# Gujarati Handwritten Character Recognition using Convolution Neural Network

MSc Research Project
Data Analytics

Dharavi Sorathiya
Student ID: x20173636

School of Computing
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Dharavi Rasikbhai Sorathiya | | |
| **Student ID:** | x20173636 | | |
| **Programme:** | MSc. Data Analytics | **Year:** | 2021-22 |
| **Module:** | MSc. Research Project | | |
| **Supervisor:** | Dr. Bharathi Chakravarthi | | |
| **Submission Due Date:** | 16th December, 2021 | | |
| **Project Title:** | Gujarati Handwritten Character Recognition using Convolution Neural Network | | |

**Word Count:7282** **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Dharavi R. Sorathiya

**Date:** 15th December, 2021

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Gujarati Handwritten Character Recognition using Convolution Neural Network

Dharavi Sorathiya
x20173636

**Abstract**

Language is important in many facets of life as it enables people to communicate in a manner that permits them to share similar views. With 22 languages 'registered' in the Indian Constitution and an approximated 780 linguistics in daily usage, India boasts a broad list of spoken languages. One of the languages is Gujarati, which is used primarily by Gujaratis and belongs to the Gujarat state of India. When it comes to character recognition, Gujarati script is among the languages for which there is limited literature is available. This paper focuses on creating a Gujarati script recognition algorithm that can recognize handwritten Gujarati characters accurately from images. Given the scarcity of data for the Gujarati language, a data collection of around 30,000 images featuring writings from various people is an essential contribution to this research. This procedure contains building a neural network that accepts a picture as an input, extracts a feature from it, then identifies the words and outputs them on the system. Three distinct models are used to train the model named Custom CNN, Xception, Inception_v3. All the implemented models have been evaluated using evaluation metrics such as accuracy, loss, recall, precision, f1-measure, and computation period. With an accuracy of 97%, 81 minutes of computation time for 15 epochs, InceptionV3 outperformed all models with regards to accuracy.

# 1 Introduction

## 1.1 Overview

Gujarati is the most spoken language in the Gujarat state of India and is originated from the Devanagari script. Gujarati script consists of 10 vowels and 34 consonants (Figure 1) a couple of more characters as well. Vowel modifiers are symbols that can be used to represent each vowel.

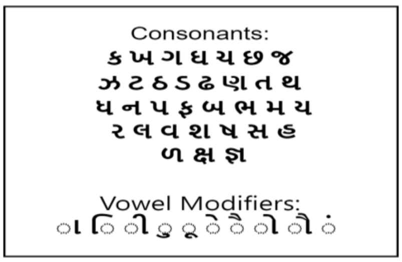


Figure 1. Consonants and vowel modifiers

The vowel modifier might occur on the character's bottom, top, left, centre, or right. Gujarati does not have a shirorekha (straight line upon the top of every phrase, like used in Devanagari language), thus all letters in a text are separated, unlike most other North Indian scripts. Gujarati has a nearly double number of characters as compared to the English language.

## 1.2 Motivation

There is a demand for digitizing the handwritten articles, documents, or papers for each language especially in a nation like India in which various languages are being utilized in daily lives. This leads to the need for computer-assisted identification of handwritten scripts since there are several dangers connected with the written content on paper, such as ripping, high sensitivity to weather patterns, and the possibility of being lost. Text recognition procedure includes the detection and identification of particular words or series of letters in a piece of a handwritten document.

Handwritten text recognition has a variety of usage areas covering government and corporate entities, along with individual businesses and sectors. There are several applications, such as the automated library and office papers, banking forms and checks procedures, document reading software, and so on, to minimize the human work involved in transforming handwritten textual content to digital information and establishing a paperless government/office. As a result, implementing such a system would aid in the automated identification of handwritten Gujarati characters while also removing the human labor that may be error-prone, time-consuming, and tiresome. Due to the unorganized and diverse writing styles of various individuals, handwritten text recognition is yet a challenging research topic.

According to research by Chaudhary et al. (2012) states that many Gujarati characters exhibit similarities. Even a little modification in the style of writings or a modest bit of extra noise might cause characters to be misclassified. Because of these idiosyncrasies, Gujarati handwritten text identification poses a difficult task for researchers. The literature review also found that, in comparison to other languages, Gujarati script receives significantly less attention. The area of text recognition is yet in its early stages of development. This plainly implies the necessity to focus on the job of recognizing handwritten Gujarati characters.

## 1.3 Research Question

**“How can Gujarati handwritten characters be recognized using deep neural networks?”**

## 1.4 Research Objectives

- A crucial analysis of a research related to the recognition of handwritten Gujarati characters.
- Data Generation and pre-processing to analyze written characters.
- Implementing deep learning algorithms to recognize characters
  - Modelling, Evaluation and examine an outcome of Custom CNN model.
  - Modelling, evaluation and examine an outcome of Xception model.

  - Modelling, evaluation and examine an outcome of InceptionV3 model.
- Comparison of implemented algorithms.

## 1.5 Outline

For this research, the dataset is created of size 30, 294 which contains handwriting from different people of various age-group. This dataset has 374 classes containing Gujarati consonants and their combination with vowel modifiers. The dataset is separated into train, test, and validation sets. This research aims to identify handwritten Gujarati characters. On this data, three different deep learning methods are implemented on this data namely, custom CNN, Xception, and InceptionV3. After the training, all the model's performances have been examined based on their accuracy, loss, recall, and precision scores on the testing dataset. At last, all models are compared based on the accuracy and computational time.

The following part of the report examines the review of various studies done about various approaches used to detect handwritten characters. The next part summarizes the research approach used for the study, which is Knowledge Discovery in the database. Following that, the implementation of different models employed is provided. At last, a discussion about this research, conclusion, and a direction to future work is provided.

# 2 Related Work

Multilingual optical character identification is a critical task to build as it requires different languages to exhibit writing and structural characters in a coherent manner. This is, however, not the case as it is complicated to simplify the segmentation procedures. This chapter will focus on different factors associated with Gujarati character recognition using deep learning algorithms. To do so, it is essential that various literature is identified that provides an in-depth understanding of Gujarati character recognition. It requires structural decomposition techniques and other online or offline types of Gujarati character recognition using low strokes and other documents that will help in the procedures.

## 2.1 Previous Findings

The area of optical character recognition is used and has acquired development of techniques that will allow the conversion of mechanical or electronic images or pictures of handwritten text to digital text. Previous litterateurs have identified the effectiveness of this study as it takes into consideration the utilization of Advanced Technologies that will not only recognize the text in pictures, photographs, or even scanned handwritten documents but also translate it to a digital text so that it can be available to the people. Every language has its structure which is unique within its culture. Much study has not been taken place to ensure character recognition in the Gujarati language (Gupta & Bag, 2019). It is therefore effective for this research to provide critical insight into the Gujarati character recognition from handwritten text or photograph or even printed text so that a digital text can be obtained to ensure the effectiveness of the study. Previous literature was also identified that this is a time-consuming process with a lot of loopholes which has not been addressed to their maximum.

