EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

Effective Tamil Character Recognition Using Supervised Machine Learning Algorithms

Dr. S. Suriya[1,*], S. Nivetha[2], P. Pavithran[2], Ajay Venkat S.[2], Sashwath K. G.[2], Elakkiya G.[2]

[1]Associate Professor, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, India
[2]UG Scholar, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, India

ABSTRACT

Computational linguistics is the branch of linguistics in which the techniques of computer science are applied to the analysis and synthesis of language and speech. The main goals of computational linguistics include: Text-to- speech conversion, Speech-to-text conversion and Translating from one language to another. A part of Computational Linguistics is the Character recognition. Character recognition has been one of the active and challenging research areas in the field of image processing and pattern recognition. Character recognition methodology mainly focuses on recognizing the characters irrespective of the difficulties that arises due to the variations in writing style. The aim of this project is to perform character recognition for of one of the complex structures of south Indian language 'Tamil' using a supervised algorithm that increases the accuracy of recognition. The novelty of this system is that it recognizes the characters of the Predominant Tamil Language. The proposed approach is capable of recognizing text where the traditional character recognition systems fails, notably in the presence of blur, low contrast, low resolution, high image noise, and other distortions. This system uses Convolutional Neural Network Algorithm that are able to exact the local features more accurately as they restrict the receptive fields of the hidden layers to be local. Convolutional Neural Networks are a great kind of multi-layer neural networks that uses back-propagation algorithm. Convolutional Neural Networks are used to recognize visual patterns directly from pixel images with minimal preprocessing. This trained network is used for recognition and classification. The results show that the proposed system yields good recognition rates.

Keywords: Computational Linguistics, Character recognition, distortions, Convolutional Neural Networks, Multi-layer neural networks, back-propagation algorithm, pixel images, preprocessing, trained network.

Received on 11 May 2020, accepted on 16 December 2020, published on 08 February 2023

doi: 10.4108/eetel.v8i2.3025

1. INTRODUCTION

Computational linguistics is the branch of linguistics in which the techniques of computer science are applied to the analysis and synthesis of language and speech. Computational linguistics is used in instant machine translation, speech recognition (SR) systems, text-to-speech (TTS) synthesizers, interactive voice response (IVR) systems, search engines, text editors and language instruction materials. The main goals of computational linguistics include:

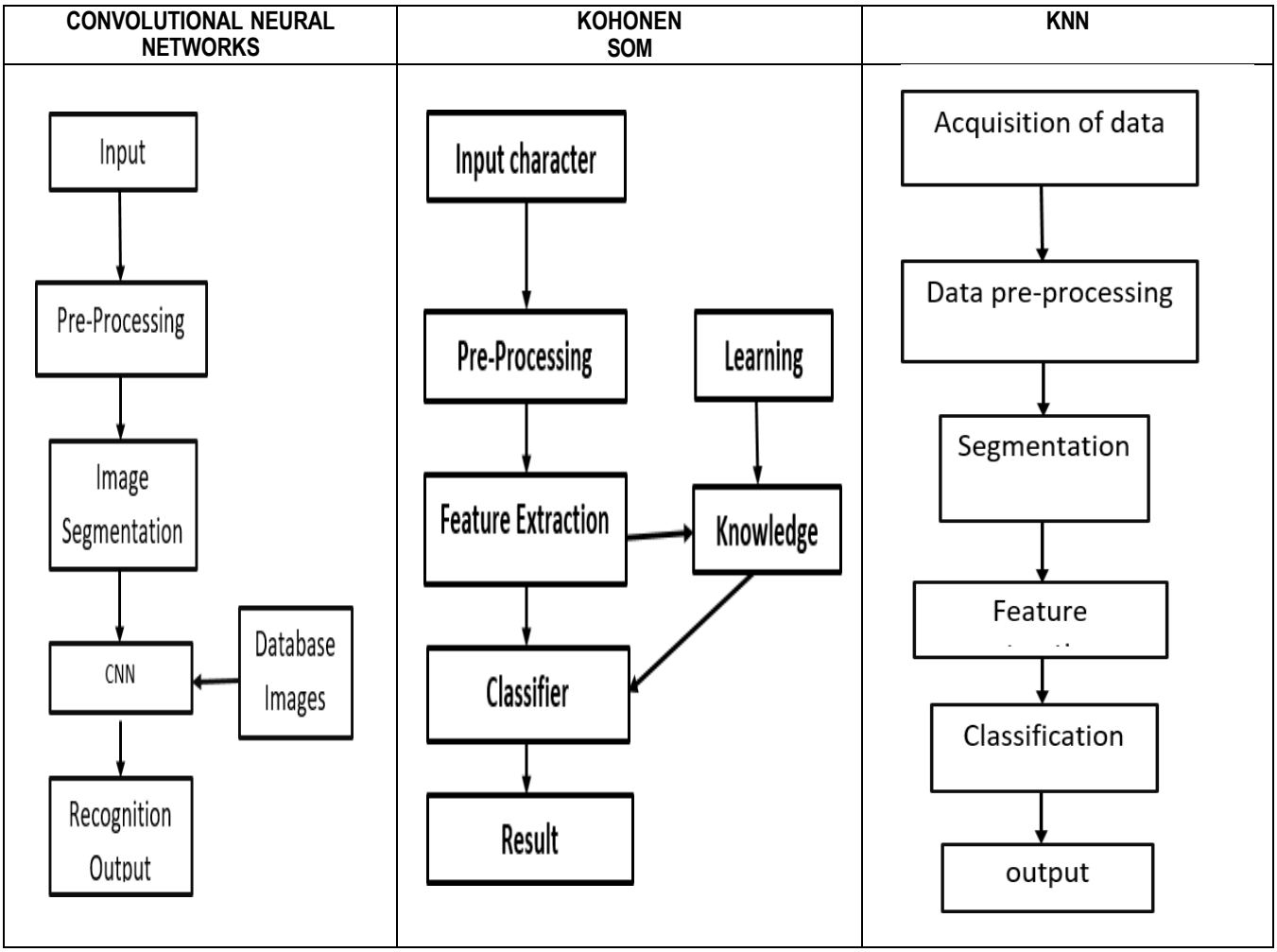
- Text-to-speech conversion
- Speech-to-text conversion
- Translating from one language to another

Out of these goals, this project deals with the first part of text-to-speech conversion, which is recognition of text, into which we focus on recognition of characters. A lot of research work has been done in this area. Most of these investigations have been on English, Chinese and Japanese

*Corresponding author. Email: suriyas84@gmail.com, ss.cse@psgtech.ac.in

character recognition. Comparatively fewer efforts are made on Indian languages such as Tamil, Telugu, Bangla, Devanagari. Handwritten character recognition is a sub area of the character recognition. Even though a lot of research has been done on this area, there is very less exposure over Tamil language. So our goal is to recognise Tamil characters for applying a best supervised algorithm to get a good accuracy of recognition.

2. BASIC STUDY OF ALGORITHMS



2.1 CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics. The role of the CNN is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

2.2 KOHONEN SELF ORGANIZING MAPS

A self-organizing map (SOM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method that helps in dimensionality reduction. Self-organizing maps differ from other artificial neural networks as they apply competitive learning instead of error-correction learning.

2.3 KNN

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm

that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity.KNN captures the idea of similarity, thus calculating the distance between points on a graph.It then adds the distance along with the index to an ordered collection, sorts it then picks the labels of first K entries. If regression, it returns mean of these labels else if classification, it returns mode of the labels.

