An Insight for Cursive Context-Specific Printed Script Recognition

Humera Rafique [1¶], Tariq Javid [1¶]

[1] Graduate School of Engineering Sciences & Information Technology,

Faculty of Engineering *Sciences & Technology, Hamdard University,*

Karachi, Pakistan

Corresponding author:

rafiqueh@hotmail.com

Abstract

The greatest challenge of machine learning problems is to select suitable techniques and resources such as tools and datasets. Despite the existence of millions of speakers around the globe and the rich literary history of more than a thousand years, it is expensive to find the computational linguistic work related to Punjabi Shahmukhi script, a member of the Perso-Arabic context-specific script low-resource language family. This paper presents a deep insight into the related work with summary statistics, advocating the popularity and success of artificial neural networks and related techniques. The paper includes support from recent trends from the authentic sources based on the top-level researchers' feedback including the machine learning frameworks. A comprehensive comparison of the most popular deep learning techniques convolutional neural network and the recursive neural network has been presented for the cursive context-specific scripts of Perso-Arabic nature. The overview of the available benchmark datasets for machine learning problems, especially for the Perso-Arabic group, is added. This paper incorporates essential knowledge contents for the researchers in machine learning and natural language processing disciplines on the selection of algorithms, architectures, and resources.

Keywords: ***Machine Learning, Pattern Analysis, Natural Language Processing, Artificial multilayered Neural Networks, Optical Character Recognition system, Benchmark dataset.***

Introduction

Background

To preserve history, knowledge, culture, facts, and figures for the present and future generations, written text is a medium of human communication for language expression with symbols. The written text is a durable and authentic alternative to human speech for centuries in various forms. The field of automated script recognition has achieved significant real-world success for daily life applications, including writer's identification, recognition of home address, written messages, signatures, legal documents and bank cheques, signature verification, to name a few. Alphabet (strokes and ligatures) recognition is the foundation of a written text identification task approaching two or more letter words' identification. The advanced approaches are to develop systems for reading complete statements in context, completing incomplete words and sentences, spelling and grammar correction, and translation to other languages. Almost every language research group around the world is engaged in the development of machine learning systems for the native language.

The scholarly work related to western languages is at its pinnacle however, Asian languages are in the way of development. Punjabi language with over 125 million speakers around the world can be written mainly in two scripts Shahmukhi and Gurmukhi. Gurmukhi is an Indo-Aryan language script, used in the Indian side of Punjab, written in Landa script, is a derivative of the Devanagari style. Shahmukhi, a Perso-Arabic script, used to write the Punjabi language, at the Pakistan side of Punjab, officially written in Nasta'leeq font, borrowed from Urdu. Fig 1. shows the word Punjabi written in two scripts. The language has an older history, more historic and scholarly work; however, a small development in machine understanding, with humble quality and lack of standardization.

Objective

To develop a quality machine learning system capable of recognizing the printed or handwritten Perso-Arabic scripts, it is desirable to know the characteristics of the systems suitable for cursive context-specific scripts. This paper presents a detailed insight into the classification and recognition methodologies specific to the Perso-Arabic family of scripts using Machine learning techniques for optical character recognition (OCR), a procedure applied to printed or handwritten texts for the representation of computer-generated file version of it.

A general OCR algorithm includes data acquisition (text scanning), local segmentation (extraction of text from the images), preprocessing, text segmentation (ligatures and strokes), feature extraction, classification, and post-processing. Its general components are shown in Fig 2.

Fig 1. Two scripts of the Punjabi language (left Gurmukhi, right Shahmukhi in Nasta'liq)

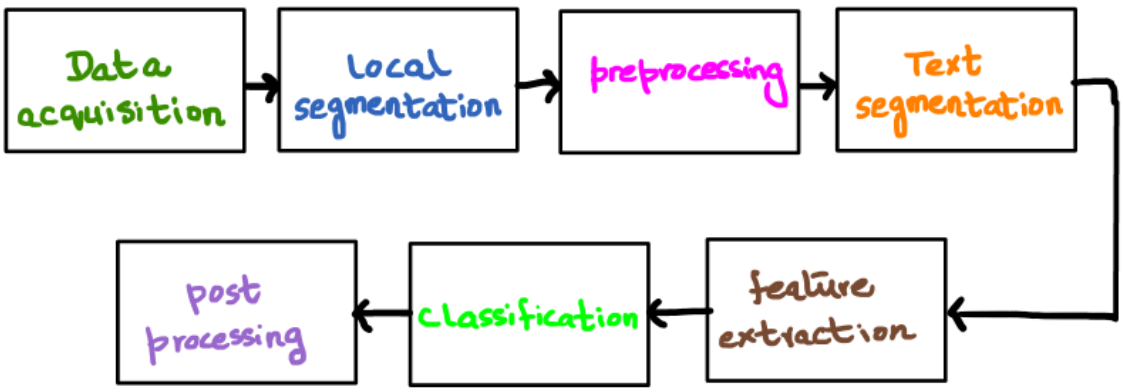

Fig 2. General components of the OCR

This paper is divided into the following sections: Introduction, Related work, Machine learning techniques suitable for cursive context-specific scripts, variants of CNN and RNN architectures, and machine learning datasets.

