algorithms

MDPI

*Article*

# ArCAR: A Novel Deep Learning Computer-Aided Recognition for Character-Level Arabic Text Representation and Recognition

**Abdullah Y. Muaad [1,2], Hanumanthappa Jayappa [1], Mugahed A. Al-antari [2,3,*] and Sungyoung Lee [3,*]**

[1] Department of Studies in Computer Science, Mysore University, Manasagangothri, Mysore 570006, India; Abdullahmuaad9@gmail.com (A.Y.M.); hanumsbe@gmail.com (H.J.)
[2] Sana'a Community College, Sana'a 5695, Yemen
[3] Department of Computer Science and Engineering, College of Software, Kyung Hee University, Suwon-si 17104, Gyeonggi-do, Korea
* Correspondence: en.mualshz@khu.ac.kr (M.A.A.-a.); sylee@oslab.khu.ac.kr (S.L.)

**Abstract:** Arabic text classification is a process to simultaneously categorize the different contextual Arabic contents into a proper category. In this paper, a novel deep learning Arabic text computer-aided recognition (ArCAR) is proposed to represent and recognize Arabic text at the character level. The input Arabic text is quantized in the form of 1D vectors for each Arabic character to represent a 2D array for the ArCAR system. The ArCAR system is validated over 5-fold cross-validation tests for two applications: Arabic text document classification and Arabic sentiment analysis. For document classification, the ArCAR system achieves the best performance using the Alarabiya-balance dataset in terms of overall accuracy, recall, precision, and F1-score by 97.76%, 94.08%, 94.16%, and 94.09%, respectively. Meanwhile, the ArCAR performs well for Arabic sentiment analysis, achieving the best performance using the hotel Arabic reviews dataset (HARD) balance dataset in terms of overall accuracy and F1-score by 93.58% and 93.23%, respectively. The proposed ArCAR seems to provide a practical solution for accurate Arabic text representation, understanding, and classification.

**Keywords:** natural language processing (NLP); deep convolutional neural network; Arabic text recognition; Arabic sentiment analysis; Arabic text computer-aided recognition (ArCAR)

check for updates

**Citation:** Muaad, A.Y.; Jayappa, H.; Al-antari, M.A.; Lee, S. ArCAR: A Novel Deep Learning Computer-Aided Recognition for Character-Level Arabic Text Representation and Recognition. *Algorithms* **2021**, *14*, 216. https://doi.org/10.3390/a14070216

Academic Editor: Francesc Pozo

Received: 26 May 2021
Accepted: 14 July 2021
Published: 16 July 2021

## 1. Introduction

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with computers and human language interactions. The goal of NLP is to process textual contents and extract the most useful information for effective real-life decisions. Indeed, text mining problems have gained much attention and have become a vital research area because of the boom in textual applications such as document recognition, social networking gates, or text identification from images or paintings [1]. Arabic text analytics are extremely important to facilitate in real life in terms of Arabic documents text classification [2], information retrieval [3], translation [4], Arabic sentiment analysis [5], spam detection [6], and news categorization [7]. The Arabic language is one of the six most global languages and is considered an official language for 26 states in the Arab world, especially in the Middle East (i.e., Arab countries) [8]. The Arabic language and its different dialects are spoken by more than 447 million native speakers [9]. The Arabic language is a semantic language that first emerged in the 1st to 4th centuries [10]. It can be categorized into three sublanguages: modern standard Arabic (MSA), colloquial or dialectal Arabic, and classical Arabic [11]. The modern standard Arabic is the official language used to varying degrees in workplaces, government, media, and newspapers. It is widely taught in schools, colleges, and universities [10]. The colloquial Arabic language varies among Arabic countries and geographical regions, whereas the classical Arabic language survives in religious scriptures and old Arabic poetry [12]. However, MSA, which is used for this study, could be understood and used by all Arabic natives. The MSA official language has 28 alphabet letters, all of which are consonants except three, which are long vowels: “أ/Alif”, “و/Waw”, and “ي/Ya” [8]. The Arabic words and sentences are written from right to left. Moreover, most Arabic alphabet letters have multiple written forms depending on their positions in the Arabic word. For example, the Arabic letter “ض/dh” could be written in different forms: ض/end of word (disconnected), ـض/end of word (connected)”, ـضـ/middle of word, or ضـ/beginning of word. In addition, diacritical marks (i.e., short vowels) highly contribute to Arabic phonology, altering the shape of the characters. For example, different combination forms of letters ب/Ba could be generated as بَ, بِ, بُ, بّ, بْ, بً, بٍ, and بٌ. This different nature of the Arabic alphabets makes Arabic word embedding and representation a major challenge [13]. Indeed, the Arabic language always faces many challenges, such as stemming, dialects, phonology, orthography, and morphology. The key is to represent the Arabic text in such a way that minimizes such difficulties [14]. Previous research studies for Arabic text classification have used conventional representation techniques such as bag-of-words and word embedding [15]. In [16], Al-Samadi et al. presented a supervised machine learning approach for aspect-based sentiment analysis (ABSA) to classify hotels’ Arabic reviews. They used different ML classifiers, including naïve Bayes, Bayesian networks, decision trees, k-nearest neighbors (KNNs), and support vector machines (SVMs) using different types of representation such as TF-IDF and N-gram. In [14], Bounhas et al. built a morpho-semantic knowledge graph from Arabic vocalized corpora. These corpora are transformed into rich graph-based representations to store for morphological and Arabic semantic knowledge. In fact, such techniques use classical classification approaches that start with engineering feature extraction, prominent feature selection, and then classification [17]. Recently, the dominant approaches for NLP tasks have been recurrent neural networks (RNNs), in particular LSTMs and convolutional neural networks (CNNs) [2,18–20]. Deep learning architectures have recently been proposed to derive a huge number of text deep features for better performance in different fields: text recognition [18,21,22], medical image diagnosis [23–27], and other computer vision techniques [27,28]. However, deep learning techniques have proven their capabilities for different recognition tasks. Such algorithms require considerable workload and attention and user intervention in each stage. Meanwhile, they require difficult preprocessing algorithms to prepare the Arabic text features. To date, almost all these representation techniques of text classification depend on words rather than character encoding. Simple statistics of some ordered word combinations (e.g., n-grams) usually perform text recognition in a better way [29]. As we mentioned above, researchers who worked in Arabic classification still have some limitations:

- First, the approach seeks to minimize the Arabic characters’ normalization by replacing the extended Arabic alphabets that have similar shapes with the basic one but with slightly different meanings such as أ, إ, and آ with ا, alphabet letter ة with ه, or letter ي with ى. Thus, the normalization process will directly affect the contextual meaning of similar shape words; for example, كرة means “football”, كره means “hatred”, علي means “the name of Ali”, على means “on the thing”, فار means “overflowed”, and فأر means “mouse”. In our character-level encoding methodology, we consider the different meanings of all of these characters by encoding them independently. This is to maintain the basic meaning of the different Arabic sentences.
- Second, to avoid the stemming problem, we need to understand the word’s root during the embedding process. The stemming problem is considered a major challenge for the Arabic language [30].
- Third, we also consider and encode stop words (i.e., من, في, إلى, على, etc.) as independent important words to keep the sentence meaning correct. The classical techniques suffered from understanding the stop words, and they removed them during the

embedding process to reduce the textual feature dimensionality, but it affected the understanding accuracy of the Arabic sentences.

- Fourth, we solve the problem of alphabet position dependence, which means that the encoding of the Arabic alphabets, based on their positions in a single word, should be different (e.g., at the beginning سـ, middle ـسـ, or at the end س, or ـس).

Such difficulties inspired us to employ the deep learning Arabic text computer-aided recognition (ArCAR) system based on the character level instead of the word or sentence level to easily understand and recognize Arabic for documents as well as Arabic sentiment analysis. The contributions of this work are summarized as follows:

- The quantization method is used to represent the Arabic text based on the character level.
- We have designed a new deep learning ArCAR system for Arabic text classification based on the capability of the deep convolutional network. The key idea of our system is to eliminate the need for Arabic text pre-processing which is very challenging and to achieve reasonable classification results.
- The deep learning ArCAR system is proposed to represent and recognize Arabic text in character level for two applications: (1) Arabic text documents recognition and (2) Arabic sentiment analysis.
- Finally, we conduct experiments and show the effectiveness of the proposed models for representation and classification to achieve excellent performance in solving many problems such as text recognition at the character level, which is better than at the word level to solve some problems such as out-of-vocabulary problems.

This will assist with fast Arabic language understanding or even solving Arabic text recognition problems. To our knowledge, this is the first study to encode all Arabic characters with number and necessary symbols that represent and classify Arabic textual contents considering many characteristics of Arabic text.

## 2. Related Works

The research on Arabic text classification systems is still very rare compared to research on English text classification. For English text classification, the authors in [31] examined and presented the latest research work methodologies and tools for multilingual sentiment analysis. They discussed the current difficulties and challenges in the area of AI-based text representation and classification. Moreover, some recommendations are also raised for future directions to especially deal with languages with scarce resources. In [32], Cambria et al. presented the affective computing and sentiment analysis for the opinion analysis based on emotion expression. In contrast, Arabic text classification still has more challenges and represents a hot research topic. Based on our research, there are multiple steps to address the problem of automatic text categorization Therefore, this literature review will cover two main subsections: Arabic text representation and Arabic text classification.

### *2.1. Arabic Text Representation (ATR)*

In fact, Arabic texts comprise unstructured data in the same way as English texts, and to be understandable for machine learning algorithms, the text must be transformed and represented by numerical values. The text can be represented in different ways using the bag-of-words (BOW) assumption or n-grams and term frequency–inverse document frequency (TF-IDF) [33]. Although these methods have shown good textual classification results, they lose the order of Arabic words, and they have limitations in capturing semantic meaning [34–36]. Guru et al. proposed a new representation approach for Arabic text called Term Class Weight-Inverse Class Frequency (TCW-ICF). Their representation is used to extract the most promising features from Arabic texts [37]. In [36], Etaiwi et al. presented a graph-based semantic representation model for Arabic text classification. Their proposed model achieved improvements of 8.60%, 30.20%, 5.30%, and 16.20% in terms of overall accuracy, sensitivity, precision, and F1-score, respectively. In contrast, other embedding techniques of Word2Vec and Glo2Vec have recently been used to represent conceptual text using deep learning approaches at the word level. Such representation is good for English text because several flexible preprocessing algorithms are available to improve the text representations compared with the structure of Arabic text [16]. In 2018, Abdul-Aziz et al. constructed Word2Vec models from a large Arabic corpus. They have worked with different ML algorithms and convolutional neural networks (CNN) for Arabic text classification. Although they used the feature selections technique to reduce the input feature dimensionality for better classification performance, some limitations \ still exist for pre-processing, removing numbers, and normalization and did not handle negation of words [38]. In 2018, Boukil et al. improved Arabic microblog retrieval with distributed representations for Arabic text classification [39]. In 2020, Almuzaini et al. proposed a method that combined document (doc2vec) embedding representation and sense disambiguation to enhance Arabic text representation. Then, they conducted experiments using the OSAC corpus dataset. They achieved an overall text classification accuracy of 90% in terms of F-measure [40]. In 2014, Y. Kim [41] proposed the first deep learning model consisting of a shallower convolutional neural network with only one convolutional layer based on the word level. Indeed, the Arabic language has many challenges for Arabic text representation [16]. The aim of this study is to represent Arabic text at the character level instead of at the word level. This is due to the scarcity of Arabic text preprocessing algorithms and to avoid the difficulties and limitations mentioned above. Arabic character-level representation is useful for deep learning ConvNet (i.e., CNN) architectures since they do not suffer from such limitations and have recently shown promising classification results for various NLP applications [42]. Moreover, ConvNet could be directly applied to distributed or discrete word embedding without pre-knowledge on the syntactic or semantic structures of that word. This is the key for developing a single reliable computer-aided recognition (CAR) system for different languages. Language characters always constitute a necessary textual construct regardless of the ability of word segmentation. Thus, working at the character level has the advantage of the ability to naturally learn abnormal character combinations of misspellings and emoticons. In 2020, Oueslati et al. presented that deep CNN was used for Arabic sentiment analysis text (SA). They represented Arabic text for sentiment analysis using character level features. This work still has some limitations such as that not all characters and numbers in Arabic texts are used, which will create misunderstanding for Arabic text [43]. We have seen in the literature review that the most commonly existing methods for representations of Arabic text categorization still use classical text representations such as the bag-of-words. Indeed, these methods are still suffering from the lack of semantics and high dimensionality of their feature space. In addition, they require complex preprocessing due to the complex morphology and nature of the Arabic language. Thus, we need to propose Arabic text representation techniques to avoid such limitations of normalization and stemming to achieve better accuracy for the Arabic text classification. On the other hand, Arabic language is more complex than other languages, and no efficient algorithms are available for English so far. For this reason, we are very excited to find a better choice for Arabic text representation to solve such difficulties.

