Journal of Intelligent System and Applied Data Science (JISADS)

Journal homepage : https://www.jisads.com
ISSN (2974-9840) Online

Leveraging Arabic Text Embedded in Images: Challenges and Opportunities in NLP Analysis

*Aws I. AbuEid [1], *, Whida mansouri[2], Ahlem Fatnassi[2] , Olfa Ben Rhaiem[2], Radhia Zaghdoud[2] , Achraf Ben Miled[2,3], Ashraf F. A. Mahmoud[2] ,Faroug A. Abdalla[2] ,Marwa Anwar Ibrahim Elghazawy[4], Mohammed Ahmed Elhossiny[4,5], Aida Dhibi[2], Firas M. Allan[2], Chams Jabnoun[2], Imen Ben Mohamed[2], , Majid A. Nawaz[2], Salem Belhaj[2]*

[1]*Faculty of Computing Studies, Arab Open University,Amman, Jordan*
[2]*Computer Science Department, Science College, Northern Border University, Arar, Kingdom of Saudi Arabia*
[3]*Artificial Intelligence and Data Engineering Laboratory, LR21ES23, Faculty of Sciences of Bizerte, University of Carthage, Tunisia*
[4]*Applied College, Northern Border University, Arar, Saudi Arabia*
[5]*Faculty of Specific Education, Mansoura University, Mansoura, Egypt.*
**Corresponding Author Email: a_abueid@aou.edu.jo*

ABSTRACT

While recent advances in scene text recognition have blossomed, research has primarily focused on languages utilizing Latin scripts, neglecting languages with unique characteristics like Arabic. This study aims to bridge this gap by delving into the under-researched domain of Arabic scene text recognition. Describing Arabic images necessitates a fusion of computer vision and natural language processing, highlighting the intricate challenges AI algorithms encounter within this cross-domain, multi-modal landscape. The objective is to generate natural language descriptions for given test images, capturing crucial details such as characters, settings, actions, and more, while adhering to natural language conventions. However, the lack of readily available open-source Arabic datasets presents a significant obstacle, as most image description research revolves around English resources. Additionally, the inherent syntactic flexibility and linguistic nuances of Arabic descriptions amplify the algorithmic implementation challenges. Consequently, research concerning image descriptions, particularly in Arabic, needs to be explored more. To bridge this gap and facilitate further research, we introduce a novel dataset, the Arabic-English Daily Life Scene Text Dataset (EvArEST). Our study demonstrates promising progress in Arabic scene text recognition, highlighting both the challenges and opportunities of multi-modal AI algorithms. We conclude by emphasizing the need for more extensive datasets and algorithmic refinements to unlock the full potential of Arabic image descriptions in the context of NLP analysis.

Keywords: Image Caption in Arabic, deep learning, text recognition, NLP

1. INTRODUCTION

Text recognition within natural scenes is a pivotal component of systems aiming to comprehend images, given that text represents one of the most prevalent forms of ubiquitous communication in our surroundings [1]. This challenge encompasses the broader context of text reading, beginning with text detection to locate text within an image and progressing to text recognition to convert these instances into legible words [2]. Scene text reading carries various practical applications in our daily lives, including developing translation systems that transcend language barriers enabling real-time reading and translation of text. Moreover, visual aid systems could significantly benefit the visually impaired by facilitating the reading of signs, ATM instructions, or books through text-to-voice systems[3]. The applications extend to intelligent inspection, multimedia retrieval, and product recognition.

Addressing scene text recognition (STR) is an intricate challenge, compounded by numerous factors distinct to text within natural scenes [4]. Beyond the conventional hurdles faced in computer vision tasks—such as image noise, scene complexity, and variations in viewpoint and brightness—the text found in natural scenes presents unique challenges [5]. This includes various font styles and shapes inherent to any language, alongside additional variations attributable to artistic effects, atypical orientations, in-plane and out-of-plane curvature, and perspective transformations. These nuances necessitate focused attention on text recognition within natural scenes, justifying its standing as a prominent and autonomous problem in research.

A typical deep-learning-based STR framework comprises four primary stages: preprocessing to facilitate recognition, feature extraction utilizing convolutional neural networks (CNN), sequence processing of extracted features, and final word prediction [6].

The current research landscape highlights the burgeoning domains of natural language processing (NLP) and computer vision (CV). NLP delves into understanding natural language, encompassing text generation, word segmentation, part-of-speech tagging, syntactic analysis, and multi-language machine translation. Meanwhile, CV revolves around comprehending images or videos and facilitating tasks such as classification, target detection, image retrieval, semantic segmentation, and human pose estimation. Recent attention has veered toward multimodal processing, integrating text and image information. Image Captioning is a critical facet of multimodal processing, enabling image-to-text transformation and aiding visually impaired individuals in comprehending image content.

Figure 1 displays a selection of examples sourced from the EvArEST dataset, specifically focusing on Arabic-English scene text samples. The dataset aims to encompass various instances involving textual elements in scenes, providing a comprehensive collection that caters to both Arabic and English..