## 2.2 Recognition of Gujarati Handwritten Characters from Text Images

Pareek et al. (2020), in their article Gujarati handwritten character recognition from text images, identified a Framework that will be used to adapt these handwritten characters and summarise them at each and every phase. The first phase requires the pre-processing period. Here the character detection procedure converts real images into formatted data sets to avoid and lessen noise and eliminate unnecessary background within the selected framework. In this process scaling, resizing noise removal and binarization skew connection is conducted. While the second series includes segmentations where the characters in the image pixels will be connected using labelling while histogram projection profile text block detection and content segmentation take place full stop the third step utilizes the process of classification with deep learning architecture using MPL and CNN models. Therefore, deep learning techniques can be utilized within the system for the Gujarati language due to the robustness and accuracy it provides.

## 2.3 Structural Decomposition Technique for Handwritten Gujarati Character Recognition

On the other hand, Sharma et al. (2019), in their article, identified as well as proposed three novel features that can help in presenting the handwritten Gujarati characters. The features included the extractions based on the structural decomposition of the characters, the zone pattern matching as well as a normalized cross-correlation that can be witnessed within the Gujarati script. To testify this the authors based their studies on the SVM and Naive Bayes method within the classification of Gujarati characters so that the proposed features could be represented to carry out character Recognition using deep learning methodologies. The identification of handwritten Gujarati numbers is the subject of the study done by Sharma et al., (2015). The zoning-based extraction of features is used to recognize Gujarati numerals. An input image is broken into four zones: which are 2x2,4x4, 8x8, and 16x16. For the categorization of numerals, naïve bayes and multilayer feed forward algorithms are used after extracting features using zoning approach. A total of 14,000 samples for every number were utilized to create a database. Overall, detection rates for this approach utilizing all the four zones mentioned are 61%, 91%, 95%, and 93%, correspondingly, while with naïve bayes algorithm are 53%, 81%, 85%, and 75%. Desai (2015) has tried to solve challenge of optical character recognition for Gujarati characters written by hand. A total of forty characters were gathered from around 189 people for the project. Aspect ratio, character extent, and picture subdivision techniques were employed as feature spaces, and a SVM was applied for classification, yielding an accuracy of around 88%. For classification, KNN is utilized also, and results are compared to those of SVM.

Goswami and Mitra (2013), identified in their paper that utilising structural feature extraction methods. The salient feature of this proposal was to identify the credibility of the characters and components. They identified that the major difference between Gujarati and other languages in India was the absence of shirorekha which is the headline running through the characters to form a word. This characteristic became the major difference between Gujarati and other Indian scripts. Therefore, using the proposed structural feature extraction method classification of subsets of Gujarati characters could be conducted (Sukhandeep et al. 2020).

Deep learning and extensive research within this Framework will not only allow effective understanding of the characters but also derive the native components to formulate it within the digital text.

## 2.4 Gujarati Handwritten Character Recognition Using Deep Learning

Joshi et al. (2018), in their research, identified the utilisation of Gujarati character under a mechanism that not only recognises the script but also makes it available for better understanding. The incorporation of deep learning as a critical methodology provides the identification of the characters through a man-made neural network. Deep learning is an important mechanism that enables in identifying a critical focus towards a right directive that allows the images to be recognised with nearly no negligence so that accuracy is obtained. According to the research, the neural Activity undertakes the concept of biological neurones that will prevail within the deep learning mechanism that has been applied within this research to ensure that accuracy and efficiency is maintained while evaluating the different characters ab Gujarati script. This is an extremely critical process towards the development of an effective program that allows understanding of the script while making it available for the people. Sharma et al. (2018) aimed to create an LSTM algorithm to recognize offline Gujarati characters. In addition, an effort was made to increase the recognition rate using model by training and learning the model using a dataset of over 58,000 pictures. The proposed approach is unique in that it identifies the whole collection of Guajarati characters accessible in the Unicode set. In this investigation, the authors utilized the LSTM network and attained around 97 percent accuracy for every character type. The purpose of this study by Rajyagor and Rakholia (2021), is to make use of the similarities that already exist in various portions of Gujarati characters. For handwritten Gujarati character identification, a new feature extraction approach relying on regulated cross association is presented. Along with the suggested feature extraction approach, average accuracy of 53%, 68%, and 66% achieved by the Naive Bayes, linear and polynomial Support Vector Machine, correspondingly. Experiments demonstrate that the suggested approach makes a considerable contribution, and that combining these characteristics with additional relevant traits may boost identification rate. The compilation of dataset of 20,500 handwritten Gujarati characters is one of the proposed work's major accomplishments.

## 2.5 Online handwritten Gujarati character recognition

According to Naik and Desai (2017), utilising online handwritten characters in Gujarati script using this mechanism is effective in classifying the strokes. This hybrid feature set encapsulates approximately 3000 samples that will be testified using 10 different writers. In order to ensure your accuracy, this methodology is applied will be tested effectively. It is extremely vital to note that the traditional keyboard is not user friendly as per the Indian scripts because of the presence of huge and complicated characters that prevails. Recognising handwritten characters are vital to induce the best possible solutions available on the internet. The article identified that the uses of the algorithms and classifiers such as SVM MLP and KNN classifier are effective in Gujarati character recognition. The accuracy remains above 91 % and takes only 0.063 seconds for the execution of the first stroke.

Later, after two years, Naik and Desai (2019), to improve the accuracy of perplexing characters, a multi-layer categorization approach is developed. With training data, an SVM algorithm with polynomial kernel is utilized in first layer classifying. For complex characters, in second layer use of SVM with linear kernel to process data is included. In both the levels of categorization, a hybrid significant features comprising of zone characteristics and a dominating point-based normalization chain code characteristic is employed. With the computation time of 0.103 seconds for each stroke, the researchers attained an average precision of 94.13 percent. Uniformity within the procedures can be witnessed as the other authors also utilised pre-processing period where the extracted features from the characters stop there after the features extracted contained a large number of drone data that was then evaluated using the different classifiers (Kumar et al., 2020). it was also recognised that many characters within the Gujarati script utilise multiple strokes within the single character and therefore recognise in these was difficult for the mechanism.

In a study done by (Gohel *et al*., 2015), it is evident that low-level stroke can act as one of the useful recognition parameters of Gujrati characters. It can be considered one of the effective ways in terms of pattern matching as a part of the deep learning technique. Although the handwritten strokes of Indian letters are full of complexities and difficulties, using suitable techniques, the research on old handwritten Gujrati documents can be one of the best options to work with Gujrati character recognition and low-level stroke can be a good measure in it. In the Gujrati scripts, handwritten words and their low-level stroke can be supposed to be one of the essential hierarchical histograms by its unique features and at a wide variety of levels, it can act as source of variation to capture the directional features.