### 2.2. *Arabic Text Classification (ATC)*

Arabic text classification is the most important phase for categorizing the different contextual Arabic contents into a proper category. Many machine learning algorithms have been proposed for Arabic text categorization, for example. In [34], El Kourdi et al. presented a Naïve Bayes classifier to classify Arabic text documents into five classes. They used TF-IDF to represent Arabic text. Boukil et al. proposed a combination method of term frequency–inverse document frequency (TF-IDF) using the capability of CNN to classify Arabic text from a large dataset [39]. In [44], Kim et al. presented a simple hybrid model for CNN and LSTM to classify English text based on character-level inputs. However, output predictions are still made based on the word level. Their evaluation results are reported using six different datasets and presented competitive and promising achievements. In [45], Romeo et al. addressed the problem of question ranking by addressing the task with machine learning algorithms for Arabic text representations. They designed an LSTM classifier to identify the most proper text segmentations in questions in a binary classification problem. This is to select the meaningful text representation and to reduce the noise as well as the computational cost. Then, they proposed a hybrid model using tree kernels built on the top of constituency parse trees. This model was first built by Farasa for Arabic word embedding based on supervised neural networks. In [46], Alayba et al. constructed Word2Vec models from a large Arabic sentiment corpus that collected over ten Arab countries. They applied different ML algorithms and convolutional neural networks with feature selection algorithms to reduce the derived feature dimensionality. The achieved accuracy was between 91.0% and 95.0% for Arabic sentiment classification using a health sentiment dataset. In [47], Al-Taani et al. used the fuzzy C-means (FCM) classifier to enhance the performance of Arabic document classification. For feature dimensionality reduction, they used singular value decomposition (SVD). Due to the problem of Arabic root words, in which one word has many possible meanings and is subject to mistakes, they used FCM to solve this issue. They achieved overall performance in terms of precision, recall, and F-measure by 60.16%, 62.66%, and 61.18%, respectively. In 2020, Elfaik et al. used the Arabic text representation based on the word level and investigated the bidirectional LSTM network (BiLSTM) to enhance the Arabic sentiment analysis [48]. They applied the forward–backward technique to encapsulate contextual information from Arabic feature sequences. They achieved the overall accuracy in terms of F1-measure by 79.41% using the LABR dataset. El-Alami et al. (2020) proposed an Arabic text categorization method using Arabic WordNet based on the bag-of-concepts and deep Auto-encoder to represent the Arabic text by eliminating the explicit knowledge that contains semantic vocabularies [49]. In 2020, Elzayady et al. proposed a hybrid model of CNN and RNN to extract the local features using CNN and then classify the Arabic sentiment via RNN [50]. In [51], Zhang et al. proposed the first deep learning character-level ConvNet for English text classification. They used two ConvNets with six convolutional layers with kernel sizes of 3 and 7, three simple max-pooling layers, and three fully connected layers. To verify their model, they used eight large-scale datasets, and they achieved the lowest testing errors for all datasets compared with the traditional methods of bag-of-means, n-grams TF-IDF, LSTM, word2vec, and lookup table ConvNet. In [18], Conneau et al. proposed a deep learning character-level architecture called a very deep CNN (VDCNN) for English text classification. They investigated the depth effectiveness of their deep learning model. They used four different structure depths via 9, 17, 29, and 49 convolutional layers with 4 max-pooling layers for all different structures. Data preprocessing and augmentation were not used in their scenario. To assess their model, eight large-scale datasets (same as in Zhang) [51] are used, and their very deep model (i.e., 29 convolutional layer depths) outperforms the ConvNet proposed by Zhang [51]. They found that the text classification performance was slightly increased when their proposed model became deeper (i.e., 29 convolutional layers). After that, they noticed that the performance was again decreased with 49 convolutional layers. In [38], Duque et al. proposed the modified version of the VDCNN, which was presented by Conneau et al. [18] and called it a squeezed very deep CNN (SVDCNN) for English text classification. Basically, their idea was to reduce the number of standard convolutional blocks used in VDCNN. In particular, the method reduced the trainable parameters and minimized the need for a high storage capacity. To do that, they modified the standard convolutional blocks of VDCNN by using temporal depthwise separable convolutions (TDSCs). Meanwhile, since 90% of the trainable parameters in a specific deep learning model are always created due to the number of dense layers [25,26], they replaced them with global average pooling (GAP) layers. In 2020, Daif et al. proposed the first deep learning model called CE-CLCNN for Arabic documents classification [18]. The deep learning CE-CLCNN model consists of two parts: a character autoencoder to encode the image-based character embedding and a character-level CNN (CLCNN) for classification purposes. It encodes each Arabic character or alphabet as a 2D image. Then, the Arabic text is represented as an array of character-based images. To handle the class imbalance problem (i.e., long-tailed data distribution problem), they used a class-balanced loss function. To evaluate their model, they created their own two datasets called the Arabic Wikipedia title (AWT) dataset and the Arabic poetry dataset (APD). The proposed model showed much classification improvement against the classical SVM machine learning classifiers by 21.23% and 4.02% in terms of the micro F1-score for the ADP and AWT datasets, respectively. In 2020, Daif et al. proposed the first deep learning structure via image-based character for Arabic documents classification, which was called AraDIC [20]. In this approach, they represented each Arabic character or alphabet as a 2D image. The proposed deep learning AraDIC consists of a 2D image-based character encoder and classifier. The character encoder is a simple CNN that consists of three 2D convolutional layers, two max-pooling layers, and two dense layers. In addition, the classifier is also a simple character-level CNN that consists of four 1D convolutional layers, two max-pooling layers, and two dense layers. Batch normalization and the ReLU activation function are used after each convolutional and dense layer. To avoid the imbalance problem, they trained their model end-to-end utilizing the weighted-class loss function. To evaluate their proposed model, they used both AWT and APD datasets. The proposed AraDIC outperforms other classical and deep learning baseline text classification techniques by 12.29% and 23.05% in terms of micro- and macro-F1-scores, respectively. In [2], Ameur et al. proposed a combined deep learning model of CNN and RNN for Arabic text documents categorization using static, dynamic, and fine-tuned word embedding. A deep learning CNN model is used to automatically learn the most meaningful representations from the space of Arabic word embedding. The proposed CNN consists of three key components: the convolutional layer, pooling layer, and fully connected or dense layer. To produce a full summary of the input Arabic text, a stacked set of bidirectional gated recurrent units (Bi-GRUs) was used. Then, multiple dense layers were utilized to finally recognize the input Arabic text into the most likely category. They evaluated their proposed deep learning model using an open source Arabic corpora (OSAC) dataset. Comparing the performance with the individual CNN and RNN models, their proposed hybridization model helped to improve the overall performance of Arabic text classification. Such methodologies still have limitations for Arabic alphabet position-dependent problems. The Arabic alphabet letters' shape and figure always depend on their position at the beginning, middle, or end of the word.

Indeed, such interesting studies for text classification inspired us to use the promising functionality of deep learning methodology to improve Arabic text classification based on the character level instead of the word level [45]. This is to tackle the complexity of the morphological analysis and limited preprocessing techniques for Arabic textual contents as well as to produce a considerably flexible and smart model to classify any Arabic text contents for documents categorization or even for sentiment analysis classification.

## 3. Materials and Methods

The proposed deep learning ArCAR framework for Arabic text classification based on the character level is presented in Figure 1. Our proposed model is applicable for both Arabic document recognition and Arabic sentiment analysis.

### *3.1. Dataset*

The initial requirement to develop any deep learning CAR systems for Arabic text recognition is to find benchmark Arabic datasets. Unfortunately, very rare and limited Arabic text datasets are publicly available. In this study, we use thirteen different datasets to perform our experiments for two applications: Arabic documents classification and sentiment analysis. For both applications, we use balanced and unbalanced datasets to show the reliability and feasibility of our proposed ArCAR system. In the following sections, the dataset details are explained in detail.

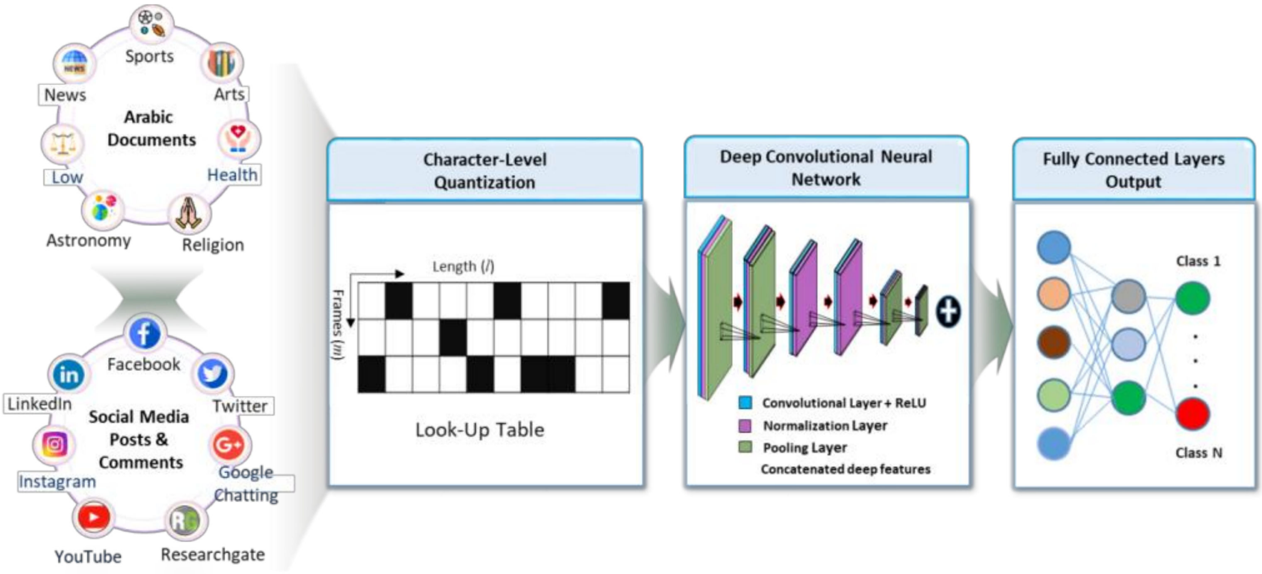


**Figure 1.** Schematic diagram of the proposed Arabic text computer-aided recognition (ArCAR) system.

#### 3.1.1. Arabic Documents Texts Datasets

To validate the classification performance of the proposed ArCAR system for Arabic text documents, we use nine different datasets: AlKhaleej, Akhbarona-balance, Akhbarona-unbalance, Alarabiya-balance, Alarabiya-unbalance, BBC Arabic corpus, CNN Arabic corpus, Open Source Arabic Corpus (OSAC), and Arabic Influencer Twitter Dataset (AITD).