2.4 SUPPORT VECTOR MACHINE

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

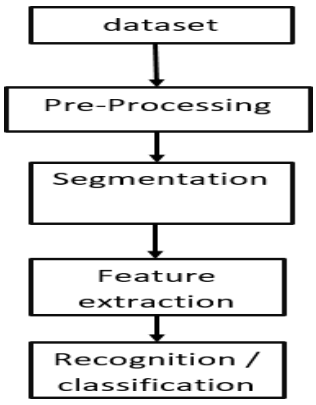
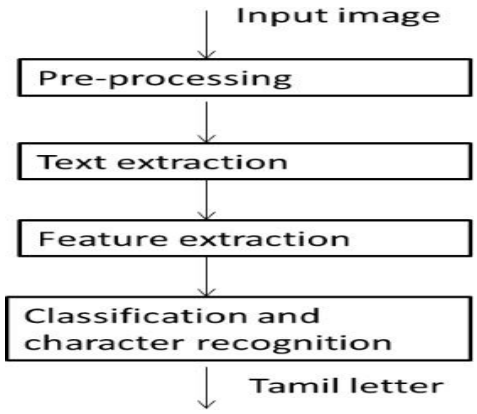
2.5 BAYES CLASSIFIER

A Bayesian classifier is a probabilistic model where the classification is a latent variable that is probabilistically related to the observed variables. Classification then become inference in the probabilistic model.Bayes Theorem provides a principled way for calculating conditional probabilities, called a posterior probability.It involves calculating the conditional probability of one outcome given another outcome, using the inverse of this relationship, stated as follows:

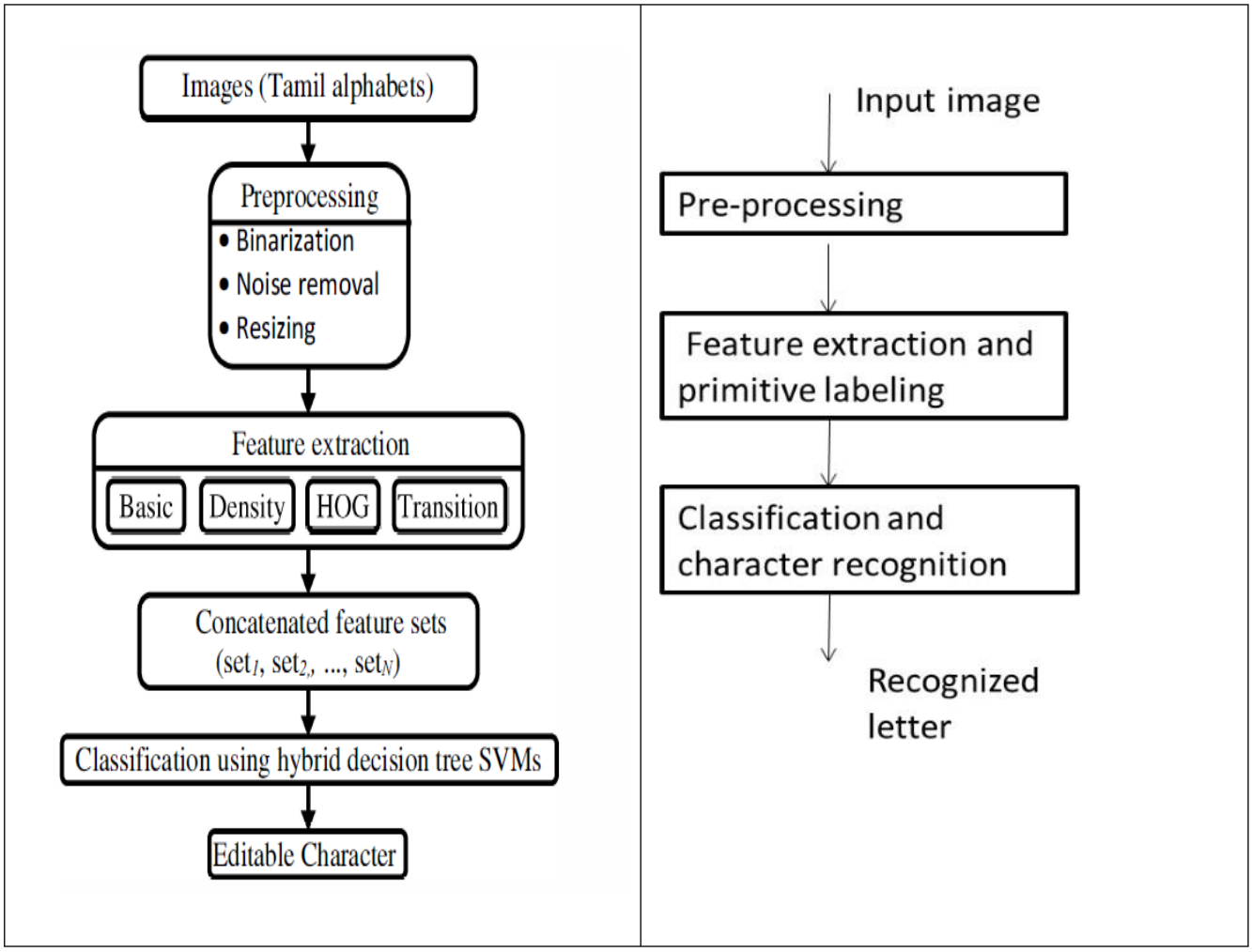
$$P(A \mid B) = (P(B \mid A) * P(A)) / P(B)$$

2.6 FUZZY APPROACH

Fuzzy logic algorithm helps to solve a problem after considering all available data. Then it takes the best possible decision for the given the input. The FL method imitates the way of decision making in a human which consider all the possibilities between digital values T and F. The term fuzzy mean things which are not very clear or vague. In real life, we may come across a situation where we can't decide whether the statement is true or false. At that time, fuzzy logic offers very valuable flexibility for reasoning.

SUPPORT VECTOR MACHINES	BAYES CLASSIFIER
	

SVM ORGANISED IN A HYBRID DECISION TREE	FUZZY APPROACH



3. LITERATURE SURVEY

Deepa et al.[1] attempts to recognize handwritten characters for Tamil alphabets without feature extraction using multilayer Convolutional neural networks(CNN). This system was proposed to work efficiently for different kinds of text appearances, including font styles and sizes. The first phase is the pre-processing that involves reading the input, image resizing, grey scale conversion, noise removal and binary image conversion. The next phase is segmentation in which the digital image is partitioned into multiple segment. The database used here has two parts, one which has images of different letters in different styles of writing and the other has the images of letters to be displayed in the output. CNN is has a series of layers which filters the input to yield the output with high accuracy of about 90.19% than any other neural network.

Vani et al.[2] proposed soft computing approaches for character credential and word prophecy analysis with stone encryptions. The proposed system focuses on recognition of eleventh-century ancient Tamil character and converting them into current-century character. Image capturing, the first step, is the procedure of making a digital image directly by using a scanner. Secondly, Image pre-processing deals with enhancing the quality of the image and making it ready for the segmentation process. The pre-processing steps include resizing of images and binarization. Then the binarized image is segmented using line and character segmentation using horizontal and vertical projection. The horizontal projection extracts the line from the stone inscription, whereas the vertical projection extracts a particular character. The segmented image undergoes a hybrid feature extraction technique, that extracts useful information from the image and omit unessential information, along with Chisquare test to check whether entire pixel in image of Zernike is bounded inside the unit circle or not, whereas ANOVA method is used for testing the significant difference between HOG feature and zoning feature. These functions are subjected to image classification and proceeded with character recognition using convolutional neural networks containing five layers with two pooling layers and two fully connected layers. Finally, the identified character is progressed into word form with the help of boggle algorithm. The hybrid feature extraction along with convolutional neural networks is achieved with 92.78% of recognition rate accurately.

Ramya et al.[3] introduces ‘AGARAM’ – a web application of Tamil characters using Convolutional

Neural Networks and Machine Learning. This paper aims to explore the scope of neural networks and apply them to recognize Tamil characters. The input of Tamil characters are pre-processed using IrfanView, an image processing tool. The IrfanView smoothens the images with the pre-processing techniques, so that the accuracy of predictions can be improved. The training of the model used for recognition of Tamil characters is done using TensorFlow.js, a machine learning platform which provides a facility to construct neural networks and run process on it. The model consists of two convolutional layer setups, then a flattening layer and finally passed to a dense layer to generate the output. The convolutional layer setup involves a layer that has 5 X 5 pixel filter. The first layer has 8 filters and the second has 16 filters. This layer is bounded by an activation function. These are then passed to a max pooling layer that pools the generated data. The model predicts what the character might be, the output generated from the model is an array of values where each value is the likelihood of the image being a certain character. The value with the highest value is the character that the image is most likely to be. The overall accuracy of this model is 80%.