Related Work

Most Asian languages are considered low-resourced languages. In the Perso-Arabic group Arabic and Persian languages have some growing trends towards computational linguistic research, but Urdu with millions of speakers worldwide, has caught less attention of the researchers comparatively, and the case becomes worst when it comes to the 'Shahmukhi' script, compared with 'Gurmukhi' [**1**]. The previous work addresses the Shahmukhi script recognition and classification is rare. Therefore, a referral approach has been adapted in which the language of the same group (Perso-Arabic) with similar scripts has been included in the literature review. The first category is the computational linguistic (CL) work of the Punjabi Shahmukhi script. The second category describes the computational linguistic work on other languages of the Perso-Arabic group, such as Arabic, and Urdu. The third category describes the database design-related work.

The research questions include objective, language/script, methodology, dataset, framework, or customized program code available, the accuracy of the system. Research questions for the third category include dataset type Printed (synthetic, or scanned), handwritten (forms; images; scanned), language/script: Perso-Arabic (Shahmukhi; Urdu; Arabic; others), Lattin (English), database type: customized dataset, available (modification: vital changes or additions), data collection methods, and characteristics (database name/title, size, type, etc.), image formats, data file types, image size, resolution, preprocessed or raw, tested, tagged and percentage of success.

In [**2**], the authors presented a Shahmukhi corpus using built-in frameworks (Word2Vec and Gensim Lib) for the named entity recognition (NER) of proper nouns using supervised learning techniques (CRF, HMM, and RNN) for the validation of their corpus with a success rate of 85% (for RNN). In [**1**], the authors presented similar work for the development of Punjabi morphology, corpus, and lexicon using the built-in (GF) framework. In [**3**]–[**5**], authors have used rule-based approaches to construct transliteration systems between two scripts of the Punjabi language

Shahmukhi and Gurmukhi. Stemming of Punjabi words is carried out in [**6**] and [**7**] using the rule-based approach and Porter's algorithms respectively with good accuracy. Punjabi stop words extraction is carried out in [**8**] using Gurmukhi. Shahmukhi mapping (lookup) method. A word extraction from sentences has been carried out in [**9**] using a rule-based approach. Using their space-based segmentation technique, this work has developed a dataset from newspapers and online resources of three million words. A two-million-word tagged corpus of semantically related words of Shahmukhi has been developed by [**10**]. In [**11**], the authors presented a survey of machine translation techniques between Punjabi and Urdu languages.

In the Perso-Arabic category, the classification of more than 3000 ligatures, printed in Urdu font 'Noori Nasta'leeq' from custom-built CLE-dataset has been presented in [**12**] using DCT sliding window and HMM (HMM Toolkit). A non-holistic approach of Arabic script recognition has been carried out in [**13**] using convolutional neural networks (CNN) implemented in Keras-LeNet-5 architecture with Nvidia Tesla GPU, using single-font images from the Arabic printed text image multi-font (APTID-MF) dataset. An MDLSTM classifier (works on extracted features by CovNet), has been trained on the MNIST dataset first and then applied to Urdu handwritten text dataset UNHD, exploiting the transfer learning technique in [**14**].

Statistics of The Related Work

The field study has been performed over six months, between October 2020 to April 2021. From 2006 to 2021 the related work has been sifted from reputable sources such as IEEE, Springer, Elsevier, and others. With 70% journal and 30% conference papers, 6% were survey papers, whereas 94% were from applied research, with 85% of classification and OCR-based work. Among these Neural networks, HMM, and SVM has been found most popular.Fig 3**,** shows the contribution of machine learning techniques related to OCR and classification. Their average percentage values of success are illustrated in Fig 4., showing that neural networks are the most suitable technique for the family of cursive context-specific scripts.

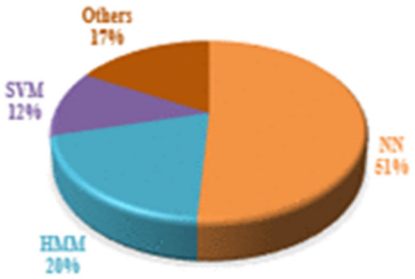

Fig 3. Contribution of Machine learning techniques in applied research work

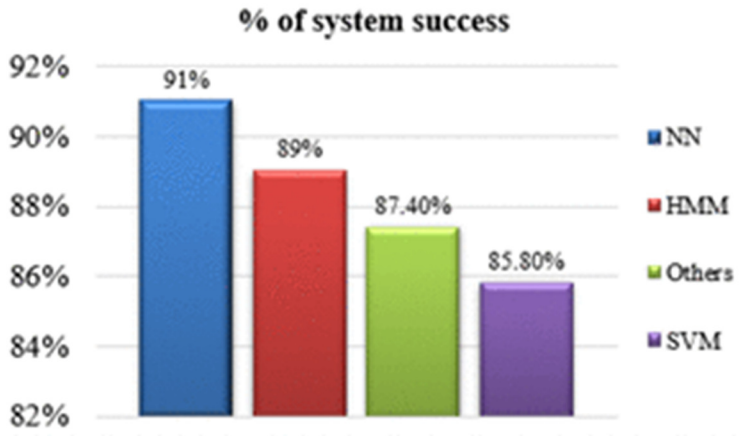

Fig 4. Success average percentage of machine learning techniques he family of cursive context-specific scripts