##### SANAD Dataset

The AlKhaleej, Akhbarona, and Alarabiya datasets were extracted from a large database called a single-label Arabic news article dataset (SANAD) [22]. The SANAD dataset is a large collection of Arabic news articles and can be used for Arabic NLP tasks, such as single-label Arabic text classification. The dataset is publically available at https://data.mendeley.com/datasets/57zpx667y9/1 (accessed on 15 July 2021) [22]. Table 1 shows the data distribution of the unbalanced SANAD datasets used in this study for each class.

**Table 1.** SANAD Dataset Distribution for each Class: AlKhaleej, Akhbarona, and Alarabiya.

| # | Class Type | AlKhaleej | Akhbarona | Alarabiya |
|---|---|---|---|---|
| 1 | Finance | 6500 | 9280 | 30,076 |
| 2 | Sports | 6500 | 15,755 | 23,058 |
| 3 | Culture | 6500 | 6746 | 5619 |
| 4 | Technology | 6500 | 12,199 | 4410 |
| 5 | Politics | 6500 | 13,979 | 4368 |
| 6 | Medical | 6500 | 12,947 | 3715 |
| 7 | Religion | 6500 | 7552 | - |

Moreover, the SANAD database has a balanced sub-dataset for AlKhaleej, Akhbarona, and Alarabiya [22]. The balanced datasets of AlKhaleej and Akhbarona collected and categorized seven different classes or categories: religion, finance, sports, culture, technology, politics, and medical. These datasets involve 6500 and 6700 Arabic text articles or documents for AlKhaleej and Akhbarona, respectively. Fortunately, as shown in Table 1, AlKhaleej has the same number of documents (i.e., 6500 Arabic articles) for each class for the balanced and unbalanced datasets. In addition, the articles of the AlKhaleej dataset were collected from the AlKhaleej news portal from 2008 until 2018, while the articles of the Akhbarona dataset were collected from the Akhbarona news portal from January 2011 until October 2018. Meanwhile, the Alarabiya balanced dataset has only five different categories:

politics, finance, medical, sports, and technology. Unfortunately, the Alarabiya-balance dataset does not have any Arabic documents for the culture class. For studying aspects of culture using Alarabiya-balanced and unbalanced datasets, we add the culture class by randomly selecting Arabic documents from the unbalanced dataset. This is to perform an acceptable performance comparison of the ArCAD system using the same classes for both balanced and unbalanced Alarabiya datasets. The Alarabiya-balance dataset has six classes: politics, finance, medical, sports, technology, and culture. In this study, each class of the Alarabiya-balance dataset involves 3700 Arabic text documents. However, the Alarabiya dataset was collected from the main website of Al-Arabiya (i.e., https://www.alarabiya.net (accessed on 15 July 2021)), which has two subdomains: "alhadath" and "aswaq".

On the other hand, the data distribution for unbalanced datasets, which belong to open source Arabic corpora such as BBC, CNN, and OSAC is explained in detail in the following sections.

BBC Arabic Corpus

This dataset was collected from the BBC website, https://www.bbc.com/arabic (accessed on 15 July 2021) and included 4763 Arabic text documents [52]. As shown in Table 2, Arabic text documents in this dataset are categorized into seven different classes (i.e., Middle East news, world news, business and economy, sports, international press, science and technology, and art and culture). In total, this dataset contains 1,860,786 (1.8 M) Arabic words and 106,733 district keywords after removing the stop words.

**Table 2.** Data Distribution per Class for BBC and CNN Arabic Corpuses.

| # | Class Type | BBC | CNN |
|---|---|---|---|
| 1 | Middle East News | 2356 | 1462 |
| 2 | World News | 1489 | 1010 |
| 3 | Business and Economy | 296 | 836 |
| 4 | Sports | 219 | 762 |
| 5 | International Press | 49 | - |
| 6 | Science and Technology | 232 | 526 |
| 7 | Entertainments | - | 474 |
| 8 | Art and Culture | 122 | - |

CNN Arabic Corpus

This dataset was collected from the CNN website, https://arabic.cnn.com (accessed on 15 July 2021) and included 5070 Arabic text documents [52]. As shortlisted in Table 2, each text document is categorized into six different classes (i.e., Middle East news, world news, business and economy, sports, science and technology, and entertainment). In total, this dataset contains 2,241,348 (2.2 M) Arabic words and 144,460 distinct keywords after stop word removal. This dataset is publically available at http://site.iugaza.edu.ps/msaad/osac-open-source-arabic-corpora (accessed on 15 July 2021) [52].

Open Source Arabic Corpus (OSAC)

This dataset was collected from multiple sources and websites to include 22,429 Arabic text documents [52]. As shown in Table 3, each text document is categorized into ten different classes (i.e., economy, history, education and family, religion, sports, health, astronomy, low, stories, and cooking recipes). This dataset contains 18,183,511 (18 M) Arabic words and 449,600 distinct keywords after removing the stop words. We can find this dataset at http://site.iugaza.edu.ps/msaad/osac-open-source-arabic-corpora (accessed on 15 July 2021) [52].

**Table 3.** OSAC Dataset Distribution For Each Class.

| # | Class Type | No. of Documents |
|---|---|---|
| 1 | Economy | 3102 |
| 2 | Religion | 3171 |
| 3 | Education and Family | 3608 |
| 4 | History | 3233 |
| 5 | Sports | 2419 |
| 6 | Health | 2296 |
| 7 | Astronomy | 557 |
| 8 | Low | 944 |
| 9 | Stories | 726 |
| 10 | Cooking Recipes | 2373 |

Arabic Influencer Twitter Dataset (AITD)

This dataset was generated by collecting different twists for 60 Arab influencers on Twitter [53]. Domain experts categorized this dataset into ten different classes, as shown in Table 4. A Twitter application programming interface (API) was used to collect the last 3200 tweets for each account. This dataset was labeled based on the aspect of the Twitter user profile. If the user profile belongs to the health category, all related tweets for that user are categorized to be in the health class. This dataset is available at this link https://github.com/shammur/Arabic_news_text_classification_datasets (accessed on 15 July 2021) [53].

**Table 4.** AIDT Dataset Distribution for Each Class.

| # | Class Type | No. of Documents |
|---|---|---|
| 1 | Spiritual | 29,554 |
| 2 | Human-Rights-Press-Freedom | 19,477 |
| 3 | Sports | 18,875 |
| 4 | Business and Economy | 12,270 |
| 5 | Health | 9456 |
| 6 | Politics | 9369 |
| 7 | Art | 6247 |
| 8 | Environment | 5010 |
| 9 | Science and Technology | 4936 |
| 10 | Education | 498 |

### 3.1.2. Arabic Sentiment Analysis Dataset

To validate and evaluate the proposed ArCAR system for Arabic sentiment analysis, we used two different datasets: book reviews in the Arabic dataset (BRAD2.0) and the hotel Arabic reviews dataset (HARD). These datasets have three different classes: negative, positive, and neutral reviews. Experts scored these articles on a scale of 1 to 5 stars, as in [54,55]. Negative reviews are scored by 1 or 2 stars, positive reviews are scored by 4 or 5 stars, and neutral reviews are scored by 3 stars. In this study, we used these datasets in the forms of binary-class and multiclass problems. The binary-class problem includes only two balanced classes: negative and positive reviews, while the multiclass problem involves all three classes, including the neutral reviews class.

Book Reviews in Arabic Dataset (BRAD2.0)

This dataset comprises 510,598 Arabic book reviews expressed in MSA as well as dialectal Arabic with three dialects: Egyptian, Levantine, and Gulf [54]. The reviews have been done for 4993 Arabic books authored by 2043 scientists. However, these reviews were collected from the website of GoodReads over two months, June and July 2016. This dataset was generated as an extension of the large Arabic book review (LABR) dataset. In the BRAD2.0 dataset, a clean-up preprocessing was performed to remove unnecessary punctuation marks and English characters. Indeed, this dataset involves 8% negative reviews, 79% positive reviews, and 12% neutral reviews. Meanwhile, this dataset has a balanced subset of 156,602 Arabic book reviews rated with 1 or 2 for the negative review class and 4 or 5 for the positive review class where the neutral reviews (i.e., rated by 3 stars) are ignored. To validate and verify the effectiveness of the proposed deep learning ArCAR system, we use both balanced and unbalanced subsets of this dataset. The data distribution for each review class in both the unbalanced and balanced subsets is shown in Table 5. This dataset is available at https://github.com/elnagara (accessed on 15 July 2021) [54].

**Table 5.** Data Distribution for both BRAD2.0 and HARD per Each Class.

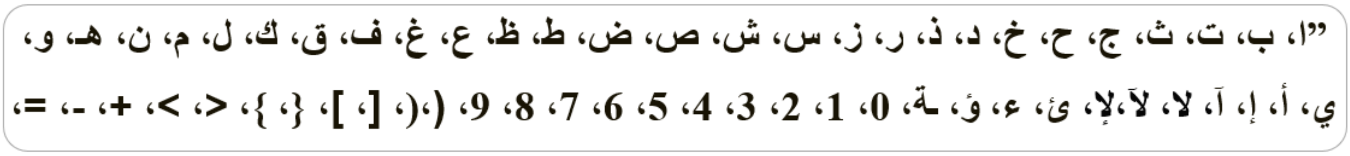
| **Class Type** | **BRAD2.0** | | **HARD** | |
|---|---|---|---|---|
| | **Multi-Class (Unbalance)** | **Binary-Class (Balance)** | **Multi-Class (Unbalance)** | **Binary-Class (Balance)** |
| Negative Reviews | 78,380 | 78,380 | 40,953 | 52,849 |
| Positive Reviews | 325,433 | 78,126 | 286,695 | 52,849 |
| Neutral | 106,785 | - | 81,912 | - |

Hotel Arabic Reviews Dataset (HARD)

The Hotel Arabic Reviews Dataset (HARD) comprises 409,562 hotel Arabic reviews in the standard Arabic language. This dataset was collected from the website of online accommodation booking (i.e., https://www.booking.com (accessed on 15 July 2021)). Each review contains the Arabic review text as well as the reviewer's rating on a scale of 1 to 5 stars [55]. Indeed, the HARD dataset has 13% negative reviews, 68% positive reviews, and 19% neutral reviews. Reviews with ratings of 1 or 2 stars were categorized as a negative review class, while reviews with rates of 4 or 5 stars were categorized as a positive review class. Meanwhile, Arabic hotel reviews with a rate of 3 stars are categorized as neutral reviews. In addition, the HARD dataset has a balanced subset and is also available online. The balanced HARD dataset consists of an almost equal number of reviews for two negative and positive classes, while neutral reviews are ignored. The dataset distribution for each class is presented in Table 5. This dataset is available at https://github.com/elnagara (accessed on 15 July 2021) [55].

