Script Identification in Handwritten Documents for Gurumukhi-Latin Script using Transfer Learning with Deep and Shallow Classifiers

Sukhandeep Kaur · Seema Bawa · Ravinder Kumar

Received: date / Accepted: date

Abstract Script identification at character level in handwritten documents is a challenging task for Gurumukhi and Latin scripts due to the presence of slightly similar, quite similar or at times confusing character pairs. Hence, it is found to be inadequate to use single feature set or just traditional feature sets and classifier in processing the handwritten documents. Due to the evolution of deep learning, the importance of traditional feature extraction approaches is somewhere neglected which is considered in this paper. This paper investigates machine learning and deep learning ensemble approaches at feature extraction and classification level for script identification. The approach here is: i. combining traditional and deep learning based features ii. evaluating various ensemble approaches using individual and combined feature sets to perform script identification iii. evaluating the pre-trained deep networks using transfer learning for script identification 'iv. finding the best combination of feature set and classifiers for script identification. Three different kinds of traditional features like Gabor filter, Gray Level Co-Occurrence Matrix (GLCM), Histograms of Oriented Gradients (HOG) are employed. For deep learning pretrained deep networks like VGG19, ResNet50 and LeNet5 have been used as feature extractor. These individual and combined features are trained using classifiers like Support Vector Machines (SVM) , K nearest neighbor (KNN), Random Forest (rf) etc. Further many ensemble approaches like Voting,Boosting and Bagging are evaluated for script classification. Exhaustive experimental work resulted into the highest accuracy of 98.82% with features extracted from ResNet50 using transfer learning and bagging based ensemble classifier which is higher as compared to previously reported work.

Keywords Script recognition · Handcrafted or non handcrafted features · Gurumukhi · English · LeNet5 · vgg19 · ResNet50 · LeNet5 · Ensemble classifier
1 Introduction

There are a large number of writing systems and scripts in the world having a wider variability in terms of character sets, writing styles, shape of characters etc. It makes the task of developing a single recognizer for all scripts and writing styles difficult (perhaps close to impossible). In the multilingual country like India, many documents contain text written in more then one language or script which makes obligatory to design Optical Character Recognition (OCR) for multilingual CR. Hence, the solution proposed by many researchers is integrating the script identification module with character recognition to select the OCR from pool of OCR corresponding to the script selected.

Automatic OCRs has three essential sub processes: text localization, character segmentation and recognition. Further multilingual OCR, has script identification sub-process for character recognition to select a particular OCR. Due to ambiguity in structure and style of leading languages, script identification in multilingual inscriptions has become a burning area of research for the pattern recognition community [1]. Moreover, Script identification facilitates many applications such as document recognition; penetrating online libraries of image documents and image document sorting etc.

Script identification methods can be broadly classified into two categories: local and Global methods. Local methods are subject to smaller components of document like characters, words and text lines while the global methods consider the larger parts of documents like pages, paragraphs etc. Global methods require the document part to be normalized and free of noises to get the correct results. On the contrary, local methods can deal with images having low quality and noise [13]. Script identification is usually performed at page, text line, word and character level. Many researchers have attempted the first found approaches but the research for script identification at character level is still a thought-provoking issue and has not been dealt with much [3].

Script recognition in handwritten documents is more challenging due to high ambiguity of resemblance between characters of different scripts in handwritten data compared to printed. To capture such variations, different kinds of features were handcrafted and extracted under explicit feature based approaches [5]. In handcrafted features, accuracy of results is biased towards the approaches applied for feature extraction and the pattern type. It is a two step process: (1i) identifying the characteristics regions of an image and (ii) finding the descriptor to distinguish each region from another. Besides handcrafted features where human experts are involved in designing a feature descriptor, in non handcrafted features only data is considered to extract the feature. But non handcrafted features need larger amount of training set resulting into higher training time. It can be resolved using some supervised or semi supervised techniques with parallel computing [6]. To resolve this issue of training time, another form of deep learning called transfer learning has been found. In transfer learning, the hyper parameters of pretrained deep networks can be reused for our new data instead of training the whole network from scratch. As for deep learning networks to optimize the hyper parameters of deep architectures, requirement of labeled, diverse and high quality training data for regional languages is still an open issue. To address this issue somehow researchers have proposed data augmentation techniques.

In pattern recognition problems, yet it is undecidable that which feature set and classifier will be best
The classifier combination techniques are less explored in script recognition area [9]. The classifier combination can be performed at feature level called early integration or at classifier(decision) level called late integration [21]. At feature level different kinds of features can be concatenated to form a new feature vector and feed into a single classifier. The new feature set can hold more information at the cost of increase in dimensionality resulting into increased training time. At recognition level, individual classifiers designed from each feature set are combined at decision level which is most popular [9]. Many strategies exist to combine at decision level as majority voting, bagging, boosting, Borda count, product, max, sum rule, Dempster-Shafer theory, Bayesian method and ROVER framework etc [11, 12, 21]. The fusion of different kinds of features and classifiers may results into better accuracy, more reliable and robust results as misclassified patterns of one classifier can be recovered by some different classifiers [10].

The proposed research considers the script identification for Bilingual documents containing Gurumukhi and Latin scripts. Gurumukhi script is the base for one of India’s 22 official languages i.e. Punjabi. The official documents of Punjab state government consist of two languages: English and Punjabi. Punjabi language contains 35 characters and 10 numerals which have resemblance with various characters and numerals of English language. English language has 26 capital and small letters with 10 numerals resulting into character set of 62.

1.1 Motivation and Challenges

To process multilingual handwritten documents, some of the challenges for script identification are as follows:

1. In multilingual text, similarities and dissimilarities are present between classes as well as within classes. Hence, results from different feature sets and classifiers vary corresponding to the pattern to be recognized.

2. Due to the depreciation of similar character shapes between Punjabi and English languages, script identification at character level is a challenging problem for these scripts.

3. In handwritten data, a single character written by individual can possess various character shapes depending upon the age, mood and gender of the writer. Many forms of noise are present like ruling lines, foreground pixels, skew and slant etc.

4. Efficient feature extraction technique to capture various patterns is one of the challenge to multilingual CR research community. Each classifier has its own pros and cons for particular pattern. To access the full potential, we need to combine the complimentary nature of different classifiers and features. This is an open issue to find an appropriate classifier combination approach.

5. Lack of benchmark datasets for multilingual CR specifically for Gurumukhi and Latin script is one of the hurdles in script identification.

6. The use of deep networks for script identification in Gurumukhi script is difficult due to lack of high quality, diverse and large training data.
The influence of colonial past and globalization across the world has made English language as second official language after regional languages in India. All of the official work has been conducted using two languages: one is English and the other is state or regional language like Punjabi in Punjab state. Many public domains such as academic, health, administrative etc has made it mandatory to use Punjabi language with English language for documentation purpose. It is the main motivation behind this research and some others are as follows:

1. The popularity of digital devices like scanners, mobile phones, cameras and digital pads etc gave birth to digitization of paper documents. The official documents contain machine printed as well as handwritten data in English and Punjabi language shown in Fig. 1. To convert such multilingual documents into machine editable form, script recognition is one of the most important sub-process.

2. Script identification at page, paragraph, text line and at word level had been the major focus in past work. Less work has been reported for script recognition in handwritten text. Moreover, there are some words (Fig. 1) in official documents consisting of characters from different scripts.

3. As deep learning, demands high training set and computation power while the traditional feature extraction needs high human expertise. The traditional features and deep learning features are needed to be combined. Considering the research from [6] article, we have tried to explore the combination of traditional and deep learning based features.

4. Recently Kumar et al [31] have conducted script recognition for handwritten text at character level using Hindi, Gurumukhi and English character set. They have reported an accuracy of 92.89% for script recognition. They have considered only character dataset not numerals for any script which has been considered in our work.

5. The potential amount of work has been done for script identification in printed text, scene and video text extraction for non-Indic scripts as compared to Indic scripts.

6. Some of the researchers have tried to combine the complementary natures of different classifiers resulting into high accuracy [9]. There main motivations for choosing ensemble classifiers over individual classifiers are statistical, representational and computational. The statistical reason arises when the training data is too small in comparison to hypothesis space. Hence, each learning algorithm produces similar results for every hypothesis generated. Here, an ensemble approach can use average of each individual classifier’s output to avoid wrong classification. Another case is, when you have large amount of data available for training. Here, some algorithms like decision tree and neural network have the possibility to stuck in local optima problem. Ensemble learning overcomes it by performing many local searches simultaneously with different initial points. The representational issue arises when the hypothesis space is unable to represent the true function. Ensemble classifier uses the weighted sum of hypothesis drawn from the original space to represent the better representational function.

7. Script identification has so many applications such as note taking in classroom, form processing and filling, document image analysis etc.
8. To save the time for training deep networks, use of some already available pretrained models is highly required.

9. The main motivation behind script identification at character level is due to two main reasons as: a. official documents contain many words with characters from different scripts as shown in Fig. 1, b. Feature extraction at character level is easy and less time consuming as compared to feature extraction at word level.

![Fig. 1: Bilingual handwritten official document in Gurumukhi-Latin script](image)

1.2 Contributions

To overcome the various challenges for script identification in Gurumukhi and Latin script for handwritten text, we have found the combination of features and classifiers based on transfer learning and traditional machine learning approaches. The major contributions of this research are enlisted as following:

1. In this research, a script identification module for multilingual handwritten text recognition has been designed for Gurumukhi-Latin script. It uses pre-segmented handwritten images of characters and numerals of Gurumukhi and Latin script.