## 2.6 Offline typed Gujarati character recognition

Macwan et al., (2015) in their study, looked at 34 consonants and only 5 vowels. In a first phase of segmentation, the architecture and linguistic features of language experienced difficulty; as a result, a novel method for segmentation is introduced. Their segmentation method is capable of properly addressing these challenges. For comparison, many methods from various domains were investigated, including Transform Domain, Geometric method, spatial domain, statistical methods. A novel approach for extracting feature vectors that combines structural and statistical methods has a high level of accuracy. These collected feature vectors were then fed into Support Vector Machines, and the resultant accuracies were evaluated by 10-fold cross validation procedure. SVM works effectively with data sets with a lot of characteristics and can handle a lot of classes. According to their findings, the transfer domain provides high accuracy, although it takes longer time. Individually, structural and statistical methods are not providing excellent accuracy, but a combination of these two methods, yields more accuracy in less time.

## 2.7 Adaptive neuro-fuzzy classifier with fuzzy hedges for Gujarati character identification

According to study done by Prasad and Kulkarni, (2015), ANFC or Adaptive Neuro-Fuzzy Classifier can be treated as one of the problems solving approaches in the character recognition of Gujrati by means of Fuzzy Hedges. Fuzzy Hedges are aligned with several useful network parameters with a lot of flexibility, an improvement approach as compared to the previous techniques used, and with a suitable approach. This measure also has a distinguishability feature with high rates in terms of overlapped classes. Using and employing the techniques of Fuzzy Hedges, the research associated with this zone can be further extended to a more in-depth study and analysis on the basis of the selection of suitable and essential features. As compared to other previous and traditional techniques, employing ANFC has been considered to be one of the important and extremely useful approaches with a high level of weightage in terms of utilization of the Fuzzy Hedges. It helps to promote the job of partitioning with essential features and aids with the provision of one of the solutions to network issues and classification problems. In addition, it can be used as one of the membership functions with a clear definition of each feature associated with it and as a combination of cooperative conditions. Overall, it is part of a concerted way of character recognition using an adaptive network.

## 2.8 Literature Gap

Gujarati script has unique structures strokes as well as characters that are not similar to others. To induce this language into a universal character recognition is not only difficult but also require years of research to understand the writing styles, size, curve and strokes of the different characters within the Gujarati script. Therefore, the literature gap that can be identified within this research is the complexity in the script which is not easy to understand let alone provide it under the character recognition techniques. Complexity within the language script create many loopholes when utilising it in the optical character recognition technology that will ultimately end up with the wrong results. On the contrary, the other gap that can be identified within this research is the limited time constraint. Understanding the writing style size curves and strokes of the different characters within the Gujarati script requires years of research and understanding however there were limitations in this research with respect to time. Along with this, the other limitation that was identified within this research was the traditional and cultural association. Traditional techniques are not only time consuming but also difficult to grasp to understand the different characters within this script. However, the utilisation of an adaptive neuro-fuzzy classifier can be useful in evaluating the different characters but also so effective in understanding this concept.

## 2.9 Summary

This chapter focused on the different factors associated with Gujarati character recognition with the help of deep learning techniques. The above Framework evaluated different literature associated with Gujarati handwritten character recognition within the deep learning Framework along with the structural decomposition techniques utilised to understand the handwritten Gujarati character. Using frameworks such as SVM, MLP, K-NN have also been identified in this chapter so that deep learning-based Gujarati handwritten character recognition can be evaluated. Gujarati script is a very complex script to understand due to the different styles, techniques, strokes as well as curves and sizes that are used. Some of the vital characteristics within this Framework has been identified that allowed to focus on the utilization of deep learning algorithms to recognize Gujarati character.

# 3 Research Methodology

To get better understanding of this research, KDD (Knowledge Discovery in Database) methodology is followed. This process is mostly useful for obtaining useful information from a wide range of data. The following diagram Figure 2 shows an overall KDD process in terms to the proposed research.

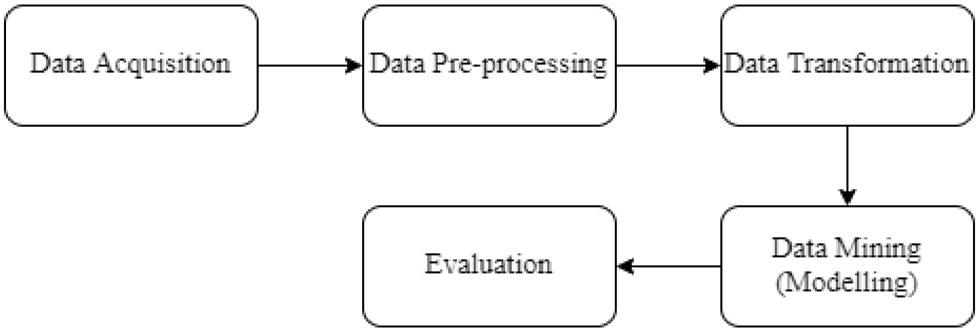


Figure 2 Gujarati Character Recognition Methodology

## 3.1 Data Acquisition

In this first stage, the goal is to gather data related to the research area. The stage involves explanation of process of collecting data from sources or generating data for particular research area. For this project, dataset is created consisting Gujarati handwritten characters by various people which is explained in detail in later part of the report.

## 3.2 Data Pre-processing

After data gathering, data undergo the process of cleaning or pre-processing. In this, data is prepared such that it can be used in implementation without generating any noises.

## 3.3 Data Transformation

Here, processed data transformed in a particular format that can be further used by the models. Transformation may include converting data from one type to other or dimension reduction of images. All above three stages are explained in detail in implementation part of the project.

## 3.4 Modelling

During this stage, whole process where patterns from dataset is extracted and used for decision making. It includes various type of algorithms from the domain of machine learning techniques as well as some complex algorithms from deep learning as well. Generally, to extract features from images convolutional neural networks are used. There are few models already exist for image classification with great ability to extract features from images. Such model can be utilized with some modification to get better predictions. Here total of three neural networks are incorporated in which two of them are Inception v3, Xception and another one is handcrafted CNN model.

### 3.4.1 Convolutional Neural Network

A CNN model is used to extract an important feature from images, which has demonstrated to be quite successful in picture classification, making this model an excellent fit. The core components which form the CNN architecture are convolution layer, pooling layers, activation functions. It's made up by stacking different layers on top of each other. On the other hand, CNN do not require fully connected networks for every pixel value but relying on suitable weights to examine small portions of images derived from data. The core functionality of architecture may be divided into four distinct parts. Input layer contains image's pixel values. Convolution layer determines outcome of neuron linked to particular sections of input vector using scalar product of its weight and area linked to input vector. The next pooling layer execute will execute down sampling of provided input, even more lowering number of features inside an activation function. The last fully-connected layer tries to generate categorization score from activations (O'Shea et al., 2015).