Machine Learning Techniques for The Cursive Context-Specific Scripts

A better way to reduce the experimentation efforts is to study the best-suited characteristics of the algorithms and architecture from the available work. **Table 1** summarizes the survey work [**15**] of text classification techniques, describing machine learning techniques are superior based on their usage popularity and accuracy compared to the rule-based techniques. **Table 2** summarizes the performance of deep learning, naïve Bayes, and SVM. [**15**].

Table 1. Comparison of text classification approaches

Approach	Usage	Working	Accuracy	Computational level
Rule-based	⇑	Handcrafted learning (linguistic) rules	Moderate; poorly compared with ML	Variable; mostly easy
Machine learning	⇑	Features based on data	High	High; easier due to available frameworks
Hybrid	~	Combination	High	Variable

Table 2. Comparison of Text Classification Approaches

Approach	**Usage**	**Data size required**	**Computational power/resources required**	**Accuracy**
Naïve Bayes	Popular	Small	Low	Good
SVM	Popular	Small	High; medium for multiclass problems	Good
Deep learning	Popular	Large	High	Best

Available Neural Network Architectures

A list of available artificial neural network (ANN) architectures is given in

Table 3 showing popular architectures used in applications like image classification, OCR, and NLP. Related research works are also mentioned where CNN, RNN, and their variants are popular architectures. CNNs are multi-layered Neural networks, to recognize visual patterns straight from image pixels with minimal preprocessing requirements. RNNs due to ‘direct-graph-temporal-sequence’ architecture are best suited for the classification of the variable-length input sequences, e.g., ligatures.

Table 3. ANN architectures

NN Architecture	Published
Perceptron (P) and multilayered perceptron	**1958**
Deep CNN, [**13**], [**17**]–[**22**]	**1980s**
Hopfield network, Kohonen Net	**1982**
Boltzmann machine; restricted BM	**1985**
Recurrent Neural Network/ MDRNN, [**2**], [**14**], [**19**], [**20**], [**23**], [**24**]	**1986**
Autoencoder/sparse ae	**1986**
Radial Bias Networks	**1988**
Long short-term memory, [**17**], [**19**], [**20**], [**24**]	**1997**
Capsule Net (CAPSNET)	**2000**
Extreme Learning Machine	**2001**
Deep Belief Network	**2006**
Deconvolutional Neural Net (DNN)	**2010**
Gated Recurrent Unit, Generative Adversarial Net (GAN), Neural Turing Machine (NTM)	**2014**
Deep Conv. Inverse Graphic Net (DC-IGN)	**2015**
Markov Chain Nnet	**2018**

Proposed Architectures

CNN and RNN the subclasses of artificial neural networks (ANNs) are layered architecture comprising of three main layers: Input, output, and middle (hidden) layers. With many hidden layers, they are called deep ANNs. These layers are composed of cells (neurons) that work together to perform pattern recognition tasks. ANNs can be further categorized as cyclic and acyclic ANNs; cyclic have feedback connections; whereas, acyclic have not, and also called feedforward networks (FFNs) [**16**].

CNN is a unique class of multi-layer feed-forward neural networks with the superb ability of typed and handwritten text recognition. It is primarily centered around on-the-spot learning. It can learn variable, complex, non-linear trends from the input images. Generally, CNN comprises two stages: feature extraction (convolution layers, pooling, and an activation function) and the classification stage (the fully connected and the classification layers). Researchers have used different CNN architectures for the text and character recognition tasks for a range of Perso-Arabic scripts. **Fig 5** shows the basic and general architecture outline of the CNN.