2. It provides a script identification module for many document image analysis applications like extracting information from magazines, articles and image searching etc.

3. In this work, Script identification has been performed at the lowest level i.e. character level.
4. Various combinations of deep learning and traditional features have been proposed and evaluated for script identification. Combining the different kinds of features together resulted into effective performance improvement by eliminating each feature shortcomings.

5. Many pretrained deep learning models have been utilized using transfer learning for script identification. The complementary natures of different classifiers using various ensemble approaches like voting, boosting, bagging etc are utilized.

6. To evaluate the proposed approach, dataset of handwritten images of characters and numerals with large variations are taken.

1.3 Research Objective

The objectives of research are given below:

1. To design a script identification module to process the handwritten text written in multilingual forms like postal, railway and passport application forms, question papers etc which uses English in conjunction with Gurumukhi script.

2. To visualize the effect of combination of traditional and deep learning features for script identification in Gurumukhi and English script.

3. To obtain an optimal combination of features and classifiers corresponding to our dataset for script identification.

4. To resolve the ambiguity in handwritten character recognition due to resemblance between characters of different scripts.

The organization of rest of the paper is as follow. Section 2 leads the latest work done for script recognition. Section 3 gives the proposed methodology while section 4 gives the experimental results. In last section 5 concludes the work and gives future directions.

2 Related Work

In this section we are reviewing the work done for script identification for Indic and non Indic scripts using traditional and deep learning models.

2.1 For Indic and non Indic scripts

In the past, texture analyses based approaches using traditional feature sets were the most used methods for script identification. Brodic et al 2016 [13] have proposed textural analysis based approach for script identification of Fraktur and Latin scripts in German Historical documents. They have considered position of letter in baseline area to determine the script in initial phase. Further, handcrafted features extracted using script specific characteristics are classified using Genetic Algorithms Image Clustering. Similarly Sharma et al 2016 [1] have designed one more approach called pixel plot and
trace and replot and re trace (PPTRPRT) to segment bilingual offline handwritten documents. It results into optimal character segmentation rate. It extracts text regions from the document and segments it into text lines. Singh et al 2018 [9] have used one shape based and two texture based features named Elliptical, HOG, Modified log- Gabor filter transform features to perform script identification of 12 major Indian scripts. A heterogeneous classifier has been created by feeding different features sets to the same classifier with fine tuning of parameters. Further, a combination strategy has been applied at feature level as well as at decision level for the classification. Similarly, Mukhopadheya et al [9] have used classifier combination approaches with Multi Layer Perceptron (MLP) for script identification considering database of handwritten document images from 12 major Indic scripts. Different kinds of shape based and texture based features are used with MLP classifier to recognize the words of different scripts. Number of classifier combination approaches like voting, borda count, sum rule, product rule and dempster-shafer theory are applied with MLP as base classifier. Kacem et al 2016 [35] have used texture analysis to discriminate between the Arabic and Latin scripts for handwritten as well as machine printed scripts at word level. Four different kinds of features based on black run length histograms are extracted to characterize the writing orientations and stroke lengths. The domain specific features are lacking in handling the small size word images due to which orientation information has been considered.

Sahare et al 2018 [2] have attempted character segmentation and recognition approach for multilingual Indian document images containing text written in Latin and Devanagari. Structural properties of characters are considered to find the segmentation points. For overlapped and joined characters, graph distance based theory has been used. They proposed three new geometrical shape based features to classify segmented characters. Further, for validation of segmented characters, SVM classifier has been used while k-nn is used for character recognition. Kacem et al 2017 [35] have used black run length based histograms to find the nature and script of the word written. Structural properties of text are used with k-nn classifier to distinguish between handwritten and machine printed text written in Latin and Arabic scripts. Structural properties like writing orientations and varying stroke length of text makes script identification at word level an easy task.

From many applications of script identification, Roy Mandal et al [15] have explored the multilingual date field extraction from documents containing data in English, Bengali and Devanagari script. The top and bottom reservoirs have been employed to gain the foreground and background information. It segments the words into primitive segments which are further classified using SVM. To identify the other components like digits and punctuation marks, gradient based features are employed with SVM. Singh et al 2018 [3] have designed a benchmark database for script identification in documents with mixed script (Bangla-Roman, Devanagari-Roman). Database consists of 150 document images for both datasets including characters, digits, texts and other symbols. The proposed database has been evaluated for script identification using modified log gabor filter based features with number of different classifiers.

With the emergence of deep learning, many researchers have tried these models for script identification and found promising results. Sarkhel et al 2017 [5] have used a new deep learning based approach for OCR of many Indic scripts. In this, multi column and multi scale based Convolutional Neural Network (CNN) architecture has been employed. In this, each column independently extracts
features and feeds to SVM classifier for recognition. The final result from all the columns is obtained using various decision combination schemes like simple voting, weighted voting and majority voting. Bhunia et al [4] have used the attention mechanism with CNN-LSTM (Long Short Term Memory) network for script identification in natural scene images. It was the first attempt which uses attention mechanism for script identification. Results have proved that attention mechanism has the ability to extract most relevant features of an image in a network. Local features have been identified using attention mechanism by calculating weights of image patches with higher significance. Further, global features were extracted from the last cell state of LSTM. To decide which features hold high weights, attention mechanism using dynamic weighting has been applied on local and global features.

Out of deep learning models, a variant of Recurrent neural network (RNN) model called LSTM has the capability to learn the whole text line without character segmentation called sequence learning capability. UL-Hasan et al 2015 [17] have employed it over traditional feature extraction methods for script identification of Greek and Latin scripts at text line level. A single target label is assigned to all characters of a particular script. The model was trained on the sequence of class label as the ground truth information instead of individual character. Another work considering LSTM has been done by Ahmed et al (2016) [18] to recognize the Urdu and Latin scripts. They employed Connectionist Temporal Connection (CTC) layer as the output layer to Bi directional LSTM (BLSTM), for sequence alignment in cursive writing recognition. Chherewal et al 2018 [12] have investigated the effect of context modeling on deep learning model for character recognition. Many researchers have observed that modeling the context information with Hidden Markov Model (HMM) results into improved recognition rate. In deep learning models, BLSTMs are those networks which train the context at feature level for long range but lacks for training the context at decision level. The authors have designed number of BLSTM models by varying the context information at label level. Distribution and direction distribution features are given to BLSTM to provide context information at feature level. Number of decision level combination strategies are implemented for word recognition using BLSTMs.

Table 1 represents the comparative analysis of deep networks and traditional approaches for script identification. Deep networks lacks only for two reasons: training cost and training time of deep networks. To overcome it, transfer learning is the new trend and many researchers have found promising results with this. Gupta et al [47] have proposed CNN based fusion free approach for multilingual handwritten digit recognition. Many handcrafted features along with traditional classifiers have been evaluated. Comparative analysis has been made for CNN and other pretrained deep networks using transfer learning. For 8 Indic and non Indic scripts, the collective accuracy of digit recognition 96.23% has been reported. Similarly Alkhawaldeh et al [48] have proposed recurrent deep transfer network for Arabic handwritten digit recognition. The hybrid network consists of two simple deep networks i.e. LSTM and CNN. To learn the deep features, VGG-16 based transfer deep architecture has been considered. Cheng et al [49] have used patch aggregator approach along with simple CNN architecture for script identification. The patch aggregator learns the deep features of scripts by combining the local patches. It results into good classification accuracy for many datasets like SIW-13, CVSI2015 and RRC-MLT2017. Pramanik et al [50] have applied transfer learning for word recognition of handwritten Bangla city names. They have evaluated many base classifiers with statistical features for comparative analysis and then a seven layer fully connected network has been trained from scratch. Many
transfer learning models like VGG-16, ResNet, AlexNet, GoogleNet, VGG-19 etc have been evaluated.