### 3.2. *Arabic Character-Level Quantization*

The input of our proposed deep learning model is a sequence of encoded Arabic characters. The Arabic encoding process is achieved by prescribing the alphabet characters of size $m$ where each character is quantized via $m \times 1$ encoding. This means the quantization process starts by tokenizing each Arabic character to encode it in a one-shot representation, and each Arabic character is encoded by one vector with size of $m \times 1$. Then, the sequence of the Arabic characters is transformed into a similar sequence of $m$-sized vectors. Each vector has a length limited by $l$. In this study, we use $m$ = 70 Arabic characters, including 28 Arabic letters, 10 numbers, 6 brackets, and 16 other characters. The total characters are as follows,



”ا، ب، ت، ث، ج، ح، خ، د، ذ، ر، ز، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ك، ل، م، ن، هـ، و،
ي، أ، إ، آ، لا، لآ،لإ، ئ، ء، ؤ، ـة، 0، 1، 2، 3، 4، 5، 6، 7، 8، 9، )،(، ]، [، }، {، >، <، +، -، =،

Other characters that are not included in these alphabets are eliminated. As in [21], the characters are quantized in backward order to make it easy for dense layers associating correlations with the latest memory. Then, the input of the proposed deep learning model is a set of frames or vectors in a 2D shape with a size of $m \times l$. The vector length is fixed to 1014, and any character exceeding this length is automatically ignored. Characteristics with lengths less than 1014 are padded by zeros.

### *3.3. Data Preprocessing*

Arabic text preprocessing represents the first stage of any text classification workflow [45,56]. It is used to clean up and prepare the unstructured text datasets to improve the overall performance of the proposed ArCAR system. The nature of the Arabic language has more structural difficulties than English, where it needs many additional preprocessing efforts, such as stemming, normalization, and difficult morphology, and many roots may be recognized for a specific single word [57]. Such difficulties make the Arabic text representation truly a difficult task and impact the overall accuracy of the classification. To minimize such difficulties, we represent the Arabic text based on the character level instead of the word level or sentence level for both applications: Arabic documents text classification and Arabic sentiment analysis. Thus, stemming and normalization are not required, and this is a key to facilitate Arabic text preparation. The preprocessing starts by splitting each Arabic word into the original characters' forms. Then, a lookup table is generated as a 2D tensor of size $(f_0, l)$ that contains the embeddings of the $l$ characters, and $f_0$ could be represented as the RGB image dimension of the input text.

### *3.4. Data Preparation: Training, Validation, and Testing*

To fine-tune and assess the proposed ArCAR system, two different datasets are used; one for Arabic documents text classification as in Section 3.1.1 and another dataset for sentiment analysis as in Section 3.1.2. For both applications, the Arabic text files (i.e., documents or sentiments) for each class are randomly split into 70% in the training set, 10% in the validation set, and 20% in the testing set [24,25,27]. The trainable parameters (i.e., network weights and biases) of the proposed convolutional neural network are optimized via the training process using the training–validation sets. After that, the overall performance of the proposed deep learning model is evaluated utilizing only the evaluation set. In addition, the proposed model is assessed over five-fold tests for training, validation, and evaluation sets. These sets are generated via stratified partitioning ensuring an equal testing rate for each text document to prevent system bias. To build a reliable and feasible CAR system for Arabic text classification, it is important to use a k-fold cross-validation strategy, especially when the size of the dataset is not large enough for training purposes [23–25].

### *3.5. The Proposed ArCAR System*

The proposed deep learning ArCAR consists of six convolutional layers and two fully connected or dense layers, as shown in Figure 1. Convolutional layers (CONVs.) are used to derive the hierarchy of deep features, and dense layers are used as a classifier to find the most proper class probability for the specific problem and produce the final output of the proposed ArCAR system. Deriving multiple deep learning automatically from the 2D input is a key to using deep learning based on CNN [49]. Indeed, a deep CNN has a better capability to directly generate deep hierarchical features from the input raw input [25,26]. Meanwhile, a logistic regression soft max layer is used to represent the output classes with different nodes based on the number of classes for each individual problem. Moreover, local response batch normalization layers are used after each convolutional layer to improve the performance of the proposed CNN model. Batch normalization layers help to vanish the gradient problem by standardizing the output of the current convolutional layer. Additionally, it prevents the restrictions of the small value of the learning rate and then speeds up the training process. The ReLU activation function is used after each convolutional and dense layer because it has a stable and faster training saturation state than sigmoid or tanh. Thus, the deep learning ArCAR system with ReLU shows better performance and a faster training process. Table 6 shows the structure of the proposed ArCAR system for Arabic text recognition in detail.

**Table 6.** Detail Structure of the Proposed ArCAD System.

| # | Layer Type | Filter Size, Maps | Pooling |
|---|---|---|---|
| 1 | Input data: 70 × 1014 (2D) | - | - |
| 2 | CONV. 1 | 7 × 7, 64 | 2 |
| 3 | CONV. 2 | 5 × 5, 128 | 2 |
| 4 | CONV. 3 | 3 × 3, 256 | NA |
| 5 | CONV. 4 | 3 × 3, 256 | NA |
| 6 | CONV. 5 | 3 × 3, 256 | 2 |
| 7 | CONV. 6 | 3 × 3, 256 | 2 |
| 8 | Dense 1 | 1024 | - |
| 9 | Dense 2 | 1024 | - |
| 10 | Softmax (Output) | Based on the number of classes | - |

### *3.6. Experimental Settings*

For training, the Adam optimizer with an initial training rate of 0.001 and weight decay of 0.0005 are used. The number of mini-batch sizes is set to 24. Meanwhile, a dropout of 0.5 is used on both dense layers to speed up the learning process and to avoid the overfitting problem. For trainable parameter initialization, we use random initialization via unbiased Gaussian distributions with a standard deviation of 0.01 [26,28,58]. The selection of the trainable parameters and epochs for each dataset is mainly done based on the criteria of the system error-based trials and achieves the best performance. To prevent system bias during the learning process due to training imbalanced datasets, the following remedies are used. First, through each mini-batch, the training dataset is shuffled to ensure that each text is only used once per epoch [27]. Second, weighted balance cross-entropy is used as a loss function [23–25,59,60]. Third, to optimize the trainable parameters, training/validation sets are used, and the testing set is only used for evaluation purposes.

### *3.7. Evaluation Strategy*

For the quantitative evaluation of the proposed ArCAR system with each fold test, weighted objective metrics, including recall or sensitivity (SE), specificity (SP), overall accuracy (Az), F1-score, Matthews correlation coefficient (MCC), precision or positive predictive value (PPV), and negative predictive value (NPV), are used. To prevent having test sets that are unbalanced with regard to all classes, the weighted-class strategy is used. The criteria for all of these metrics are defined as follows:

$$\text{Recall/Sensitivity (SE)} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{1}$$

$$\text{Specificity (SP)} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \tag{2}$$

$$\text{F1} - score\ (\text{DICE}) = \frac{2{\cdot}\text{TP}}{2{\cdot}\text{TP}+\text{FP}+\text{FN}}, \tag{3}$$

$$\text{Overall accuracy (Az)} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{TN}+\text{FP}}, \tag{4}$$

$$\text{MMC} = \frac{\text{TP}{\cdot}\text{TN} - \text{FP}{\cdot}\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}, \tag{5}$$

$$\text{Precision/PPV} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{6}$$

$$\text{NPV} = \frac{\text{TN}}{\text{TN}+\text{FN}}, \tag{7}$$

where TP, TN, FP, and FN are defined to represent the number of true positive, true negative, false positive, and false negative detections, respectively. To derive all of these parameters, a multidimensional confusion matrix is used. Finally, to avoid having test sets that are unbalanced with regard to all classes, we used the weighted-class strategy with each dataset to calculate the evaluation indices [27].

### *3.8. Implementation Environment*

To perform all experiments in this study, we used a PC with the following specifications: Intel R © Core(TM) i7-6850 K processor with 16 GB RAM, 3.360 GHz frequency, and one NVIDIA GeForce GTX 1080 GPU. The presented ArCAR system is implemented using Python 2.7.14 on the Ubuntu 16.04 operating system and back-end libraries of TensorFlow and Keras.

## **4. Results**

In this section, the evaluation results of the proposed ArCAR system for Arabic text recognition are presented for both applications of documents classification and sentiment analysis. For both applications, different datasets with multiple classes were used to test the reliability of the proposed ArCAR system. For document classification, the number of documents in each dataset varies from 5 k to 200 k, while the number of classes varies between 5 and 10 classes. For Arabic sentiment analysis, the number of posts or comments varies between 165 k and 500 k, while the dataset has two and three balance and unbalance classes, respectively. The evaluation results shown in this section are recorded as an average over 5-fold test for each dataset. Given a specific dataset, the same model with deep architecture and training settings is used to achieve reliable results and show the goal of the proposed ArCAR system.

### *4.1. Arabic Documents Text Recognition*

Table 7 shows the average evaluation results of the proposed ArCAR system for Arabic documents text classification. These results are derived as an average over 5-fold test using only the testing datasets. These results show the reliability and capability of the proposed ArCAR system, achieving promising evaluation results with different datasets. The overall classification performance using the AlKhaleej dataset was 92.64%, 98.28%, 97.47%, and 92.63% in terms of sensitivity, specificity, accuracy, and F1-score, respectively.

**Table 7.** Evaluation Result of the proposed ArCAD system for Arabic documents text classification as an average over 5-fold test.

| DATASET | SE | SP | Az | F1-Score | MCC | PPV | NPV |
|---|---|---|---|---|---|---|---|
| AlKhaleej balance | 92.64 | 98.28 | 97.47 | 92.63 | 91.55 | 92.75 | 98.28 |
| Akhbarona balance | 88.99 | 97.67 | 96.43 | 88.98 | 78.31 | 89.16 | 97.67 |
| Akhbarona unbalance | 88.08 | 97.93 | 96.68 | 88.28 | 86.55 | 88.60 | 97.94 |
| Alarabiya balance | 94.08 | 98.50 | 97.76 | 94.09 | 93.01 | 94.16 | 98.50 |
| Alarabiya unbalance | 83.80 | 95.88 | 94.43 | 76.87 | 74.40 | 73.01 | 96.69 |
| BBC | 69.02 | 85.44 | 81.63 | 69.62 | 57.43 | 71.18 | 83.43 |
| CNN | 74.72 | 94.58 | 91.56 | 75.43 | 70.72 | 77.46 | 84.86 |
| OSAC | 91.26 | 98.49 | 97.60 | 91.40 | 90.85 | 93.10 | 98.39 |
| AIDT | 90.15 | 97.11 | 96.59 | 90.17 | 88.58 | 90.73 | 98.18 |

Similar encouraging results were obtained using all the reset datasets. All evaluation metrics were derived using the averaged multiclass confusion matrix over a 5-fold test. Figures 2 and 3 show the averaged multiclass confusion matrices for Arabic documents text recognition over each dataset. The sum of each row in the confusion matrix represents the testing set for each class.