Kowsalya et al.[4] proposed an approach for the Recognition of Tamil handwritten character using modified neural network with aid of elephant herding optimization. In order to overcome low accuracy in recognition due to variation in size, the proposed technique utilizes effective Tamil character recognition. The proposed method has four main process such as pre-processing process, segmentation process, feature extraction process and recognition process. For pre-processing, the input image is first fed to Gaussian filter to remove the noise due to digital image processing. Binarization process for transforming the gray scale image into a black and white image through threshold method. Skew detection technique to detect the deviation of the text image from horizontal and vertical axis which includes dimension reduction and skew estimation. Segmentation is used to verify objects and boundary. It helps to focus only on the object. From the segmented output, the features are extracted. After that the feature extraction, the Tamil character is recognized by means of optimal artificial neural network. Here the traditional neural network is modified by means of optimization algorithm. The proposed method utilized 80% of Tamil character image for training process; the remaining 20% of Tamil character images are used for testing process. The proposed approach will be implemented in MATLAB. The performance of the proposed method is assessed with the help of the metrics namely Sensitivity, Specificity and Accuracy. The recognition rate is 93% which is greater than any other neural network techniques.

Kavitha et al.[5] proposed an approach for offline Handwritten Tamil Character Recognition using Convolutional Neural Networks. Character recognition is of two forms: Offline and online. Online methods converts the tip movements of the digital pen, whereas offline method uses scanned images. An offline HCR system is first trained with the set of characters and later when a new character image is given as input, the system should be able to recognize it accurately. The dataset taken consists of 82,929 images. The proposed method consists of two parts: training part and the recognition part. The training part include the data pre-processing, building the network architecture and training the network with pre-processed data. The recognition part includes pre-processing of the input image and recognizing the character using the trained model. The pre-processed image is first passed through the convolutional layer 1 that has 16 filters, the convolutional layer 2 that has 16 filters then followed by max pooling of stride 2 X 2. The image is then passed through layer 3 and layer 4 with 32 layers each and through a max pooling of stride 2 X 2. The image is then passed through layer 5 with 64 filters then to a fully connected layer containing 500 neurons then another layer containing 200 neurons. The output of the final layer is the character recognized. The accuracy of 97.7% is achieved using this approach.

M. Antony Robert Rajet al. [6] describes the Tamil handwritten character recognition using feature extraction. This paper deals with three ways of feature predictions that are used to grasp features from various Tamil characters possessing variations in style. This algorithm is designed for recognizing the characters which has curvy nature and capable to address all Tamil characters. The general pre-processing steps that includes binarization, noise removal, skeletonization and normalization that help to get a noiseless, thinned and standardized image are used. Here the feature extraction algorithm is applied by dividing the entire character portion into different images at the microlevel. Junction points have been used for the separation, for which eight directional chain code algorithms are preferred. For feature extraction three techniques have been used:quad tree, strip tree and Z-ordering. The features that are extracted from quad tree, strip tree and Z-ordering are discrete values. The discrete sequential information is gathered and given as input to the SVM classifier. SVM is used in hierarchical manner with divide- and-conquer approach to classify the correct character obtained from those three ways of features. Thus this combination can address unique 100 characters, but it will be challenging when the level of complexity is high.

Subashini et al. [7] describes a method for recognition of hand-written Tamil characters by using a set of SIFT feature vectors and K-Means Clustering. The first phase is the pre-processing of images by reducing noise, normalizing size, extraction and segmentation. The feature vectors of the images are generated by the SIFT algorithm based on a keypoint. 48 X 48 sized images were used as they yielded best results. K – means clustering is used to create a codebook based on nearest neighbour condition and centroid condition for optimality. A k- Nearest Neighbour classifier is used for classification and the highest accuracy was obtained for k=1.The algorithm were trained with 6000-character images belonging to 20 classes and tested for a set of 2000 characters. The best results were obtained for a codebook size of 1024 with a recognition rate of 87%.

Bhattacharya et al. [8] describes a two-stage off-line handwriting recognition by using K-Means Clustering and MLP classifier. The first stage deals with the classification of an input character into a small number of groups using K-Means Clustering. The second stage deals with the computation of chain code histogram features and a distinct MLP classifier is trained for each of the groups. The value for K in K-Means Clustering was taken as 25 since it was acceptable choice in terms of various factors. The overall recognition accuracy is 92.77% and 89.66% on the training set and test set respectively.

Elakkiya et al. [9] proposed an approach for Tamil text recognition by using KNN classifier. This involves a template creation stage where images of every letters are gathered and split into connected component images. They are converted to a common dimension and labelled. The character recognition stage involves pre-processing of images followed by segmentation and feature extraction. The classification is then performed using the correlation coefficient. The maximum correlated character is declared as the tested character. The k-nearest neighbour is used to classify the object being assigned to the class most common among its k nearest neighbours. This yielded an accuracy rate of 91%.

Liyanage et al. [10] used Tesseract engine helped them for developing robust tamil OCR. Preparation of data involves generation of OCR alphabets and it is defined by considering various glypgs in tamil letters. They created images of different size with same DPI value as resolution has no effect in accuracy. The character segmentation involves creation of text file which contains sufficient information of training image and the file is called as box file created in tesseract. They trained the modules for each and every data sets using training images of different font size and type and combined those data sets. Training the modules requires training image of all three font size and corresponding box files were used. With this training data training process is done and file which contains features of the letter ,then at next stage of process file font properties and the unicharset file were generated the former is used to enhance accuracy. To evaluate the system, we compared its performance with the existing Tamil module in Tesseract. They found that it was inappropriate for some letters. Their system gave an accuracy of 81%. This is 12.5% improvement on existing Tesseract Tamil module's performance.

Manisha et al. [11] uses Bayes classifier to classify and recognise letters. At first the image is pre-processed to reduce noise, this is done using median filter and then binarization is done since letters are present at foreground of the image. At next phase they extract the text and segment them, here space based techniques are used to extract lines and words. Each characters are identified using Bounding box technique. Other characters are removed using Morphological operator such as dilation. The segmented words are made into same size. Features like horizontal and vertical lines, curves are found using Sobel mask. After all these steps the letter is found using Bayes theorem. For each class universal probability is found and test character's probability is found using the feature extracted and this value is compared with universal value and then Unicode is used for recognition and for dispalying. The disadvantage is similar looking characters are not classified properly. The accuracy was 96.3% (measured by F score).

Ramanan et al. [12] describes a novel approach for muIticlass classification to recognise Tamil characters using binary support vector machines (SVMs) organised in a hybrid decision tree. The first phase involves pre-processing which has three phases such as Binarization, noise removal and resizing. The second phase is the feature extraction phase which extracts the basic, Density, HOG and Transition features. The last phase involves Classification where the extracted feature vectors are analyzed using novel hybrid decision tree of DAG and UDT SVMs. They have taken about 12400 samples of data for their algorithm. This algorithm gives about 98.08% accuracy.

Shivsubramani et al. [13] describes a method for recognition of hand-written Tamil characters by using MultiClass Hierarchical support vector machines. The first phase involves Pre-processing in which binarization was performed based on a threshold value applied on the image. Second phase will be Segmentation that will be divided into line and character segmentations. Third phase will be Feature extraction that explores the information across an entire image. Fourth phase will be Hierarchical labelling which will be done for similar characters recognition. They have taken about 20 sample training data for a particular class. The accuracy of the algorithm will be around 96.23 to 96.86 depending among the number of characters provided.

Stephen et al.[14] proposed a novel method for pattern recognition problems in terms of linear regression. They developed a linear regression classification algorithm that works on the nearest subspace approach. The images are converted into greyscale and split into N number of distinguished classes. The images are down-sampled to a particular order and converted into a vector by using column concatenation. The subspace of each vector is created. Each class is represented by a vector subspace called as regressor or predictor. The image to be classified is pre processed and if the image belongs to a particular class, then it should be represented as a linear combination of training images from the same class. The distance between the predictor vector and the class vector is calculated using Euclidian Distance. The class which provides the minimum distance is selected. They created their own dataset with each image scanned at a resolution of 300 dpi. The dataset consists of 100 images of a single character for 12 characters which totals to 1200 images. They achieved an accuracy of 91% with a better time complexity and space complexity.

Suresh et al. [15] proposed an approach for Tamil text recognition using fuzzy technique. This involves procedure for segmentation and then classification of character. Segmentation is done on basis of apriori knowledge of characters and features considered for classification. The

character is converted to two tone image. By membership function the segmented image is classified either as line or arc. For 16 directions distance from the frame to the point where the direction hits the image is measured, thus a vector like (d1,d2,…,d16) is found. Similarly, through the two tone converted character, five different vectors are obtained from five different positions in the frame. Then they normalized the feature vector and the resulting vector is called as normalized feature vector (NFV). Following are methods used by them to classify the characters.