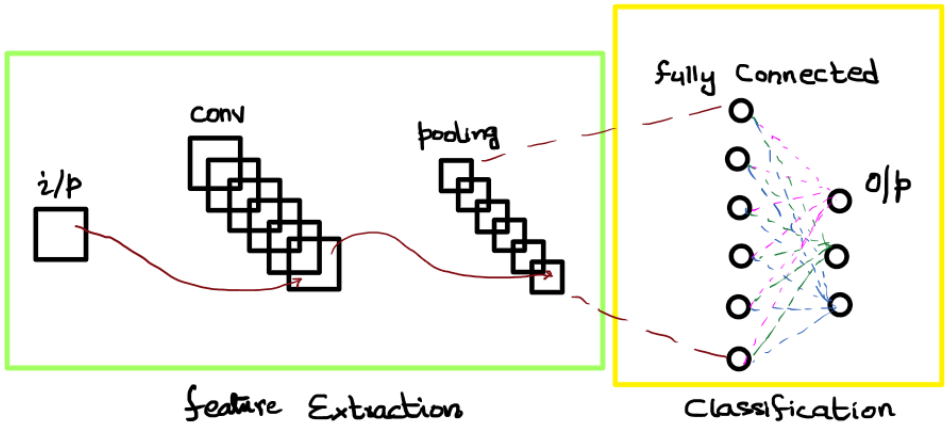

Fig 5. Outline of CNN Architecture [https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697]

This architecture shows distinct features for word and character recognition of cursive context-specific scripts and fonts:

CNNs do not require hand-crafted features. To a certain limit, they can learn noise impermeable, shift, translation, and shape distortions invariant, silent features from the training data. [**13**], [**19**]. This feature outperforms many other traditional techniques with a considerable error rate reduction [**18**].

Low chances of overfitting and reduced computational complexities due to CNN weight sharing and local connectivity properties, thus better learning capacity [**19**]. Due to convolution operation, CNN keeps the spatial features of the input image intact, thus used for temporal information (visual data) classification and feature extraction. Due to large data size requirements and expensive computational cost, the use of CNN may be limited to feature

extraction or to use a transfer approach of using a pre-trained CNN [**19**], such as (AlexNet, VGG16, VGG19, GoogleNet, InceptionV3, and ResNet101) [**18**].

The presence of the pooling stage in CNN reduces the dimensionality of the feature maps. In the training phase, the filter weights' adjustment to loss function provides customized optimal filter layers for a specific machine learning problem.

The RNN works with a minimum of one feedback element thus activation circulates in a loop. The recurring path between hidden cells causes a generative and sequential learning modeling characteristic [**24**], useful in speech recognition, machine translation, handwriting recognition, and generative modeling (e.g., simulated datasets). The basic architecture is shown in Fig 6**.**

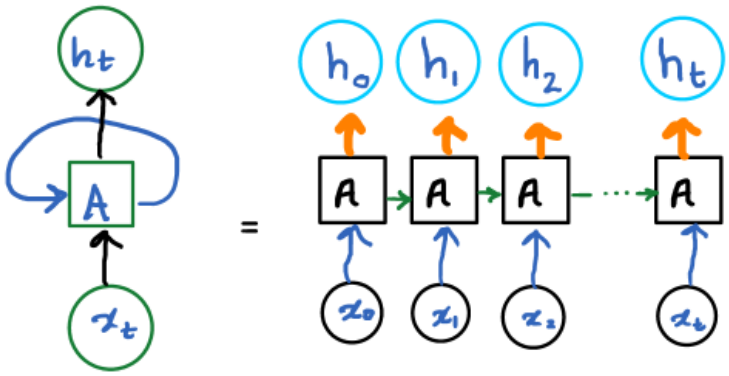

Fig 6. The basic architecture of RNN [pydeeeplearning.weebly.com]

Some of the characteristics are given:

When previous information is needed to comprehend the recent (or delayed) stage, RNN is the best choice. It can handle variable lengths of input sequences thus suitable for machine learning tasks of NLP and the network gets unaffected in terms of its model size. Its popular applications are sentiment analysis in text and speech, name entity recognition, and machine translation.

RNNs are suitable for temporal information classification as CNNs; however, owing to their feedback property, they can handle sequences like video frames in a better way comparatively. RNNs are computationally efficient; however, generally harder to train compared with CNN, thus requiring pre-extracted features or a CNN stage before the classification stage [**20**].

During backpropagation, an RNN might suffer the vanishing gradient problem where the gradient becomes very small, inappropriate to modify the weights during the training phase. The solution is long-short-term-memory RNN (LSTM- RNN). Fig 7 shows the simple architecture of the LSTM network. LSTM-RNN can filter the required information as per the requirements of the task during the training phase using its memory units [**20**]. These gates are responsible to control the activation of the internal unit. The RNN-LSTM is best suited with the CTC layer. A transfer learning approach (Tl-Deep-MDLSTM) can produce promising results [**14**].