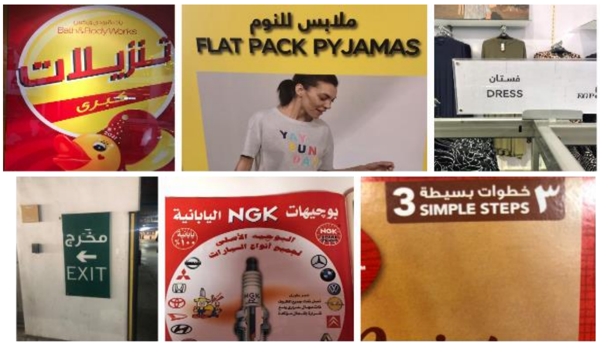

Figure 1: EvArEST: Arabic-English scene text samples.

Additionally, Figure 1 highlights the diverse and abundant text variations within scenes, reflecting the richness in multi-lingual settings.

Presently, research predominantly focuses on generating image abstracts in English, with limited exploration into Arabic abstract generation methods. The complexity of Arabic words and sentence structures further accentuates the difficulty in describing images in Arabic. This underscores the need for innovative approaches based on diverse datasets.

2. RELATED WORK

A The current common method is based on the extension of neural networks, most composed of an encoder and decoder. The image is encoded using a pre-trained deep convolutional neural network (CNN), and then the image is embedded into a recurrent neural network (RNN). The corresponding description sequence is finally output as a description.

Gaafar et al. [8] employed a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) architecture for training on both textual and image datasets. The evolution of training and validation accuracy during the learning process was presented. Notably, strong correlations were observed between the two metrics, particularly during the initial training epochs (1 to 10). In the textual domain, the LSTM-RNN model achieved an accuracy of 85.69% for classifying 1000 words into five distinct classes. However, the training and validation processes were slower, requiring 18.25 minutes. The study concluded that the LSTM-RNN achieved better results for image classification regarding both speed and accuracy. This was attributed to the inherent complexity of hidden patterns within visual data compared to textual information.

Du et al. [8] proposed SVTR, a novel single visual model for scene text recognition that bypasses the conventional hybrid architecture of feature extraction and sequence modeling. Instead, SVTR utilizes a patch-wise image tokenization framework, decomposing text images into character components. Hierarchical mixing stages capture intra- and inter-character relationships, enabling recognition without sequential modeling. This approach demonstrates competitive accuracy on English datasets. It significantly outperforms existing methods on Chinese datasets, Attention maps generated by the SVTR-T model provide further evidence of achieving faster inference times. The effectiveness of SVTR's multi-grained character component perception. These maps reveal the model's ability to recognize sub-character, character-level, and cross-character features, providing deeper insights into its recognition process.SVTR offers a versatile solution with two variants: SVTR-L and SVTR-T. SVTR-L balances accuracy, speed, and cross-lingual capabilities, making it suitable for diverse application needs. Conversely, SVTR-T prioritizes resource efficiency, offering excellent performance in resource-limited scenarios. Additionally, SVTR presents a compelling single visual model for scene text recognition. It achieves competitive accuracy and speed

information is collected [6]. Consider the limitations: Acknowledge any limitations of the framework and how they may affect the interpretation of the results. It is important to note that a theoretical framework should not be confused with a literature review. The literature review provides background information and context for the research, while the theoretical framework provides a structure for understanding the relationships between the studied variables.

While scene text recognition has garnered significant research attention, existing literature reviews often need to pay more attention to the unique challenges and advancements associated with recognizing Arabic text in images. This study addresses this gap by focusing on Arabic script and its associated complexities.

Challenges of Arabic Script: Arabic script presents distinct hurdles for recognition systems due to its inherent characteristics. These include:

Cursive nature: Unlike Latin script with predominantly disconnected characters, Arabic script features characters that connect in various ways, impacting segmentation and recognition.

Variable ligatures: The way certain Arabic characters connect can vary depending on their position within a word, posing challenges for accurate character identification.

Contextual forms: Arabic characters can alter their appearance based on their position within a word (beginning, middle, or end), further complicating recognition.

Addressing the Gap: Previous Research in Arabic Text Recognition

To comprehensively understand the current state of Arabic text recognition, this study delves into various relevant research endeavors:

Comparative Analysis of Advancements: This research compares different approaches and algorithms employed for Arabic text recognition, highlighting their strengths and weaknesses. Such analysis provides valuable insights into the current state-of-the-art techniques in this domain.

Review of Challenges and Opportunities: This work identifies key challenges specific to Arabic text recognition, such as variable ligatures, diacritics (vowel markings), and font variations. It further explores potential solutions and future research directions to address these issues effectively.

Survey of Arabic Text Recognition: This research delves specifically into scene text recognition for Arabic images. It provides an overview of existing datasets, benchmark metrics used for performance evaluation, and the state-of-the-art approaches currently employed. This comprehensive survey serves as a valuable resource for researchers developing Arabic text recognition systems.