Let x = (nx1 ,…,nx 16) be the NFV of an input character. Let μ(x,y) = 1- δ [∑16 (nxi –nyi)1/2] where y∈YL if x is of

line pattern y∈YA if x is of arc pattern and Number of segments in x for which match is found in y and

δ = $Number\ of\ segments\ in\ x\ for\ which\ match\ is\ found\ in\ y.$

Total number of segments in x

x is classified as the prototype character y for which μ(x,y) is maximum. The main advantage is it could recognize characters even when it is tilted upto 30 degrees. Efficiency of this approach were from 88% to 100%.

Suresh et al. [16] proposed an approach for Tamil text recognition using fuzzy technique. This involves pre-processing of input image, feature selection and then recognition is done. For feature selection intersection, curvarure, loops, etc are identified. The membership functions μh, μv for horizontaland vertical strokes are found. Algorithm for fuzzy context free grammar inference is defined and the input for this algorithm is set of sample texts and output is production rules for the grammar and then algorithm for fuzzy membership value is defined whereininput for this is productions generated from former algorithm. For designing parser they just modified Earley's parsing algorithm, modification includes assigning weight to new value, computation weight is considered and accounting for cases where grammar G is ambiguous. During character matching phase if the character is not matched with any pre-existing character in database the character is sent to parser for recognizing the character and it declares character class. The approach used by them is two-stage recognition technique wherein at first stage for feature selection fuzzy logic is used and for second stage fuzzy grammar parsing is done. The efficiency ranged from 90% to 100%.

Rituraj et al. [17] proposed technique of classifying tamil characters through online semi-supervised learning. They have pre-processed the input image and then feature selection is done. For solving the problem of unlabelled data they used Expectation Maximization (EM). Posterior probability of data points is calculated at E-step and in M-step they have calculated parameters of learning model. Since the proposed model is online they have used procedure which updates the model in an evolutionary and a continual manner. They have introduced a regulating constant

λ which helped them in moderating the reducing the learning rate (η) of unlabelled data and hence the weight of the unlabelled samples during step M. Due to higher posterior value qk, η increases for correct class sample. They trained Random Naive Bayes (RNB) online with a few labelled training samples and then following procedure is repeated for labelled or unlabelled data. For E-step, if input sample is unlabelled then they used trained classifier to find the posterior $qk = (Y = yk|X)$ for all k where k represents class. Else qk =1 then for M-step they found following parameters $ck = ck + qk\lambda$, $\eta k = qk((1-\alpha\ /ck)+\alpha)\lambda$, $\mu ik(t) = (1-\eta k)\mu ik(t-1)+\eta k x j i \delta(Yj = yk)$. In this method the efficiency is more than Naive Bayes classifier.

Gandhi et al. in [18] proposed the usage of Kohonen Neural Network based Self Organizing Maps to recognize Handwritten Tamil Character. Data samples were collected and pre-processed. Pre-processing includes resizing the images to 205 x 250 pixels and removal of noise by spatial filter. The preliminary classification grouped the characters into two groups namely, Crux Characters and Exhaustive characters which was further divided into Ascending and Descending. For Feature Extraction, the images are scaled to a size of 32 x 32 pixels using a bilinear interpolation technique. Unwanted portions are removed using Sobel's mask. The vectors are created for each image. The input vector is clustered by calculating the weight of the neuron and the weight vector that comes close to the input vector is chosen as the output. The neurons within a certain neighbourhood of the output neuron are updated. The dataset consists of 200 training samples and 800 testing examples. Accuracy ranging from 89.5% to 98.5% was achieved based on the number of datasets trained and tested.

Banumathi et al.[19] proposes an approach to recognize the handwritten Tamil characters using artificial neural networks approach. In the proposed system the scanned image is pre-processed and segmented into paragraphs, paragraphs into lines, lines into words and words into character image glyph. The first phase of the algorithm is Scanning which involves obtaining a digitized image from a real world source. And then comes the Pre-processing phase where the scanned copies are pre processed and this procedure involves three steps such as Binarization,noise removal and Skew correction.Third phase will be the segmentation phase where the noise free image and skew corrected image is passed,where the image is decomposed into individual characters. The next phase to segmentation is feature extraction where each character is represented as a feature vector, which becomes its identity. Feature extraction forms the backbone of the recognition process.The last phase will be SOFM which is Kohonen's self Organizing feature map.This phase involves classification of documents.This algorithms has f=given more than 80% accuracy for the samples tested.

Dr.J.Venkatesh et al.[20] proposed the Tamil characters recognition using the Kohonon's self Organizing Map.This algorithm finds applications in document analysis where the handwritten document can be converted to editable printed document. This character recognition will be

considered under the following phases. The first phase is Character recognition Functions I, which includes scanning, pre-processing, segmentation and feature extraction.In the Scanning phase the printed document will be scanned using OCR.Then comes the pre-processing phase where the scanned image pre-processed for noise removal. Here the image is first brightened and binarized.After that, Skew detection and Correction will take place. The angle of the Skew will be measured here. After pre-processing, the noise free image is passed to the segmentation phase, where the image is decomposed into individual characters. The next phase to segmentation is feature extraction where individual image glyph is

considered and extracted for features. The second phase of the Character Recognition functions consists of classification and Unicode mapping and recognition strategies.Kohonon's SOM will be done along with the Unicode mapping.

3.1 ADVANTAGES AND DISADVANTAGES IN SUPERVISED ALGORITHMS

The following table represents the pros and cons of the various supervised algorithms. These results are obtained from the literature survey.

ALGORITHM	CNN	KOHONEN SOM	KNN CLASSIFIER	FUZZY APPROACH
PROS	+ Provides high accuracy in image recognition problems of about 90.19%	+Data mapping is easily interpreted +Capable of organizing large, complex data sets	+ No assumptions about data + Simple algorithm + High accuracy -91%	+similar to human reasoning. +Based on linguistic model. +high precision +rapid operation
CONS	-High computational cost. - If you don't have a good GPU they are quite slow. -They use to need a lot of training data.	-Difficult to determine what input weights to use -Mapping can result in divided clusters -Requires that nearby points behave similarly	-Computationally expensive (Stores all of the training data). -High memory requirement -Prediction stage might be slow	- Lack of real time response - Restricted number of usage of input variables. - The lower speed and also longer run time of the system

ALGORITHM	BA YES	SVM	K-MEANS
PROS	+It is easy and fast to predict class of test data set. +It also perform well in multi class prediction. + accuracy obtained was 96.3%	+SVM is more effective in high dimensional spaces. +SVM is effective in cases where number of dimensions is greater than the number of samples. +SVM is relatively memory efficient. +accuracy obtained was 96.23%	+ Relatively simple to implement. +Scales to large data sets. + Easily adapts to new examples. + accuracy – 88%
CONS	-If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction known as "zero frequency"	-SVM algorithm is not suitable for large data sets. -SVM does not perform very well, when the data set has more noise i.e. target classes are overlapping. In cases where number of features for each data point exceeds the number of training data sample, the SVM will under perform.	- Choosing k manually. - Being dependent on initial values. - Scaling with number of dimensions.

4. DESIGN OF PROPOSED SYSTEM

System Design is the process of defining the architecture for a system to satisfy the specified requirements. System design is the process of designing the elements of the system such as the architecture, modules and the components of the system, the different interfaces of those components and the data that goes through the system.