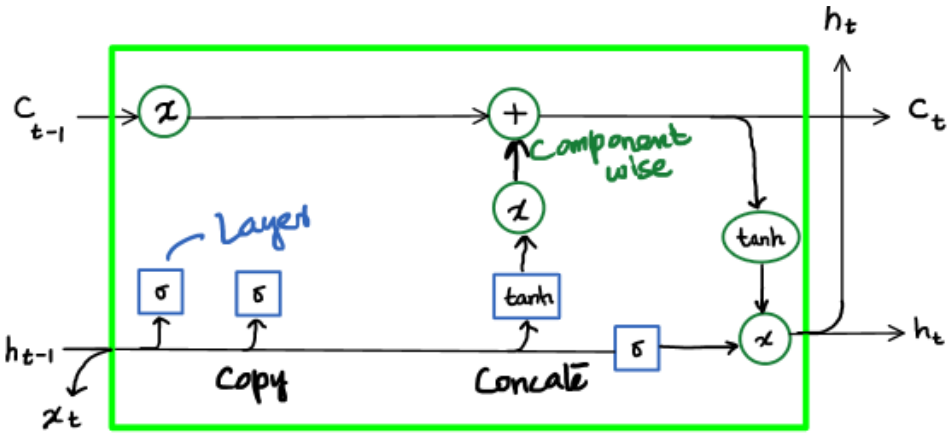

Fig 7. LSTM architecture [Wikipedia.com: LSTM]

Bidirectional RNN (BRNNs) are two independent RNN connected in forward and reverse directions. Thus, a BRNN gets trained twice to the input sequence. Preservation ability of temporal and contextual characteristics of the events, RNN and its variants BD-LSTM and BLSTM are very popular for cursive writing recognition [**24**]. Integration of BDRNN and BLSTM results in MDLSTM architecture, shown in Fig 8. The MDLSTM is recommended for cursive context-specific scripts as it can handle almost all complexities of this family. [**16**]. An output CTC layer supplies a tremendous result.

MDRNN can handle multidimensional data due to multiple recurrent connections. The variants of RNN perform better than RNN at high computational costs.

Architectures of CNN and RNN

Some of the popular architectures have been presented here, used in cursive context-specific script recognition problems.

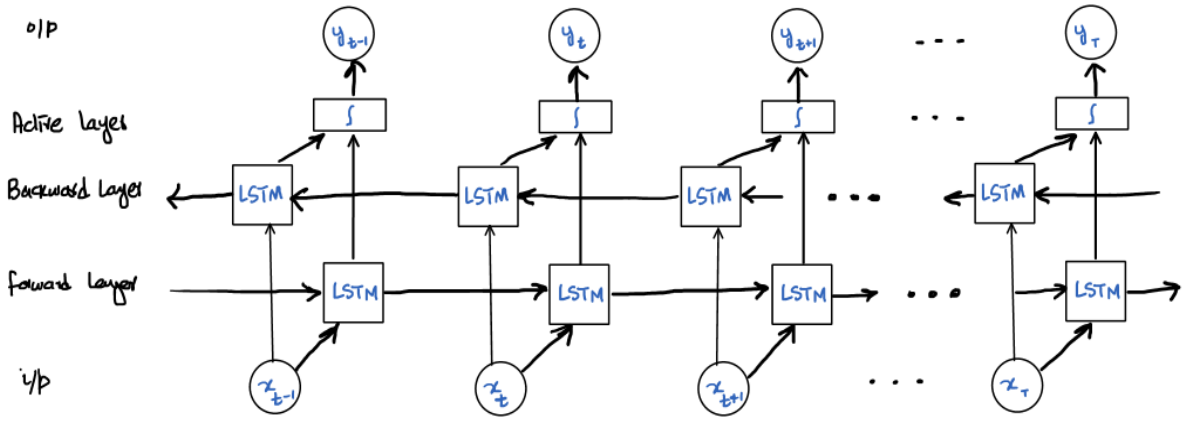

Fig 8. MD-LSTM architecture [https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/]

CNN Architectures

LeNet-5: Invented in 1989, principally a classic CNN architecture and simplest one. With two convolutional (and pooling) layers and three fully connected layers (total 5), with around 60,000 parameters. In [**13**] authors have used this architecture for Arabic text segmentation.

AlexNet: CNN architecture was invented in 2012, with around 60 million parameters, five convolutional (and pooling), and three fully connected, eight in total layers [**25**]. The authors (K. Alex, S. Ilya, and Hinton G.) first introduced rectified linear unit (ReLU) as an activation function. [**18**] used AlexNet as a pre-trained network for their system of identification of Urdu text ligatures from video frame captions.

VGG16: 138 parameters, 13 convolutional and 3 fully connected layers, five max-pooling layers, and one SoftMax layer, from visual geometry group in 2014 for the large-scale image recognition, the VGG16 was a deeper network compared to its predecessors [**25**]. There are other versions of VGG, such as VGG11, VGG16, and VGG19 with different layers. In [**18**], the authors used VGG16 and VGG19 (sixteen convolutional layers, five pooling, and three fully-connected layers) as pre-trained networks for their system of identification of Urdu text

ligatures from video frame captions.

ResNet-50: Residual network, a 50-layered deep CNN, is capable of handling millions of images, is based on many computer vision algorithms. The recent versions of ResNet have more than 100 layers. [**18**], used this architecture for translational learning in Urdu text recognition.

RNN Architectures

RNNs are especially suitable for NLP, and time series applications as they model sequences. The specific architecture of RNN controls the information among various cells (neurons), therefore a selection of specific architecture is very important for the success of a task.

