

Visual News Ticker Surveillance Approach from Arabic Broadcast Streams

Moeen Tayyab¹, Ayyaz Hussain^{2,*}, Usama Mir³, M. Aqeel Iqbal⁴ and Muhammad Haneef⁵

¹Department of Computer Science and Software Engineering, International Islamic University, Islamabad, 44000, Pakistan

²Department of Computer Science, Quaid-i-Azam University, Islamabad, 44000, Pakistan

³Department of Computer Science, Senior Member IEEE, University of Windsor, N9B 3P4, Canada

⁴Department of Software Engineering, Foundation University Islamabad, Islamabad, 44000, Pakistan

⁵Department of Electrical Engineering, Foundation University Islamabad, Islamabad, 44000, Pakistan

*Corresponding Author: Ayyaz Hussain. Email: ayyaz.hussain@qau.edu.pk

Received: 23 July 2022; Accepted: 22 September 2022

Abstract: The news ticker is a common feature of many different news networks that display headlines and other information. News ticker recognition applications are highly valuable in e-business and news surveillance for media regulatory authorities. In this paper, we focus on the automatic Arabic Ticker Recognition system for the Al-Ekhbariya news channel. The primary emphasis of this research is on ticker recognition methods and storage schemes. To that end, the research is aimed at character-wise explicit segmentation using a semantic segmentation technique and words identification method. The proposed learning architecture considers the grouping of homogeneous-shaped classes. This incorporates linguistic taxonomy in a unified manner to address the imbalance in data distribution which leads to individual biases. Furthermore, experiments with a novel Arabic News Ticker (AI-ENT) dataset that provides accurate character-level and character components-level labeling to evaluate the effectiveness of the suggested approach. The proposed method attains 96.5%, outperforming the current state-of-the-art technique by 8.5%. The study reveals that our strategy improves the performance of low-representation correlated character classes.

Keywords: Arabic text recognition; optical character recognition; deep convolutional network; SegNet; LeNet

1 Introduction

Besides the applicability, machine visual reading of texts is an active research domain. The recognition of daily broadcast news tickers is one such application. The focus of recent advancements has been on English recognition tickers. New tickers in cursive text, on the other hand, continue to necessitate considerable thought. Because of the limited availability of labeled datasets and processes described specifically for cursive scripts, recognizing news tickers in the Arabic language may be a difficult task. Recently, several strategies [1] for recognizing cursive text on an ad-hoc basis have been



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

proposed. Most of these techniques are intended to identify scanned printed text images in a controlled manner.

Real-world data from broadcasts, on the other hand, is noisy and difficult to recognize using machine learning techniques. Furthermore, various factors such as signal-to-noise ratio, transmission, image resolution, and other compression errors can produce image/video distortions. The lack of these artifacts in presently available datasets creates a challenge to develop an effective learning model. This study addresses these challenges with deep learning-based techniques on the customized dataset and will be further discussed in the next sections.

This study presents an innovative Al-Ekhbariya Arabic text news ticker recognition method based on deep learning. A combination of semantic segmentation with text formation and storage methods is proposed. Due to a large number of ligature classes, the current study focuses on explicit methods and precludes holistic approaches. It incorporates character families for more precise and robust segmentation to restrain inter-class correlations and imbalance in data distributions. To generate identified characters into linguistic form, the technique employs a text formatting method. The proposed idea is based on the innovative Al-Ekhbariya Arabic News Ticker (Al-ENT) dataset, which provides character-level and character component-level labeling. Testing assessments on Al-ENT indicate improved performance from the previous explicit Arabic characters recognition approach. Fig. 1 illustrates a visualization of the proposed Arabic news ticker recognition process.

The following are the highlights of the presented investigation.

- Generation of the novel Al-Ekhbariya News Ticker dataset with pixel-level labeling for explicit characters recognition.
- Investigation of proposed semantic segmentation approach by contemplating character families to prevent inter-class correlations and data diversity.
- Demonstration of text formation technique and storage scheme from the recognized characters. Performance evaluation of proposed method on a custom-built dataset.
- Significant improvements in recognition performance, particularly for correlated character classes with low representations.

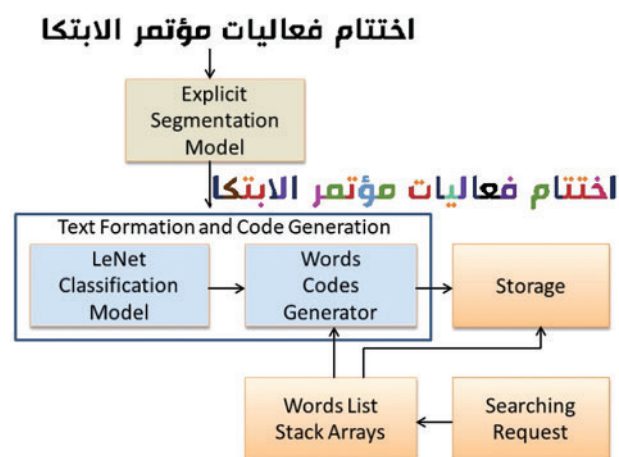


Figure 1: Proposed model for Arabic news ticker recognition: Explicit segmentation architecture for character segmentation. Input text image with the corresponding segmented image. Text formation, word codes generation model for textual recognition and storage. Best viewed in color

The remaining sections of the paper are organized as follows: The background and related work in the area of cursive text recognition are described in Section 2. The suggested mechanism for the proposed Arabic news ticker model is discussed in Section 3. Section 4 presents the experiments and results, as well as a comparative analysis with the existing techniques for character recognition, while Section 5 concludes the research.