Evaluations of Deep Learning Approaches: This study explores the performance of various deep learning architectures on Arabic text recognition tasks. Analyzing their effectiveness in handling challenges like text skew, noise, and background clutter offers valuable insights for future developments.

Discussions on Future Directions: This research focuses on developing robust and scalable systems specifically for Arabic text recognition. Highlighting important avenues for future research is crucial for continued progress in this field.

3. THE EVAREST DATASET

The EvArEST dataset, available at https://github.com/HGamal11/EvArEST-dataset-for-Arabic-scene-text, is a valuable resource for researchers and developers working on Arabic scene text recognition. It comprises two distinct datasets:

1. Text Detection Dataset:

This dataset consists of 510 images containing one or more instances of text in this study; the sample is 133 from 510 images. Each word annotation is provided as a four-point polygon, starting from the top left corner and proceeding clockwise. The dataset also includes a text file for each image, specifying the polygon's four points and the language of the word. This structure facilitates the task of text localization within natural scenes.

Moreover, the EvArEST dataset exhibits diversity in terms of text appearance, orientation, font styles, and background complexity, making it suitable for evaluating the robustness and generalization capabilities of Arabic text recognition algorithms across various real-world scenarios. This dataset serves as a benchmark for assessing the performance of text detection algorithms specifically tailored for Arabic script, thereby facilitating advancements in the field of Arabic scene text recognition.

2. Recognition Dataset:

This dataset offers 7232 cropped word images featuring Arabic and English text. The ground truth data is provided as a text file, where each line associates an image filename with its corresponding text. This dataset primarily aims to support the development of Arabic text recognition models but can also be extended to bilingual text recognition tasks.

4. METHODOLOGY AND EXPERIMENTS

As previously indicated, the process of text prediction in STR involves four key stages. This section details the distinct methods adopted for each stage within the STR system

4.1 PREPROCESSING

The preprocessing phase is crucial in scene text recognition pipelines by optimizing images for subsequent feature extraction. This involves addressing various inherent challenges commonly found in text images. Algorithm 1, employed in this work, tackles these challenges by performing the following:

1. Image Dimension Standardization: All images are resized to a standardized dimension of 255x255 pixels. This facilitates efficient processing and ensures consistent input to subsequent pipeline stages.

2. Irregular Text Handling: A specialized algorithm detects and corrects irregularity in text layouts, such as skewed or distorted characters. This improves the readability of the text and promotes consistent character representation across the image.

3. Font Style Normalization: Techniques are implemented to address the diverse font styles encountered in natural scenes. These techniques aim to normalize and standardize different font types, enhancing the uniformity of character appearance and improving recognition accuracy.

4. Background Noise Reduction: Advanced noise reduction methods are applied to eliminate unwanted background noise that might obscure or interfere with the text. Techniques such as denoising filters ensure clear and unobstructed text visibility, facilitating accurate character identification.

5. Inclination and Illumination Correction: Algorithmic approaches are used to rectify inclination and illumination disparities within the images. These processes involve techniques to adjust text orientation and normalize uneven illumination, ensuring consistent and balanced lighting across the image.

The preprocessing phase effectively addresses the diverse challenges inherent in text images through the sequential application of these algorithms and techniques. By mitigating issues like irregular text structures, variations in fonts, background noise interference, and inconsistencies in inclination and illumination, the preprocessed images become significantly more amenable to subsequent feature extraction and analysis. This, in turn, contributes to enhancing the accuracy and reliability of text recognition and analysis algorithms.

It is important to note that the specific algorithms and techniques employed in the preprocessing phase may vary depending on the specific characteristics of the dataset and the desired performance metrics. However, the overall goal remains consistent: to optimize the images for robust and accurate text recognition by addressing the inherent challenges in scene text data.

Algorithm 1: preprocessed image

```
Input:
- dataset in jpg or png format
Output:
-preprocessed image

Import necessary libraries:
 Load the dataset:
resized_image=Resize(image, 255x255)
gray_image= ConvertToGrayscale(resized_image)
thresholded_image= ApplyThresholding(gray_image)
denoised_image= Denoise(thresholded_image)
adjusted_image= AdjustIllumination(denoised_image)
```

Algorithm 1 for image preprocessing begins by defining a function to process individual images. Within this function, it performs a sequence of operations: reading the image, resizing it to standardized 255x255 dimensions, converting it to grayscale, applying thresholding for irregular text handling, denoising to reduce background noise, and adjusting illumination for overall enhancement. Figure 2 illustrates the transformation from the raw or original image (before preprocessing) to the processed image (after applying the preprocessing steps).