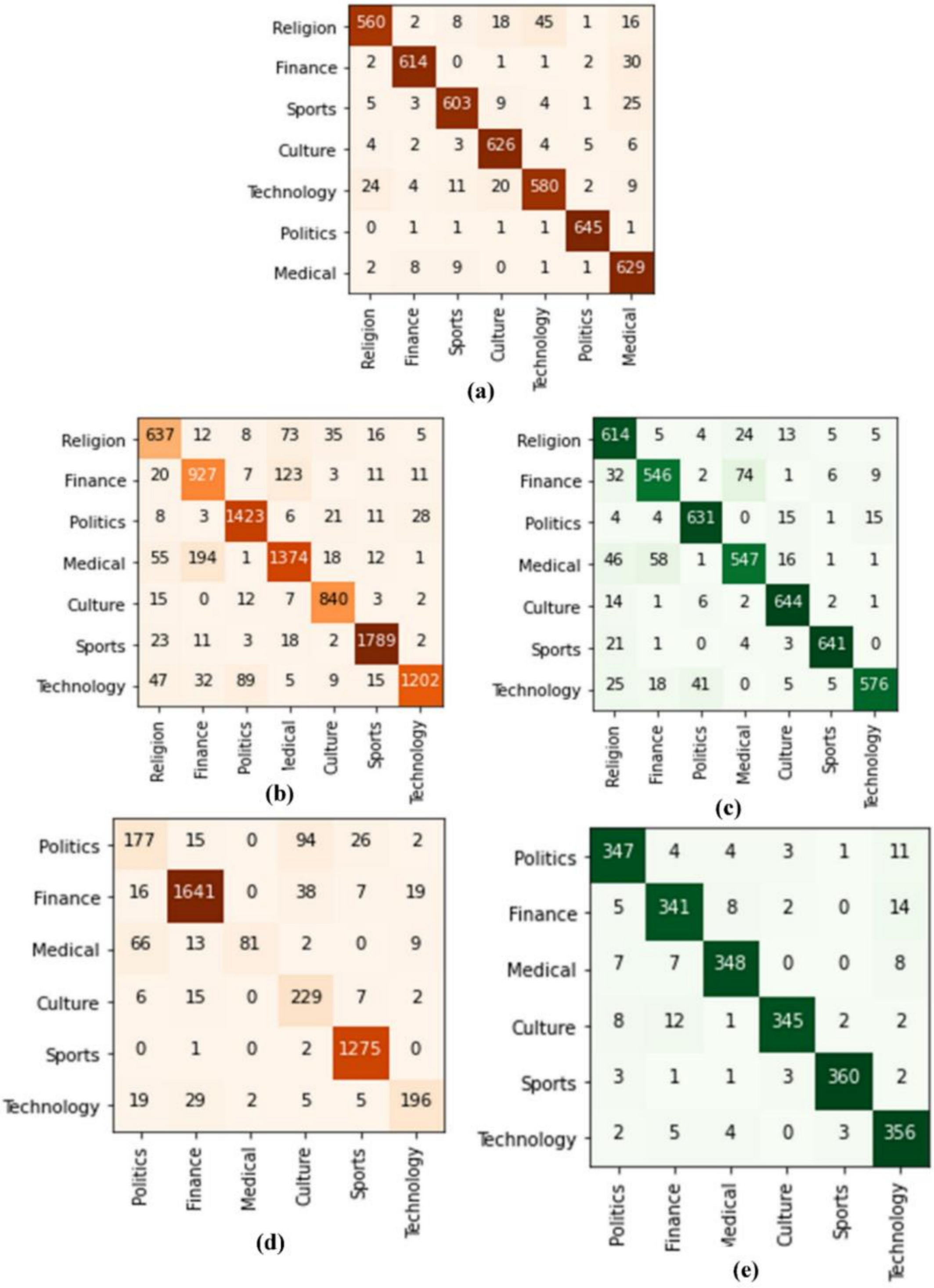


**Figure 2.** Averaged multiclass confusion matrices of Arabic document text recognition for different datasets with corresponding classes: (**a**) AlKhaleej, (**b**) Akhbarona-unbalance, (**c**) Akhbarona-balance, (**d**) Alarabiya-unbalance, and (**e**) Alarabiya-balance.

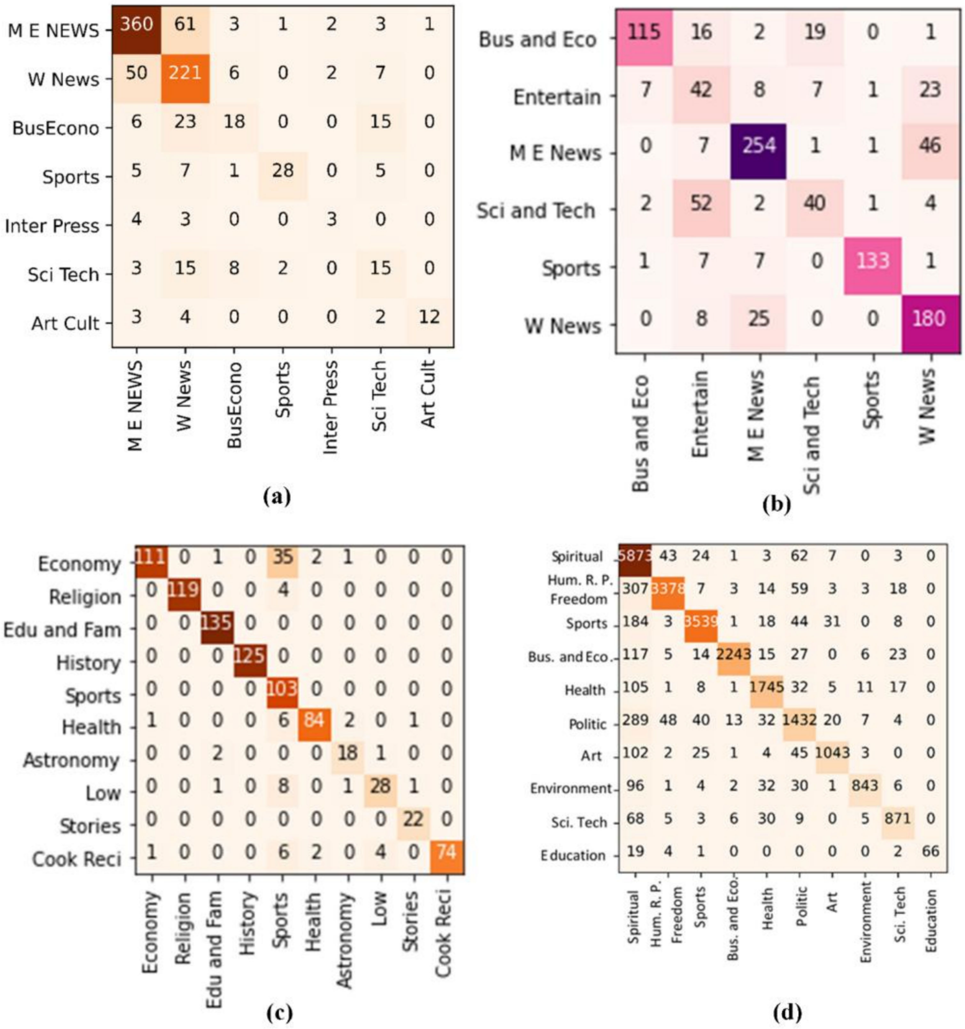


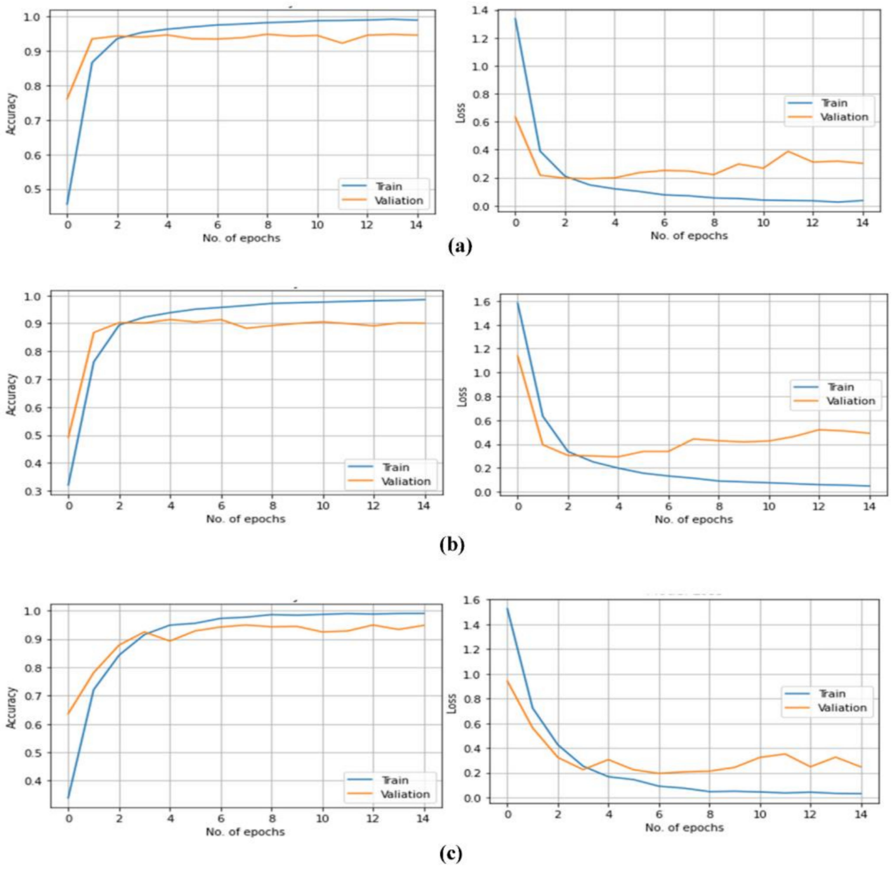
**Figure 3.** Averaged multiclass confusion matrices of Arabic documents text recognition for different datasets with corresponding classes: (**a**) BBC Arabic corpus, (**b**) CNN Arabic corpus, (**c**) OSAC, and (**d**) AIDT.

Moreover, the ArCAR system performance in terms of computation training and testing cost is recorded corresponding to each dataset and listed in Table 8. The computation time for deep learning models depends on the deep structure, learning settings (epochs, batch size, etc.), the size of the training set, and specifications of the PC. The deep learning structure affects the training and testing time prediction cost. This is because the number of trainable parameters is proportionally increased with the depth of the deep learning model, making the prediction cost high [18,19]. Thus, accurate and rapid predictions are required for more reliable and feasible text recognition systems. In this study, the proposed ArCAR system has a reasonable depth and achieves promising recognition results in terms of Arabic document recognition and Arabic sentiment analysis. The number of trainable parameters (weights and baises) of the proposed model are recorded to be 11.30 million. For training, an example of the training ArCAR system in terms of training–validation accuracy and loss function behaviors is shown in Figure 4. It is clearly shown that the ArCAR system is trained well, achieving good behaviors without any bias or overfitting. As shown in Table 8, the training and testing processing times are recorded based on the total number of training and testing datasets for all classes with respect to the specific dataset. The training processing time is recorded for each epoch. Indeed, the variation in the dataset sizes directly affects the required training and testing times. This means that the number of Arabic documents in the training set affects the required training time to finalize the learning processes and trainable parameter optimization. For testing, the required time is much less than the training time because the testing sets represent only 20% of the total size of the specific dataset. For example, the proposed ArCAR system requires 1.2 s to perform the recognition process of 4550 Arabic text documents from the AlKhaleej testing set, as shown in Table 8. Each class of the AlKhaleej testing set has 650 Arabic text documents, as shown in Figure 2a. Since the same deep learning ArCAR model is used, the processing testing time for each document is $2.64 \times 10^{-4}$ s.