2 Background and Related Work

The use of news tickers to display headlines or other information has become a regular feature of several news networks. The ticker has been used in a variety of ways on multiple channels. News ticker applications are quite useful in e-business, as previously stated. The performance of OCR engines is important for the high efficiency of such applications. The product level and real-time applications necessitate efficiency, and excellent recognition algorithms based on comprehensive real-world datasets. Another aspect affecting OCR's effectiveness is the nature of the language. Due to the obvious occurrence of connecting and overlapping letters in cursive scripts, character segmentation is the most crucial, difficult, and computational step of any OCR system. Most of the emphasis has always been on various ways of recognizing cursive text using simple scanned images. Advanced machine learning and pattern recognition techniques are proposed as evolutionary advancements to OCR problems. For the recognition of Arabic text from videos, [2] used a long short-term memory (LSTM). References [3,4] introduced the Multidimensional Long Short-Term Memory (MDLSTM) output layer with Connectionist Temporal Classification (CTC). Reference [5] demonstrated a rudimentary work using Bidirectional Long Short-Term Memory (BLSTM) to recognize Urdu News Ticker. Reference [6] suggested a deep learning model based on a Convolutional Neural Network (CNN) and LSTM combination. Experiments are conducted on 12,000 text lines culled from 4,000 video frames from Pakistani news stations. Reference [7] proposed a comparable UrduNet model, which is a hybrid approach of CNN and LSTM. On a self-generated dataset of almost 13,000 frames, a complete set of experiments are carried out. The following sections describe some related work to meet the challenge of Arabic cursive text recognition methods.

Arabic text recognition methods can be divided into two types i.e., 1) analytical and 2) holistic approaches [8]. An analytical technique relies on the explicit or implicit segmentation of ligatures or words into characters. Instead of letters, holistic techniques use ligatures as recognition units. Many investigations commonly use structural and statistical characteristics. Structural features rely on the structure of the patterns under study characters or ligatures i.e., information of curves, endpoints, loops, and distribution of vertical and horizontal lines, etc. Statistical features are based on statistics derived from the placement of pixels in an image formation. Statistical features normally involve projections, profiling, moments, pixel densities, etc. The following are some more characteristics of analytical and holistic approaches.

2.1 Holistic Approach

As previously stated, the holistic approach works on the full ligature, hence the model is trained and recognized on ligatures. Reference [9] used a holistic approach to analyzing and modifying the Tesseract engine for the Nastaleeq style typeface. They increased the search space technique and font sizes to a range between 14 and 16. A self-generated dataset of almost 1.4 million ligatures was used. The images used were segmented and cleaned. They have created a Tesseract-based Urdu OCR that supports font sizes ranging from 14 to 44. They developed four separate recognizers to accommodate varied text sizes for this objective. Reference [10] suggested CNN recognize Nastaleeq font style

ligatures. The technique analyzed 18,000 ligatures over 98 classes and achieved a recognition rate of 95%. Reference [11] introduced a model incorporating raw pixel value as a feature instead of a human-crafted feature. On the Urdu Printed Text Image (UPTI) dataset with 43 classes and aligned input, they introduced the Gated Bi-directional Long Short-Term Memory (GBLSTM) based model, claiming 96.71% accuracy. The model is trained on un-degraded and tested on unseen data. Reference [12] proposed model comprises CNN and Bi-directional Long Short-Term Memory (BDLSTM). Using a custom build dataset, results in 97.5% on clean image data. The networks can perform word-level recognition.

By these holistic approaches, each ligature is recognized as a distinct model class. This causes a large number of ligature classes, and samples per class and increases complexity in the training models. For applications like news ticker recognition, such techniques are ineffective.

2.2 Analytical Approach

The analytical approach relies on the segmentation of text into characters. Implicit and explicit segmentation of analytical approaches is discussed below.

2.2.1 Implicit Segmentation

Implicit segmentation is a basic approach that is generally language agnostic. Their accuracy is determined by their classifying abilities. Reference [13] presents the architecture of the Quran-seq2seq and Quran-Full-CNN models. These are the combination of CNN and LSTM networks. Customized Quran Text Image Dataset (QTID) is used for experiments. 98.90% is found as the best character level accuracy. Reference [14] proposed CNN-RNN Attention Model. The best result on the ALIF dataset is 98.73% at a character level. [15,16] employ BLSTM and CTC to demonstrate an implicit technique using the UPTI Nastaleeq font style scripts dataset. References [17,18] utilized MDLSTM with the UPTI dataset, and it outperformed state-of-the-art approaches by a significant margin. Reference [18] used statistical features and MDLSTM to attain improved accuracy rates with the same dataset. Along with the low complexity, high efficiency, and high speed with large information, [19] proposed zoning characteristics with the combination of Two Dimensional Long Short-Term Memory networks (2DLSTM). References [17–19] used a sliding window procedure to extract features, which are then supplied to the recognizer.

Several researchers have effectively used an implicit method. Classification methods that are based on implicit segmentation necessitate a huge amount of training data. The total number of segments for a word is chosen as a decisive criterion in this method. Less number of segments reduces computations but it leads to an under-segmentation problem on widely written words. Increasing the number of segments leads to more computation and more junk sequences, which the model must deal with [1]. This causes under and over-segmentation.

2.2.2 Explicit Segmentation

In this approach, the words are split into smaller individual components, like ligatures, letters, strokes, or hypotheses which have been analyzed to measure the segmentation points' validity. By integrating CNN and MDLSTM, [20] developed a hybrid technique based on explicit feature extraction. The CNN extracts low-level translational invariant features, which are then fed to the MDLSTM architecture to learn. Experiments were conducted using a 44-class with UPTI Nastaleeq cursive scripted database. They used CNN to extract features from the MNIST dataset, which does not adequately represent Nastaleeq font text features. This suppresses the textual features. In the

context of the scripted cursive language, [21] claimed that CNN is appropriate to analyze styles of visual images. They recommend that the strategy of explicit segmentation and feature extraction is appropriate, although extracting features is difficult to achieve the desired precision. They used Arabic images from the English-Arabic Scene Text (EAST) dataset for this study.