4.1 CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

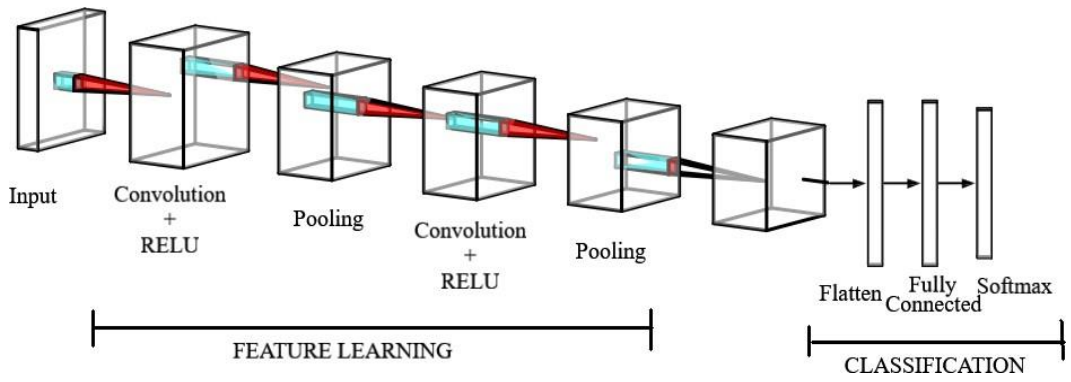


Figure 1. Convolutional Neural Network Architecture

4.2 BLOCK DIAGRAM FOR CONVOLUTIONAL NEURAL NETWORK

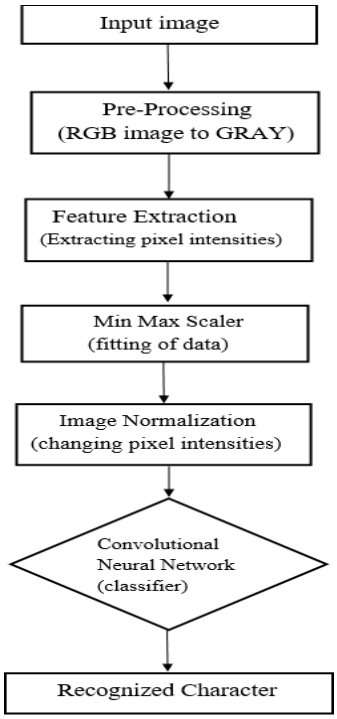


Figure 2 Block Diagram For Character Recognition System

The input image is passed to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolutional layer extract relevant features from the input image to pass further. Pooling layers are then added to further reduce the number of parameters. Several convolution and pooling layer are added before the prediction is made. Convolutional layer help in extracting features. The output layer in a CNN is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network. The output is then generated through the output layer and is compared to the output layer for error generation. A loss function is defined in the fully connected output layer to compute the mean square loss. The error is then back propagated to update the filter (weights) and bias values. One training cycle is completed in a single forward and backward pass.

4.3 UML USECASE DIAGRAM

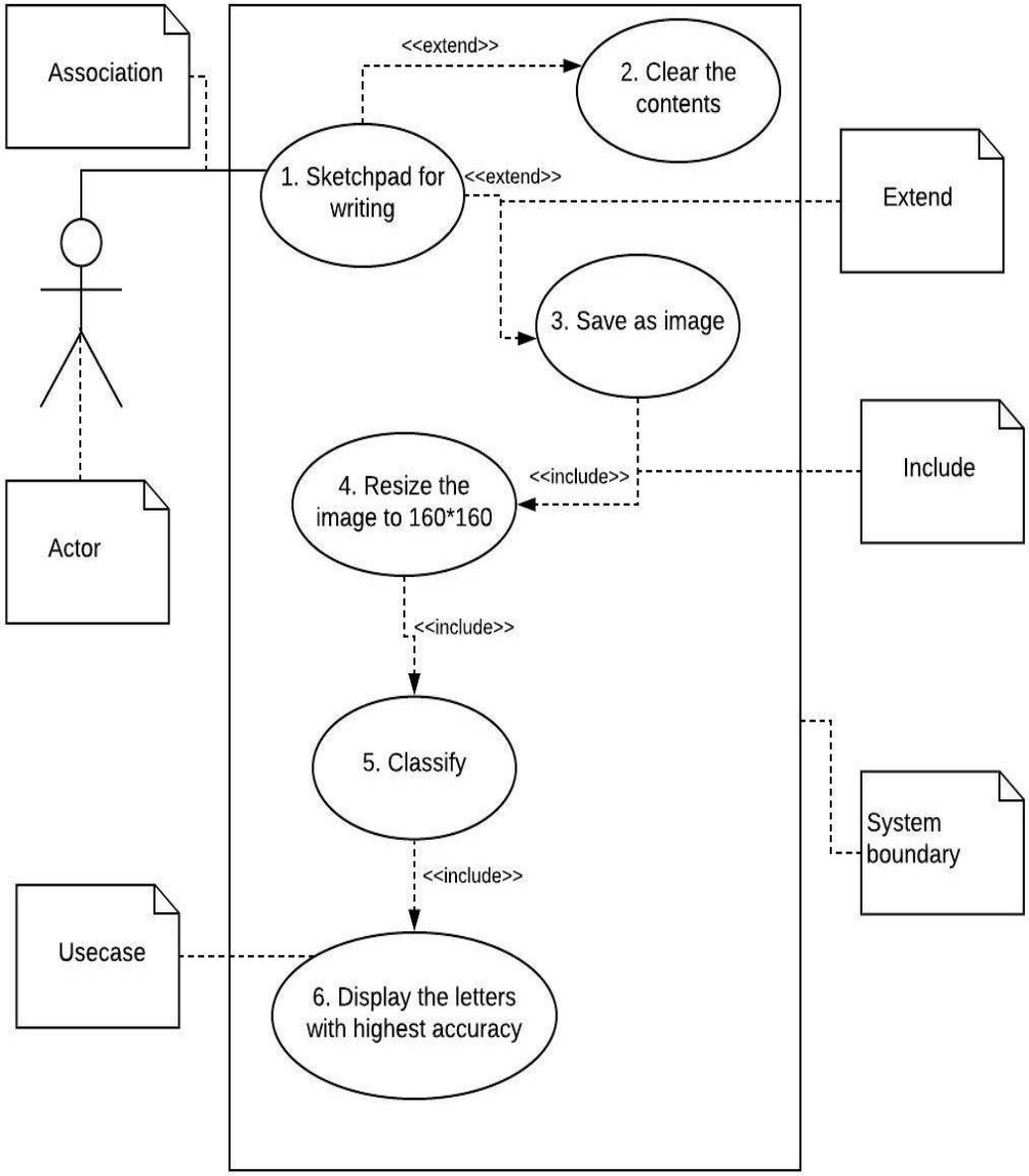


Figure 3. UML Use case diagram for Character Recognition System

5. EXPERIMENTAL RESULTS

Usecase Number	1	2	3	4	5	6
Brief Description	User will be able to write the letters	User can clear the contents in the scratch pad	User can save the letter in the sketch pad as image	The saved image is resized for prediction	Trained model which helps in classifying letters	Output is displayed for the user
Actor involved	User	User	User	User	User	User
Pre-condition	User should have entered the system	User should have written something	User should have written a letter	User should have pressed save option	User should have written a letter	Testing image should have been resized
Main flow	User can write a letter	Clears the contents	Letter in scratch pad is saved as image	Image is re- sized	Comparison with features extracted	Display of result
Post-condition	Proceed to work with the system	Allows user to write new letter	The image is given for resizing	The resized image is given to testing for classification	Letter is predicted	Can close the system
Alternate flow	-	-	-	-	-	-

System implementation is the process of defining how a system should be built, ensuring that the system is operational and is easy to be used, and also ensuring that the system meets the quality standard (i.e. Quality assurance). An implementation is an realization of a technical specification or an algorithm as a program or as a software component, or as other computer system through computer programming and deployment.

MODULE 1 – COLLECTION OF DATASET

This Tamil character recognition project requires a lot of training data. The dataset consisting of approximately 51,800 were collected from the sources available. The dataset approximately contains 185 Tamil characters each having 280 samples, written by the native Tamil writers including school children, college students and adults.

Figure 4. Handwritten Tamil character dataset

TAMIL LANGUAGE

Tamil is a classical language and one of the major languages of the Dravidian language family. Tamil language is spoken predominantly by Tamils living in India, Sri Lanka, Malaysia, and Singapore. Furthermore, there are small communities of Tamil speaking people living in many other countries. As of 1996, it was the eighteenth most spoken language in the world, with over 74 million speakers worldwide. It is one of the official languages of India, Sri Lanka as well as Singapore. Tamil alphabet has 12 vowels, 18 consonant, combination vowels and consonant 216, and one Ayutha letter, totally 247 letters in Tamil 10 numerical symbols.

MODULE 2 – PREPROCESSING

This project focusses on the recognition of the vowels part of the Tamil characters. Hence those sample images are filtered from the dataset. All the images are initially converted to greyscale images. Some of the sample images were extremely damaged, which when used for training will actually worsen the learned model. Hence those samples were discarded. The original unequally sized rectangular images were resized to 160 X 160 sized square images using an online image resizer and stored as a JPG file.