### 3.4.2 Xception

Xception refers to "extreme inception". First, it applies filters to each depth maps, then shrinks input data by performing 1x1 convolution over the depth. This approach is almost equivalent to depth-wise separable convolutions, which is utilized in building of neural networks. This architecture has 36 layers, which serves as the base for extraction of features. Besides the first and last units, all 36 layers are divided into 14 units, which contain continuous residual connections surrounding them. Furthermore, it follows three-flow technique, with data first passing through the entering flow, following that intermediate flow where it gets repeated for 8 times, and lastly an exit flow. An architecture's interactive nature makes it simpler to design and change by utilizing number of frameworks like Keras (Chollet, 2017).

### 3.4.3 InceptionV3

The Inception v3 is commonly utilized image recognizing algorithm on Imagenet dataset. The model represents the results of several concepts explored over time by number of scholars. This model is comprised of 48 layers and provided training over a million pictures from ImageNet. The network designed from Inception group which includes Label Smoothing, 7x7 factorized convolutional layers, also an additional classifier to transport labelled data deeper down to network. This architecture was referred to as GoogLeNet when introduced. Soon after, it was improved in number of ways, from which one is inclusion of batch normalization called Inception v2. Further factorization concepts were added later to architecture in next iteration, which can be called as Inception-v3 (Szegedy et al. 2016).

## 3.5 Evaluation

The assessment and analysis of several variables included after developing a model is thoroughly performed in this step to be assured enough to meet the study goals appropriately. In this step, models that have been implemented would be analysed to obtain informed insights. Evaluation metric used for an evaluation are accuracy, loss, precision, recall and f1-score.

# 4 Design Specification

The segment discusses the flow of the research and requirements that comes under the research study. As shown in Figure 3, the process is divided into mainly two parts: Dataset generation, and Training of the models. All blocks of implementation are explained in detail below.

The purpose of this paper is to recognize Guajarati handwritten characters. At present there is no such dataset exist on internet which can be utilized to train the models. So, dataset must be handcrafted from scratch. Dataset generation is performed in the first part of implementation which includes various age group of people to write Gujarati characters on paper, camera to capture handwritten character, various scripts to pre-process, transformation, parsing and splitting dataset. As a result of dataset generation process, three directories are created for training, validation and testing.

Next part of process includes training of various models. Three different models are used to train the model. Generally, pre-trained model is used for transfer learning. Similar concept is used to train two of three models. Trained models on images classification have good ability to extract features. Here Inception version 3 and Xception models trained on ImageNet dataset are used with some dense layers to train Gujarati handwritten dataset. Third model is crafted from scratch. Each model's execution time, train accuracy and validation accuracy are recorded to get statistics.

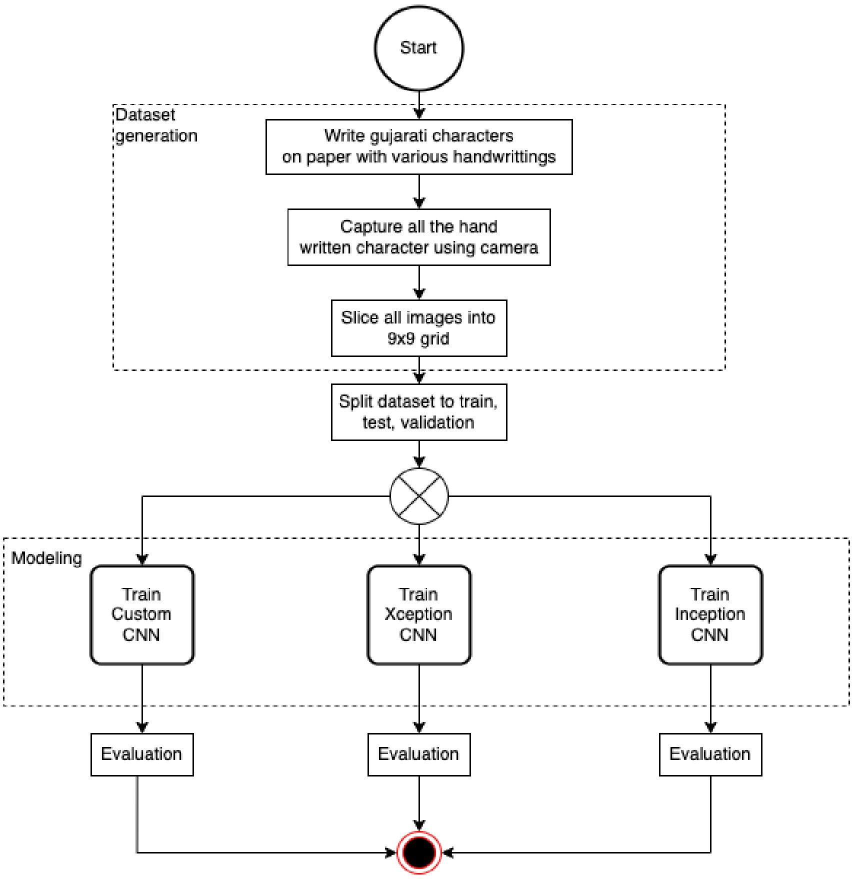


Figure 3 Process Flow Chart

Python's Tensorflow library is used to train the models along with various supporting libraries such as NumPy, sklearn, time, etc. Last part of implementation is an evaluation. This part includes testing of all three model on testing dataset as well as some conclusions to check goodness of models. The next section goes through all of the processes in detail.

# 5 Implementation

This section provides implementation detail in depth from dataset generation to configuring the models to train the data. This section is divided into two parts: Dataset Generation and Training the models.

## 5.1 Dataset Generation

The dataset of Gujarati handwritten characters is created by writing each character 81 times on a single page by different people from various age group.

The whole process from writing characters on pages, capturing images from camera and storing onto system and dividing them into training, testing and validation sets is explained thoroughly with Diagram as shown in Figure 4.

- Character writing on pages:
  On each page of 374 pages, a rectangular box of size 18cm is drawn having 81 sub-boxes of size 2cm per sub-box. Hence, every character is written 81 times on a single page by different people having different handwritings. The same process is followed for all the 374 pages.
- Converting physical dataset to digital:
  Each image is captured through the mobile's camera covering only a rectangular box (four edges of box) which has dimensions of 2992x2992.