2.2 For Gurumukhi script

For Gurumukhi scripts, the first attempt for script identification has been made by Rani et al [30] for printed dataset containing English and Gurumukhi data. Gabor and Gradient based feature extraction approaches were applied with multi class SVM classifier. They had considered both characters and numerals resulting into 4 class classification problem. The dataset consist of 19,448 characters with multi fonts having multi sized characters and received the accuracies around 98.9% for Gabor and 99.45% for Gradient based features. Kumar et al [31] have presented a multilingual script recognition system for English, Hindi and Gurumukhi scripts using pre-segmented handwritten characters. Various features like zoning, diagonal and horizontal peak extent etc have been used with K-nn, linear SVM, and MLP classifiers. The highest accuracy of script recognition reported was 92.89% for character dataset. Dataset of 4920 samples having 4 different classes (English upper and lower case characters, Hindi and Gurumukhi characters) is used for experimentation. Pandey et al [29] have explored script identification for Gurumukhi, Hindi and Urdu scripts for handwritten data using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) feature extractions. The script recognition accuracy of 82.70% has been reported for dataset of 961 handwritten samples.
| Author                        | Text mode            | Scripts                              | Feature set                  | classifier | Dataset                        | Accuracy                                      |
|-------------------------------|----------------------|--------------------------------------|------------------------------|------------|--------------------------------|-----------------------------------------------|
| Sharma et al [?]              | Handwritten          | Hindi and English                    | Pixel Plot and Trace based   | FFNN       | 68,000 characters              | 99.78% segmentation accuracy                  |
| Sahare et al [2]              | Printed and handwritten | Latin and Devanagari               | geometrical shape based features | k-nn     | Tobacco 800 and proprietary database | 97.71% and 98.06% segmentation rate for Tobacco 800 and proprietary dataset respectively |
| Singh et al [3]               | Handwritten          | Bangla-Latin, Latin-Devanagari      | Modified Log-Gabor filter    | MLP        | 300 document pages            | 92.32%, 95.30% and 93.78% for Bangla-Roman, Devanagari-Roma and Bangla-Roman-Devanagari respectively |
| Bhunia et al [4]              | scene images and videos | Indic and non-Indic scripts        | deep learning                | CNN-LSTM  | SIW-13, CVSI2015, ICDAR-17, MLe2e | 96.50%, 97.75%, 90.23%, 96.70% script recognition accuracy |
| Mukhopadhyay et al [9]        | handwritten          | 12 Indic scripts                    | Shape based and texture based | MLP and classifier combination | 7200 words | 95-98% recognition accuracy |
| Shi et al [14]                | natural images       | 13 script indic and non indic       | deep learning                | DiscCNN    | SIW-13                         | 89% for script identification                |
| Author          | Text mode     | Scripts                          | Feature set                      | Classifier | Dataset                                      | Accuracy                  |
|-----------------|---------------|----------------------------------|----------------------------------|------------|----------------------------------------------|---------------------------|
| Modal et al [15]| Handwritten   | English, Devanagari, Bangla      | gradient based and profile based | DTW and SVM| 1100, 820, 1240 lines of Bangla, English, Devanagari | 96.69% word wise script recognition accuracy |
| Tian et al [16] | scene text    | English, Chinese and Bengali     | HOG and convolutional HOG        | SVM        | 6,182 Chinese, 19,530 Bengali, ICDAR 2003, SVT and IIIT 5K-word datasets for English | 81.70%, 92.20% and 71% for ICDAR, Bengali and Chinese respectively |
| Ul-Hasan et al [17] | printed      | English and Greek                | deep learning                    | LSTM       | 9500 text line images                        | 98.186% script recognition |
| Ahmed et al [18] | synthetic or printed | Urdu and Latin                        | deep learning                    | BLSTM with CTC | UCOM and UNLV-ISRI | 99.17% and 88.94% for Latin and Urdu respectively |
| Angadi et al [22] | printed      | Hindi, Kannada, English, Malyalam and Tamil | wavelet based and horizontal run statistics based | Fuzzy inference system | 1200 low resolution images of display boards | 94.33% script identification |
| Chanda et al [46] | printed      | 11 scripts                       | zernike moment based and gradient features | SVM        | 142 torn documents                           | 81.39%                     |
| Chanda et al [23] | printed      | Thai and English                 | structural, overlapping, water reservoir, topological features | SVM        | 6110 words                                  | 99.36% script identification accuracy |
| Author            | Text mode     | Scripts                        | Feature set                                      | Classifier | Dataset                  | Accuracy                  |
|-------------------|---------------|--------------------------------|--------------------------------------------------|------------|--------------------------|---------------------------|
| Pati et al [26]   | printed       | 11 indic and non indic scripts | gabor and DCT features                          | SVM        | 20,000 words             | 94.80% average script recognition accuracy |
| Tan et al [27]    | handwritten   | Arabic, Roman and Latin        | Horizontal and vertical inter-stroke direction, average stroke length, stroke density and the reverse directions | Information retrieval based | 39 documents             | 93.30% average accuracy   |
| Kavitha et al [28] | printed       | English, Kannada, Tamil, Telugu, Hindi and Gujarati | Connected components and dominant points         | template based | 700 scanned document images | 96.10% classification accuracy |
| Kacem et al [35]  | handwritten and printed | Arabic and Latin | Texture features | K-nn | IAM, IFN-ENIT, AHTID/MW and APTI | 98.92% |
| Pal et al [37]    | printed       | English, Chinese, Arabic, Devanagari, Bangla | Shape based and statistical features | rule based | 700 document images | 97.33% |
| Roy et al [40]    | handwritten   | Bangla and English             | run length smoothing based and water reservoir based | Neural network | 4342 handwritten and 650 printed words | 97.62% |
| Author       | Text mode | Scripts                  | Feature set     | Classifier     | Dataset                      | Accuracy |
|--------------|-----------|--------------------------|-----------------|----------------|------------------------------|----------|
| Pan et al [41] | printed   | Chinese, Japanese, Korean and English | gabor filter    | feed           | 600 samples for English and 550 for each of the other | 98.5%    |
3 Proposed Methodology

3.1 Problem Statement

Let \( D(x) = \{d_1, d_2, ..., d_n\} \) be a set of handwritten document images where each \( d_i = \{C^S, Y^S\} \) consists of set of characters and numerals i.e. \( C \) belonging to different scripts, and \( Y \) is the set of corresponding script labels. Set \( S = \{S_1, S_2, S_3, S_4, S_5\} \) is a collection of different scripts where 

- \( S_1 = \) English lowercase character
- \( S_2 = \) English uppercase characters
- \( S_3 = \) Gurumukhi character
- \( S_4 = \) Gurumukhi numerals
- \( S_5 = \) English numerals

Further, dataset of images of characters and numerals i.e. \( C_{new} \) consist of two kinds real and augmented dataset i.e. \( C_r \) and \( C_s \) respectively. Now on this new dataset \( C_{new} \), we defines \( S \) be the set of labels. Let for \( C_i \) image associated with label \( S_j \) in \( S \), we defines the classification function as:

\[
f: (C_i \rightarrow S_j \mid S_j \in S)
\]

Hence, the problem can be formulated as we are given with a set \( C_{new} \) of pre segmented handwritten character images of any script or language \( s_j \mid S \), find the most efficient and accurate classification function which maps the images to the most appropriate label. Figure 2 shows the block diagram of proposed approach.

3.2 Data Augmentation

Data augmentation has recently become popular to meet the need of large training data, for deep neural network. Two types of data augmentation approaches are available: geometric transformation based and task specific based data augmentation. Many approaches come under these like, flipping, rotation, scaling, elastic deformation, patch extraction etc. Images generated using simple data augmentation approaches are unable to extract the deep features of an image during training of deep networks. To overcome this issue, recent trends in deep learning for data generation are Generative adversarial networks. Here in our research we are using simple data augmentation approaches based
on rotation and flipping. Main motive of data augmentation is to make the model transformation invariant and to solve the class imbalance problem. Here we have used random rotation with angle of 20 deg to generate the random imbalance images. Further zoom operation has been applied with range 0.2 on randomly selected images. Data augmentation easily overcomes the problem of class imbalance by randomly generating artificial images for each class.

3.3 Feature extraction using Handcrafted and deep learning features

In any pattern recognition problem, feature extraction plays a vital role for classification. In character recognition, various kinds of feature extraction methods have been used such as statistical, structural, deep learning etc. Today, with the emergence of deep learning there comes a one more category called handcrafted v/s non handcrafted features. The handcrafted features use the traditional approaches and need human expertise to characterize any pattern while deep learning features have capability to automatically extract the features (Fig. 3).

3.3.1 Traditional/handcrafted features

Many methods like SIFT, HOG, GLCM, etc available to extract the low level features (line, dots, curves etc.) come under traditional approaches. These can be extracted using single layer convolutional filters, gabor filters, HOGs etc. to represent geometrical and surface characteristic properties of text. Further these can be divided into statistical and structural categories of feature extraction. The statistical feature extraction includes zoning, projection and profile, crossing and distance and texture features etc. In zoning, an image is divided into N*M zones to capture the features from each zone resulting into local features of an image. Number of variants to zoning features like density and direction features etc. exit. Projection profiles and histograms methods calculate the number of pixels in the row and column of image which makes them noise and deformation tolerance.

Structural features describe the template or shape of character with geometrical properties of small...
components making the character. Aspect ratio, curves, edges, lines, crossing point, loops and branch points etc comes under structural features. The third category based on global transformations and moments are texture features like GLCM, Gabor filter, gradient, cosine, wavelet transformations, zernike moments etc. These features are invariant to object scaling and rotations.

The other structural features considered are water reservoir principle based which are one of the highest used methods for structural features. Water reservoir principle uses cavity region of the component with direction of water poured into the cavity region. To characterize the character shape, it uses number of parameters as top reservoir, left reservoir, water flow level, reservoir base-line, height of reservoir etc. The in-depth details of features extracted for the work has been given as following:

3.3.2 Gabor feature

The gabor features use the Gabor filters to differentiate between textured and non textured region of an image. These are the sinusoidal function which uses various orientation and frequencies to captures the features of an image. The filters generated corresponding to orientations and frequency captures the edges of an image as shown in Fig. 4(a). These filters capture strong response to those orientation for which direction of image structure goes. Many researchers have tried gabor filters for script identification with different orientation and frequency with good response [41, 26, 3]. The even and odd symmetric filters used in our work are given by Eq. 2 and Eq. 3.

$$G_c[i,j] = e^{-\frac{1}{2\sigma^2}(i^2+j^2)} \cos(2\pi f(i \cos \theta + j \sin \theta))$$ \hspace{1cm} (2)

$$G_s[i,j] = e^{-\frac{1}{2\sigma^2}(i^2+j^2)} \sin(2\pi f(i \cos \theta + j \sin \theta))$$ \hspace{1cm} (3)

where i, j are the spatial coordinates, f is the frequency of sinusoidal component of the gabor filter, \(\sigma\) represents the frequency of Gaussian envelop along the principal axis and \(\theta\) is the angle of orientation.