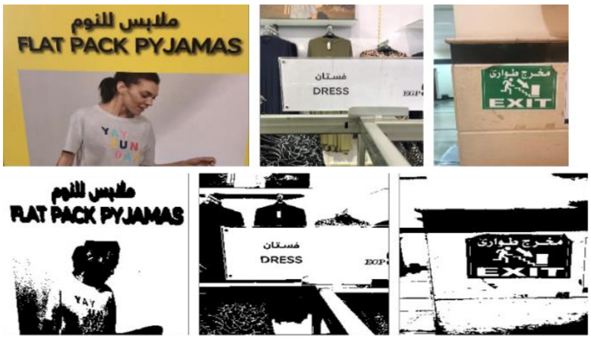

Figure 2: Comparison of Sample Image Before and After Preprocessing

On the other hand, the "after processing" image demonstrates enhancements: the irregular text might be clearer and more structured, different fonts could be more standardized, background noise reduced, and the overall illumination might be more uniform. The differences between the two images emphasize the effectiveness of the preprocessing steps in improving the image quality for subsequent analysis.

4.2 Feature Extraction through Stroke Width Transform

The Stroke Width Transform (SWT) plays a pivotal role in the feature extraction stage of the text recognition pipeline [21]. This powerful algorithm is particularly adept at text detection and image localization tasks. Its effectiveness stems from its ability to analyze and exploit stroke width variations across text regions [22]. This is particularly valuable for scenarios involving diverse font styles and sizes, as are prevalent in Arabic text.

SWT identifies regions where stroke widths exhibit consistent patterns, leveraging this information to precisely delineate and isolate text segments from complex backgrounds or cluttered scenes. This transformative technique forms a foundational step in text recognition systems. By enabling accurate localization and segmentation of text regions, SWT is an essential tool in computer vision and optical character recognition (OCR) applications [23].

4.3 Sequence Processing: Bridging the Gap between Visual Features and Textual Meaning

Sequence processing plays a critical role in modern text recognition systems, bridging the gap between the extracted visual features and the prediction of meaningful information. This crucial step utilizes the predictive potential of features acquired through Stroke Width Transform (SWT) during the feature extraction stage. These features, derived from individual words within the image, encapsulate intricate details of the text's visual attributes, facilitating a comprehensive understanding of the sequence.

By leveraging SWT-generated features, the sequence-processing phase transcends the limitations of raw visual data. It allows the system to analyze the features within their contextual relationships, uncovering hidden patterns and dependencies between individual elements within the sequence. This enables the system to predict the textual content with increased accuracy and reliability.

Algorithm 2, presented above, outlines a comprehensive approach for extracting salient features from images. It leverages various techniques to capture essential characteristics that contribute to the image's visual representation:

1. Corner Features: Corner detection algorithms identify and quantify distinct corners within the image. This provides a robust measure of the distribution and

prevalence of significant image elements.

2. Edge Features: The image's edges are identified and delineated using edge detection algorithms. The total number of edges present in the image is calculated by counting non-zero pixels. This information captures valuable insights into the structural and textural properties of the image.

3. Mean Channel Features: This component extracts the mean value for each channel in the image. This concisely represents the image's overall intensity and color distribution.

Algorithm 2: Image Feature Extraction

```
Input:
- preprocessed image dataset output of Algorthim1
Output:
- dataset contains extracted features of the
preprocessed image dataset

1. Define function extract_image_features
   taking a preprocessed image dataset with
   the output of Algorthim1 as input
2. Extract corner features using a corner
   detection algorithm, counting the number of
   corners found.
3. Extract edge features using an edge
   detection algorithm, calculating the number
   of edges.
4. Calculate the mean  features of one channel
5. Use placeholder values for sharpness and
   texture features.
6. Save the DataFrame of
   extract_image_features as a CSV file
```

By extracting these distinct features, Algorithm 2 offers a valuable set of characteristics that provide crucial insights into the images' structural, textural, and other related attributes. This information is vital for subsequent analysis, prediction, classification, or other computational tasks.

Table 1 compares diverse image features extracted from three sample images in datasets: Image 5, Image 20, and Image 119, shown in Figure 2. It comprehensively captures key attributes such as:

- Corner Features: Represents the number of distinct corners identified within the image.
- Edge Features: Represents the number of detected edges within the image.
- Mean Color Features: Represents the average intensity for each color channel (Blue, Green, Red).
- Processed Corner Features: Represents the number of corner features after applying post-processing techniques.
- Processed Edge Features: Represents the number of edge features after applying post-processing techniques.
- Processed Mean Color Feature: Represents the average intensity for one color channel after post-processing.

Image 5: This image exhibits a high number of features, with 100 Corner Features, 245,935 Edge Features, and mean color values of 149.4791 (Blue), 156.4568 (Green), and 152.4374 (Red). Post-processing reduces the Edge Features to 8,889 but leaves the Corner Features unchanged. Additionally, the Mean Color Feature from one channel is transformed to 136.6164552.

Image 20: This image exhibits Corner Features (100) similar to Image 5 but a significantly lower number of Edge Features (33,289). The mean color values differ slightly, with Blue at 160.3355, Green at 149.8498, and Red at 156.4595. After processing, the Edge Features are further reduced to 5,371, and the Mean Color Feature from one channel becomes 142.2783545.

Image 119: This image exhibits a lower number of features compared to the previous two, with 100 Corner Features, 120,589 Edge Features, and mean color values of 83.26154 (Blue), 87.75472 (Green), and 89.2075 (Red). Post-processing significantly reduces the Corner Features (52) and Edge Features (1,556). The processed Mean Color Feature from one channel is 32.73571703.