Mirza et al. [7] present an implicit technique for new tickers from video frames by combining CNN and LSTM. This can be seen that even on its own training dataset, the model has difficulty in training with low accuracy. The explicit-based technique provides the advantage of reducing the number of training classes and minimizing the problem of under-segmentation. It outperforms less complicated implicit segmentation methods by a small margin [22]. To extract characters, it is necessary to have a complex understanding of letter patterns, as well as their beginning and ending positions inside ligatures or words. As different alphabets have distinct features, methods are computationally complicated and language-dependent. Furthermore, the performance of pre-processing techniques, particularly binarization and normalization processes, has a significant impact on accuracy.

For the proposed Arabic news ticker cursive text recognition engine, we present a new approach with a combination of deep learning and text formation methods. Deep convolutional neural networks are employed to learn alphabets from text line images explicitly. For the obtained segmented characters, the text formation method is used, which leads to a linguistically recognized form in the next phase. This also generates word code for storage. Model simplification, not only in architectural terms but also in parametric reduction and computation gains, are the advantages of the innovative contribution of the proposed methodology as compared to excessively complicated traditional techniques. The findings of the newly presented method are superior to those of other methods.

3 Proposed Arabic News Ticker Model

This section presents the flow of the proposed Arabic New Ticker Recognition system, which is an expert application for e-business. The presented paper mainly focuses on news ticker recognition models and storage schemes. The study targets the Al-Ekhbariya TV news channel, which is widely viewed in Saudi Arabia. To this rationale, detected Arabic tickers are segmented by the proposed robust deep learning-based architecture in a unified manner for high accuracy from the existing techniques. The ticker text image is semantically segmented into the components of the character. These components are concatenated to form characters followed by a LeNet-based classification method. The text format generation method arranges the recognized letters string into sentence formation of words and generates word codes for effective storage in the next stage. The study also addresses Arabic text segmentation challenges regarding training data distributions. Fig. 2 shows the proposed ticker recognition system and the stages are further discussed in the following sections.

3.1 Ticker Segmentation

Qaroush et al. [22] recently presents an explicit method of Arabic character segmentation. The segmentation is performed in three steps namely segmentation by vertical projection, connected components, and baseline removal. In the first step, dots and diacritics are removed from a binary text image using the connected component method. Practically the video data is difficult. It contains noisy artifacts as described earlier, causing deformation, and merging of dots, diacritic marks, and letters. The suggested method malfunctioned towards accuracy. The connected components method is also applied in the second stage of segmentation, aiming to segment the unconnected and vertically overlapped characters. In the final segmentation stage, the ligatures are segmented by identified vertical projection. Accurate identification of the baseline with no minor error is obvious. Most of

the assumptions for segmentation are based on the baseline. It could underperform on less squarish fonts as fonts must fulfill the assumptions as per the methodology. Robust binarization, especially in small fonts is very essential as assumption sometimes goes wrong even on squarish fonts as well. Unfair class data distribution for feature extraction using LeNet-5 is also a factor towards accuracy. Less number of classes with de-shaped samples up to a certain limit (per class) is a way towards model simplicity. Reference [23] also used a heuristic method for segmentation with previously mentioned limitations. Such contemporary methods may work on clean data that is unfeasible for applications like Arabic news ticker recognition.