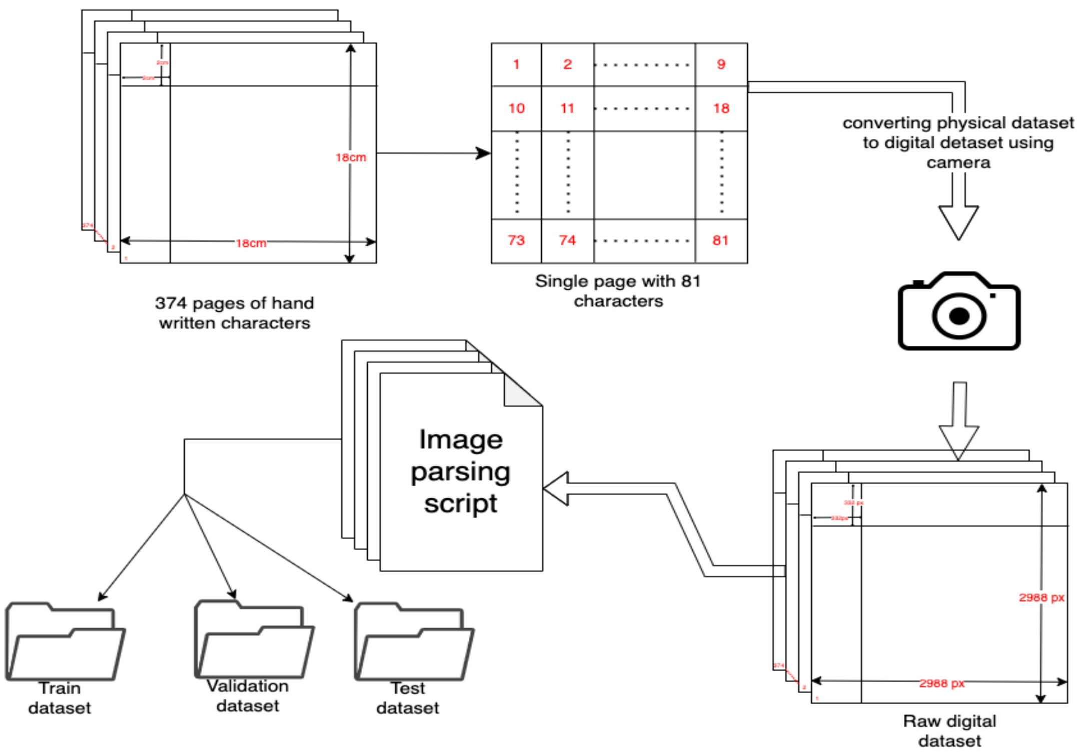


Figure 4 Dataset Generation

- Storing raw dataset:
  For every character, a separate folder is created, that stores characters along with the combination of characters with 10 vowel modifiers. Total 34 folders are created for 34 characters, where each folder has 11 images, one the character itself, and the remaining 10 images with the vowel modifiers.
- Storing parsed images:
  After executing the script on images, every page will create 81 sub-images. To store them, each folder has 11 sub-folders, where every folder contains 81 images. Out of 11 sub-folders, one is for the characters and 10 sub-folders for the character with vowel modifiers. Overall, all 34 folders have 11 sub-folders. Every sub-folder contains 81 images of the same character in different writings.
- Divide the data in train, validation, and test set:
  For test dataset, randomly one image is moved from every folder of parsed dataset. In the end, testing set has 374 images having an image of every character from each class. For validation dataset, any two images from the parsed dataset are moved to validation set, which makes a total of 748 images in validation data. Remaining data used as training set having a total number of images 29,172. Overall, from 30,294 images in total, 374 images moved to testing dataset, 748 images for validation set and remaining 29,173 images in training dataset.
- Data Transformation:
  Initially, the original dimension of every image is 2992*2992 in RGB form. So, there is a need to convert it from RGB to greyscale for further use. Along with that, the input image dimension has been set to 2998*2998 to avoid fractional pixels that make every image 332*332. Also, horizontal, and vertical margins have been reduced by 10 pixels. Hence, in the end, an input image has a 312*312 dimension, which is used to train models.

Following Figure 5(1) and 5(2), shows how an example of how raw-image looks like and how parsing the image generates sub-images for a character.

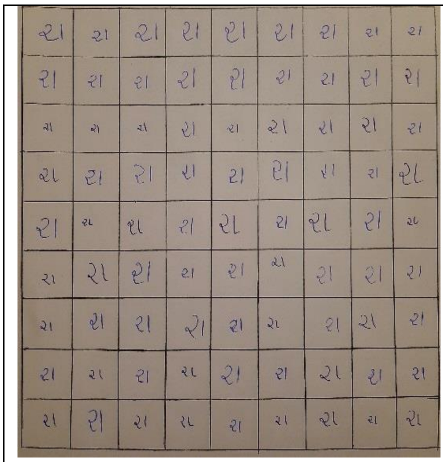

Figure 5(1) Raw Image

Figure 5 (2) Parsed Image

## 5.2 Modelling

The second section of the implementation part is further divided into two parts, where first part elaborates model architectures of all three models and second part explains model configurations done to train the data.

### 5.2.1 Model Architectures

#### 5.2.1.1 Custom Convolution Neural Network

The architecture of CNN is displayed in Figure 6 below. The model has 3 convolutional layers, 3 maxpooling layers, flatten layers, 2 dense layers, one output layers. The input shape for this model is 30 * 30 pixel of grey scale Gujarati alphabet image. Image passes through all the layers till the output layer. Proposed model has almost half million of trainable parameters. Input layer contains 16 convolution frames with shape of 3x3 which outputs 28x28x16 image. Same process is repeated for next two convolutional layers.

Generally convolutional layers are used to extract the feature from images followed by dense layers to select proper extracted features. Proposed network extract feature from image in first three convolutional layers. Then two dense layers with 512 neurons each are used to select accurate features. Relu activation function is used for all trainable layers except last. Output layers has 374 neurons associated with each class. Softmax activation is used for output layer. Proposed network is the lightest weight model due to a smaller number of parameters which leads to fast training as well as fast prediction.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 12, 12, 32)        4640
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 6, 6, 32)          0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 4, 4, 64)          18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_6 (Dense)              (None, 512)               131584
_________________________________________________________________
dense_7 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_8 (Dense)              (None, 374)               191862
=================================================================
Total params: 609,398
Trainable params: 609,398
Non-trainable params: 0
_________________________________________________________________
```

Figure 6 Model Design – Custom CNN

#### 5.2.1.2 InceptionV3

Inception v3 is the popular CNN architecture for image classification. It’s an opensource model that is trained on the largest image dataset named “Imagenet” which has 1000 classes. Such trained model has excellent ability to extract features from images. Such ability can be utilized to extract valuable feature from Gujarati character input image. Training new model by utilization pre trained model is called transfer learning. Here third version of inception is used to extract feature and one dense layer attached to the end of the trained model to select extracted features.