Figure 5 Preprocessed image

MODULE 3 – TRAINING AND TESTING DATASET SPLIT UP

This module deals with splitting up of the dataset thus preprocessed into training and testing set. Since this project deals with the recognition of the vowels in Tamil language, a folder is created for each vowel, with each folder consisting of nearly 90 – 100 preprocessed images This forms the training dataset.

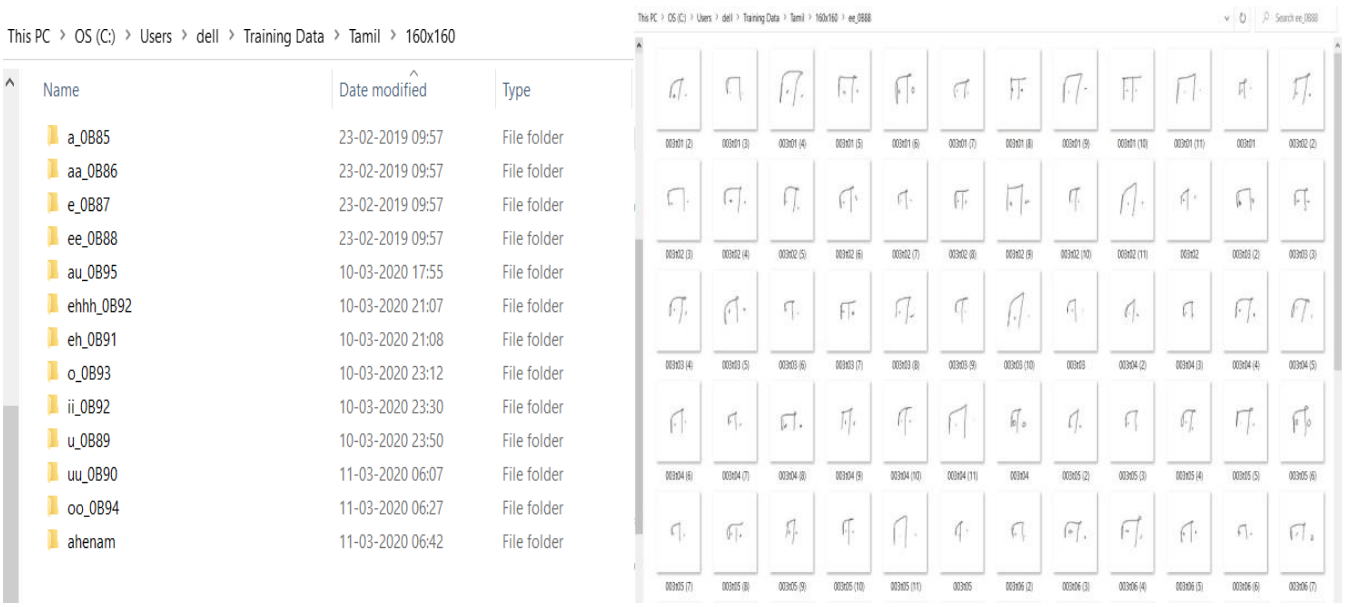


Figure 6. Training dataset

Similarly the testing dataset can also be prepared. Handwritten Tamil vowels written by an author were scanned and stored in another folder that can be used for testing the model. This testing dataset was created to test the model in the background, without the user interface.

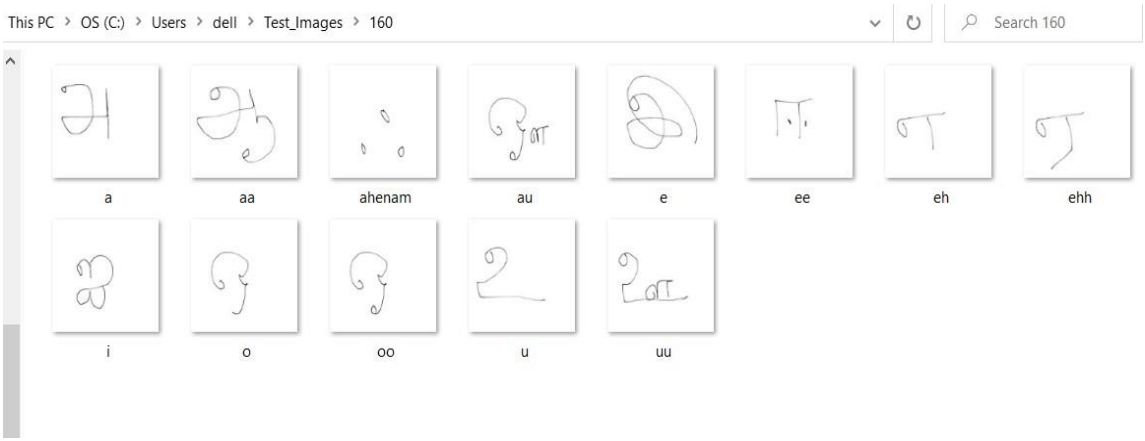


Figure 7. Testing dataset

MODULE 4 – CHOOSING SUITABLE MODEL

The Convolutional Neural Network algorithm is realized into a python program. There are two python scripts: one for training the dataset and the other is for classifying the test characters.

CONVOLUTIONAL NEURAL NETWORKS

Human brain is a very powerful machine. We see multiple images every second and process them without realizing how the processing is done. But, that is not the case with machines. In simple terms, every image is an arrangement of dots (a pixel) arranged in a special order. If you change the order or color of a pixel, the image would change as well. A weight matrix is defined which extracts certain features from the image. Sometimes when the images are too large, we would need to reduce the number

of trainable parameters. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is done for the sole purpose

of reducing the spatial size of the image. Pooling is done independently on each depth dimension; therefore the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling. Three hyper parameter would control the size of output volume.

(i) The number of filters – The depth of the output volume will be equal to the number of filter applied. Remember how we had stacked the output from each filter to form an activation map. The depth of the activation map will be equal to the number of filters.
(ii) Stride – When we have a stride of one we move across and down a single pixel. With higher stride values, we move large number of pixels at a time and hence produce smaller output volumes.
(iii) Zero padding – This helps us to preserve the size of the input image. If a single zero padding is added, a single stride filter movement would retain the size of the original image.

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size m−n+1. Each map is then subsampled typically with mean or max pooling over p x p contiguous regions where p ranges between 2 for small images (e.g. MNIST) and is usually not more than 5 for larger inputs. Either before or after the subsampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN consisting of convolutional and subsampling sublayers. Units of the same color have tied weights.

TRAINING THE MODEL

This module deals with training the model using the training dataset images that are preprocessed. Back propagation is the technique used in training the Convolutional Neural Network.

BACK PROPAGATION

Back propagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e.previous iteration). Proper tuning of the weights reduces the error rates and makes the model more reliable by increasing its generalization. It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

Let δ(l+1) be the error term for the (l+1)-st layer in the network with a cost function J(W,b;x,y) where (W,b) are the parameters and (x,y) are the training data and label pairs. If the l-th layer is densely connected to the (l+1)-st layer, then the error for the lth layer is computed as

$$\delta(l)=((W(l))T\delta(l+1))\cdot f'(z(l))$$

and the gradients are

$$\nabla W(l)J(W,b;x,y)\nabla b(l)J(W,b;x,y)=\delta(l+1)(a(l))T,=\delta(l+1)$$

.

If the l-th layer is a convolutional and subsampling layer then the error is propagated through as

$$\delta(l)k=upsample((W(l)k)T\delta(l+1)k)\cdot f'(z(l)k)$$

Where k indexes the filter number and f′(z(l)k) is the derivative of the activation function. The upsample operation has to propagate the error through the pooling layer by calculating the error w.r.t to each unit incoming to the pooling layer. For example, if we have mean pooling then upsample simply uniformly distributes the error for a single pooling unit among the units which feed into it in the previous layer. In max pooling the unit which was chosen as the max receives all the error since very small changes in input would perturb the result only through that unit.

