CNN-BLSTM Model for Arabic Text Recognition in Unconstrained Captured Identity Documents

Nabil Ghanmi(✉), Amine Belhakimi, and Ahmad-Montaser Awal

AI & ML Center of Excellence, IDNOW, Rennes, France
{nabil.ghanmi,amine.belhakimi,montaser.awal}@idnow.io

Abstract. Optical Character Recognition (OCR) for Arabic text (printed and handwritten) has been widely studied by researchers in the last two decades. Some commercial solutions have emerged with good recognition rates for printed text (on white or uniform backgrounds) or handwritten text with limited vocabulary. In addition to being naturally cursive, the Arabic language comes with additional challenges due to its calligraphy resulting in a variety of fonts and styles. In this work, recent advances in recurrent neural networks are explored for the recognition of Arabic text in identity documents captured in the wild. The unconstrained captures bring additional difficulties as the text has to be first localized before being able to recognize it. Various pre-processing steps are introduced to overcome the difficulties related to the Arabic text itself and also due to the capturing conditions. The presented approach outperforms existing solutions when evaluated using a private dataset and also using the recent MIDV2020 dataset.

Keywords: Arabic Text Recognition · Identity Document · Convolutional Neural Network · Long Short-Term Memory · Connectionsit Temporal Classification · Character Error Rate

1 Introduction

In an economic environment that is increasingly being digitalized, it is becoming essential for companies to rethink their customer journey in order to offer them fast, reliable and ergonomic online services. This particularly concerns the customer onboarding process. For such a purpose, companies have turned to completely digital and remote onboarding as it is a flexible way to acquire new customers without meeting them physically in offices or stores. Typically, the remote access to a service or a product (e.g. telephone or internet contracts, banking, e-gambling, etc.) requires that the customer sends or uploads copies (photos or scans) of his identity documents (or any relevant documents). Then, the data in the document images should be automatically extracted and verified in order to know the customer identity and check the validity of the document as well as the conformity of the extracted data with that provided by the customer

G. L. Foresti et al. (Eds.): ICIAP 2023 Workshops, LNCS 14365, pp. 106–118, 2024.
https://doi.org/10.1007/978-3-031-51023-6_10

in the subscription form. This digital processes was initially enabled by image processing techniques and is now powered by deep learning based approaches.

To make that work properly, a full automatic reading system, from document classification and localization to text reading and verification, is required. Even that this problem has been widely studied and much progress have been made for some alphabet such as latin, reading arabic documents still far from being resolved. It is true that many steps of document reading system are generic and could be directly applied on arabic documents without much difficulty. Nevertheless, the text recognition (aka OCR) is language-dependent and applying, for example, latin OCR to arabic alphabet is not straightforward.

In this paper we focus on arabic text recognition in identity documents. We will present a deep learning based OCR that takes into account the arabic text specifities and the complexity of the identity document background.

2 Related Works

Optical character recognition (OCR) goal is to extract text contained in images (scanned documents or photos taken by a camera) and represent them with a standard encoding like ASCII. OCR systems are usually composed of text detection and recognition steps organized sequentially in most cases. The first step aims at locating text in the input images (scanned documents or photos taken by cameras). Every detected text is then used as an input to the text recognition step.

Most text recognition approaches use a segmentation-based transcription where the input image (containing text blocks) is segmented into text lines, then into words or patches and finally into characters. Many works were focusing on the segmentation task like [8,17] which are based on peaks detection.

Some other methods perform the transcription at word or sentence level. A text line is segmented into words using Hidden Markow Models (HMM) or Artificial Neural Network (ANN) models. The most recent approaches use Recurrent Neural Network (RNN) with Connectionist Temporal Classification (CTC) (also called sequence learning) [11] [15].

In [16], Yousfi et al. presented an Arabic video text recognition system. Features have been extracted from the input images using deep Belief networks and multi layer perceptron. A BLSTM network was used to map the sequence to characters, followed by a CTC output layer. In [4], the authors use a multidimensionnal LSTM and apply a dropout operation on its first layer. Their system was tested on the OpenHaRT (145.000 Arabic handwritten text) with an accuracy of 90.1%.