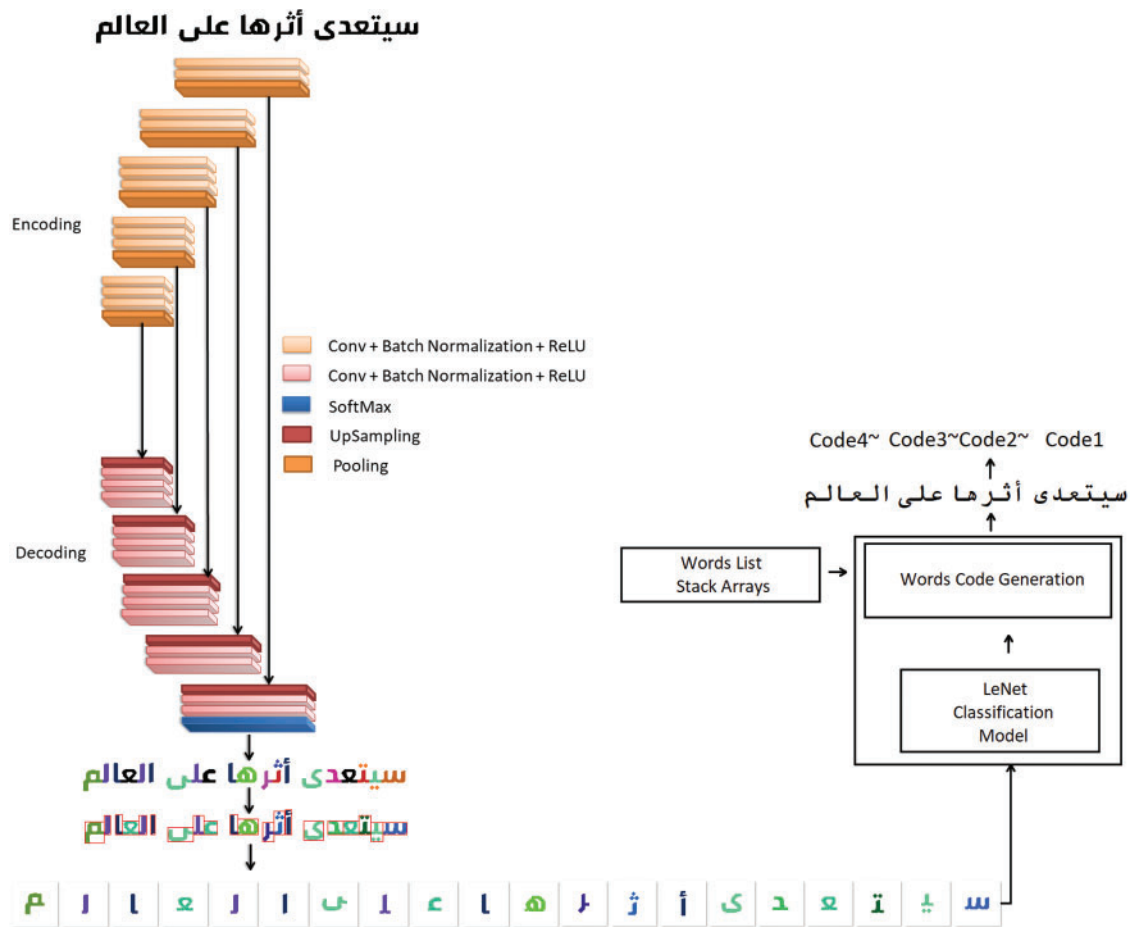


Figure 2: Proposed model for Arabic news ticker recognition: SegNet model presenting down-sample and up-sampling architecture for character segmentation. Input text image with the corresponding recognized image. LeNet classification and words code generation model for textual recognition and storage. Best viewed in color

The present research utilizes a robust SegNet deep learning model to propose an explicit technique of segmentation. SegNet's design, like Fully Convolutional Networks (FCN), is divided into two parts: an operating encoder and a decoder. An encoder first downsamples the text image first, using the same CNN architecture as FCN or Residual Network (ResNet). Decoding is a CNN method performed backward, with up-sampling layers instead of down-sampling layers. At the deepest encoding level, SegNet discards fully connected layers. In comparison to other recently proposed designs, this also

minimizes the number of parameters [24,25]. On a probabilistic basis, the decoding result is given to a softmax layer, which generates a predicted label map of the same dimension as the source images. We used eighteen-layer encoder architecture under this investigation. The encoder layer corresponds to the decoder layer in the same way as the VGG16 network has thirteen convolutional layers. Hence the model contains a decoder with eighteen layers as shown in Fig. 2.

The network learns to classify each pixel based on the orientation of the alphabet component, its spatial location, and its surroundings. Instead of being a vector, the recognized output image with segmented components has the same dimensions as the input image. Convolution using filters is performed by the encoder layer. The outputs are batch normalized before being rectified element-by-element. Eq. (1) is used to accomplish the ReLU process. Eq. (2) predicts label maps probabilistically using the Softmax classifier. Where x is the batch normalization value, φ_k represents the values at the output layer and c is the number of outputs.

$$\Phi(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (1)$$

$$\text{Soft max}(\Phi_k) = \frac{\exp(\Phi_k)}{\sum_j^c \exp(\Phi_j)} \quad (2)$$

The softmax classifier produces a k -channel image of probabilities, with k denoting the number of classes. At each pixel, the predicted segmentation corresponds to the class with the highest likelihood. All network settings are set at random. The network's pooling window dimensions and stride values are also factors.

Labeled news ticker data at two levels, i.e., in text image character's component-wise labeling as well as in text form, is required to develop the proposed methodology of character learning. For the research community, no such dataset exists, particularly for Al-Ekhbariya's news tickers Arabic text recognition. An innovative Al-Ekhbariya News Ticker (Al-ENT) dataset is developed in order to carry out the ongoing research hypothesis. It accurately labels characters and character components at the symbolic level. The following Dataset Generation and Labeling Section 3.2 discusses data collection and processing.

3.2 Dataset Generation and Labeling

The development of an Al-ENT dataset is a challenging task in the investigation. The video data of telecast streams of the famous Saudi news network Al-Ekhbariya TV are captured. The videos have a resolution of 1080p. One-third of the bottom portion of video frames is designated as ticker zones in order to focus on data collection tasks. This bottom portion, based on presumption, is sufficient because it mostly contains news ticker lines. Bright text on a dark background is also assumed. As indicated in Eq. (3), the selected area is clamped with threshold epsilon ϵ to form matrix A , where ϵ is a 75-pixel value. Gx is a vertical gradient that incorporates vertical derivative estimations at each point and is clamped with ϵ as shown in Eqs. (4) and (5). The normalized moving average representation B locates the text region and approximates the font size. Tickers of appropriate font size are selected. Font sizes ranging from 25pt to 42pt are used in this investigation. With the help of distance and vertical lines at both ends of the ticker image, the areas other than ticker text are removed. Fig. 3 shows some tickers that are extracted.

$$A = \begin{cases} 0 & I \leq \epsilon \\ I & I > \epsilon \end{cases} \quad (3)$$

$$G_x = h_x * A \quad (4)$$

$$B = \begin{cases} 0 & G_x \leq \epsilon \\ G_x & G_x > \epsilon \end{cases} \quad (5)$$



Figure 3: Extracted text line images of Urdu news ticker

Selected text images are labeled both characters components-wise and in textual form. 2100 news tickers are captured. Fig. 4 shows a few lines from the AI-ENT dataset together with their character components-wise label interpretations. Tickers can also contain numeric characters, such as 0 to 9, which are included in the dataset. Characters such as semicolons, slashes, and brackets are also included. Datasets are labeled in order to build our model more effectively and robustly by grouping character classes on the basis of character families. The diacritic marks (3 in number) are recognized as independent classes. Low symbolic representations are comforted by character grouping methods, which lowers data inequity and diversity. This decreases pixel recognition biases towards more frequent symbols. Segmented diacritic marks are concatenated with the corresponding prime component by a restrained distance formula within the width of the prime component of characters in the post-segmentation procedure. The number of classes (from 42 to 35) and model parameters (k of softmax classifier) are also reduced significantly using this strategy.



Figure 4: Visualization of character components-wise labeled Arabic news tickers. Best viewed in color

3.3 Text Formatting

The word identification process of the recognized string of letters is essential for complete text recognition. The textual formation process is performed in two steps as follows. Words codes are the outputs of this process to be stored for the searching operation.