Finally, to calculate the gradient the filter maps, we rely on the border handling convolution operation again and flip the error matrix δ(l)k the same way we flip the filters in the convolutional layer.

$$\nabla W(l)kJ(W,b;x,y)\nabla b(l)kJ(W,b;x,y)=\sum i=1m(a(l)i)*rot90(\delta(l+1)k,2),=\sum a,b(\delta(l+1)k)a,b.$$

Where a(l) is the input to the l-th layer, and a(1) is the input image. The operation (a(l)i)∗δ(l+1)k is the “valid” convolution between i-th input in the l-th layer and the error w.r.t. the k-th

The results of any classification task can be better analyzed using the following evaluation metrics.

ACCURACY

Accuracy is one of the evaluation metrics for text classification. Accuracy is the proportion of true results among the total number of cases examined. The formal definition for accuracy can be given as:

Accuracy = Number of correct predictions / Total number of predictions

For binary classification this can be given as,

Accuracy = (TP+TN) / (TP+FP+TN+FN_ where

- TP refers to True Positive
- TN refers to True Negative
- FP refers to False Positive
- and FN refers to False Negative.

F1-MEASURE

The F1 score is the harmonic mean of precision and recall. The F1 score can be given as follows: F1 = (2 * precision * recall) / (precision + recall)

AVERAGED PRECISION

The precision describes the ability of the classifier to not label a negative sample as positive. The precision score can be given as,

Precision = TP / TP+FP

The averaged precision score is used to summarize a precision recall curve as the weighted mean of precisions for each threshold with the increase in recall in the previous threshold used as the weight.

Averaged Precision = $\sum_{n}$ ($R_n - R_{n-1}$) P_n

where P_n denote the precision value at the n-th threshold R_n denote the recall value at the n-th threshold.

ROC CURVE

A Receiver Operating Curve is a graph plotted against the true positive and false positive rates which describes the performance of the classification model at all classification thresholds. True Positive Rate can be defined as follows:
True Positive Rate = TP / (TP + FN) False Positive Rate can be given as follows:
False Positive Rate = FP / (FP + TN)
AUC represents the Area Under the ROC curve. AUC ranges in value from 0 to 1. It provides an aggregate measure of performance across all possible classification thresholds.
Once the model is trained it creates separate folder for each character trained under the bottlenecks folder. These folders hold the feature points that are extracted from every sample image.

Figure 8. Folders in Bottleneck

A text document ‘retrained_labels’ containing the folder names of sample images that were trained, is created as the result of training the model. This text file is referred to while testing the model, in order to fetch those folders that must be taken into consideration for classifying the images. After referring to this text file, the corresponding folders present in the bottleneck folder is referred. The testing image is then classified into that character whose feature points matches it.

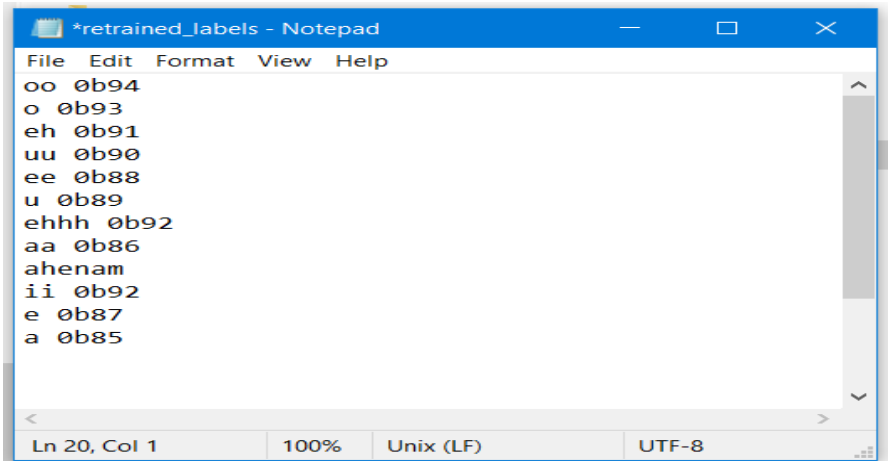


Figure 9. Retrained_labels

The trained model is stored in a PB File 'retrained_graph'. This file can be integrated in order to train the model in another system.

TESTING THE MODEL

The model was initially testing in the background. The testing dataset that was initially created can be used as testing images for testing the model. These images are fetched one after the other and tested. The output of testing each character will be three vowels which has highest accuracy.

To make the system more interactive, a user interface is designed. The user interface is a webpage that is designed using flask, HTML and JavaScript. This webpage allows the user to give a handwritten Tamil character as input, that can be tested the background to classify the input into the correct Tamil character. In order to get handwritten input form the user, a sketchpad is created. A sketchpad is similar to that of the white board, where the user could write any character using the mouse or the touchpad.

The webpage contains three buttons, clear image, to erase the input given in the sketchpad, save image, to accept the image and resize the image to a required size and classify image, that triggers the background execution.

Once the handwritten input is given by the user, the input is extracted as an image and the image is resized to 160 x 160. The resized image is stored in the directory C:\Users\dell\Test_Images\160, so that the image can be fetched for classification. The resized image is fetched from the directory and is tested in the background. Once the image is tested, as in case of background execution, the five characters whose accuracy is highest is stored in a text file 'temp.txt', which is then fetched by the frontend code and is displayed in the table in webpage.

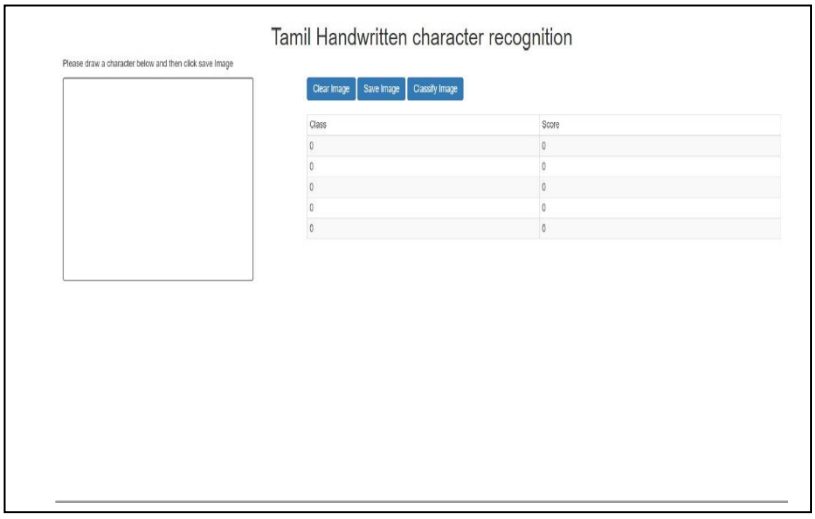


Figure 10. User interface

6. CONCLUSION

The paper discusses in detail all advances in the area of Tamil character recognition. The most accurate solution provided in this area directly or indirectly depends upon the quality and accuracy provided by the method. Various techniques have been described in this paper for character recognition in Tamil character recognition system. A comparison is shown between the different methods proposed the table. From the study done so far, it is analysed that the selection of the classification as well as the feature extraction techniques needs to be proper in order to attain good rate in recognizing the character. Studies in the paper reveals that there is still scope of enhancing the algorithms as well as enhancing the rate of recognition of characters.

A lot of research works exist in the survey for Handwritten Tamil character recognition. However, there is standard solution to identify all Tamil characters with reasonable accuracy. Various methods have been used in each phase of the recognition process. Challenges still prevails in the recognition of normal as well as abnormal writing, slanting characters, similar shaped characters, joined characters, curves and so on during recognition process. In this paper, our team has projected various aspects of each phase of the Tamil character recognition process. This project mainly focusses on a particular part of Tamil characters (i.e. uyir eluthukkal). Coverage is not given for different writing styles and font size issues. The following key challenges can be further explored in the future. As a result, the proposed system has been found to yield the highest recognition accuracy of 95.3%. The handwritten Tamil character recognition system described in this paper will find potential applications in handwritten character recognition. The proposed architecture has shown enhanced performance in recognizing the Tamil character.