**Table 8.** Computation Measurements for the Proposed Deep Learning ArCAR System with Different Datasets: Arabic Text Documents Recognition.

| Dataset | * Train Time/Epoch (s) | No. of Epochs | * Testing Time/Testing Set (s) |
|---|---|---|---|
| AlKhaleej | 36.89 | 14 | 1.2 |
| Akhbarona unbalance | 1065.096 | 10 | 7.651 |
| Akhbarona balance | 1134.466 | 14 | 3.611 |
| Alarabiya unbalance | 307.265 | 10 | 3.689 |
| Alarabiya balance | 556.531 | 14 | 1.706 |
| BBC | 8.10 | 20 | 0.22 |
| CNN | 44.76 | 25 | 0.4 |
| OSAC | 179.251 | 10 | 0.746 |
| AIDT | 109.67 | 10 | 2.90 |

* The time is computed considering all datasets from all classes.



(a)

(b)

(c)

**Figure 4.** Training-validation behavior of the proposed ArCAR system in terms of accuracy and loss functions with respect to the number of epochs using (**a**) AlKhaleej, (**b**) Akhbarona-balance, and (**c**) Alarabiya-balance.

### *4.2. Arabic Sentiment Analysis*

The evaluation results of the proposed ArCAR system for Arabic sentiment analysis are shown in Table 9. These results are derived as an average of the 5-fold test. As shown in Table 9, the proposed ArCAR system presented its reliability and capability for Arabic sentiment analysis. It is clearly shown that using the remedies of the unbalanced datasets, the proposed ArCAR system achieves similar evaluation results in terms of all evaluation indices. Using the balance (i.e., case of binary class) and unbalance (i.e., case of multiple class) HARD datasets, overall accuracies of 93.23% and 93.58% are achieved. However, the overall system accuracy using the binary BRAD dataset (i.e., balance set) is slightly better by 4.33% than that using the multiclass BRAD dataset (i.e., unbalance set). Similarly, all evaluation metrics are derived using the averaged multiclass confusion matrix over a 5-fold test, as shown in Figure 5.

**Table 9.** Evaluation Result of the proposed ArCAD system for Arabic Sentiment Analysis as an Average over 5-fold Test.

| Dataset | SE | SP | Az. | F1-Score | MCC | PPV | NPV |
|---|---|---|---|---|---|---|---|
| BRAD Binary-class | 81.44 | 81.44 | 81.46 | 81.45 | 62.92 | 81.48 | 81.48 |
| BRAD Multiclass | 68.35 | 71.61 | 77.13 | 67.25 | 40.87 | 66.77 | 75.00 |
| HARD Binary-class | 93.23 | 93.23 | 93.58 | 93.23 | 86.49 | 93.26 | 93.26 |
| HARD Multiclass | 80.98 | 94.35 | 93.23 | 81.63 | 76.34 | 82.33 | 95.00 |

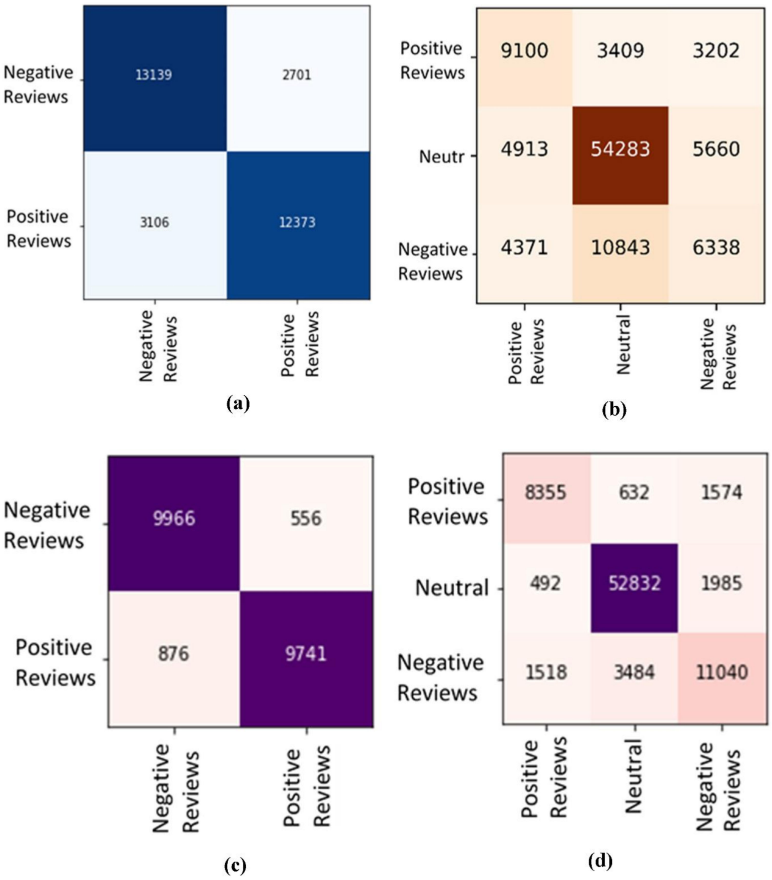


**Figure 5.** Averaged binary and multiclass confusion matrices of the Arabic sentiment analysis for (**a**) BRAD: binary-class, (**b**) BRAD: multiclass, (**c**) HARD: binary-class, and (**d**) HARD: multiclass.

Similarly, the training and testing processing times are proportionally affected by the number of training and testing Arabic reviews in the HARD and BRAD datasets, as presented in Table 10. For example, using the total number of 31,319 testing reviews in the balanced BRAD (i.e., binary class) testing set, the ArCAR system requires 3.3 s to complete the testing recognition process. Meanwhile, for the BRAD multiple classes problem (i.e., unbalance set) using the total number of 102,119 testing reviews, the ArCAR system requires 10.76 s. This means for one Arabic sentiment (i.e., review), the ArCAR system needs only $1.054 \times 10^{-4}$ s.

**Table 10.** Computation Measurements for the Proposed Deep Learning ArCAR System with Different Datasets: Arabic Sentiment Analysis.

| Dataset | * Train Time/Epoch (s) | No. of Epochs | * Testing Time/Epoch (s) |
|---|---|---|---|
| BRAD Binary-class | 1594.02 | 11 | 3.3 |
| BRAD Multiclass | 3511.13 | 10 | 10.76 |
| HARD Binary-class | 93.03 | 10 | 2.44 |
| HARD Multiclass | 648.03 | 12 | 9.83 |

* The time is computed considering all datasets from all classes.

## 5. Discussion

Deep learning computer-aided recognition based on convolutional networks has gained much research attention to improve the overall recognition performance with different NLP applications, medical imaging applications, computer vision approaches, etc. This is due to the capability of deep learning based on CNNs to extract a huge hierarchy of deep feature knowledge to learn about the low-level as well as high-level deep features end-to-end directly from the input data. Here, we employed the functionality of the deep learning CNN to represent and recognize Arabic text at the character level for two applications: Arabic text documents recognition and Arabic sentiment analysis. The idea of using the character level instead of the word level or the sentence level for Arabic text classification is inspired to avoid stemming and to minimize the preprocessing complex techniques, especially for the Arabic language. This is to maintain and represent the Arabic contextual text meaning correctly. In contrast to recent works on Arabic text classification, the proposed ArCAR system can efficiently handle Arabic text documents and Arabic sentiment at the character level, as presented in Tables 7 and 9.

### *5.1. Arabic Documents Text Recognition*

For the first application of Arabic documents text classification, the evaluation results show the reliability of the ArCAR system, achieving promising recognition of the best results using the Alarabiya-balance dataset with 97.76% and 94.09% in terms of overall accuracy and F1-score, respectively. Using the AlKhaleej and OSAC datasets, evaluation results of 97.47% and 97.60% in terms of overall accuracies are obtained, respectively. Meanwhile, the evaluation results using Akhbarona and AIDT depict overall accuracies of 96.68% and 96.59%, respectively. The lowest accuracy achieved by this system is recorded using the BBC dataset with 81.63%. As shown in Table 7 and Figures 2 and 3, the proposed ArCAR system shows its capability and reliability to handle different datasets regardless of the number of classes. For example, the ArCAR system performs well to achieve promising results using AIDT with ten different classes, achieving overall accuracy, F1-score, MCC, precision, and NPV results of 96.59%, 90.17%, 88.58%, 90.73%, and 98.18%, respectively. Similarly, the proposed ArCAR system achieves encouraging evaluation results using the OSAC dataset with ten classes. Using the OSAC dataset, it achieves an overall accuracy and F1-score of 97.60% and 91.40%, respectively. In addition to the promising overall accuracy achieved by the proposed ArCAR, a rapid recognition time is also required. As shown in Tables 7 and 8, the proposed ArCAR system presents the capability and feasibility of achieving encouraging and rapid recognition rates for Arabic text documents. As shown in

Table 8, the ArCAR system needs less than 0.264 msec (i.e., for AlKhaleej as an example) to complete the recognition process for each Arabic text document.

The proposed remedies of the ArCAR system due to the unbalanced datasets for training and testing help to train the deep learning model in a stable way. In addition, it helps to achieve more stable evaluation results for balanced and unbalanced datasets, as shown in Figure 6. Although some unbalanced training and testing remedies are used, the performance of the proposed ArCAR system is still better with balanced datasets. This is clearly shown with the evaluation F1-score index using the Alarabiya dataset, where it always measures the system evaluation regardless of the imbalance issue in the testing datasets. Unfortunately, other datasets of AlKhaleej, BBC, CNN, OSAC, and AIDT do not have balanced subsets to show the system performance. However, it is clearly shown that using the remedies of the unbalanced datasets, the proposed ArCAR system achieves similar evaluation results in terms of all evaluation indices. Using the balanced and unbalanced HARD datasets, overall accuracies of 93.23% and 93.58%, respectively, are achieved. However, the overall system accuracy using the balanced dataset is slightly better by 4.33% than that using the unbalanced dataset.

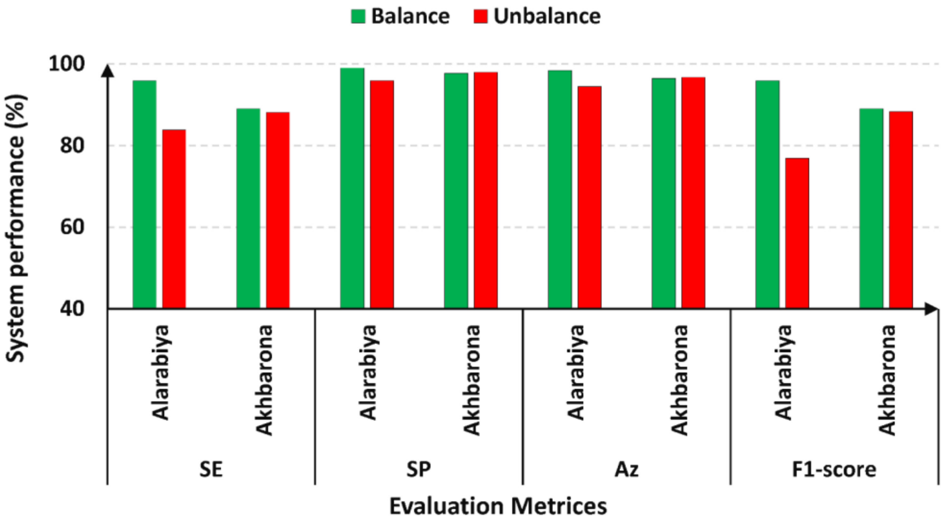


**Figure 6.** Evaluation performance comparison of the proposed ArCAR using the balanced and unbalanced Alarabiya and Akhbarona datasets.