Table 1 Comparative Analysis of Extracted Image Featurests

Image No	Corner Features	Edge Features	Mean Color Features (B)	Mean Color Features (G)	Mean Color Features (R)	Corner Features	Edge Features after a process	Mean one channel Features after a process
5	100	245935	149.4791	156.4568	152.4374	100	8889	136.6164552
20	100	33289	160.3355	149.8498	156.4595	100	5371	142.2783545
119	100	120589	83.26154	87.75472	89.2075	52	1556	32.73571703

Table 1 provides valuable insights into the variations observed in corner detection, edge detection, and color feature extraction across different images. It is a reference for understanding how these features change after applying post-processing techniques, highlighting their impact on the extracted characteristics. Additionally, Figure 3 presentation provides a comprehensive overview of image characteristics before and after processing.

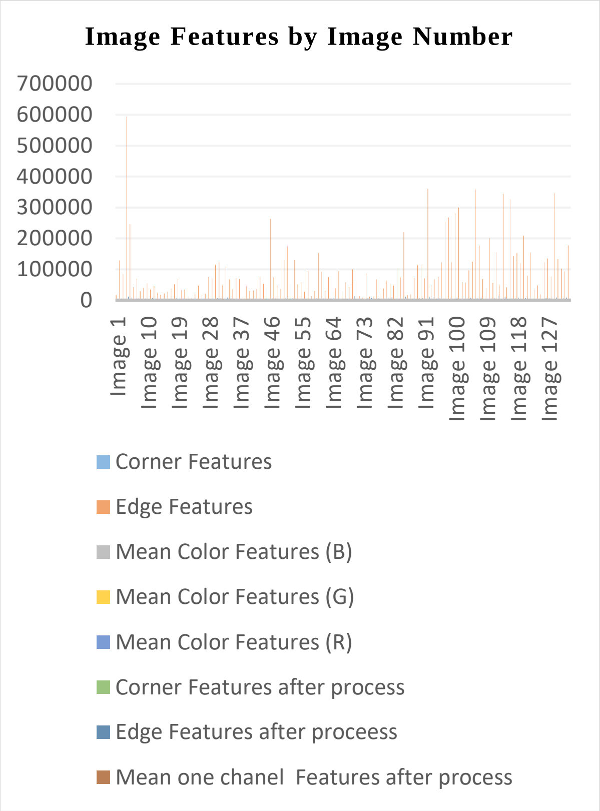

Figure 3: a comprehensive overview of image characteristics before and after processing.

Figure 3 illustrates a detailed breakdown of different image features and their corresponding values. The data is organized into two distinct sections. The first section includes columns such as Image No, Corner Features, Edge Features, Mean Color Features (B), Mean Color Features (G), and Mean Color Features (R). The second section demonstrates processed values, displaying the changes after a specific process. For instance, it displays Corner Features after processing, Edge Features after processing, and Mean one-channel Features after processing. Each row represents an individual image (from Image 1 to Image 133) and its respective numerical values for the features above. This structured presentation provides a comprehensive overview of image characteristics before and after processing.

4.4 prediction

Optical Character Recognition (OCR) and text recognition tasks rely heavily on two main methods for the prediction stage: attention mechanisms and Connectionist Temporal Classification (CTC) loss. Both methods have proven highly effective in these applications, as evidenced by their widespread adoption [24].

1. Attention Mechanisms in OCR
Inspired by the human visual attention system, attention mechanisms in OCR dynamically focus on specific regions of the input image when decoding the output sequence (text) [26]. This allows the model to allocate resources more efficiently, improving accuracy and robustness.

2. Connectionist Temporal Classification (CTC) Loss
CTC loss is a popular function in various sequence prediction tasks, including OCR. It effectively handles scenarios where the alignment between inputs and outputs is not readily available. CTC loss offers significant advantages by enabling the model to learn directly from input-output pairs without requiring explicit alignment information [27].
This study investigates the potential of hybrid models combining attention mechanisms and CTC loss. By leveraging the strengths of both approaches, these hybrid models aim to achieve even greater performance in OCR and Arabic text recognition tasks.
Algorithm 3 utilizes Tesseract OCR, OpenCV, and Pandas libraries in Python to extract text from the dataset. It processes each image, extracts text using OCR, stores the results in a Data Frame and saves the extracted text along with their respective image numbers for further analysis or usage.

Algorithm 3: Arabic Text Extraction

Input:
- preprocessed image dataset output of Algorthim1
Output:
- dataset contains image No ,Extracted Text

```
1.  Import necessary libraries
2.  Read an image using OpenCV.
3.  Utilize Tesseract via `pytesseract` to extract
    text with Arabic language settings.
4.  Return the extracted text.
5.  Read an image using OpenCV.
6.  Utilize Tesseract via `pytesseract` to extract text
    with Arabic language settings.
7.  Save the Data Frame of image No ,Extracted
    Text as a CSV file
```

Algorithm 3 leverages three powerful open-source libraries to efficiently extract and manage textual information from images:

1. Tesseract OCR:

This open-source library performs optical character recognition (OCR), converting the text embedded within an image into machine-readable text [28]. Algorithm 3 utilizes Tesseract to bridge the gap between the visual and textual domains, enabling further analysis and storage of the extracted information.