ERNN: Elman recurrent neural network, is the basic and simplest architecture of RNNs. It can be used to learn grammar from a dataset without annotation, to predict potential words. The input and output layers are feedforward layers, whereas the hidden layers are recurrent.

GRU: Gated recurrent unit, a gated architecture combines the input and forget gates into an update gate and was introduced by K. Cho et al in 2014. It is a variation of LSTM architecture with forget gate and no output gate. Its usage and performance are like LSTM with a lesser number of parameters.

Machine Learning Datasets

There are various datasets available for different machine learning problems free of cost, paid, and free upon request. Some of the popular datasets have been given below along with the basic information required to get fundamental guidelines. These datasets are available online for non-commercial use.

Benchmark Datasets

MNIST dataset: A modified National Institute of Standards and Technology database (created in 1998) used for the training and testing of several machine learning algorithms. It contains images of single digits from 0-9. It is good for learning as now it is considered a resolved dataset for advanced machine learning algorithms.

CIFAR10: Canadian Institute for Advanced Research, an image classification dataset for object recognition, comprising of 10 classes such as images of airplanes, automobiles, birds, cats, dogs, deer, frogs, horses, ships, and trucks.

CIFAR100: A modified version of CIFAR10, contains 100 classes of objects including subcategories within a class such as types of trees.

ImageNet: A large-scale object recognition dataset with 1000 classes, 1,281,167 training images 50k validation images, and 100k test images of variable sizes. The dataset is designed following WordNet (a lexical database of English maintained by Princeton University).

Omniglot: A large dataset of handwritten 1623 categories and 20 versions of each category (character). Characters are 50 (real and imaginary) alphabets from different scripts along with different strokes and position coordinates. Languages are Malay, Hebrew, Korean, Latin, Greek, and many others. **Table** 4 summarizes the important characteristics of the discussed datasets and Fig 9**,** Fig 10**,** Fig 11**,** Fig 12**,** and Fig 13 show samples of these datasets. These are benchmark datasets and are useful to (a) train the algorithm for transfer learning, (b) study the dataset design, and (c) do practice if it is intended to work with a customized dataset.

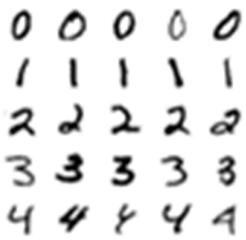

Fig 9. MNIST dataset sample

Fig 10. CIFAR10 dataset sample

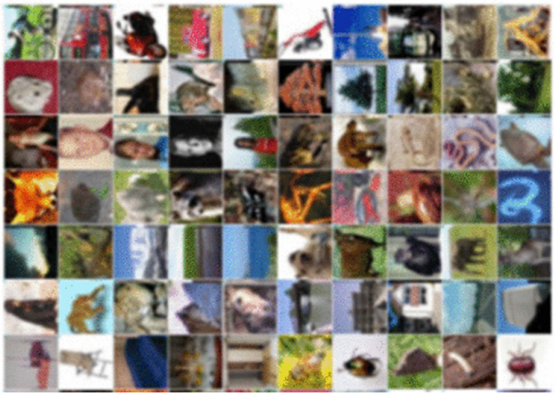

Fig 11. CIFAR100 dataset sample

Fig 12. ImageNet dataset sample

Fig 13. Omniglot dataset sample

Table 4. Image classification dataset - summary

Title	Classes	Type	Size
MNIST	10 Labeled	28 x 28 Grayscale image	50k: Training, 10k: Test
CIFAR10	10 Labeled	32 x 32 RGB image	50k: Training, 10k: Test, 6k/class
CIFAR100	100 Labeled	32 x 32 RGB image	50k: Training (500/class)10k: Test (100/class) 20 (5/class) superclasses
ImageNet	1000 Labeled	Variable	1.3mil: Training, 50k: Validation (50/class), 100k: Test (100/class)
Omniglot	1623 categories (50 alphabets)	-	20 images/category

Perso-Arabic Datasets

There are a limited number of benchmark datasets for Perso-Arabic scripts. Some of them have been presented here:

UPTI: Urdu Printed Text Image dataset, used by [**19**], [**20**], [**24**], [**26**]–[**29**], comprised of 10000 Urdu text synthetic lines in Nasta'liq font covering a variety of topics.

CLE: A linguistic resource by College of Language Engineering, University of Engineering and Technology Lahore, Pakistan. This comprehensive database is available online for purchase. It is comprised of image, speech, and text corpora in different font sizes.

CENPARMI: A dataset developed by Centre for Pattern Recognition and Machine Intelligence, Concordia University, Canada comprising of digits, numeral strings, special symbols, and isolated characters, and Urdu words and Urdu dates in different styles [**30**]. Although the first dataset of its kind till that time, this handwritten offline dataset has a limited number of data samples, 44 individual characters, and 57 Urdu words, focusing on one field mostly, the financial terms.