3.3.3 HOG feature

Histogram of oriented gradient captures the key characteristics of an image by converting the pixel based representation to gradient based. These are the widely adopted features due to its ability to encode and match the strong gradients in characters. Moreover these are robust to illumination
variation and invariance to the local geometric transformation. With all these, the HOG lacks in capturing the spatial information with respect to neighboring pixels as only the orientation of isolated pixels considered. It places a square grid over the image which divides it into blocks. Each block is further divided into cells. The horizontal and vertical gradients are calculated for each pixel in the cell. Further, for each cell a histogram is constructed for all the orientations as shown in Fig. 5(a).

Hence, the final feature vector concatenates all the block vectors. Sobel operator is used to calculate the horizontal and vertical gradients of image pixel. Equation 4 and 5 gives the magnitude and orientation of gradient calculated. Fig. 5(b) shows the feature extracted image using HOG.

\[
g = \sqrt{g_x^2 + g_y^2} \tag{4}
\]

\[
\theta = \arctan \frac{g_x}{g_y} \tag{5}
\]

3.3.4 GLCM feature

Grey level co occurrence matrix based features use the spatial relationship of pixels to characterize the texture of an image. It considers the degree of correlation between the neighboring pixel or pixels. The other texture features provides the statistical view of an image but lacks in obtaining the spatial relationship between pixels of an image. The graycomatrix function used in GLCM creates a GLCM matrix by calculating the co occurrences of pixel having some intensity value to another pixel having intensity value too present in some specific spatial relationship. The element (i,j) of an GLCM matrix is the sum of all the instances present in input images with intensity value i for pixel of interest and intensity values j for neighboring pixel. To optimize this matrix calculation in terms of complexity, graycomatrix function performs scaling to reduce the number of intensity values in gray scale image. The offset considered from pixel with interest to another pixel during spatial relationship is the major parameter for GLCM feature vector by default is one i.e. neighboring pixels. As the correlation occurs usually at small distances, the value of d is kept small. In GLCM, the linear distance between pixels are used with combination of orientation called \(\theta\) between them. To capture the texture of an image, GLSM with single offset is not sufficient. So, here we are considering varying offsets in distance and direction resulting into 12 GLCM with 3 distances and 4 orientations. Fig. 6(a) represents the offsets considered in this work and Fig.6(b) represents the calculated GLCM from an input image with offset value of 2 and direction 45 \(\circ\). For an image with dimensions m*n, GLCM(i,j) matrix can be used to extract various statistics like energy, contrast, homogeneity, correlation to characterize the...
Fig. 6: GLCM calculation: (a) shows the offset values considered for pixel of interest corresponding to varying directions and distances; (b) shows GLCM matrix obtained with direction $\theta=45^\circ$ and distance=2

Table 2: Parameter details for handcrafted features

| Feature name | Parameter          | Values         | length | used as                          |
|--------------|--------------------|----------------|--------|----------------------------------|
| Gabor        | f=frequency, $\theta$=angle of orientation | f=0.4, $\theta$=16 | 32     | Energy and mean amplitude        |
| HOG          | orientation        | $\theta$=6, block size=4*4, cell size=4*4 | 2400   | Energy, contrast, homogenity     |
| GLCM         | distance, orientation | d=1,2,3 and orientation=0deg, 45deg, 90deg, 135deg | 48     | Energy, Contrast, Correlation, homogenity |

image. Energy feature represents textual uniformity in terms of sum of squared elements of GLCM represented in Eq. 6 and contrast captures the local variations in the GLCM matrix (Eq. 7). The homogeneity given by Eq. 8 defines the closeness of grey level in the spatial distribution over image while correlation as Eq. 9 shows how correlated a reference pixel to its neighbors over an image. In our experiment we are considering three different distances with four different orientations. To analyze all the properties of GLCM matrix, we are extracting all the features resulting into feature vector of 48 dimensions.

\[
Energy = \sum_{i=1}^{m} \sum_{j=1}^{n} (GLCM(i,j))^2 \tag{6}
\]

\[
contrast = \sum_{i=1}^{m} \sum_{j=1}^{n} (i-j)^2 GLCM(i,j) \tag{7}
\]

\[
Homogeneity = \sum_{i=1}^{m} \sum_{j=1}^{n} \left( \frac{GLCM(i,j)}{1 + |i-j|} \right) \tag{8}
\]

\[
correlation = \sum_{i=1}^{m} \sum_{j=1}^{n} \left( \frac{(i \cdot j)GLCM(i,j) - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y} \right) \tag{9}
\]

The parameter details for each handcrafted feature used in the experiments have been represented by Table 2.
3.3.5 Deep learning or non handcrafted features

Handcrafted features use human designed features which need human expertise. Hence, it may loss some useful information due to biasness. To extract the high level features of an image like edges, corners etc., we need more deeper layers of filters. Recently, deep CNN has been found to be outperform as a feature extractor for many pattern recognition problems. CNN is an end-to-end system which extracts the local features of a pattern by jointly learning the parameters of the network. Each layer adds an abstraction to the representation of an image with series of convolutional and max pooling operations performed on the input. For lower level layers, information extracted resembles with the one extracted using gabor filter like edges and texture. However, by cascading this information at each level can result into more complex representations. Many recherches have found promising results with CNN as feature extractor. Unlike handcrafted features, it automatically extracts the features of an image using convolutional layers with sub sampling and pooling operations. These extracted features can be used in number of ways rather then putting into fully connected layer and applying the softmax layer for classification. In this paper we are extracting the CNN features from the pretrained model like LeNet5, Vgg19, ResNet50 for further classification purposes. In depth analysis of transfer learning has been done in section 3.4.

3.3.6 Combining deep learning and traditional features

The recent advances in deep learning have proved that it outperforms as feature extractor as compared to traditional features for many areas. However CNN gives less significance to the discriminative local patches of an image which are crucial elements in script identification for scripts with subtle differences. Secondly, it lacks in handling the images having arbitrary aspect ratio [14]. To make feature extractor more robust in terms of rotations and geometrical in-variances, traditional feature extraction methods are needed to be considered. Nanni et al [6] have combined the handcrafted and non handcrafted features for computer vision applications using CNN and LBP,LTP etc. Ngyun et al [25] have combined the handcrafted features like LTP, HOG, LBP etc and deep learning features CNN and MLP for presentation attack detection in face recognition.

Moreover, it is an open issue to find the appropriate features corresponding to the data type. To benefit from the complementary nature of different feature extraction methods, we are combining the traditional and deep learning feature extraction methods. Three traditional feature extraction methods HOG, Gabor, GLCM are used in combination with one of the deep learning feature extraction method appropriate to our dataset type i.e. LeNet. LeNet has been trained on our dataset from scratch out of other deep models considered in the paper. Features extracted from the fifth layer of trained LeNet model are combined with traditional methods and trained on individual and ensemble classifiers. The combination of features results into 15 different kinds of features using handcrafted and non handcrafted features.
3.4 Transfer learning

Training efficient deep networks requires bulks of training data while collecting large training data is highly time consuming. Further, training such models on such large data takes many hours or days. Hence, transfer learning is a way of reusing the pre-trained models on large dataset for some task, to a new dataset for some other task, both tasks belonging to same domain. The formal definition of transfer learning is: Model M has been efficiently trained for dataset $D_m$ for source task. Similarly another model N has dataset $D_n$ which is smaller then $D_m$ resulting into inefficient training of model N. Now the part of model M will be used by N to perform the tasks efficiently. In our research we are using many pretrained models for training our script identification module for bilingual datasets.

### 3.4.1 LeNet5 feature

LeNet5 is the special kind of deep CNN architecture designed for handwritten or machine printed character recognition with minimum preprocessing needed. For LeNet, size of input image is restricted to 32*32 as the key distinguishable features like strokes, end points, corners etc can lie in the center of receptive fields of higher level convolutional layers. The major change done in the LeNet 5 architecture is at third layer i.e. convolutional layer with various feature maps is connected only to a subset of S2’s feature maps instead of every feature map of S2. It has two main reasons as : 1) It keeps number of connections in reasonable bound i.e. computationally less demanding and 2) to put the complementary features with different feature sets so that network can learn different patterns. The Tanh sigmoid activation functions are used to update the weights of network.

![Fig. 7: Filtered image at fifth layer of Lenet](image)

Fig. 7 represents the filtered images generated after second convolutional layer from the trained LeNet. The fifth level layer of LeNet before the fully connected layer, has been used for feature extraction, resulting into 400 dimensional vector.