2. OpenCV:

OpenCV, a popular library for computer vision and machine learning, provides a comprehensive toolbox for image and video processing [29]. Algorithm 3 employs OpenCV for crucial preprocessing steps, such as reading and manipulating the input images, ensuring optimal performance for the subsequent Tesseract OCR process.

3. Pandas:

This powerful Python library facilitates data manipulation and analysis [30]. Algorithm 3 leverages Pandas' data structures, particularly Data Frames, to handle and analyze the extracted text effectively. This enables efficient organization, management, and exploration of the retrieved information, facilitating further investigation and utilization.

Combining these complementary libraries' strengths, Algorithm 3 provides an efficient and robust solution for extracting and managing textual information from images.

4.4.1 Merging Diverse Data Sources for Enhanced Analysis

This study employed a data fusion approach to construct a comprehensive dataset suitable for further analysis and exploration. The process involved merging three distinct sources of information:

1. Extracted Text:

Utilizing Algorithm 3 and Tesseract OCR, Arabic text was extracted from preprocessed image data generated by Algorithm 2. This extracted textual information formed the initial data source for the merged dataset.

2. Extracted Features:

The second data source consisted of extracted features derived from preprocessed image data, also generated by Algorithm 2. These features captured essential visual characteristics of the images.

3. External Textual Information:

Additional textual information was acquired from an external source, the EvAREST Dataset to enrich the dataset further. This textual data was meticulously matched to the corresponding image numbers based on pre-existing association information.

By merging these three data sources, a comprehensive dataset was constructed. Each entry in the dataset now contained four key components:

- Image number.
- Extracted features (see Table 1).
- Extracted text.
- External textual information

This unified dataset offers a powerful resource for further analysis and exploration to investigate the relationships between visual content, extracted features, and external textual data, leading to a richer understanding of the information embedded within images. Additionally, this comprehensive dataset paves the way for enhanced prediction and training models. Table 2 shows a sample of the extracted features from various images along with the corresponding extracted text and external textual information. Each row represents a specific image with its associated features and textual data.

Table 2: Image Features and Extracted Textual Information

Img No	Corner Features	Edge Features	Mean Color Features (B)	Mean Color Features (G)	Mean Color Features (R)	Corner Features	Edge Features after a process	Mean one channel Features	Extracted text	External textual information
1	100	16605	110.4525	116.3103	156.9713	95	3469	32.71055748	للمفرونثات الراقيه بحجفله وقطاعى	للمفروشات الراقية جملة
29	100	72270	178.2424033	177.2633509	158.6950827	100	4420	173.2581007	القاهرة الجديدة بيعلا الرحاب ,	القاهرة الجديدة الرحاب
33	100	108981	132.8643507	131.7163183	139.0303376	100	8885	106.4061207	أقساط, بالعاصمة ,جع الإدارية,	شركتك عيادتك محلك وسط حي

4.4.2 Utilizing TF-IDF for Text Feature Extraction in Natural Language Processing

Within Natural Language Processing (NLP), Feature Extraction occupies a crucial position, transforming text data into a form readily interpretable and exploitable by machine learning algorithms [31]. TF-IDF (Term Frequency-Inverse Document Frequency) stands out as a fundamental pillar among the various techniques employed in this process.

TF-IDF quantifies the importance of individual words within a corpus of documents by considering two key factors:

Term Frequency: This metric assesses the frequency with which a word appears within a document. The higher the frequency, the more central the word will likely be to the document's meaning.

Inverse Document Frequency: This metric measures the rarity or commonality of a word across the entire corpus. Words appearing in only a few documents are considered more informative due to their specificity.

By combining these two factors, TF-IDF assigns a weight to each word, reflecting its significance within a document and across the entire dataset [31].

Leveraging TF-IDF facilitates extracting meaningful features from the raw text, enabling the construction of robust models for various NLP tasks. These tasks encompass classification, clustering, and information retrieval, thereby unlocking the hidden potential of textual information for diverse applications. Table 3 presents a transformed representation of text data, likely obtained using the TF-IDF (Term Frequency-Inverse Document Frequency) technique for Feature Extraction in Natural Language Processing (NLP).

Each column represents an individual word or term extracted from the original text data, and each row corresponds to a distinct piece of text or document from the original dataset.

Within each row, the presence and relevance of the extracted terms are scored based on their TF-IDF values. These values are calculated for each term by considering two key factors:

- Term Frequency (TF): This metric measures the frequency of a specific term within the corresponding document. Higher TF values indicate greater importance of the term within that particular document.
- Inverse Document Frequency (IDF): This metric assesses the rarity of a term across the entire corpus. Terms appearing in only a few documents are deemed more informative due to their exclusivity, resulting in higher IDF values.