```
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Functional)    (None, 1, 1, 2048)        21802784

flatten_1 (Flatten)          (None, 2048)              0

dense_2 (Dense)              (None, 512)               1049088

dense_3 (Dense)              (None, 374)               191862
=================================================================
Total params: 23,043,734
Trainable params: 1,240,950
Non-trainable params: 21,802,784
```

Figure 7 Model Design – InceptionV3

As Figure 7 shows, Inception layer has 21 million non-trainable parameters followed by two dense layers including one output layer. This model contains 1.2 million trainable parameters.

#### 5.2.1.3 Xception

The Xception is an inception architectural modification that uses depth wise separable convolutions instead of the normal inception model. It is structured in linear stack way which makes it faster in compared to Inception network. Xception is used to extract features from train dataset and two dense layers are used to manipulate selected features.

```
Layer (type)                 Output Shape              Param #
=================================================================
xception (Functional)        (None, 3, 3, 2048)        20861480

flatten_2 (Flatten)          (None, 18432)             0

dense_4 (Dense)              (None, 512)               9437696

dense_5 (Dense)              (None, 374)               191862
=================================================================
Total params: 30,491,038
Trainable params: 9,629,558
Non-trainable params: 20,861,480
```

Figure 8 Model Design – Xception

As Figure 8 shows, Xception layer has 20 million non-trainable parameters followed by two dense layers including one output layer. This model contains 9.6 million trainable parameters.

### 5.2.2 Model Configuration

The dataset is utilized by three models. Each model has different configuration in terms of trainable parameters, layer, activation function, epochs and input image shape as shown in Table 1. Training a model is very crucial part because wrong configuration can lead to classical problems of training such overfitting or underfitting or not learning at all. It is very important to select most appropriate activation function for each layer, proper optimizer for backpropagation and most efficient loss function. There is total 374 classes that each model has to predict. Therefore, each model’s output layer contains that much of neurons. Each model design is already explained in previous section. Tensorflow library is used to train the model. Each model gets the data from train directory in the batch of 64 images at a time. After each epoch training and validation accuracy is calculated. Validation data contains 748 images (2 images from each class) from 374 images. There is a callback which always check for best training accuracy at the end of each epoch. If greater accuracy is achieved then such model is saved to specific path for prediction. Inception and Xception networks are very large network. It is not preferable to train whole network again instead pre-trained model weights can be used to extract feature. Therefore, for those networks, some model's layers training is disabled. All accuracy is registered for evaluation at the end of learning.

| CNN name | Epochs | Optimizer | Learning time (in minutes) | Input shape | Batch size | Trainable parameters |
|---|---|---|---|---|---|---|
| Custom CNN | 35 | Adam | 53 | (30, 30, 1) | 64 | 609,398 |
| Inception V3 | 15 | Adam | 81 | (75, 75, 3) | 64 | 1,240,950 |
| Xception | 15 | Adam | 131 | (71, 71, 3) | 64 | 9,629,558 |

Table 1 Model Configuration Table

# 6 Evaluation

This section provides the results of algorithms implemented on a dataset. After training the data with the above three models, results are evaluated with the help of accuracy, f1-score, precision, and recall score on testing data. Precision indicates how much of the expected positive values were accurately predicted out of all of them. The recall states that out of all the positive classes, how many were really projected to be positive. When there is no apparent difference between accuracy and recall, the F1-measure is often utilized. It combines the two trends into a single number. Also, a training and validation accuracy graph has been shown per epoch to analyse the performance of each model.

## 6.1 Custom CNN

With 35 epochs in 50 minutes, custom CNN gives an accuracy of 93% and 90% on training and test set respectively. With the test set this model achieves 92% accuracy. Precision, recall and f1-score all score 92% on testing dataset proving the model performance is excellent.

Table 2 Accuracy- Custom CNN

| Training | Validation | Test |
|---|---|---|
| 93.39 | 90.11 | 92.24 |