3.3.1 Classification Model

By inserting space between words, recognized character representations are brought into a sentence of words. For this objective, the research employs a LeNet-5 model with minor modifications in the final layer to classify more than 10 classes. The network only contributes to the recognition of consecutive characters and the word's terminal character to add space after each word. Depending on where it appears in the word, Arabic letters contain multiple shapes. Normally the shape of the last character of the word is almost similar to the independent letter shape and can be differentiated from the other shapes. This method does not assure that identified symbols will be appropriately arranged

into word formation sentences. For further corrections, a searching-based syntax rectification technique is developed, which will be addressed in the next Word code Generator section.

Minor segmentation misclassifications are removed by a statistical modulus operation using a window size of 3×3 pixels. Segmented characters are then provided for classification via a concatenation process. For this purpose, prime components and diacritic marks from the recognized image are relabeled in a sequential sequence. The segmented image's recognized letters are ordered from right to left. After the concatenation of prime components with diacritic markings, these characters are sorted based on the left borders and placed in 32×32 pixels dimension buffers, as illustrated in Fig. 2. The character that is greater than 32×32 pixels in height, width, or both are scaled down to a maximum of 32 pixels in height, width, or both while maintaining the aspect ratios. The classification model utilizes these character candidates as input. This process only determines the ending and successive characters. The letters, which have similar shapes at the end of the words and ligatures, are not addressed in this model.

3.3.2 Words Code Generator

The suggested syntax correction approach entails a search for entire words inside the identified string. Words lists are organized in ascending order. A track of these key lists is maintained in a separate array of pointers. This pointer array is characterized as a sparse array because it contains information about word lists. Each list has a fixed number of words, n , ranging from 2 to m , where the maximum word length in data is m . These lists are arranged to perform an efficient search of words [26]. Using a search procedure, we attempted to find the whole word by starting with the longest word. Unless a word is not found in the list, continue searching in a list of words of length $(n-1)$ until the word list reaches two characters. Failing at each step is finally an isolated character.

For sentence organization, the proposed classification and syntax correction reinforce each other. In searching methodology, without the classification method, a sequence of incorrect textual recognition followed by a single miss-classified character is possible.

The unique code of the word is the combination of the list number and the index number within the list. This code is used to save data for later retrieval. The unsigned short data type of 16-bit can be used to store codes. This technique is storage space efficient because most words have more than two letters. The approach decreases the use of storage resources, while also allowing for effective searches using these key codes. Our study recommends this efficient scheme of storage and searching for future implementation. Fig. 2 describes the layout of the proposed Arabic New ticker architecture.

4 Results and Experiments

A total of 2100 news tickers are employed (1170 for training, 130 for validation, and 800 for testing for segmentation). These news tickers are single channels, contained in a 205×1365 pixel buffer with character-level labeled images in correspondence with 205×1365 pixel size (ground truths). Experiments are carried out using a computer system with an Intel® Core™ i3-8100 processor with 3.60 GHz x 4, 8GB DDR-5 RAM and SSD media. The processing is carried out on a GeForce RTX™ 2070/PCIe/SSE2 NVIDIA Graphics Processing Unit (GPU).

From the analysis perspectives of the proposed methodology [22], the technique is implemented in the context of explicit segmentation. For this, news ticker data without numeral characters are selected from the training and testing parts of the AI-ENT dataset. LeNet-5 convolutional network is implemented on MATLAB for feature extraction and character/ligature recognition with the

mentioned model parameters. The LeNet-5 model used with minor modifications in the final layer consists of 78 numbers of classes, containing a combination of two characters or ligatures, and isolated characters from our particular dataset. Several segmentation limitations were observed as discussed earlier. Unfair class data distribution is also noticed while data preparation from AI-ENT. The network is trained for 3220 iterations using learning rates of 0.01 and batch size of 32. We selected 11311 images (32×32) from the training part of tickers. The training details are tabulated in Table 1. The training graph of LeNet is shown in Fig. 5. Final validation accuracy is 90.30%.

Table 1: LeNet training using the AI-ENT dataset

LeNet parameters	Values
Number of training characters	11311
Number of validation characters	1000
Batch size	32
Learning rate	0.01
Training time	1 m 13 sec

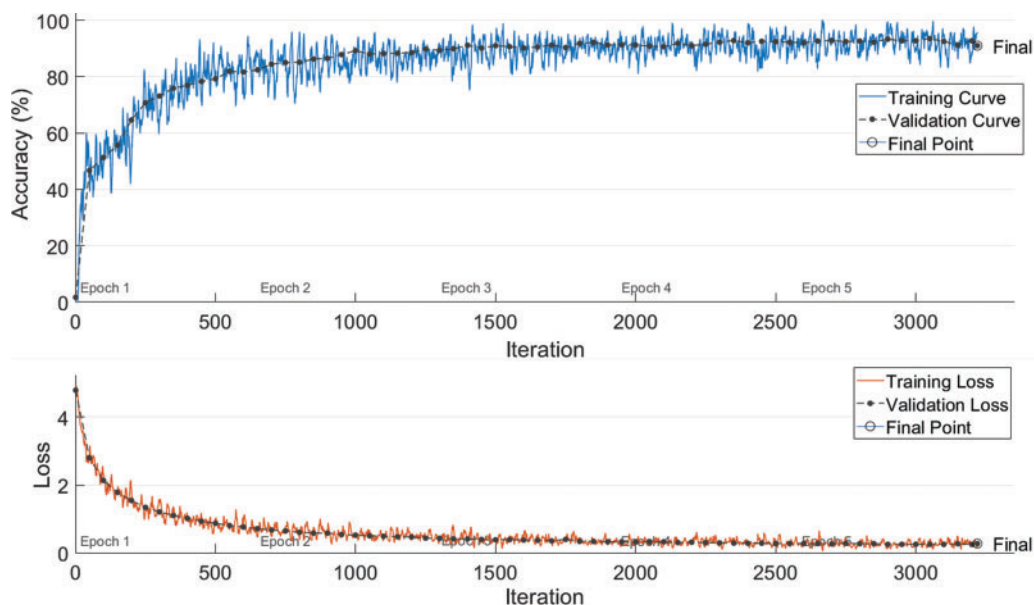


Figure 5: Graphical representation of LeNet validation accuracy, training loss, and validation loss curves. Best viewed in color

4.1 Proposed Segmentation Experiments

The investigations are carried out by taking into account the learning model, which groups character into families based on their homogeneous shapes. SegNet, a segmentation algorithm written in C++ and Python, is used to research. This contains eighteen convolutional layers in the encoder, and each encoder layer has a corresponding decoder layer. As an activation function, Rectified Linear Units (ReLU) are employed. The final decoder output is connected to a multi-class softmax classifier

to provide class probabilities for each pixel individually. Tables 2 and 3 provide the parameters and matrix shapes for the encoding and decoding layers in the network. The total number of parameters includes the number of bias parameters. The learning rate is set to sigmoid decay.