The direct comparison results using the same datasets with different machine learning classifiers and different Arabic text representation (i.e., word-level and character-level) is presented as in Table 11. Due to the use of many datasets in our work, we compare only two datasets (i.e., unbalanced CNN and AlKhaleej). As is clearly shown in Table 11, our proposed ArCAR achieved comparable and better results compared with the traditional machine learning algorithms. For Arabic text representation, we have used bag-of-words (BOW), term frequency-inverse document frequency (TFIDF), and character-level representation.

To compare the proposed ArCAR system performance with the latest works as well as the conventional machine learning (CML) techniques, we summarized the comparison results regarding the evaluation metrics, as shown in Table 12. Compared with other recent studies and different methodologies, text recognition in Arabic documents at the character level achieves comparable and promising evaluation results with different datasets. This means that the proposed approach seems to be helpful for real practices of Arabic documents classification with a more accurate recognition rate. It is also shown that the proposed system outperforms CML techniques such as multinomial naïve Bayes, Logistic Regression and SVC, and linear SVC. In this study, we performed multinomial naïve Bayes to show a direct comparison using the same datasets, as presented in Table 12. For example, using the AlKhaleej dataset, the proposed ArCAR system outperforms other deep learning models of DL-CNN-GRU and machine learning models of multinomial naïve Bayes by 7.47% and 1.47% in terms of overall accuracies, respectively. Thus, the proposed system may help people understand and analyze the Arabic language in a sufficient and desirable way. This also approves the methodology of Arabic character-level representation and recognition.

**Table 11.** Comparison Evaluation Results using Different Conventional Machine Learning Classifiers with different Representation Techniques.

| **Classification** | **World-Level and Character-Level Representations** | | | | | |
|---|---|---|---|---|---|---|
| | **CNN Dataset (Unbalance)** | | | **AlKhaleej Dataset (Balance/Unbalance)** | | |
| | **Word-Level Representation** | | **Char Level Representation** | **Word-Level Representation** | | **Char Level Representation** |
| | **TFIDF** | **BOW** | | **TFIDF** | **BOW** | |
| MultinomialNB | 64.00 | 88.00 | - | 91.00 | 93.00 | - |
| BernoulliNB | 61.00 | 61.00 | | 87.00 | 85.00 | |
| LogisticRegression | 90.00 | 91.30 | | 94.00 | 96.00 | |
| SGDClassifier | 91.20 | 91.00 | | 94.00 | 95.00 | |
| Support Vector Classifier(SVC) | 90.00 | 90.00 | | 95.00 | 96.00 | |
| Linear SVC | 91.00 | 91.00 | | 94.00 | 96.00 | |
| Proposed ArCAR | - | - | 91.56 | - | - | 97.47 |

**Table 12.** Comparison Evaluation Results with the Latest Works for Arabic Text Documents Recognition.

| **Dataset** | **Method** | **Precision/PPV** | **Recall/SE** | **F1-Score** | **Accuracy** |
|---|---|---|---|---|---|
| AlKhaleej | DL-CNN-GRU<br>Elnagar et al. [9] | - | - | - | 96.0 |
| | Multinomial Naïve Bays<br>Our Conventional | 90.0 | 90.0 | 90.0 | 90.0 |
| | The proposed ArCAR System | 92.75 | 92.64 | 92.63 | 97.47 |
| Akhbaronabalance | DL-CNN-GRU<br>Elnagar et al. [9] | - | - | - | 94 |
| | The proposed ArCAR System | 89.16 | 88.99 | 88.98 | 96.43 |
| Alarabiyabalance | DL-CNN-GRU<br>Elnagar et al. [9] | - | - | - | 97.0 |
| | The proposed ArCAR System | 94.16 | 94.08 | 94.09 | 97.76 |
| BBC | Multinomial Naïve Bays<br>Our Conventional | 64.0 | 37.0 | 42.0 | 74.0 |
| | The proposed ArCAR System | 71.18 | 69.02 | 69.62 | 81.63 |
| CNN | Fuzzy C-mean and SVD<br>Kowsari et al. [15] | 60.0 | 61.0 | 62.0 | - |
| | Multinomial Naïve Bays<br>Our Conventional | 77.0 | 67.0 | 75.43 | 91.0 |
| | The proposed ArCAR System | 77.46 | 74.72 | 75.43 | 91.56 |
| OSAC | Multi-Layer Perceptron<br>Saad et al. [43] | 91.0 | 90.0 | 90.0 | - |
| | Multinomial Naïve Bays<br>Our Conventional | 97.0 | 88.0 | 89.0 | 87.0 |
| | The proposed ArCAR System | 93.10 | 91.26 | 91.40 | 97.62 |
| AIDT | BERT model<br>Chowdhury et al. [52] | 86 | - | - | - |
| | The proposed ArCAR System | 90.15 | 97.11 | 90.17 | 96.59 |

Abbreviations: DL-CNN-GRU: Deep Learning CNN with gated recurrent unit, SVD: singular value decomposition, BERT: Bidirectional Encoder Representations Transformers, and ArCAR: Arabic text computer-aided recognition.

### *5.2. Arabic Sentiment Analysis*

For this application, the proposed ArCAR system was evaluated using BRAD and HARD datasets for binary and multiclass problems. Fortunately, the BRAD and HARD datasets for the binary-class problem have balanced positive and negative review datasets. As shown in Table 9, the best performance is achieved using HARD-balance datasets (i.e., binary-class problem) with overall accuracy, recall, precision, and F1-score of 93.58%, 93.23%, 93.26%, and 93.23%, respectively. Using the BRAD dataset, the ArCAD systems achieved better performance for the binary-class problem by 4.33%, 14.2%, 13.09%, and 14.71% in terms of overall accuracy, F1-score, recall, and precision, respectively. Similarly, using the HARD-balance dataset, the ArCAR system achieves better performance in terms of recall, precision, and F1-score by 12.25%, 10.93%, and 11.60%, respectively. Despite the unbalanced remedies used for training and testing purposes, the ArCAR system always achieves better performance using a balanced dataset (i.e., binary-class problem). With the balanced datasets, the evaluation performance is more stable over all evaluation metrics. As depicted in Table 9 and Figure 5, the proposed ArCAR system shows its reliability and feasibility to handle the BRAD and HARD datasets with binary and multiclass problems with more than 500 thousand Arabic article reviews.

As listed in Tables 9 and 10, the ArCAR system achieves encouraging evaluation results for Arabic sentiment reviews and requires less than 0.105 msec for each Arabic article recognition. For computation time cost, ArCAR systems need less recognition time in the case of sentiment analysis compared with Arabic document texts. This is because the size of each Arabic text document is larger than that of a single Arabic review.

For performance comparison of the proposed ArCAR system performance with the latest work, the comparison results regarding the evaluation metrics are summarized in Table 13. The proposed ArCAR system for Arabic sentiment analysis using character-level representation shows promising comparable recognition results with the latest studies using different ML approaches [13,37]. For example, using the HARD balanced dataset, the proposed ArCAR system outperforms the recognition system using SVM presented in [54] by 15.23%. Meanwhile, the ArCAR system performs better evaluation performance than the random forest presented in [54] by 5.58% using the multiclass HARD dataset.

**Table 13.** Comparison Evaluation Results with the Latest Works for Arabic Sentiment Analysis.

| Dataset | Method | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|---|
| BRAD Binary-class | SVM<br>Elnagar et al. [46] | - | - | 80.0 | 82.0 |
| | The proposed ArCAR System | 81.48 | 81.44 | 81.45 | 81.46 |
| BRAD Multiclass | SVM<br>Elnagar et al. [46] | - | - | 71.0 | 78.0 |
| | The proposed ArCAR System | 66.77 | 68.35 | 67.25 | 77.13 |
| HARD Binary-class | SVM<br>Elnagar et al. [47] | - | - | 81.0 | 78.0 |
| | The proposed ArCAR System | 93.26 | 93.23 | 93.23 | 93.23 |
| HARD Multiclass | Random Forest<br>Elnagar et al. [47] | - | - | 51.0 | 88.0 |
| | The proposed ArCAR System | 82.33 | 80.98 | 81.63 | 93.58 |

Although the proposed ArCAR system showed promising Arabic text recognition results for documents recognition and sentiment analysis, some drawbacks and limitations are noted. First, the sizes of training and testing annotated Arabic text documents and sentiments are still limited. Thus, the data augmentation strategy may help to improve the performance of the ArCAR system. Second, even though some remedies are utilized to avoid system bias due to unbalanced datasets, it seems insufficient since deep learning models perform well with balanced datasets.

Future plans to improve the performance of the proposed ArCAR system are addressed as follows. First, Arabic data augmentation could be used to enlarge the training datasets and improve the overall performance. Second, a hybrid system to represent Arabic text at the character level and word level may assist in more recognition improvements. Third, the challenging problem of multi-label text categorization could be addressed for future directions to better understand the Arabic language. Fourth, the sentiment analysis should also address the issue of neutrality or ambivalence as well.

## 6. Conclusions

In this study, a new deep learning Arabic text computer-aided recognition (ArCAR) is proposed for character-level Arabic text classification in two applications: Arabic text document classification and Arabic sentiment analysis. This approach represents Arabic text at the character level to minimize preprocessing drawbacks such as stemming, search for the root word, normalization, etc.

To provide a rapid and more accurate recognition of Arabic text, we employ deep learning based on a convolutional neural network due to its capability to generate huge hierarchal deep features without user interventions. For Arabic text document classification, we use twelve different datasets in the multiclass problem to show the reliability and capability of the ArCAR system regardless of the number of classes. For Arabic sentiment analysis, we use four datasets to show the feasibility of the ArCAR system for Arabic sentiment recognition. In this case, the proposed ArCAR system is evaluated regarding binary-class and multiclass problems. A comparison study using balanced and unbalanced datasets shows the slightly stable performance of the ArCAR system, especially with some remedies for unbalanced datasets. Such deep learning ArCAR systems would be beneficial to provide AI-based practical solutions for better understanding of the Arabic language.

**Author Contributions:** Conceptualization, A.Y.M. and M.A.A.-a.; methodology, A.Y.M. and M.A.A.-a.; software, A.Y.M.; validation, A.Y.M. and M.A.A.-a.; formal analysis, A.Y.M.; investigation, H.J. and A.Y.M.; resources, A.Y.M. and H.J.; data curation, A.Y.M.; writing—original draft preparation, A.Y.M. and M.A.A.-a.; writing—review and editing, A.Y.M. and M.A.A.-a.; visualization, M.A.A.-a.; supervision, S.L.; project administration, M.A.A.-a.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** Ministry of Science and ICT, South Korea: IITP-2021-2017-0-01629, Korea Government (MSIT): 2017-0-00655, Institute for Information & Communications Technology Promotion (IITP): IITP-2021-2020-0-01489, NRF: NRF-2019R1A2C2090504.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program under Grant IITP-2021-2017-0-01629, and in part by the Institute for Information & Communications Technology Promotion (IITP), through the Korea Government (MSIT) under Grant 2017-0-00655 and IITP-2021-2020-0-01489 and Grant NRF-2019R1A2C2090504.