As per the below graphs, loss values of train and test dataset goes down as epochs increase. While the accuracy rate follows an upward trend with increasing epochs value. Validation accuracy has experienced some fluctuations in the accuracy rate after 12$^{th}$ epoch, and in the end the rate goes down than the training accuracy.

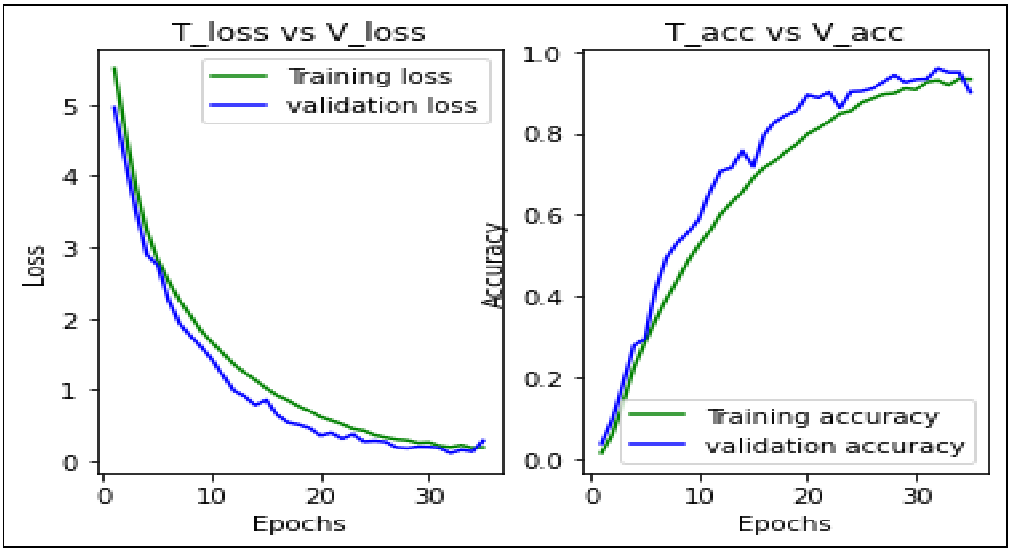

Figure 9 Accuracy and Loss Value vs Epoch – Custom CNN

## 6.2 Inception V3

Results of InceptionV3 are as given in below Table 3, according to which training and validation accuracy are 97 and 98%, respectively. Moreover, recall, precision, f1-score and accuracy on test dataset is almost 92%, which means that model is performing well.

Table 3 Accuracy - InceptionV3

| Training | Validation | Test |
|---|---|---|
| 97.16 | 98.80 | 92.78 |

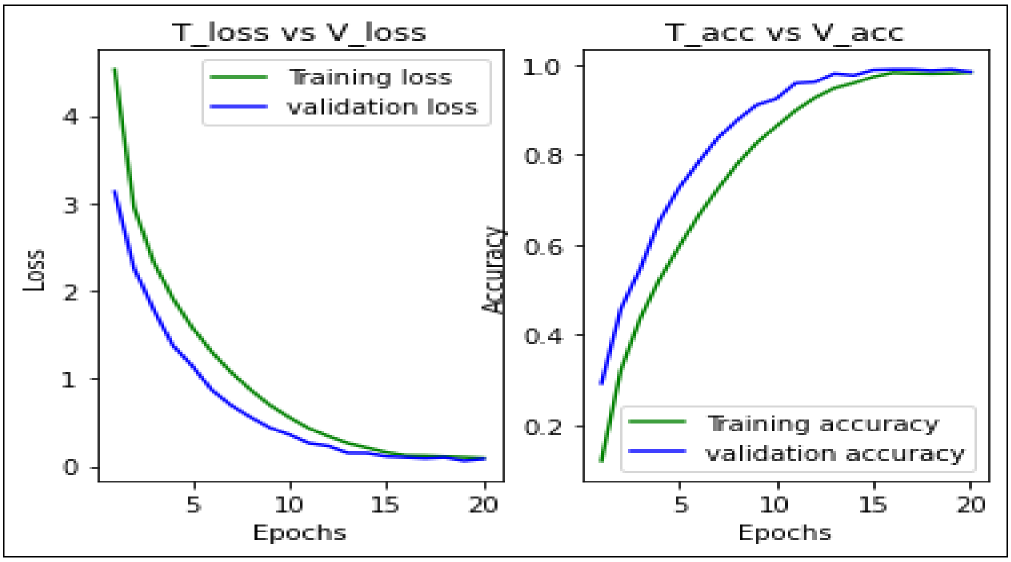

Figure 10 Accuracy and Loss vs Epoch: InceptionV3

Above graphs shows training and validation loss per epoch. Training loss is high in 1st epoch compared to the validation loss. As the epoch increases both loss value starts to decrease and at the end, both has almost the same low loss value. On the other hand, accuracy rates of training and validation data increases as per epoch. In the beginning, training accuracy is high compared to the validation, but in the end both the sets end up sharing almost same accuracy rate.

## 6.3 Xception

Xception model provides an accuracy of 91% on training dataset, and 94% on validation test set after 15 epochs as shown in Table 4. According to that testing accuracy is 89%, which is quite good. Moreover, recall, precision and f1_score is also near to the 90%, which describes that performance of model is excellent.

Table 4 Accuracy-Xception

| Training | Validation | Test |
|---|---|---|
| 91.70 | 94.65 | 89.30 |

The below graphs depicts the accuracy and loss model experinced while training the data. As per the loss grpah training loss is high in starting epochs, but as it goes high loss values is decresaing. Similary, on validation set, loss is high in the early epochs but not as high as in training and as epochs goes higher it keeps going down. However, as epoch value goes higher accuracy of both the data, training and testing increaed. At the end of the epochs validation accuracy is higher than accuracy of training set.

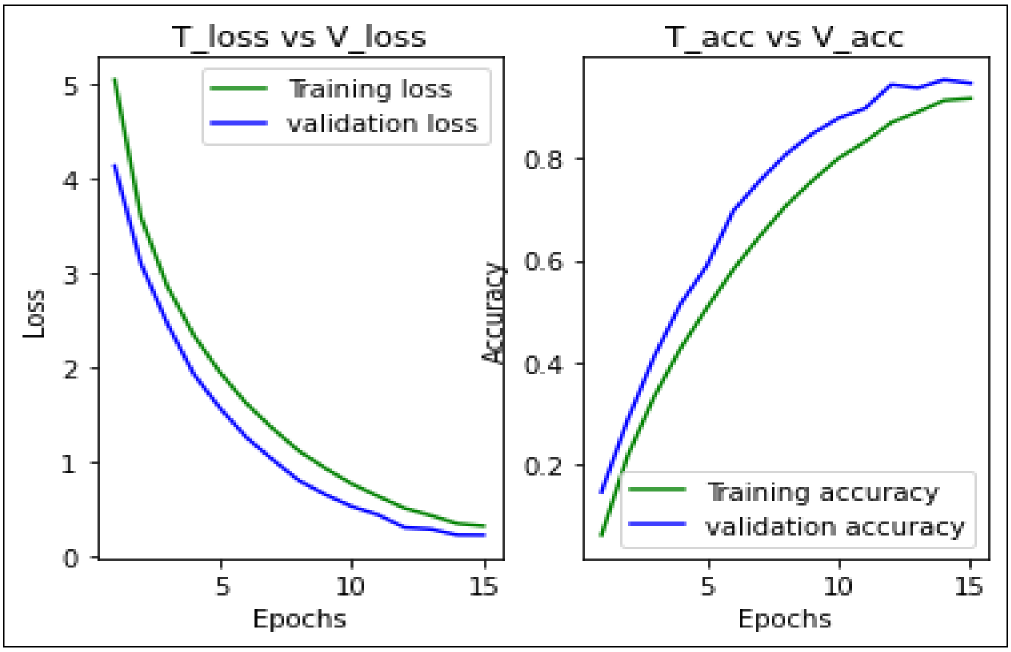


Figure 11 Accuracy and Loss values vs Epoch: Xception

# 7 Discussion and Model Comparison

During this study, for recognition of handwritten Gujarati characters dataset is created first, and then three deep learning approaches are used in order to get promising results and compare the results with each other. The proposed algorithms used for this purpose have performed quite well concerning accuracy and computation time.

As Pareek et al. (2020), created dataset of 10,000 pictures having 59 classes, while in this research total of 374 classes are created making the total dataset size of 30,294 to further train the models. Moreover, there was a comparison in their research paper with two deep learning models named MLP and CNN, where the accuracy was 64% and 97% on test data. While with suggested algorithms in this research, custom CNN performs quite commendable with about 93% accuracy on training and testing dataset, and 90% with a validation set along with the 53 minutes of computation time. However, InceptionV3 and Xception get 97% on the testing dataset with a computation time of 81 and 131 minutes accordingly.

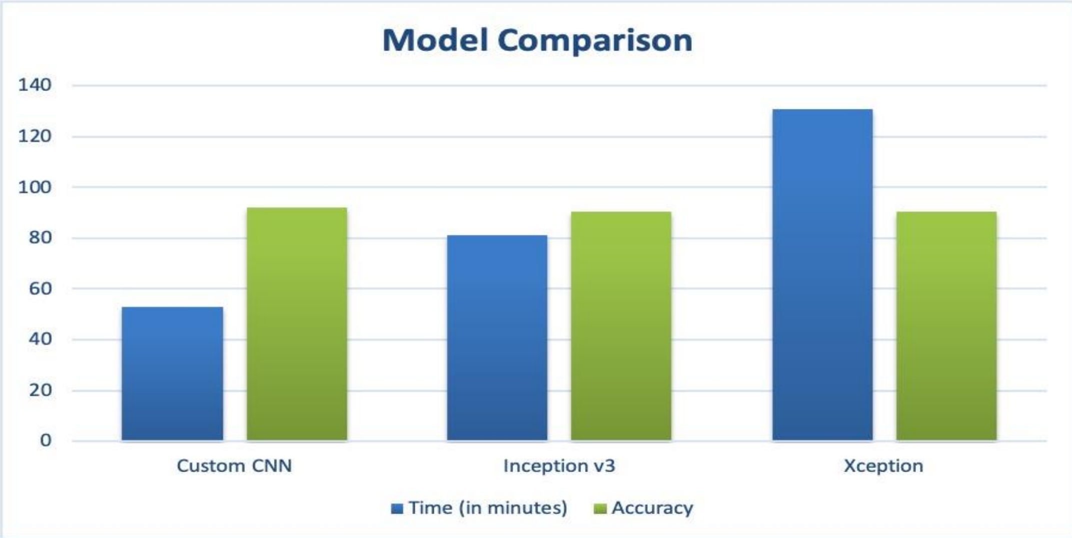


Figure 12 Model Comparison

# 8 Conclusion and Future Work

The research here demonstrates the use of various deep learning methods to recognize handwritten Gujarati characters. A primary goal was to implement algorithms to detect Gujarati handwritten characters and compare their outcomes. For that, thorough research is done relating to handwritten character identification. Based on that, the models for implementation have been selected. To fulfil the objectives, firstly dataset was created of size 30, 294 images. Then dataset was partitioned into train, test, and validation sets. Following that, the dimensions of images changed as per the need for algorithms. Three different algorithms are implemented on data namely, Custom CNN, InceptionV3, and Xception. Out of these three models, InceptionV3 model seems to be the best fit as it provides higher accuracy of 97% on training data, and 92% on testing data. On the other hand, Custom CNN and Xception provides 93% and 91% accuracy on training data, respectively. However, in terms of computation time, in which InceptionV3 and Xception take 81 and 131 minutes, accordingly. Here, Custom CNN takes 53 minutes of computation time providing 93% accuracy on training ser and 92% on testing data.