Table 2: Arabic SegNet encoding layers with parameters and matrix shapes

Encoding layers	Kernel size	Number of filters	Convolution output	Batch normalization	Mask pooling	BN + Bias parameters	Total number of parameters
Convolution	3	64	$64 \times 205 \times 1365$	Yes	No	128	768
Convolution	3	64	$64 \times 205 \times 1365$	Yes	No	128	37056
Max pooling	2	-	-	No	Yes	-	-
Convolution	3	64	$64 \times 103 \times 683$	Yes	No	128	37056
Convolution	3	64	$64 \times 103 \times 683$	Yes	No	128	37056
Max pooling	2	-	-	No	Yes	-	-
Convolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Convolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Convolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Max pooling	2	-	-	No	Yes	-	-
Convolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056
Convolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056
Convolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056
Max pooling	2	-	-	No	Yes	-	-
Convolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Convolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Convolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Max pooling	2	-	-	No	Yes	-	-

Table 3: Arabic SegNet decoding layers with parameters and matrix shapes

Decoding layers	Kernel size	Number of filters	Deconvolution output	Batch normalization	Mask pooling	BN + Bias parameters	Total number of parameters
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 13 \times 86$	Yes	No	128	37056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 26 \times 171$	Yes	No	128	37056

(Continued)

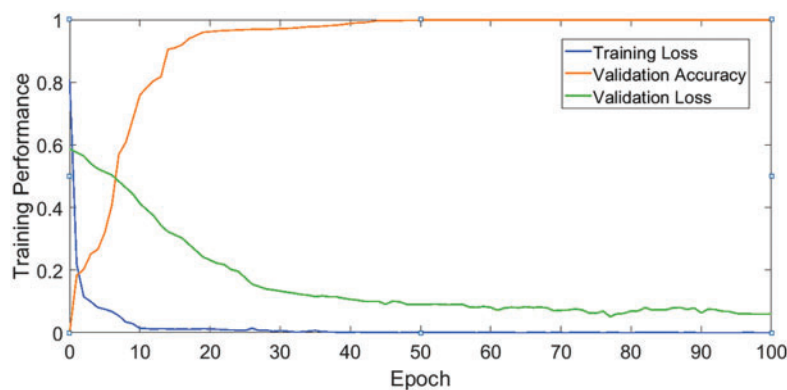
Table 3: Continued

Decoding layers	Kernel size	Number of filters	Deconvolution output	Batch normalization	Mask pooling	BN + Bias parameters	Total number of parameters
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 52 \times 342$	Yes	No	128	37056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	$64 \times 103 \times 683$	Yes	No	128	37056
Deconvolution	3	64	$64 \times 103 \times 683$	Yes	No	128	37056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	$64 \times 205 \times 1365$	Yes	No	128	37056
Deconvolution	3	36	$36 \times 205 \times 1365$	Yes	No	-	20900

Table 4 sums up the details of the training. The training took 58 h. The training procedure is depicted graphically in Fig. 6 (blue, orange, and green curves of training loss, training loss, and validation loss respectively).

Table 4: SegNet training details comparison with the proposed technique using the AI-ENT dataset

SegNet training details	Training with proposed method
Symbol classes	35
Training epochs	100
Batch size	1
Learning rate	0.001
Normalization	Pixel
Training time	58 Hrs

**Figure 6:** Graphical representation of SegNet validation accuracy, training loss, and Validation loss curves. Best viewed in color

4.2 Proposed Text Formation Model Experiments

As earlier described, with the use of the LeNet model implemented on MATLAB, all the candidates are placed in a 32×32 pixels buffer dimension following the concatenation procedure. As illustrated in the design Fig. 2, these candidates are trained with their corresponding ground truth. For this 46 classes are used. The network is trained for 3620 iterations using learning rates of 0.01 and batch size of 32. We select 12615 images (32×32) from the training part of tickers. Training details for the LeNet models are tabulated in Table 5. Training curve for the model is shown in Fig. 7. The final validation accuracy is 95.70%.

Table 5: LeNet training using the AI-ENT dataset

LeNet parameters	Values
Number of training characters	12615
Number of validation characters	1000
Batch size	32
Learning rate	0.01
Training time	1 m 19 sec