### 3.4.2 VGG-19

VGG network was created by Visual Geometry Group in 2014 after the success of its predecessor Alexnet with some modifications into the later. It is considered the simplest network as all the 3*3 convolutional layers are stacked on one another and two fully connected layers with 4096 nodes were
Table 3: Parameter details for deep network features

| Network name | Pre-trained | Fine Tuning | Depth | Image size | no of convolutional layers | Activation | no of Pooling layers | Fully connected layer | Feature vector size |
|--------------|-------------|-------------|-------|------------|--------------------------|------------|---------------------|---------------------|--------------------|
| LeNet5       | No          | Complete    | 5     | 32x32x1   | 3                        | Tanh       | 2                   | 1                   | 400                |
| Vgg19        | yes         | Partial     | 19    | 224x224x3 | 16                      | ReLu       | 5                   | 3                   | 25088              |
| ResNet50     | yes         | Partial     | 50    | 224x224x3 | 48                      | ReLu       | 1                   | 1                   | 100352             |

stacked on top of convolutional layers. It takes images of size 224x224 in RGB. To provide non-linearity in the model for better classification, ReLu activation has been adopted, and Max pooling with stride 2 has been applied to reduce the volume size.

3.4.3 ResNet50

ResNet are the residual networks which allows training the extremely deep networks upto 152 layers. The difference between ResNet and other networks is the use of skip connections between the convolutional layers relying on residual mapping instead of stacked learning. Skip connections manages the vanishing gradient problem when the depth of network increases highly. ResNet 50 has 48 convolutional layers with 1 maxpooling and 1 average pooling layer.

Table 3 represents the parameters details for deep networks used.

3.5 Ensemble Approaches

Ensembling can be done in various ways as: manipulating the training parameters of classifiers, using different subsets of classifiers, different feature subsets and manipulating the output of classifiers etc [19]. There are two major requirements for an ideal ensemble classifier. First, it needs highly diverse base classifiers in order to get low correlation across different classifiers for misclassified patterns. Second, recognition rate of ensemble classifier needs to be higher then one produced by individual classifiers. Bagging and Boosting are the two popular approaches used for ensemble creation.

Moreover, rather then using simple CNN features there are many other approaches to use these such as transfer learning, combining these features with strong handcrafted features for better results etc. Here, we are working on both feature and classification phase to boost the performance of script identification. For feature extraction we are analyzing the 17 different kinds of individual and ensemble features using deep learning and shallow learning approaches while for classification multiple classifier system and ensemble classifiers have been used.
3.5.1 Voting

The voting based ensemble is the simplest one which takes output predictions (votes) of each individual classifier. The final prediction will be the one which has highest votes. The two main variants of this are: majority voting (final prediction will be the one which has more than half of votes) and weighted voting (assigns a weight to one or more best individual classifiers as it will run multiple times).

\[ y_i = \text{mode}\{C_1, C_2, ..., C_k\} \]  

(10)

Lets take four classifiers, where \( C_1 \) predicts class 0, \( C_2 \) predicts class 1, \( C_3 \) predicts class 1 and \( C_4 \) predicts class 1 as.

\[ y_i = \text{mode}\{0, 1, 1, 1\} \]  

(11)

Hence the majority voting based ensemble will result class 1 as output. In our work, we are using majority voting with base classifiers as SVM, Random forest and K-nn for each feature set.

3.5.2 Manipulating training samples

In this, various classifiers are generated corresponding to different subsets of training samples. It includes three types of ensemble approaches mainly: bagging, boosting and cross validation based on the criteria of selecting training subsets [34]. The unstable algorithms (where output of classifier changes with small change in the training samples) are the major candidates for this type ensemble category.

i. Bagging

In bagging, \( N \) individual classifiers are generated choosing \( N \) training subsets randomly from the pool of original training data with size \( n \). It is also named as bootstrap aggregating as the training subsets taken from original training set are called bootstrap replicate. Each training subset consists of an average of 63.2\% of original training data with repetitions [32]. For our work we are using SVM, K-nn and Random forest as base classifiers for bagging.

iii. Cross validation

Another ensemble approach is creating the disjoint subsets of the training data. For example we are creating 5 overlapping training subsets by leaving out a disjoint subset in each training set. This procedure is named as K fold cross validation where \( K \) training sets are designed with one different subset in each fold. The ensemble designed based upon this is called cross validated committees [34]. To evaluate this approach for our work, we are considering 3 fold cross validation in each experiment.

ii. Boosting

Adaboost [33] generates multiple training sets with some initial weights assigned to each training example. In each iteration \( i \), the learning algorithm employed on training sets produces hypothesis \( h_i \). The weights of training sets get updated based on the classification results of hypothesis \( h_i \). The strategy to assign weights, places higher weights to misclassified training examples and lower weight to correctly classified training examples. For Adaboost, we are considering SVM, decision tree and random forest.
3.5.3 Manipulating Input features

This is an open issue in machine learning to identify the best combination of particular feature set with classifier corresponding to the dataset. Hence, researchers have found an alternate to this called ensemble learning. It combines the complementary nature of different classifier and features extracted from the dataset, to boost the performance of pattern recognition. Various subsets of features were used individually and in combination to create the ensembles. In our proposed script identification approach we have three traditional and one deep learning feature extraction methods. By manipulating these input features we have designed 15 different feature sets which are classified using various classifiers.

3.5.4 Manipulating output target labels

This ensemble approach provides solution for classification problems with large number of output classes. In this, number of output classes K are divided into smaller subsets of classes as $A_l$ and $B_l$ and input data is relabeled corresponding to the label assigned to new class subset. The learning algorithm is trained on the relabeled input data with the new derived output labels. This process repeats L times, by generating different subset of classes for $A_l$ and $B_l$ each time. Hence L number of ensemble classifiers are generated [34]. We are not considering this ensemble approach in our experiments as we have only small number of output classes i.e. 5.

4 Experimental results and Discussion

In this research we have explored and implemented various shallow and deep learning models for script identification to find the best combination of classifiers and features. To extract deep features many pre-trained models with the help of transfer learning have been used. To train and implement the deep networks, Keras a deep learning library with tensorflow as backend has been used. To fine tune the pretarined deep network GPU based system has been used. All the experiments have been done in jupyter notebook using sklearn and keras for basic machine learning and deep learning operations respectively. System with Intel Xeon(R) processor having NVE7 graphic card with 8 GB RAM has been used for experimental work.

Many works have been reported for Gurumukhi and English script, but this is the first kind of work combining various deep learning features with traditional features using transfer learning models for script identification in Gurumukhi and English text. To explore more, many ensemble approaches have been used to evaluate in-dept effect of traditional and deep learning features. In this section along with recognition accuracies for each feature and classifier set, confusion matrix for each feature set has been designed, feature visualization has been done for comparison of different feature sets.

4.1 Experimental Dataset

There is no benchmark multilingual dataset available for Gurumukhi and English pre-segmented handwritten characters. Hence for English numerals and characters we are using Extended-MNIST dataset [45] while for Gurumukhi numerals and characters dataset from Chandan et al [44] have been
considered. The bilingual dataset designed using these two datasets contains 5 different output classes as: Gurumukhi characters (GC), English uppercase characters (EC_u), English lower case characters (EC_l), Gurumukhi Numerals (GN), English Numerals (EN). For Gurumukhi script, we are considering 6174 samples of 35 characters and 1152 samples of 10 numerals for evaluation. Similarly, to avoid the class imbalance problem for English script, 2470 samples of 26 upper case letters, 1980 samples of 26 lower case letters and 1199 samples of 10 numerals are taken. It results into total dataset of 12,977 sample images. Some of the confusing character pairs present within the scripts and between the scripts are shown in Fig. 8 from the dataset. All dataset have been divided into training, testing and validation set as per the needs of experiments.

Fig. 8: sample images of confusing characters such as (a) two from Gurumukhi numerals and two from English numerals (b) dadda from Gurumukhi and kakka from Gurumukhi (c) three from English numerals and tatta from Gurumukhi numerals (d) five from English numerals and S from English characters (e) hahaha from Gurumukhi characters and J from English characters

4.2 Training models

To fine tune the pre-trained models, we have used data augmentation on training and testing datasets. All preprocessed images have been resized to the sizes as per the need of individual network. The binarization has been performed to convert gray scale images to black and white. For deep networks, training dataset with 8475 images have been used to extract the features of dataset which are validated using 3633 images of validation data. Further to test the trained model, unseen 1020 test images have been given. While for traditional classifiers like SVM, Rf, Knn and decision tree 3-fold cross validation with 11679 training and 1298 validation images has been applied. For SVM, radial basis kernel function
Table 4: Parameter details for training deep networks

| Network name | learning rate | optimizer | epoch | batch size | loss function | Activation | no. of trainable parameters |
|--------------|---------------|-----------|-------|------------|---------------|------------|-----------------------------|
| LeNet5       | 0.01          | Adam      | 11    | 128        | categorical cross entropy | Tanh       | 61, 281                     |
| Vgg19        | 0.001         | SGD       | 50    | 32         | categorical cross entropy | Relu       | 21,240,010                  |
| ResNet50     | 0.001         | SGD       | 50    | 32         | categorical cross entropy | Relu       | 26,323,082                  |

has been considered. Two parameters called: regularization parameter C and parameter gamma of SVM are optimized with the help of GridsearchCV for values of gamma and C lies between (-3,0,4) and (-1,1,3) respectively. It results into the optimal values of C and gamma for each feature set. For K-nn again we have applied GridsearchCV for optimal values of K with range between (1,25). For random forest the number of estimators used are 1000. The complete details of optimized parameters have been represented in Table 4 and 5.

4.3 Results

We have conducted the exhaustive experiments to find the most significant feature and classifier combination for script identification on our dataset.

