year 2014. she has done her B.E, CSE under Anna University Affiliation College in the year 2012. Mrs.A.Sunitha has 08 years of teaching experience and 8 publications in International Journals and Conferences. Her area of interests includes Network Security, Computer Networks, Data Science and Machine Learning. She is an active member of CSI. She has organized various International Conferences, workshops and Seminars in the area of Computer Networks, Cloud Computing & Machine Learning respectively.

AUTHOR 2

Mr. T. M. Midhun B.E., Student of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops and Seminars in the area of Python

AUTHOR 3

Mr. A. Herman Josh B.E., Student of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops and Seminars in the area of Python and Machine Learning.