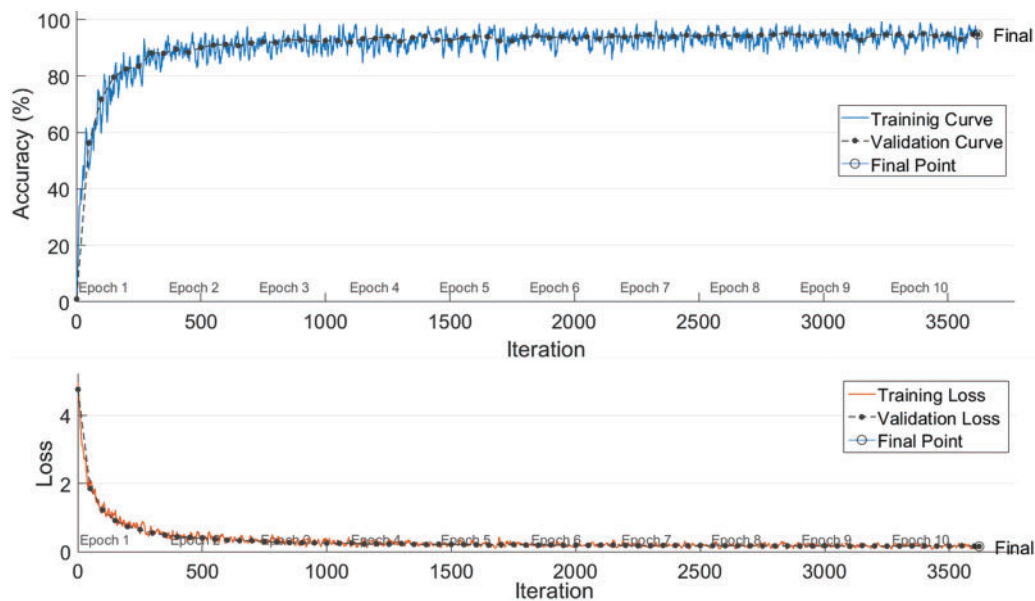


Figure 7: Graphical representation of LeNet validation accuracy, training loss, and Validation loss curves. Best viewed in color

5 Result Analysis

The suggested SegNet training model's final validation accuracy for 32 classes is 99.98%, with validation and testing losses of 0.0011 and 0.00030, respectively. Pixel-level segmentation refers to the classification of unique characters or component pixels from the rest, in terms of semantic segmentation. Segmenting character or character components as a unit within a ligature or word is

referred to as character or character component-level segmentation. After a filtration process that removes miss-classified pixels based on area under-recognized segments, the concatenation segmented letter pixels are boundaries as prediction masks. The appropriate information retrieval measures, F-Score and Accuracy, are given below in Eqs. (6)–(9) to evaluate each of the predicted masks with each of the available target masks of specified input. The scores are based on masks that overlap by at least 50%.

$$Accuracy = \frac{TotalCorrectPixels}{TotalNumberOfPixels} \quad (6)$$

$$precision = \frac{TP}{TP + FP} \quad (7)$$

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$$F - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (9)$$

where True Positive is TP, False Positive is FP, and False Negative is FN. On test data, the proposed segmentation model has an overall pixel-wise accuracy of 89.94%. The proposed approach's detailed performance is shown in Table 6. On the other hand, the reference method achieves 88.0% accuracy in segmentation observed manually.

Table 6: Detailed Arabic characters segmentation performance of proposed method

Character class	F-Score	Precision	Recall	Character class	F-Score	Precision	Recall
آ	0.95645	0.95325	0.95968	ك	0.99186	0.98982	0.99391
ب	0.98084	0.98479	0.97693	ل	0.98213	0.97634	0.98799
ت	0.95466	0.94598	0.96351	م	0.97331	0.97245	0.97418
ث	0.93248	0.91037	0.95571	ن	0.97699	0.97483	0.97916
ج	0.96612	0.94043	0.99327	و	0.96489	0.95002	0.98024
ح	0.98626	0.98983	0.98272	ه	0.97941	0.97343	0.98546
خ	0.97686	0.97253	0.98124	هـ	0.98333	0.97112	0.99587
د	0.93409	0.91037	0.95909	ء	0.98180	0.98276	0.99385
ذ	0.91471	0.90614	0.92345	ى	0.97621	0.96254	0.99028
ر	0.96152	0.94309	0.98070	ے	0.98981	0.98952	0.99011
ز	0.94988	0.94922	0.95056	1	0.94295	0.90037	0.98976
س	0.98136	0.97317	0.98969	2	0.94869	0.90239	1
ش	0.96315	0.96856	0.95781	3	0.95961	0.92237	1
ص	0.98003	0.97681	0.98328	4	0.97088	0.94342	1
ض	0.96998	0.96473	0.97529	5	0.95813	0.91964	1
ط	0.97976	0.96034	1	6	0.94310	0.89234	1
ظ	0.94008	0.93776	0.94242	7	0.98807	0.97644	1
ع	0.95678	0.92563	0.99012	8	0.94691	0.89918	1
غ	0.91065	0.83674	0.99890	9	0.99212	0.98438	1

(Continued)

Table 6: Continued

Character class	F-Score	Precision	Recall	Character class	F-Score	Precision	Recall
ف	0.97700	0.96466	0.98967	0	0.99337	0.98684	1
ق	0.99093	0.98634	0.99558	,,?	0.93050	0.88025	0.98685
				Average	0.96534	0.94883	0.98326

The accuracy of the text formation method is 96.26%. In the LeNet classification process, erroneous detection is usually de-shaped and hard to identify. Notably, the recognition rate performs better on the average font size of the AI-ENT dataset. Digits have more recall rate due to their uncorrelated shape and orientation. Due to the grouping of prime components and the simplicity of the learning model, the proposed technique benefits from a higher recognition rate of low-frequency characters. An explicit method in the context of Arabic characters recognition analysis is tabulated in [Table 7](#).

Table 7: Performance comparison with the AI-ENT dataset

Recent works	Accuracy in percentage
Supriana et al. [27]	82.0%
Qaroush et al. [22]	88.0%
This study	96.5%

6 Conclusion

This study suggested the Arabic New Ticker Recognition system, an expert application. The primary focus of the article is on novel ticker recognition models and storage schemes. This proposed an innovative explicitly learning-based strategy for recognizing cursive text of news tickers. The work focuses on the Al-Ekhbariya TV news channel, which is extensively watched in Saudi Arabia. The ticker text image is semantically segmented into letters, followed by LeNet based word identification method, and generates word codes for effective storage. Strong representation of data symbols with font sizes is essential. Data curation and augmentation, with the idea of character families incorporated to improve fairness. Evaluation results in a better character recognition rate. The technique proves the performance on recognition and has significance. Other learning models like UNet etc. can be used for performance evaluation. Moreover, the concept of an Arabic dictionary at the syntax correction stage can improve performance levels.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," *IEEE Access*, vol. 6, pp. 46019–46046, 2018.