4.3.1 Experiment 1: Script identification using Transfer learning and handcrafted features

Table 6 shows the script identification accuracy using various feature sets and classifiers. The pretrained models Vgg19 and ResNet50 trained on imagenet dataset are further fine tuned for our script identification dataset. Fine tuning of pretrained models on script identification dataset has been done to get the efficient results. While LeNet5 has been pretrained on script identification dataset from scratch. The pretrained networks are used in two ways as classifier and as deep network feature extractor for classification. Vgg19 has received the highest recognition accuracy out of pretrained models as a classifier using softmax layer i.e. 95.88%. Further to use the transfer learning, SVM, Rf and KNN classifiers have been used with features extracted from pretrained models. For feature extraction, the fully connected layers of the pretrained models have been removed and output of last convolution layer has been used as feature set. In the second scenario of transfer learning i.e pretrained models as feature extractor, features from ResNet50 have performed well with RF classifier resulting into testing accuracy of 98.01%. From traditional feature sets, HOG has performed well again with RF classifier resulting into accuracy of 94.45%. Class wise accuracy for each model has been well represented in Figure 9 and 10 using confusion matrix.
Table 5: Parameter details for training machine learning models

| feature set | SVM(c, gamma) | Rf(n_estimators) | Knn(n_estimators) |
|-------------|---------------|------------------|-------------------|
| Gabor       | 50,5          | 500              | 13                |
| GLCM        | 50,5          | 1000             | 14                |
| HOG         | 10, 0.01      | 1000             | 8                 |
| LeNet5      | 10, 0.05      | 1000             | 5                 |
| Vgg19       | linear        | 1000             | 6                 |
| ResNet50    | linear        | 1000             | 3                 |

Fig. 9: Confusion matrix for classification of data using: (a) vgg19; (b) Lenet5; and, (c) Resnet50
Table 6: script recognition accuracy for multiple classifiers using deep features and handcrafted features

| Feature   | Feature vector | SVM    | RF    | k-nn    | Softmax |
|-----------|----------------|--------|-------|---------|---------|
| Vgg19     | 25088          | 97.10  | 97.68 | 97.91   | 95.88   |
| ResNet50  | 100352         | 97.43  | 98.01 | 73.6    | 95.58   |
| LeNet5    | 400            | 94.37  | 93.22 | 92.45   | 92.30   |
| Gabor(F1) | 64             | 78     | 77    | 75.35   | -       |
| GLCM (F2) | 48             | 86     | 86    | 82.94   | -       |
| HOG(F3)   | 2400           | 93.28  | 94.45 | 92.63   | -       |

4.3.2 Experiment 3: Combining various features for script identification

To analyze the effect of combining different kinds of features, we have designed 11 combinations using deep learning and traditional features. In this experiment we are designing multiple classifiers using each feature set with three base classifiers like SVM, RF and K-nn resulting into 33 different classifiers. Table 7 represents the 3-fold cross validation accuracies obtained for multiple classifier system. SVM as base classifier has resulted highest accuracy with combination of all three traditional features i.e. gabor, HOG, GLCM while RF can result into an accuracy equal to SVM with combination of just GLCM and HOG. KNN has weak performance as compared to SVM and RF almost for all the feature sets with accuracy range lies in between 92-93 except for gabor and GLCM combination i.e. 83.63.

Table 7: script recognition accuracy for Multiple classifiers with combination of different feature sets

| Feature Set          | Feature vector | SVM    | RF    | k-nn    |
|----------------------|----------------|--------|-------|---------|
| LeNet+Gabor          | 464            | 92.61  | 90.21 | 92.62   |
| LeNet+GLCM           | 484            | 94.26  | 91.83 | 93.20   |
| LeNet+HOG            | 2800           | 93.22  | 94.14 | 92.73   |
| F2+F3                | 2448           | 93.93  | 94.53 | 92.95   |
| F1+F3                | 2464           | 93.73  | 94.14 | 92.91   |
| F1+F2                | 112            | 87.49  | 85.28 | 83.63   |
| F1+F2+F3             | 2512           | 93.93  | 94.37 | 92.97   |
| F1+F2+F4             | 512            | 94.57  | 91.83 | 93.11   |
| F1+F2+F3+F4          | 2912           | 94.04  | 94.06 | 92.97   |
| F1+F3+F4             | 2864           | 93.95  | 93.83 | 92.91   |
| F2+F3+F4             | 2848           | 94.01  | 94.11 | 93.12   |

4.3.3 Experiment 4: Script identification using Ensemble approaches

Voting based ensemble approach

To combine the complimentary nature of different classifiers designed above in multiple classifier system, we are evaluating voting based ensemble approach. Table 8 represents the cross validation
Fig. 10: Confusion matrix for classification of data using: (a) gabor features; (b) GLCM features; and, (c) HOG features

accuracies achieved for each feature set using SVM, random forest and k-nn as base classifiers. Voting based classifier gives the highest accuracy of 95.99% with combination of Gabor and HOG. Accuracy of some other feature sets also get improved. The best performing feature set combinations are LeNet-GLCM (95.53%), LeNet-HOG (95.68%), GLCM-HOG (95.83%), Gabor-HOG (95.99%), Gabor-GLCM-HOG (95.76%), Gabor-GLCM-LeNet (95.45%), Gabor-GLCM-HOG-LeNet (95.76%), Gabor-HOG-LeNet (95.83%), GLCM-HOG-LeNet (95.83%). Note that combination of traditional feature extraction i.e. Gabor and HOG have performed well from traditional features for our dataset. Accuracy of pretrained deep network models decreases as compared to individual models.

Bagging and Boosting
To further experiment with other ensemble approaches we had considered Bagging and Boosting approaches. In bagging based ensemble experiment three base classifiers linear SVM, k-nn and random forest have been used for each feature set. Table 9 represents the accuracies reported for various feature
Table 8: script recognition accuracy using voting based classifier

| Feature       | Feature vector | SVM  | RF   | k-nn  | voting based |
|---------------|----------------|------|------|-------|--------------|
| Gabor(F1)     | 64             | 78   | 77   | 75.35 | 75.19        |
| GLCM (F2)     | 48             | 86   | 86   | 82.94 | 87.59        |
| HOG(F3)       | 2400           | 93.28| 94.45| 92.63 | 95.14        |
| LeNet(F4)     | 400            | 94.69| 93.22| 92.68 | 92.75        |
| LeNet+Gabor   | 464            | 92.61| 90.21| 92.62 | 93.83        |
| LeNet+GLCM    | 484            | 94.26| 91.83| 93.20 | 95.53        |
| LeNet+HOG     | 2800           | 93.22| 94.14| 92.73 | 95.68        |
| F2+F3         | 2448           | 93.93| 94.53| 92.95 | 95.83        |
| F1+F3         | 2464           | 93.73| 94.14| 92.91 | 95.99        |
| F1+F2         | 112            | 87.49| 85.28| 83.63 | 86.97        |
| F1+F2+F3      | 2512           | 93.93| 94.37| 92.97 | 95.76        |
| F1+F2+F4      | 512            | 94.57| 91.83| 93.11 | 95.45        |
| F1+F2+F3+F4   | 2912           | 94.04| 94.06| 92.97 | 95.76        |
| F1+F3+F4      | 2864           | 93.95| 93.83| 92.91 | 95.83        |
| F2+F3+F4      | 2848           | 94.01| 94.11| 93.12 | 95.83        |
| Vgg19         | 25088          | 97.10| 97.68| 97.91 | 95.27        |
| ResNet50      | 100352         | 97.43| 98.01| 73.60 | 94.96        |

sets using bagging based ensemble under different base classifiers. For bagging k-nn has performed good comparable to linear SVM and random forest for most of the feature sets especially for traditional features. However, It was performing weakly as individual classifier. Hence, it proves that bagging ensemble approach increases the accuracy of weak individual classifiers. Some of the good accuracies achieved by K-nn as base classifier with best performing feature sets are: HOG(93.30%), LeNet-GLCM (93.92%), LeNet-HOG (93.54%), Gabor-HOG (93.23%), Gabor-GLCM-HOG (93.36%), Gabor-HOG-LeNet (93.35%). The worst case accuracy has been reported with SVM (individually strong classifier) as base classifier and feature set of LeNet and Gabor i.e. 67.24%. Further for pretrained models , we have noticed the highest accuracy of 98.82% by ResNet50 features with linear SVM classifier.