Table 3: TF-IDF Feature Representation for Extracted Text Data

Original image	**أقساط**	**الادارية**	**الجديدة**	**الراقيه**	**الرحاب**	**القاهرة**	**بالعاصمة**	**للمفرونشات**	**محلك**	**قطاعى**
للمفروشات الراقية جملة قطاعي تركي	0	0	0	0.5	0	0	0	0.5	0	0.5
القاهرة الجديدة الرحاب	0	0	0.5	0	0.5	0.5	0	0	0	0
شركتك عيادتك محلك وسط حي الوزارات	0.447214	0.447214	0	0	0	0	0.447214	0	0.447214	0
Extracted text	**أقساط**	**الادارية**	**الجديدة**	**الراقيه**	**الرحاب**	**القاهرة**	**بالعاصمة**	**للمفرونشات**	**محلك**	**وقطاعى**
للمفرونشات الراقيه بحجفله وقطاعى	0	0	0	0.408248	0	0	0	0.408248	0	0.408248
القاهرة الجديدة بيعلا الرحاب	0	0	0.47363	0	0.622766	0.622766	0	0	0	0
أقساط جع بالعاصمة الإدارية محلك	0.293884	0.293884	0.223506	0	0	0	0.293884	0	0.293884	0

Table 3 transforms the raw text data into a numerical representation that captures its semantic significance. TF-IDF enables the effective utilization of machine learning models. This technique provides a robust and informative representation of textual information, facilitating various NLP tasks such as classification, clustering, and information retrieval.

5. RESULTS AND DISCUSSION

The study aimed to extract text from images, perform feature extraction using TF-IDF, and integrate datasets to explore text data using NLP techniques. The results highlight both successful endeavors and challenges encountered in these processes.

1. Text Extraction from Images

The application of Tesseract OCR and OpenCV showcased successful text extraction from multiple images. However, challenges were noted in maintaining the integrity of the image-to-text conversion process, particularly in handling diverse image qualities and textual variations.

2. Feature Extraction with TF-IDF

An attempt was made to employ TF-IDF for feature extraction from text data. Challenges emerged in handling empty vocabularies and pruning, indicating potential issues with the data quality or the parameter settings used in the extraction process. This highlights the need for further optimization and data preprocessing.

3. Combining Datasets

Efforts were directed towards combining datasets containing text features extracted from images with existing datasets. This integration aimed to augment the information available for analysis, potentially incorporating image-related data or additional contextual information. However, the merging process might necessitate further consideration of data compatibility and alignment.

4. Utilizing NLP Techniques

The intent to apply natural language processing techniques to the extracted text data was established. The application of NLP could offer deeper insights and facilitate advanced analysis. Challenges encountered in the initial steps underline the importance of refining preprocessing steps and parameter adjustments for successful application.

In summary, this study reflects efforts to extract text from images, conduct feature engineering, and integrate datasets for comprehensive text analysis. While successful in demonstrating text extraction capabilities and the intent to apply NLP techniques, encountered obstacles emphasize the necessity for refining methodologies, data preprocessing, and parameter optimization for a robust and accurate analysis of text data sourced from images.

6. CONCLUSION AND FUTURE WORK

This study investigated the extraction of Arabic text from images, explored TF-IDF for feature extraction, and examined the integration of diverse datasets for Natural Language Processing (NLP) analysis. The aim was to explore and leverage Arabic text data embedded within images for comprehensive analysis and insights.

The application of Tesseract OCR and OpenCV successfully extracted Arabic text from images. However, challenges emerged regarding consistency across varied image qualities and handling Arabic text variations. These findings emphasize the need for robust preprocessing and image quality enhancement techniques.

Utilizing TF-IDF for feature extraction demonstrated promising avenues for analyzing Arabic text data. Despite encountering challenges related to empty vocabularies and pruning, this technique exhibited potential for extracting significant features. Further refinement and optimization are necessary to improve its effectiveness.

Integrating datasets containing Arabic text features extracted from images with existing datasets showcased the potential to augment analytical capabilities. However, aligning and harmonizing disparate datasets necessitate careful consideration to ensure compatibility and meaningful integration.

Challenges and Opportunities of NLP with Arabic Text

The application of NLP techniques to Arabic text data presented both opportunities and challenges. While NLP promises advanced insights, encountered limitations during the initial stages highlighted the importance of refining preprocessing methods and parameter tuning for effective application.

Conclusion

This study laid the groundwork for Arabic text extraction, feature engineering, and dataset integration, marking crucial steps towards comprehensive Arabic text analysis from images. Challenges encountered signify areas requiring further refinement and optimization to fully unlock the potential of Arabic text data sourced from images.

Future Work

Building upon the findings of this study, future research endeavors should focus on the following:

Enhanced image preprocessing and quality enhancement techniques to address challenges related to variations in image quality and Arabic text representation.

Advanced feature extraction methods beyond TF-IDF.