- [2] S. A. B. Yousfi and C. Garcia, "Contribution of recurrent connectionist language models in improving lstm-based arabic text recognition in videos," *Pattern Recognition, Elsevier*, vol. 64, pp. 245–254, 2017.
- [3] S. Naz, A. I. Umar, R. Ahmed, S. B. Ahmed, S. H. Shirazi *et al.*, "Offline cursive nastaliq script recognition using multidimensional recurrent neural networks," *Neurocomputing*, vol. 177, pp. 228–241, 2016.
- [4] S. B. 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.
- [5] S. Rehman, B. Tayyab, M. F. Naeem, A. Hasan and F. Shafait, "A Multi-faceted ocr framework for artificial urdu news ticker text recognition," in *13th IAPR Int. Workshop on Document Analysis Systems*, Vienna, Austria, pp. 211–216, 2018.
- [6] A. Mirza and I. Siddiqi, "Impact of pre-processing on recognition of cursive video text," in *Iberian Conf. on Pattern Recognition and Image Analysis*, Madrid, Spain, pp. 565–576, 2019.
- [7] M. A. Mirza, O. Zeshan and I. Siddiqi, "Detection and recognition of cursive text from video frames," *Journal on Image and Video Processing (EURASIP)*, vol. 2020, no. 1, pp. 1–19, 2020.
- [8] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, S. A. Madani *et al.*, "The optical character recognition of urdu-like cursive scripts," *Pattern Recognition, Elsevier*, vol. 47, no. 3, pp. 1229–1248, 2014.
- [9] Q. A. Akram, S. Hussain, A. Niazi, U. Anjum and F. Irfan, "Adapting tesseract for complex scripts: An example for urdu nastalique," in *Workshop on Document Analysis Systems (DAS)*, Tours, France, pp. 191–195, 2014.
- [10] I. S. N. Javed, S. Shabbir and K. Khurshid, "Classification of urdu ligatures using convolutional neural networks-a novel approach," in *Int. Conf. on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, pp. 93–97, 2017.
- [11] I. Ahmad, X. Wang, Y. Mao, G. Liu, H. Ahmad *et al.*, "Ligature based urdu nastaleeq sentence recognition using gated bidirectional long short term memory," *Cluster Computing*, vol. 21, no. 1, pp. 703–714, 2017.
- [12] M. Fasha, B. Hammo and N. Obeid, "A hybrid deep learning model for arabic text recognition," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 122–130, 2020.
- [13] M. Badry, M. Hassanin, A. Chandio and N. Moustafa, "Quranic script optical text recognition using deep learning in IoT systems," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 1847–1858, 2021.
- [14] H. Butt, M. R. Raza, M. J. Ramzan, M. Junaid Ali and M. Haris, "Attention-based CNN-RNN Arabic text recognition from natural scene images," *Forecasting*, vol. 3, pp. 520–540, 2021.
- [15] S. B. Ahmed, S. Naz, M. I. Razzak, S. F. Rashid, M. Z. Afzal *et al.*, "Evaluation of cursive and non-cursive scripts using recurrent neural networks," *Journal Neural Computing and Applications*, vol. 27, pp. 603–613, 2016.
- [16] A. Hasan, S. B. Ahmed, S. F. Rashid, F. Shafait and T. M. Breuel, "Offline printed urdu nastaleeq script recognition with bidirectional LSTM networks," in *Proc. of the Twelfth Int. Conf. on Document Analysis and Recognition (ICDAR)*, Washington, DC, USA, pp. 1061–1065, 2013.
- [17] S. Naz, A. I. Umar, R. Ahmad, M. I. Razzak, S. F. Rashid *et al.*, "Urdu nastaliq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks," *SpringerPlus*, vol. 5, no. 1, pp. 1–16, 2016.
- [18] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed and S. H. Shirazi, "Urdu nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features," *Neural Computing and Applications*, vol. 28, no. 2, pp. 219–231, 2017.
- [19] R. A. Saeeda Naz, S. B. Ahmed and M. I. Razzak, "Zoning features and 2dlstm for urdu text-line recognition," in *Int. Conf. on Knowledge Based and Intelligent Information and Engineering Systems*, York, United Kingdom, pp. 16–22, 2017.
- [20] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi and S. B. Ahmed, "Urdu nastaliq recognition using convolutional recursive deep learning," *Neurocomputing*, vol. 243, pp. 80–87, 2017.
- [21] S. Naz, S. B. Ahmed, M. I. Razzak and R. Yousaf, "Deep learning based isolated arabic scene character recognition," *1st Int. Workshop on Arabic Script Analysis and Recognition*, Nancy, France, pp. 46–51, 2017.

- [22] A. Qaroush, A. Awad, M. Modallal and M. Ziq, "Segmentation-based, omnifont printed arabic character recognition without font identification," *Journal of King Saud University Computer and Information Sciences*, vol. 34, no. 6, pp. 3025–3039, 2020.
- [23] H. Osman, K. Zaghaw, M. Hazem and S. Elsehely, "An efficient language-independent multi-font OCR for arabic script, November," in *10th Int. Conf. on Advances in Computing and Information Technology*, London, United Kingdom, pp. 57–71, 2020.
- [24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd Int. Conf. on Learning Representations*, San Diego, CA, USA, pp. 1409–1556, 2015.
- [25] A. K. V. Badrinarayanan and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Tran on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 1–14, 2016.
- [26] S. A. Hassan, I. Haq, M. Asif, M. B. Ahmad and M. Tayyab, "An efficient scheme for real-time information storage and retrieval systems: A hybrid approach," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, pp. 427–431, 2017.
- [27] I. Supriana and A. Nasution, "Arabic character recognition system development," in *Int. Conf. on Electrical Engineering and Informatics*, Bangi, Selangor Darul Ehsan, Malaysia, pp. 334–341, 2013.