Although the dataset has been tested to achieve a success rate of 98.61%, the limited number of words from one field is a compromise over the generalization. Further, only one method has been used as a data classifier (SVM), instead of using more methods to validate the results and performance, whereas no standard error analysis method has been described.

To the best of the author's knowledge, most of the discussed datasets are not available to researchers at all or are available online at a considerable cost.

Conclusion

The paper has described the detailed review of the related work in the field of Perso-Arabic text recognition mainly in three categories: Shahmukhi computational-linguistic-based research available work, Shahmukhi (and other Perso-Arabic scripts) recognition systems, and benchmark database development work related to Perso-Arabic script recognition. The research findings include review categories, work type classification, distribution of techniques, machine learning techniques, the success rate for machine learning techniques, and research article category. The paper also presented summaries of comparisons of common text classification approaches and machine learning-based text classification techniques. Recent trends showing the popularity of Convolutional neural networks and Recurrent neural networks in methods and algorithms; and dominant Python-based tools for machine learning frameworks have also been presented. A review of neural network architectures for optical character recognition, natural language processing, and image processing has been described. Deep learning architectures such as CNN and RNN and their variants have been described with their important characteristics useful for the selection of a machine learning algorithm. Available benchmark datasets and low-resource language datasets have been described.

References

[1] M. Humayoun and A. Ranta, "Developing Punjabi morphology, corpus and lexicon," *PACLIC 24 - Proc. 24th Pacific Asia Conf. Lang. Inf. Comput.*, pp. 163–172, 2010, Accessed: Oct. 14, 2021. [Online]. Available: https://gup.ub.gu.se/publication/131250

[2] M. T. Ahmad *et al.*, "Named Entity Recognition and Classification for Punjabi Shahmukhi," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 4, pp. 1–13, Jul. 2020, doi: 10.1145/3383306.

[3] M. G. A. Malik, "Punjabi machine transliteration," in *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2006, vol. 1, pp. 1137–1144. doi: 10.3115/1220175.1220318.

[4] T. S. Saini, G. S. Lehal, and V. S. Kalra, "Shahmukhi to Gurmukhi transliteration system," in *Coling 2008 - 22nd International Conference on Computational Linguistics, Proceedings of the Conference*, 2008, vol. 1, pp. 177–180. Accessed: Oct. 14, 2021. [Online]. Available: https://www.researchgate.net/publication/221102825_Shahmukhi_to_Gurmukhi_Transliteration_System

[5] G. S. Lehal, T. S. Saini, and S. K. Chowdhary, "An Omni-font Gurmukhi to Shahmukhi Transliteration System," *Coling*, vol. 3, no. December 2012, pp. 313–320, 2012.

[6] A. Mateen, M. Kamran Malik, Z. Nawaz, H. M. Danish, M. Hassan Siddiqui, and Q. Abbas, "A Hybrid Stemmer of Punjabi Shahmukhi Script," 2017. [Online]. Available: http://cloud.politala.ac.id/politala/1. Jurusan/Teknik Informatika/19. e-journal/Jurnal Internasional TI/IJCSNS/2017 Vol. 17 No. 08/20170813 Reviewing and modeling the optimal output velocity of slot linear diffusers to reduce air contamination in the sur

[7] M. IRFAN, J. A. MAHAR, H. SHAIKH, and F. A. SURAHIO, "Stemming Software Application of Shahmukhi Script Using Porter's Algorithm," *Sindh Univ. Res. J. -Science Ser.*, vol. 49, no. 004, pp. 813--816, Dec. 2017, doi: 10.26692/surj/2017.12.63.

[8] J. Kaur and J. R. Saini, "Punjabi stop words: A Gurmukhi, Shahmukhi and roman scripted chronicle," in *ACM International Conference Proceeding Series*, 2016, vol. 21-22-Marc,

pp. 32–37. doi: 10.1145/2909067.2909073.

[9] G. S. Lehal and T. S. Saini, “A transliteration based word segmentation system for Shahmukhi script,” in *Communications in Computer and Information Science*, vol. 139 CCIS, 2011, pp. 136–143. doi: 10.1007/978-3-642-19403-0_22.

[10] M. A. Hashmi, M. A. I. Mahmood, and M. A. I. Mahmood, “Analysis of Lecxico-Semantic Relations of Punjabi Shahmukhi Nouns: A Corpus Based Study,” *Int. J. English Linguist.*, vol. 9, no. 3, p. 357, May 2019, doi: 10.5539/ijel.v9n3p357.

[11] N. Bansal and A. Kumar, “Machine translation survey for Punjabi and Urdu languages,” in *Proceedings - 2017 3rd International Conference on Advances in Computing, Communication and Automation (Fall), ICACCA 2017*, Sep. 2018, vol. 2018-Janua, pp. 1–11. doi: 10.1109/ICACCAF.2017.8344667.

[12] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, “Segmentation free nastalique Urdu OCR,” *World Acad. Sci. Eng. Technol.*, vol. 46, pp. 456–461, 2010, doi: 10.5281/zenodo.1080342.