Similarly to evaluate boosting based ensemble approach we have chosen Adaboost ensemble classifier. Linear SVM, Decision tree and random forest have been used as base classifier for Adaboost ensemble. Each feature set has been trained on each Adaboost ensemble classifier and reported the accuracy in table 10. Adaboost classifier has performed good with random forest as base classifier with maximum accuracy of 92.44% having HOG as feature extractor from the category of traditional classifiers. For SVM, Adaboost has reported the lowest accuracies as compared to other approaches used in the work for traditional methods. Decision tree as base classifier reports the minimum accuracy of 31.04% for Deep learning features extracted. For pretrained models, Adaboost has achieved maximum accuracy of 98.43% with combination of ResNet50 and SVM.
Table 9: script recognition accuracy using bagging based ensemble classifier

| Feature          | Feature vector | SVM    | knn    | rf     |
|------------------|----------------|--------|--------|--------|
| Gabor(F1)        | 64             | 77.02  | 67.62  | 74.51  |
| GLCM (F2)        | 48             | 75.33  | 83.53  | 83.16  |
| HOG(F3)          | 2400           | 84.80  | 93.30  | 91.88  |
| LeNet(F4)        | 400            | 88.28  | 92.45  | 90.59  |
| LeNet+Gabor      | 464            | 67.24  | 92.20  | 88.35  |
| LeNet+GLCM       | 484            | 76.85  | 93.92  | 89.71  |
| LeNet+HOG        | 2800           | 85.36  | 93.54  | 92.06  |
| F2+F3            | 2448           | 84.10  | 92.90  | 92.48  |
| F1+F3            | 2464           | 84.08  | 93.23  | 92.13  |
| F1+F2            | 112            | 74.63  | 84.00  | 83.20  |
| F1+F2+F3         | 2512           | 85.55  | 93.36  | 92.33  |
| F1+F2+F4         | 512            | 72.48  | 92.81  | 89.79  |
| F1+F2+F3+F4      | 2912           | 84.33  | 92.87  | 91.17  |
| F1+F3+F4         | 2864           | 83.70  | 93.35  | 92.01  |
| F2+F3+F4         | 2848           | 83.84  | 92.91  | 92.21  |
| vgg19            | 25088          | 98.61  | 97.76  | 98.16  |
| ResNet50         | 100352         | 98.82  | 70.50  | 98.57  |

4.3.4 Experiment 6: comparison of pre trained deep networks and traditional classifiers using traditional feature set

In this work we have evaluated different kinds of pretrained models with varying depths and the other side many traditional classifiers. In pretrained models, LeNet5 has the smallest depth as compare to Vgg19 and ResNet50, which results into the lower accuracy for our dataset. When we go deeper into the network by increasing the depth to 19 with Vgg19, marginal improvement has been noticed in accuracy for almost all the classifiers. Further by moving more deeper into the network with ResNet50 having depth of 50 layers, accuracy has improvement of approximately by 1% . But, increasing the number of layers in network increases the training cost of network in terms of trainable parameters and time taken for training. Hence, we observed that, increasing the layers in deep networks i.e. switching from LeNet5 to Vgg19 is a good trade-off between accuracy and training cost while increasing the depth after some extent such as from Vgg19 to ResNet50 is not reasonable as training cost is higher than increase in accuracy. Number of trainable parameters for Vgg19 was 21,240,010 while for ResNet50 was 26,323,082. On other hand for traditional features, the accuracy was not so good and parameter optimization was another major challenge. Somehow gridsearchCV has helped in finding the optimal parameters but it was time consuming task. Moreover, from feature visualization (Fig. 11) we found that features extracted from deep networks are enough capable to differentiate the classes without
Table 10: script recognition accuracy using boosting based ensemble classifier

| Feature       | Feature vector | SVM  | rf   | Decision Tree |
|---------------|----------------|------|------|---------------|
| Gabor (F1)    | 64             | 49.15| 76.34| 68.56         |
| GLCM (F2)     | 48             | 52.54| 83.97| 69.26         |
| HOG (F3)      | 2400           | 65.87| 92.44| 68.87         |
| LeNet (F4)    | 400            | 59.01| 90.60| 31.04         |
| LeNet + Gabor | 464            | 52.91| 85.82| 68.25         |
| LeNet + GLCM  | 484            | 53.01| 87.75| 68.56         |
| LeNet + HOG   | 2800           | 63.11| 91.91| 48.68         |
| F2 + F3       | 2448           | 52.96| 87.15| 71.18         |
| F1 + F3       | 2464           | 58.22| 90.83| 69.02         |
| F1 + F2       | 112            | 51.55| 82.89| 68.25         |
| F1 + F2 + F3  | 2512           | 53.88| 91.37| 68.33         |
| F1 + F2 + F4  | 512            | 52.45| 90.21| 67.41         |
| F1 + F2 + F3 + F4 | 2912 | 53.77| 91.67| 70.10         |
| F1 + F3 + F4  | 2864           | 52.78| 91.83| 71.64         |
| F2 + F3 + F4  | 2848           | 62.18| 92.37| 70.03         |
| Vgg19         | 25088          | 96.12| 97.27| 75.57         |
| ResNet50      | 100352         | 98.43| 97.60| 82.24         |

traditional classifiers while for traditional feature sets the same was not true (Fig. 12). Traditional feature sets further need machine learning classifiers to classify the objects.

4.3.5 Experiment 6: Comparative analysis with previous work

In this section we are performing the comparative analysis for the performance of various models with the proposed one. Table 6 to 10 gives the complete analysis of various results found using different feature sets with different classification techniques. The proposed approaches have been compared in Table 11 with existing state-of-the-art approaches for script identification. Table 11 is dictating that for script identification at character level for Gurumukhi and English scripts, our method is performing well for handwritten data. Improvement of almost 6% has been reported for handwritten dataset at character level over the existing accuracy reported by [31] i.e. 92.89%.
Fig. 11: Feature visualization using t-SNE: (a) Vgg19 features; (b) ResNet50 features; and, (c) LeNet5 features
Fig. 12: Feature visualization using t-SNE for traditional feature set: (a) gabor features; (b) glcm features; and, (c) hog features
### Table 11: Comparative Analysis of the proposed work with state-of-the-art

| Author            | Text mode | level       | Feature set          | Script                        | Dataset          | Accuracy        |
|-------------------|-----------|-------------|----------------------|-------------------------------|------------------|-----------------|
| Pati et al [26]   | Printed   | word level  | Gabor and DCT        | 11 Indic and non Indic scripts| 20,000 words     | 94.80 average script recognition accuracy% |
| Ul-Hasan et al [17] | printed  | character   | deep learning LSTM   | English and Greek             | 2,89,013         | 98.186%         |
| Pandey et al [29]  | handwritten | word level  | DCT and DWT          | Hindi, Urdu and English       | 961 handwritten samples | 82.70%          |
| Rani et al [30]    | printed   | character   | Gabor and Gradient   | Gurumukhi and English         | 19,448 characters | 98.9% and 99.45% |
| Kumar et al [31]   | handwritten | character   | statistical and structural | English, Hindi and Punjabi | 92.89%          |
| proposed           | handwritten | character   | Transfer learning and traditional features | English and Gurumukhi characters and numerals | 12,977          | 98.82%          |
4.4 Feature visualization for Deep learning and Handcrafted features

Major focus of research was to explore the different kinds of features, As pretrained model have power to automatically extract the features using number of layers from data. It makes interesting to visualize the kind of features extracted by deep networks and how they differentiate each other in multi classes. Here in our experiments we have applied t-SNE approach for feature visualization. t-SNE visualization approaches construct the pairwise similar and dissimilar objects using probability distributions. To find the similarity in data some kind of distance like Euclidean distance has been used. Fig 11 shows the features learned using various pre trained models. As with Vgg19 and ResNet50 features, a clear separation between the various classes can be found. It depicts the high accuracy achieved by these models during classification. On the other hand in LeNet5, we see that data of some classes overlaps with other classes showing the misclassification of data. Class 3 and 4 has clear separation from class 0,1,2 which shows high classification while classes 0,1,2 have mixing data showing the misclassification.

4.5 Discussion

We have evaluated the script identification in Gurumukhi and English script for numeral as well as character dataset with 5 different classes. Many handcrafted and deep learning features have been used to find the best match with efficient accuracy. We proposed the best combination of feature sets and classifiers for script identification. We observed that deep features extracted from pretrained models have performed much better as compared to traditional feature for classification. The use of transfer learning has saved a lot of time for feature extraction using deep models. The pretrained models was trained on imagenet dataset which were further fine tuned on handwritten character dataset. Another high point of deep features was parameter learning during training of network unlike in handcrafted features. Real data are non linear in nature and deep networks are best in generalizing such non linear data due to the use of activation function which is not possible in handcrafted features. Further, the results in the form of confusion matrix as well as from feature visualization graphs depicts that, the miss classification mostly occurs in between upper case English characters as well as lower case English characters. But the major identification between English and Gurumukhi is satisfiable. Results in the form of table reveal that boosting based ensemble methods are performing worse than other approaches. For our datasets, Adaboost has reported lower accuracy even using the weakest base classifier i.e. decision tree compared to voting and bagging based classifiers. Adaboost reports an improvement of 20.80% and 26.57% with random forest as base classifier and combination of traditional and deep learning features(LeNet and HOG) over decision tree and SVM as base classifiers respectively.

For bagging based ensemble approach K-NN as base classifier has performed well almost for each feature set as compared to other base classifier SVM and random forest (table 9). Here, K-NN as base classifier had performed 7.99% better then SVM and 1.06% better then random forest. Combination of deep learning and traditional features(HOG) has given highest results with K-nn. For SVM and random forest as base classifier in bagging works good with combination of traditional feature extraction methods (HOG, GLCM, Gabor and GLCM, HOG respectively).
For multi classifier system voting based ensemble with SVM, random forest and k-nn as base classifier has outperformed compared to all other ensemble classifiers in this work. Combination of traditional feature extraction approaches i.e. Gabor and HOG with voting based ensemble approach has reported an improvement of 1.3%, 1.46% and 2.79% over individual classifier SVM, random forest and k-nn respectively.