APPROACH	ACCURACY	PURPOSE
offline Handwritten Tamil Character Recognition using Convolutional Neural Networks[5]	Overall accuracy of 97.7%	to utilize the CNN technique to achieve good recognition results on both training and testing datasets.
Tamil handwritten character recognition using feature extraction[6]	Around 85%	Deals with three feature extraction techniques in order to grasp features from various Tamil characters possessing variations in style and shape
Tamil text recognition by using KNN classifier[9]	Overall 91%	to get an efficient output and this approach has increased the speed and accuracy of character recognition.
Effective Printed Tamil Text Segmentation and Recognition Using Bayesian Classifier[11]	Overall accuracy of 96.3%	To recognize Tamil characters irrespective of the characteristics of the text such as font style, color, and size.
novel approach for multiclass classification to recognise Tamil characters using binary support vector machines[12]	About 98.08%	Each node of the hybrid decision tree exploits optimal feature subset in classifying the Tamil characters effectively.
novel method for pattern recognition problems in terms of linear regression[14]	Around 91%	To effectively recognize the Tamil characters using Linear Regression that works on nearest subspace approach
Tamil text recognition using fuzzy technique[16]	Accuracy ranged from 90%-100%	to recognize cursive Tamil handwritten words with fuzzy logic
Kohonen Neural Network based Self Organizing Maps to recognize Handwritten Tamil Character[18]	Accuracy ranges from 89.5% to 98.5%	To yield promising and feasible output withhigher performance than other existing techniques.

7. FUTURE WORK

This survey was mainly done for choosing an algorithm that suites best for the recognition of Tamil characters irrespective of the variations in size, style, etc. The best algorithm is chosen based on training and testing accuracy. The algorithm that gives a higher training and testing accuracy is chosen from those that give good accuracy found through literature survey, by executing the algorithms. The future enhancement will be recognition of

Tamil characters in an effective way yielding a good accuracy.

This paper deals with the first part of text-to-speech conversion, which is recognition of text, into which our team focus more on recognition of characters. This paper can be further enhanced to recognize all the other characters in Tamil language. The future prospect of this paper would be creating the experimental datasets and recognizing the words and sentences. Once sentences are recognized, the project would be further enhanced to implement computational linguistics which is text-to-speech conversion.

REFERENCES

[1] Vani, V. and Ananthalakshmi, S.R., Soft computing approaches for character credential and word prophecy analysis with stone encryptions. *Soft Computing*, pp.1-14

[2] Ramya, J., Kumar, G.K.R. and Peniel, C.J., 2019, March. 'Agaram'–Web Application of Tamil Characters Using Convolutional Neural Networks and Machine Learning. In *International Conference on Emerging Current Trends in Computing and Expert Technology* (pp. 670-680). Springer, Cham

[3] Kowsalya, S. and Periasamy, P.S., 2019. Recognition of Tamil handwritten character using modified neural network with aid of elephant herding optimization. *Multimedia Tools and Applications*, *78*(17), pp.25043-25061.

[4] Kavitha, B.R. and Srimathi, C., 2019. Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. *Journal of King Saud University-Computer and Information Sciences*.

[5] Raj, M.A.R. and Abirami, S., 2019. Structural representation-based off-line Tamil handwritten character recognition. *Soft Computing*, pp.1-26.

[6] Subashini, A. and Kodikara, N.D., 2011, August. A novel SIFT-based codebook generation for handwritten Tamil character recognition. In 2011 6th International Conference on Industrial and Information Systems (pp. 261- 264). IEEE.

[7] Bhattacharya, U., Ghosh, S.K. and Parui, S., 2007, September. A two-stage recognition scheme for handwritten Tamil characters. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) (Vol. 1, pp. 511-515). IEEE.

[8] Elakkiya, V., Muthumani, I. and Jegajothi, M., 2017. Tamil text recognition using KNN classifier. Advances in Natural and Applied Sciences, 11(7), pp.41-46

[9] Liyanage, C., Nadungodage, T. and Weerasinghe, R., 2015, August. Developing a commercial grade Tamil OCR for recognizing font and size independent text. In *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 130-134). IEEE

[10] Manisha, S. and Sharmila, T.S., 2017. Effective Printed Tamil Text Segmentation and Recognition Using Bayesian Classifier. In *Computational Intelligence in Data Mining* (pp. 729-738). Springer, Singapore.

[11] Ramanan, M., Ramanan, A. and Charles, E.Y.A., 2015, August. A hybrid decision tree for printed Tamil character recognition using SVMs. In *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 176-181). IEEE.

[12] Shivsubramani, K., Loganathan, R., Srinivasan, C.J., Ajay, V. and Soman, K.P., 2007. Multiclass hierarchical SVM for recognition of printed Tamil characters. *TC*, *2*, p.2.

[13] Stephen, P. and Jaganathan, S., 2014, March. Linear regression for pattern recognition. In 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE) (pp. 1-6). IEEE.

[14] Suresh, R.M., Arumugam, S. and Ganesan, L., 1999, September. Fuzzy approach to recognize handwritten Tamil characters. In *Proceedings Third International Conference on Computational Intelligence and Multimedia Applications. ICCIMA'99 (Cat. No. PR00300)* (pp. 459-463). IEEE.

[15] Suresh, R.M. and Arumugam, S., 2007. Fuzzy technique based recognition of handwritten characters. *Image and Vision Computing*, *25*(2), pp.230-239.

[16] Kunwar, R., Pal, U. and Blumenstein, M., 2013, November. Semi-supervised online learning of handwritten characters using a bayesian classifier. In *2013 2nd IAPR Asian Conference on Pattern Recognition* (pp. 717-721). IEEE.

[17] Gandhi, R.I. and Iyakutti, K., 2009. An attempt to recognize handwritten Tamil character using Kohonen SOM. *International Journal of Advanced Networking and Applications*, *1*(3), pp.188-192.

[18] Banumathi, P. and Nasira, G.M., 2011, July. Handwritten Tamil character recognition using artificial neural networks. In *2011 International Conference on Process Automation, Control and Computing* (pp. 1-5). IEEE.

[19] Venkatesh, J. and Sureshkumar, C., 2009. Tamil handwritten character recognition using kohonon's self organizing map. *International Journal of Computer Science and Network Security*, *9*(12), pp.156-161.

WEB REFERENCES

[1] https://link.springer.com/article/10.1007/s00500-019-03978-5

[2] https://searchenterpriseai.techtarget.com/definition/computational-linguistics

[3] https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

[4] https://www.researchgate.net/post/What_is_the_pros_and_cons_of_Convolutional_neural_networks

[5] https://link.springer.com/article/10.1007/s00500-019-03978-5

[6] https://en.wikipedia.org/wiki/Convolutional_neural_network

[7] https://plato.stanford.edu/entries/computational-linguistics/

[8] http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-4-131-133.pdf

[9] https://www.google.com/search?q=block+diagram+for+the+character+recognition

[10] https://www.slideshare.net/nikbharat/project-report-of-ocr-recognition

[11] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[12] https://www.researchgate.net/profile/Elviz_Ismayilov/publication/338420222_Parallel_Solution_Of_Features_Subset_Selection_Process_For_Hand-Printed_Character_Recognition/

[13] https://www.geeksforgeeks.org/project-idea-character-recognition-from-image/

[14] https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589

[16] https://www.researchgate.net/post/What_are_pros_and_cons_of_decision_tree_versus_other_classifier_as
[17] _KNN_SVM_NN
[18] https://www.coursehero.com/file/ps5vu3/What-are-the-Pros-and-Cons-of-Naive-Bayes-Pros-httpswwwanalyticsvidhyacomwp/
[19] https://www.researchgate.net/publication/323547763_Handwritten_Character_Recognition_HCR_USING_NEURAL_NETWORK
[20] https://arxiv.org/pdf/1804.02864.pdf
[21] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[22] https://www.slideshare.net/chiranjeeviadi/hand-written-character-recognition-using-neural-networks
[23] https://www.slideshare.net/chiranjeeviadi/hand-written-character-recognition-using-neural-networks
[24] https://www.academia.edu/258529/Requirements_for_the_Design_of_a_Handwriting_Recognition_Based_Writing_Interface_for_Children
[25] https://senior.ceng.metu.edu.tr/2016/teamtrio/docs/srs.pdf
[26] https://www.researchgate.net/figure/Advantages-and-disadvantages-of-fuzzy-logic-control-techniques_tbl7_323441631
[27] https://electricalvoice.com/kohonen-self-organizing-maps-algorithm-advanatges/
[28] https://plato.stanford.edu/entries/computational-linguistics/
[29] https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=+literature+survey+on+Tamil+character+recognition&btnG=
[30] https://www.guru99.com/what-is-fuzzy-logic.html
[31] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
[32] http://www.academia.edu/Documents/in/Literature_Review