Development of robust and optimized NLP models specifically designed for Arabic text analysis, addressing issues related to empty vocabularies and parameter tuning.

Construction of large-scale and diverse Arabic text datasets extracted from images, facilitating further research and development in Arabic NLP.

Future research aims to address existing challenges and unlock the full potential of Arabic text data embedded within images, contributing to advancements in NLP and related fields.

REFERENCES

[1] Nahar, K. M., Alsmadi, I., Al Mamlook, R. E., Nasayreh, A., Gharaibeh, H., Almuflih, A. S., & Alasim, F. (2023). Recognition of Arabic Air-Written Letters: Machine Learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques. Sensors, 23(23), 9475.

[2] Naosekpam, V., & Sahu, N. (2022). Text detection, recognition, and script identification in natural scene images: A Review. International Journal of Multimedia Information Retrieval, 11(3), 291-314.

[3] Messaoudi, M. D., Menelas, B. A. J., & Mcheick, H. (2022). Review of Navigation Assistive Tools and Technologies for the Visually Impaired. Sensors, 22(20), 7888.

[4] Zheng, C., Li, H., Rhee, S. M., Han, S., Han, J.

J., & Wang, P. (2022). Pushing the performance limit of scene text recognizer without human annotation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14116-14125).

[5] Zhang, C., Zhang, C., Zhang, M., & Kweon, I. S. (2023). Text-to-image diffusion model in generative ai: A survey. arXiv preprint arXiv:2303.07909.

[6] Harizi, R., Walha, R., Drira, F., & Zaied, M. (2022). Convolutional neural network with joint stepwise character/word modeling based system for scene text recognition. Multimedia Tools and Applications, 1-16.

[7] Gaafar, A. S., Dahr, J. M., & Hamoud, A. K. (2022). Comparative Analysis of Performance of Deep Learning Classification Approach based on LSTM-RNN for Textual and Image Datasets. Informatica, 46(5).

[8] Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., ... & Jiang, Y. G. (2022). Svtr: Scene text recognition with a single visual model. arXiv preprint arXiv:2205.00159.

[9] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, ‘‘ASTER: An attentional scene text recognizer with flexible rectification,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[10] C. Luo, L. Jin, and Z. Sun, ‘‘MORAN: A multi-object rectified attention network for scene text recognition,’’ Pattern Recognit., vol. 90, pp. 109–118, Jun. 2019.

[11] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, and X. Bai, ‘‘Symmetry-constrained rectification network for scene text recognition,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9147–9156

[12] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, ‘‘What is wrong with scene text recognition model comparisons? Dataset and model analysis,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4715–4723.

[13] Zhang, R., Chang, S., Wei, Z., Zhang, Y., Huang, S., & Feng, Z. (2022). Modulation classification of active attacks in internet of things: Lightweight mcbldn with spatial transformer network. IEEE Internet of Things Journal, 9(19), 19132-19146.

[14] C. Luo, Q. Lin, Y. Liu, L. Jin, and C. Shen, ‘‘Separating content from style using adversarial learning for recognizing text in the wild,’’ Int. J. Comput. Vis., vol. 129, no. 4, pp. 960–976, Apr. 2021.

[15] Krichen, M. (2023). Convolutional neural networks: A survey. Computers, 12(8), 151.

[16] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, ‘‘Decoupled attention network for text recognition,’’ in Proc. AAAI, 2020, pp. 12216–12224.

[17] H. Li, P. Wang, C. Shen, and G. Zhang, ‘‘Show, attend and read: A simple and strong baseline for irregular text recognition,’’ in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 8610–8617.

[18] Bouraoui, A., Jamoussi, S., & Hamadou, A. B. (2022). A comprehensive review of deep learning for natural language processing. International Journal of Data Mining, Modelling and Management, 14(2), 149-182. trends for corporate foresight,” *J. Bus. Econ.*, vol. 88, no. 5, pp. 643–687, 2018.

[19] S. Goria, “The search for and identification of routine signals as a contribution to creative competitive intelligence,” *Intell. J.*, no. 3, pp. 1–12, 2013.

[20] H. M. Alzoubi, M. In’airat, and G. Ahmed, “Investigating the impact of total quality management practices and Six Sigma processes to enhance the quality and reduce the cost of quality: the case of Dubai,” *Int. J. Bus. Excell.*, vol. 27, no. 1, pp. 94–109, 2022, doi: 10.1504/IJBEX.2022.123036.

[21] V. Rieuf, C. Bouchard, and A. Aoussat, “Immersive moodboards, a comparative study of industrial design inspiration material,” *J. Des. Res.*, vol. 13, no. 1, pp. 78–106, 2015.

[22] K. Kohn, “Idea generation in new product development through business environmental scanning: the case of XCar,” *Mark. Intell. \& Plan.*, 2005.

[23] A. Gordon, R. Rohrbeck, and J. O. Schwarz, “Escaping the" faster horses" trap: bridging strategic foresight and design-based innovation,” *Technol. Innov. Manag. Rev.*, vol. 9, no. 8, pp. 30–42, 2019.