5 Conclusion and Future directions

In this paper we have performed script recognition at character level for Bilingual handwritten documents containing English and Gurumukhi script with finding the best combination of feature set and classifiers. For feature extraction, 17 different kinds of features including traditional (Gabor, HOG, GLCM) and deep learning have been extracted. To obtain the potential of traditional features, combination of deep learning and traditional features are considered. To utilize the benefits of transfer learning, pretrained models have been fine tuned to be used as classifier and feature extractor for script identification. Many classifiers such as SVM, random forest, k-nn, decision tree have been trained on different features. Further, ensemble based classifier are employed to boost the recognition accuracy for script identification. The best result of 98.82% has been found with ResNet50 using ensemble based bagging approach. In future, the proposed combination approach can be used for character recognition of individual script and for many similar other scripts. Moreover, other feature and classifier combinations can be proposed to improve the script recognition accuracy.
Compliance with ethical standards

Ethical Approval This article does not contain any studies with human participants by any of the authors.

Funding Details This research was supported by Council of Scientific and Industrial Research (CSIR), India under Grant number 09/677(0031)/2018/EMR-I funded by the Ministry of Science and Technology, Government of India.

Conflict of interest The authors declare that they have no conflict of interest.

Informed Consent Informed consent was obtained from all individual participants included in the study.

Authorship Contributions

Authors contribution All authors contributed to the study, conception, data collection and design of the research. The first draft of manuscript was written by SK and all other authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

Sharma, Manoj Kumar, and Vijay Pal Dhaka. "Pixel plot and trace based segmentation method for bilingual handwritten scripts using feedforward neural network.” Neural Computing and Applications 27.7 (2016): 1817-1829.

Sahare, Parul, and Sanjay B. Dhok. "Multilingual character segmentation and recognition schemes for Indian document images.” IEEE Access 6 (2018): 10603-10617.

Singh, Pawan Kumar, et al. ”Benchmark databases of handwritten Bangla-Roman and Devanagari-Roman mixed-script document images.” Multimedia Tools and Applications 77.7 (2018): 8441-8473.

Bhumia, Ankan Kumar, et al. ”Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network.” Pattern Recognition 85 (2019): 172-184.

Sarkhel, Ritesh, et al. ”A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts.” Pattern Recognition 71 (2017): 78-93.

Nanni, Loris, Stefano Ghidoni, and Sheryl Brahnam. ”Handcrafted vs. non-handcrafted features for computer vision classification.” Pattern Recognition 71 (2017): 158-172.

Nguyen, Dat Tien, et al. ”Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors.” Sensors 18.3 (2018): 699.

Hosseini, Sepidehsadat, Seok Hee Lee, and Nam Ik Cho. ”Feeding hand-crafted features for enhancing the performance of convolutional neural networks.” arXiv preprint arXiv:1801.07848 (2018).

Mukhopadhyay, Anirban, et al. ”A study of different classifier combination approaches for handwritten Indic Script Recognition.” Journal of Imaging 4.2 (2018): 39.
Rahman, Ahmad Fuad Rezaur, and Michael C. Fairhurst. "Multiple classifier decision combination strategies for character recognition: A review." Document Analysis and Recognition 5.4 (2003): 166-194.

Kumar, Pradeep, et al. "A lexicon-free approach for 3D handwriting recognition using classifier combination." Pattern Recognition Letters 103 (2018): 1-7.

Chherawala, Youssouf, Partha Pratim Roy, and Mohamed Cheriet. "Combination of context-dependent bidirectional long short-term memory classifiers for robust offline handwriting recognition." Pattern Recognition Letters 90 (2017): 58-64.

Brodić, Darko, Alessia Amelio, and Zoran N. Milivojević. "Identification of Fraktur and Latin scripts in German historical documents using image texture analysis." Applied Artificial Intelligence 30.5 (2016): 379-395.

Shi, Baoguang, Xiang Bai, and Cong Yao. "Script identification in the wild via discriminative convolutional neural network." Pattern Recognition 52 (2016): 448-458.

Mandal, Ranju, et al. "Multi-lingual date field extraction for automatic document retrieval by machine." Information Sciences 314 (2015): 277-292.

Tian, Shangxuan, et al. "Multilingual scene character recognition with co-occurrence of histogram of oriented gradients." Pattern Recognition 51 (2016): 125-134.

Ul-Hasan, Adnan, et al. "A sequence learning approach for multiple script identification." 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015.

Ahmed, Saad Bin, et al. "Evaluation of cursive and non-cursive scripts using recurrent neural networks." Neural Computing and Applications 27.3 (2016): 603-613.

Rahman, Akhlaqur, and Sumaira Tasnim. "Ensemble classifiers and their applications: a review." arXiv preprint arXiv:1404.4088 (2014).

Günter, Simon, and Horst Bunke. "Ensembles of classifiers for handwritten word recognition." Document Analysis and Recognition 5.4 (2003): 224-232.

Kessentini, Yousri, Sana BenAbderrahim, and Chawki Djeddi. "Evidential combination of svm classifiers for writer recognition." Neurocomputing 313 (2018): 1-13.

Angadi, S. A., and M. M. Kodabagi. "A fuzzy approach for word level script identification of text in low resolution display board images using wavelet features." 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2013.

Chanda, Sukalpa, Oriol Ramos Terrades, and Umapada Pal. "SVM based scheme for Thai and English script identification." Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). Vol. 1. IEEE, 2007.

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
Nguyen, Dat Tien, et al. "Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors." Sensors 18.3 (2018): 699.

Pati, Peeta Basa, and A. G. Ramakrishnan. "Word level multi-script identification." Pattern Recognition Letters 29.9 (2008): 1218-1229.

Tan, Guo Xian, Christian Viard-Gaudin, and Alex C. Kot. "Information retrieval model for online handwritten script identification." 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009.

Kavitha, S., et al. "A robust script identification system for historical indian document images.” Malaysian Journal of Computer Science 28.4 (2015): 283-300.

Pandey, Akhilesh, et al. ”Handwritten script recognition using soft computing.” Int J Adv Res Comput Sci Electron Eng 1.6 (2012): 6-11.

Rani, Rajneesh, Renu Dhir, and Gurpreet Singh Lehal. "Script identification of pre-segmented multi-font characters and digits." 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.

Kumar, Munish, and Simpel Rani Jindal. "A Study on Recognition of Pre-segmented Handwritten Multi-lingual Characters.” Archives of Computational Methods in Engineering (2019): 1-13.

Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting.” Journal of computer and system sciences 55.1 (1997): 119-139.

Dietterich, Thomas G. "Ensemble methods in machine learning.” International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000.

Kacem, Afef, and Asma Saïdani. "A texture-based approach for word script and nature identification.” Pattern Analysis and Applications 20.4 (2017): 1157-1167.

Shi, Baoguang, Xiang Bai, and Cong Yao. "Script identification in the wild via discriminative convolutional neural network.” Pattern Recognition 52 (2016): 448-458.

Pal, U., and B. B. Chaudhuri. "Automatic identification of english, chinese, arabic, devnagari and bangla script line.” Proceedings of Sixth International Conference on Document Analysis and Recognition. IEEE, 2001.

Pati, Peeta Basa, and A. G. Ramakrishnan. "Word level multi-script identification.” Pattern Recognition Letters 29.9 (2008): 1218-1229.

Rani, Rajneesh, Renu Dhir, and Gurpreet Singh Lehal. "Performance analysis of feature extractors and classifiers for script recognition of English and Gurmukhi words.” Proceeding of the workshop on Document Analysis and Recognition. ACM, 2012.
Roy, K., U. Pal, and B. B. Chaudhuri. "Neural network based word-wise handwritten script identification system for Indian postal automation." Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing, 2005.. IEEE, 2005.

Pan, W. M., Ching Y. Suen, and Tien D. Bui. "Script identification using steerable Gabor filters." Eighth International Conference on Document Analysis and Recognition (ICDAR’05). IEEE, 2005.

Pal, Umapada, et al. "Handwritten numeral recognition of six popular Indian scripts." Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). Vol. 2. IEEE, 2007.

Busch, Andrew, Wageeh W. Boles, and Sridha Sridharan. "Texture for script identification." IEEE Transactions on Pattern Analysis and Machine Intelligence 27.11 (2005): 1720-1732.

Aggarwal, Ashutosh, and Chandan Singh. "Zernike moments-based Gurumukhi character recognition." Applied Artificial Intelligence 30.5 (2016): 429-444.

Cohen, Gregory, et al. "EMNIST: an extension of MNIST to handwritten letters." arXiv preprint arXiv:1702.05373 (2017).

Chanda, Sukalpa, Katrin Franke, and Umapada Pal. "Identification of Indic scripts on torn-documents." 2011 International Conference on Document Analysis and Recognition. IEEE, 2011.

Gupta, Deepika, and Soumen Bag. "CNN-based multilingual handwritten numeral recognition: A fusion-free approach." Expert Systems with Applications 165 (2021): 113784.

Alkhawaldeh, Rami S. "Arabic (Indian) digit handwritten recognition using recurrent transfer deep architecture." Soft Computing 25.4 (2021): 3131-3141.

Cheng, Changxu, et al. "Patch aggregator for scene text script identification." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

Pramanik, Rahul, and Soumen Bag. "Handwritten Bangla city name word recognition using CNN-based transfer learning and FCN." Neural Computing and Applications (2021): 1-13.