[13] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, “Segmentation-based, omnifont printed Arabic character recognition without font identification,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2020, doi: 10.1016/j.jksuci.2020.10.001.

[14] S. Bin Ahmed, I. A. Hameed, S. Naz, M. I. Razzak, and R. Yusof, “Evaluation of handwritten Urdu text by integration of MNIST dataset learning experience,” *IEEE Access*, vol. 7, pp. 153566–153578, 2019, doi: 10.1109/ACCESS.2019.2946313.

[15] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, “Comparing automated text classification methods,” *Int. J. Res. Mark.*, vol. 36, no. 1, pp. 20–38, Mar. 2019, doi: 10.1016/j.ijresmar.2018.09.009.

[16] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi, and M. I. Razzak, “Urdu Nasta’liq text recognition system based on multi-dimensional recurrent neural network and statistical features,” *Neural Comput. Appl.*, vol. 28, no. 2, pp. 219–231, 2017, doi: 10.1007/s00521-015-2051-4.

[17] A. Mirza, I. Siddiqi, U. Hayat, M. Atif, and S. G. Mustufa, “Recognition of Cursive Caption Text Using Deep Learning - A Comparative Study on Recognition Units,” in *Lecture Notes*

in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12068 LNCS, pp. 156–167. doi: 10.1007/978-3-030-59830-3_14.

[18] U. Hayat, M. Aatif, O. Zeeshan, and I. Siddiqi, "Ligature recognition in Urdu caption text using deep convolutional neural networks," 2019. doi: 10.1109/ICET.2018.8603586.

[19] S. Naz *et al.*, "Urdu Nastaliq recognition using convolutional–recursive deep learning," *Neurocomputing*, vol. 243, pp. 80–87, 2017, doi: 10.1016/j.neucom.2017.02.081.

[20] A. Naseer and K. Zafar, "Meta features-based scale invariant OCR decision making using LSTM-RNN," *Comput. Math. Organ. Theory*, vol. 25, no. 2, pp. 165–183, 2019, doi: 10.1007/s10588-018-9265-9.

[21] M. J. Rafeeq, Z. ur Rehman, A. Khan, I. A. Khan, and W. Jadoon, "Ligature categorization based Nastaliq Urdu recognition using deep neural networks," *Comput. Math. Organ. Theory*, vol. 25, no. 2, pp. 184–195, 2019, doi: 10.1007/s10588-018-9271-y.

[22] A. A. Chandio, A. A. Chandio, M. Asikuzzaman, and M. R. Pickering, "Cursive Character Recognition in Natural Scene Images Using a Multilevel Convolutional Neural Network Fusion," *IEEE Access*, vol. 8, pp. 109054–109070, 2020, doi: 10.1109/ACCESS.2020.3001605.

[23] N. Sabbour and F. Shafait, "A segmentation-free approach to Arabic and Urdu OCR," in *Document Recognition and Retrieval XX*, 2013, vol. 8658, p. 86580N. doi: 10.1117/12.2003731.

[24] S. Naz *et al.*, "Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks," *Neurocomputing*, vol. 177, pp. 228–241, 2016, doi: 10.1016/j.neucom.2015.11.030.

[25] R. Karim, "10 CNN Architectures," *Towards Data Science*, 2019. https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d%0Ahttps://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#6872

[26] I. Ud Din, I. Siddiqi, S. Khalid, and T. Azam, "Segmentation-free optical character recognition for printed Urdu text," *Eurasip J. Image Video Process.*, vol. 2017, no. 1, 2017,

doi: 10.1186/s13640-017-0208-z.

[27] N. H. Khan, A. Adnan, A. Waheed, M. Zareei, A. Aldosary, and E. M. Mohamed, “Urdu ligature recognition system: An evolutionary approach,” *Comput. Mater. Contin.*, vol. 66, no. 2, pp. 1347–1367, 2020, doi: 10.32604/cmc.2020.013715.

[28] I. Ahmad, X. Wang, R. Li, M. Ahmed, and R. Ullah, “Line and Ligature Segmentation of Urdu Nastaleeq Text,” *IEEE Access*, vol. 5, pp. 10924–10940, 2017, doi: 10.1109/ACCESS.2017.2703155.

[29] S. S. R. Rizvi, M. A. Khan, S. Abbas, M. Asadullah, N. Anwer, and A. Fatima, “Deep Extreme Learning Machine-Based Optical Character Recognition System for Nastalique Urdu-Like Script Languages,” *Comput. J.*, Jun. 2020, doi: 10.1093/comjnl/bxaa042.

[30] M. W. Sagheer, C. L. He, N. Nobile, and C. Y. Suen, “A new large urdu database for off-line handwriting recognition,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5716 LNCS, Springer, Berlin, Heidelberg, 2009, pp. 538–546. doi: 10.1007/978-3-642-04146-4_58.