As a future direction, the dataset can be extended by adding complex characters such as joints can be added to the dataset to strengthen robustness over variations in handwriting. Also, different deep learning algorithms can be applied to detect such characters and to further minimize computation time. The live camera also can be added to minimize the computing time and get better accuracy. These methods can also be used for other languages to get promising outcomes. Gujarati digits can also be added to the dataset. Considering modifiers or special symbols as separate classes and correctly allocating them to the base characters, also utilizing Natural Language Processing methods in order to increase unit accuracies by rectifying any incorrect word in the identified text are also included.


# Acknowledgement

I am thankful to my professor Dr. Bharathi Chakravarthi for their support and guidance throughout the research. I would also like to express my gratitude to my parents and friends for helping me with the collection of the dataset.


# References


Chaudhary, M., Shikkenawis, G., Mitra, S.K. and Goswami, M., 2012, December. Similar looking Gujarati printed character recognition using Locality Preserving Projection and artificial neural networks. In *2012 Third International Conference on Emerging Applications of Information Technology* (pp. 153-156). IEEE.

Prasad, J.R. and Kulkarni, U., 2015. Gujarati character recognition using adaptive neuro fuzzy classifier with fuzzy hedges. *International Journal of Machine Learning and Cybernetics*, *6*(5), pp.763-775.

Pareek, J., Singhania, D., Kumari, R.R. and Purohit, S., 2020. Gujarati Handwritten Character Recognition from Text Images. *Procedia Computer Science*, *171*, pp.514-523.

Sharma, A.K., Thakkar, P., Adhyaru, D.M. and Zaveri, T.H., 2019. Handwritten Gujarati Character Recognition Using Structural Decomposition Technique. *Pattern Recognition and Image Analysis*, *29*(2), pp.325-338.

Gupta, D. & Bag, S. 2019, "Handwritten multilingual word segmentation using polygonal approximation of digital curves for Indian languages", *Multimedia Tools and Applications,* vol. 78, no. 14, pp. 19361-19386.

Kumar, M., Jindal, M.K., Sharma, R.K. & Jindal, S.R. 2020, "Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: a study", *The Artificial Intelligence Review,* vol. 53, no. 3, pp. 2075-2097.

Sukhandeep, K., Seema, B. & Kumar, R. 2020, "A survey of mono- and multi-lingual character recognition using deep and shallow architectures: indic and non-indic scripts", *The Artificial Intelligence Review,* vol. 53, no. 3, pp. 1813-1872.

Goswami, M. and Mitra, S., 2013, December. Structural feature based classification of printed Gujarati characters. In *International Conference on Pattern Recognition and Machine Intelligence* (pp. 82-87). Springer, Berlin, Heidelberg.

Gohel, C.C., Goswami, M.M. and Prajapati, V.K., 2015, December. On-line handwritten Gujarati character Recognition using low level stroke. In *2015 Third International Conference on Image Information Processing (ICIIP)* (pp. 130-134). IEEE.

Naik, V.A. and Desai, A.A., 2017, July. Online handwritten Gujarati character recognition using SVM, MLP, and K-NN. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

Joshi, D.S. and Risodkar, Y.R., 2018, February. Deep learning based Gujarati handwritten character recognition. In *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)* (pp. 563-566). IEEE.

O'Shea, K. and Nash, R., 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

Macwan, S.J. and Vyas, A.N., 2015, August. Classification of offline Gujarati handwritten characters. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1535-1541). IEEE.

Desai, A.A., 2015. Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space. *CSI transactions on ICT*, *2*(4), pp.235-241.

Sharma, A.K., Adhyaru, D.M., Zaveri, T.H. and Thakkar, P.B., 2015, November. Comparative analysis of zoning based methods for Gujarati handwritten numeral recognition. In *2015 5th Nirma University International Conference on Engineering (NUiCONE)* (pp. 1-5). IEEE.

Sharma, A.K., Adhyaru, D.M. and Zaveri, T.H., 2018. A novel cross correlation-based approach for handwritten Gujarati character recognition. In *Proceedings of First International Conference on Smart System, Innovations and Computing* (pp. 505-513). Springer, Singapore.

Rajyagor, B. and Rakholia, R., 2021, September. Isolated Gujarati Handwritten Character Recognition (HCR) using Deep Learning (LSTM). In *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1-6). IEEE.

Naik, V.A. and Desai, A.A., 2019. Multi-layer classification approach for online handwritten Gujarati character recognition. In *Computational Intelligence: Theories, Applications and Future Directions-Volume II* (pp. 595-606). Springer, Singapore.