**Conflicts of Interest:** There are no conflict of interest associated with publishing this paper.

## References

1. Aggarwal, C.C.; Zhai, C. A survey of text classification algorithms. In *Mining Text Data*; Springer: New York, NY, USA, 2012; pp. 163–222.
2. Ameur, M.; Belkebir, R.; Guessoum, A. Robust Arabic Text Categorization by Combining Convolutional and Recurrent Neural Networks. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2020**, *19*, 1–16. [CrossRef]
3. Larkey, L.S.; Connell, M.E. Arabic information retrieval at UMass in TREC-10. In Proceedings of the Tenth Text Retrieval Conference, Gaithersburg, MD, USA, 13–16 November 2001.
4. Mohammed, O.A.; Salah, A. Translating Ambiguous Arabic Words Using Text Mining. *Int. J. Comput. Sci. Mob. Comput.* **2019**, *8*, 130–140.

5. Harrat, S.; Meftouh, K.; Smaili, K. Machine translation for Arabic dialects (survey). *Inf. Process. Manag.* **2019**, *56*, 262–273. [CrossRef]
6. El-Halees, A.M. Filtering Spam E-Mail from Mixed Arabic and English Messages: A Comparison of Machine Learning Techniques. *Int. Arab. J. Inf. Technol.* **2009**, *6*, 1.
7. Shehab, M.A.; Badarneh, O.; Al-Ayyoub, M.; Jararweh, Y. A supervised approach for multi-label classification of Arabic news articles. In Proceedings of the 2016 7th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 13–14 July 2016; pp. 1–6.
8. Hakak, S.; Kamsin, A.; Tayan, O.; Idris, M.Y.I.; Gilkar, G.A. Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges. *Inf. Process. Manag.* **2019**, *56*, 367–380. [CrossRef]
9. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
10. Al-Sabahi, K.; Zhang, Z.; Long, J.; Alwesabi, K. An enhanced latent semantic analysis approach for arabic document summarization. *Arab. J. Sci. Eng.* **2018**, *43*, 8079–8094. [CrossRef]
11. Hasanuzzaman, H. Arabic language: Characteristics and importance. *Echo J. Humanit. Soc. Sci.* **2013**, *1*, 11–16.
12. Salah, R.E.; binti Zakaria, L.Q. A comparative review of machine learning for Arabic named entity recognition. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, *7*, 511–518. [CrossRef]
13. Marie-Sainte, S.L.; Alalyani, N.; Alotaibi, S.; Ghouzali, S.; Abunadi, I. Arabic natural language processing and machine learning-based systems. *IEEE Access* **2018**, *7*, 7011–7020. [CrossRef]
14. Bounhas, I.; Soudani, N.; Slimani, Y. Building a morpho-semantic knowledge graph for Arabic information retrieval. *Inf. Process. Manag.* **2020**, *57*, 102124. [CrossRef]
15. Kowsari, K.; Meimandi, K.J.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]
16. Al-Smadi, M.; Al-Ayyoub, M.; Jararweh, Y.; Qawasmeh, O. Enhancing aspect-based sentiment analysis of Arabic hotels' reviews using morphological, syntactic and semantic features. *Inf. Process. Manag.* **2019**, *56*, 308–319. [CrossRef]
17. Al-antari, M.A.; Al-masni, M.A.; Metwally, M.K.; Hussain, D.; Park, S.-J.; Shin, J.-S.; Han, S.-M.; Kim, T.-S. Denoising images of dual energy X-ray absorptiometry using non-local means filters. *J. X-ray Sci. Technol.* **2018**, *26*, 395–412. [CrossRef]
18. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very deep convolutional networks for text classification. *arXiv* **2016**, arXiv:1606.01781.
19. Duque, A.B.; Santos, L.L.J.; Macêdo, D.; Zanchettin, C. Squeezed Very Deep Convolutional Neural Networks for Text Classification. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 193–207.
20. Daif, M.; Kitada, S.; Iyatomi, H. AraDIC: Arabic Document Classification using Image-Based Character Embeddings and Class-Balanced Loss. *arXiv* **2020**, arXiv:2006.11586.
21. Zhang, X.; LeCun, Y. Text understanding from scratch. *arXiv* **2015**, arXiv:1502.01710.
22. Einea, O.; Elnagar, A.; al Debsi, R. Sanad: Single-label arabic news articles dataset for automatic text categorization. *Data Brief* **2019**, *25*, 104076. [CrossRef]
23. Al-Masni, M.A.; Al-Antari, M.A.; Choi, M.T.; Han, S.M.; Kim, T.S. Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks. *Comput. Methods Programs Biomed.* **2018**, *162*, 221–231. [CrossRef]
24. Al-Masni, M.A.; Al-Antari, M.A.; Park, J.-M.; Gi, G.; Kim, T.-Y.; Rivera, P.; Valarezo, E.; Choi, M.-T.; Han, S.-M. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Comput. Methods Programs Biomed.* **2018**, *157*, 85–94. [CrossRef] [PubMed]
25. Al-Antari, M.A.; Al-Masni, M.A.; Choi, M.T.; Han, S.M.; Kim, T.S. A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. *Int. J. Med Inform.* **2018**, *117*, 44–54. [CrossRef]
26. Al-Antari, M.A.; Kim, T.-S. Evaluation of Deep Learning Detection and Classification towards Computer-aided Diagnosis of Breast Lesions in Digital X-ray Mammograms. *Comput. Methods Programs Biomed.* **2020**, *196*, 105584. [CrossRef] [PubMed]
27. Al-antari, M.A.; Hua, C.-H.; Bang, J.; Lee, S. Fast deep learning computer-aided diagnosis of COVID-19 based on digital chest x-ray images. *Appl. Intell.* **2020**, *51*, 2890–2907. [CrossRef]
28. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
29. Gambäck, B.; Sikdar, U.K. Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 4 August 2017; pp. 85–90.
30. Habash, N.Y. Introduction to Arabic natural language processing. *Synth. Lect. Hum. Lang. Technol.* **2010**, *3*, 1–187. [CrossRef]
31. Lo, S.L.; Cambria, E.; Chiong, R.; Cornforth, D. Multilingual sentiment analysis: From formal to informal and scarce resource languages. *Artif. Intell. Rev.* **2017**, *48*, 499–527. [CrossRef]
32. Cambria, E.; Das, D.; Bandyopadhyay, S.; Feraco, A. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–10.
33. Durou, A.; Aref, I.; Al-Maadeed, S.; Bouridane, A.; Benkhelifa, E. Writer identification approach based on bag of words with OBI features. *Inf. Process. Manag.* **2019**, *56*, 354–366. [CrossRef]

34. El Kourdi, M.; Bensaid, A.; Rachidi, T.-E. Automatic Arabic document categorization based on the Naïve Bayes algorithm. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, Switzerland, 28 August 2004; pp. 51–58.
35. Al-Harbi, S.; Almuhareb, A.; Al-Thubaity, A.; Khorsheed, M.; Al-Rajeh, A. Automatic Arabic Text Classification; In Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data, Lyon, France, 12–14 March 2008.
36. Etaiwi, W.; Awajan, A. Graph-based Arabic text semantic representation. *Inf. Process. Manag.* **2020**, *57*, 102183. [CrossRef]
37. Suleiman, D.; Awajan, A.; al Etaiwi, W. The use of hidden Markov model in natural arabic language processing: A survey. *Procedia Comput. Sci.* **2017**, *113*, 240–247. [CrossRef]
38. Al-Ayyoub, M.; Khamaiseh, A.A.; Jararweh, Y.; Al-Kabi, M.N. A comprehensive survey of arabic sentiment analysis. *Inf. Process. Manag.* **2019**, *56*, 320–342. [CrossRef]
39. Boukil, S.; Biniz, M.; el Adnani, F.; Cherrat, L.; el Moutaouakkil, A.E. Arabic text classification using deep learning technics. *Int. J. Grid Distrib. Comput.* **2018**, *11*, 103–114. [CrossRef]
40. Almuzaini, H.A.; Azmi, A.M. Impact of stemming and word embedding on deep learning-based arabic text categorization. *IEEE Access* **2020**, *8*, 127913–127928. [CrossRef]
41. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
42. Abdul-Mageed, M. Modeling arabic subjectivity and sentiment in lexical space. *Inf. Process. Manag.* **2019**, *56*, 291–307. [CrossRef]
43. Oueslati, O.; Cambria, E.; HajHmida, M.B.; Ounelli, H. A review of sentiment analysis research in Arabic language. *Future Gener. Comput. Syst.* **2020**, *112*, 408–430. [CrossRef]
44. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-aware neural language models. *arXiv* **2015**, arXiv:1508.06615.
45. Romeo, S.; Martino, G.d.; Belinkov, Y.; Barrón-Cedeño, A.; Eldesouki, M.; Darwish, K.; Mubarak, H.; Glass, J.; Moschitti, A. Language processing and learning models for community question answering in arabic. *Inf. Process. Manag.* **2019**, *56*, 274–290. [CrossRef]
46. Alayba, A.M.; Palade, V.; England, M.; Iqbal, R. Improving sentiment analysis in Arabic using word representation. In Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), London, UK, 12–14 March 2018; pp. 13–18.
47. Al-Taani, A.T.; Al-Sayadi, S.H. Classification of Arabic Text Using Singular Value Decomposition and Fuzzy C-Means Algorithms. In *Applications of Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 111–123.
48. Elfaik, H. Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text. *J. Intell. Syst.* **2020**, *30*, 395–412. [CrossRef]
49. El-Alami, F.-Z.; El Mahdaouy, A.; El Alaoui, S.O.; En-Nahnahi, N. A deep autoencoder-based representation for arabic text categorization. *J. Inf. Commun. Technol.* **2020**, *19*, 381–398. [CrossRef]
50. Elzayady, H.; Badran, K.M.; Salama, G.I. Arabic Opinion Mining Using Combined CNN-LSTM Models. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 25–36. [CrossRef]
51. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 649–657.
52. Saad, M.K.; Ashour, W.M. Osac: Open Source Arabic Corpora. In Proceedings of the 6th International Conference on Electrical and Computer Systems (EECS'10), Lefke, North Cyprus, 25–26 November 2010.
53. Chowdhury, S.A.; Abdelali, A.; Darwish, K.; Soon-Gyo, J.; Salminen, J.; Jansen, B.J. Improving Arabic Text Categorization Using Transformer Training Diversification. In Proceedings of the Fifth Arabic Natural Language Processing Workshop, Barcelona, Spain, 1 December 2020; pp. 226–236.
54. Elnagar, A.; Einea, O. Brad 1.0: Book reviews in arabic dataset. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; pp. 1–8.
55. Elnagar, A.; Khalifa, Y.S.; Einea, A. Hotel Arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent Natural Language Processing: Trends and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 35–52.
56. Alsharhan, E.; Ramsay, A. Improved Arabic speech recognition system through the automatic generation of fine-grained phonetic transcriptions. *Inf. Process. Manag.* **2019**, *56*, 343–353. [CrossRef]
57. Farha, I.A.; Magdy, W. A comparative study of effective approaches for Arabic sentiment analysis. *Inf. Process. Manag.* **2021**, *58*, 102438. [CrossRef]
58. Park, H.-G.; Bhattacharjee, S.; Deekshitha, P.; Kim, C.-H.; Choi, H.-K. A Study on Deep Learning Binary Classification of Prostate Pathological Images Using Multiple Image Enhancement Techniques. *J. Korea Multimed. Soc.* **2020**, *23*, 539–548.
59. Agarap, A.F. Deep Learning using Rectified Linear Units (ReLU). *arXiv* **2018**, arXiv:1803.08375.
60. Li, Y.; Xu, L.; Rao, J.; Guo, L.; Yan, Z.; Jin, S. A Y-Net deep learning method for road segmentation using high-resolution visible remote sensing images. *Remote Sens. Lett.* **2019**, *10*, 381–390. [CrossRef]