A combination of CNN and GRU (Gated Recurrent Unit, which is a type of RNN architecture) was used by [14] to recognize Arabic license plates numbers without segmentation. The system achieved an accuracy of 90%. A Similar approach was proposed in [12] where the authors built an end-to-end deep CNN

- RNN model to solve the text-based CAPTCHA images with distortion, rotation and noisy background. This model achieved 99% accuracy but on a fully generated data set.

On the industrial domain, a few OCR systems manage Arabic script recognition such as Sakhr OCR, ABBYY, Nuance, etc. As we are dealing with identity documents, a special focus has been laid on the commercial OCRs dedicated to identity document reading. In that context, we have identified ID reader as an OCR engine that extracts automatically information form several identity documents [5]. It can be applied on 300 dpi images with various fonts, sizes and resolution. Another commercial solution that deals with Arabic identity documents is sky IDentification which is based on deep learning technique.

All these existing OCRs (as well as some others) were evaluated on our private data set of many thousands of fields extracted from various Arabic identity documents. The obtained results were unsatisfactory and did not exceed 43% of accuracy.

Since we focus on deep learning approaches, data sets are a very important aspect for OCR systems. Like for any deep learning application, a good amount of data is required in order to learn the model how to recognize the text in images. Most of the previous works showcase a high accuracy but on private data sets without a fair comparison. This is due to the lack of public data sets and standard bench-marking. In the following, we try to outline the most used Arabic data sets.

Several public data sets are available such as KHATT [6], APTI [13], APTID/MF [2] and PATDB [3]. Even if these data sets present a large variability in terms of capture conditions, image quality, text properties and lexicon, the identity document challenges and characteristics still not considered. In fact, an identity document has a highly textured background, a constrained layout and also a very large lexicon that may contains words not necessarily belonging to the Arabic vocabulary (proper names). Furthermore, this kind of document requires a very accurate recognition as it contains sensitive data.

For our knowledge, few identity document data sets are publicly available. Moreover, they generally contain partial information, or synthetic data. The most known data sets are LRDE IDID [9], BID [7], MIDV [1]. The LRDE IDID contains a few quantity of document samples, which is not useful for significant benchmarking and deep analysis of the processing methods. The BID dataset is composed of synthetic images that are generated by writing text field values on automatically masked document regions, which leads to several imperfections making the data set very different from a real one. As for MIDV data set, even if it contains a lot annotated images, it presents the disadvantage of lack of variability as for each document type, only five different samples are collected under slightly variable conditions. This data set might be useful for document localization, classification or tracking but it is less useful for OCR evaluation as it contains few variable data. Nevertheless, we used a part of this data set in our evaluation.

3 Problem Statement

3.1 Identity Document Reading

Identity documents, such as ID cards, passports, driving licences, resident permits, etc. are a particular type of documents that have a precise design and several security features. They are usually issued by governments to define, prove and verify the holder's identity. The content of these documents is composed of static fields representing the document template and variable fields containing its holder personal data such as his first name, last name, birth date, birth place etc. Extracting these information is of great interest as it allows an easy authentication of the document holder, a robust identity verification and an automated form filling. Nevertheless, this task is very challenging due to the complex background texture that makes text localization and recognition very hard. In fact, the graphical components of the texture (several geometric forms with various styles and colors, complicated patterns, etc.) can be easily confused with textual components and thus disturb the text extraction and recognition system.

3.2 Arabic Text Recognition

Arabic text holds additional difficulties to recognition systems due to its specific style. It is written from right to left and consists of 28 basic letters that connect to each others in order to form words in printed and handwritten text. Thus, Arabic text is always cursive even in its machine-printed form. Most of these letters has 4 different forms according to their positions in the word (beginning, middle, end, or isolated), as shown in Fig. 1.

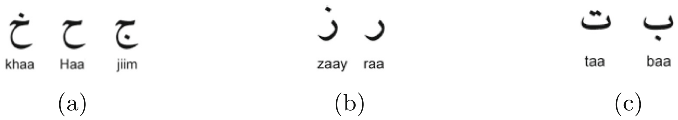
(a) (b) (c) (d)

Fig. 1. Different forms of the letter *'seen'* according to its position: (a) isolated, (b) at the end, (c) in the middle and (d) in the beginning of the word.

Some of these letters do not connect from the left (as for example *'raa'*, *'zaay'*, etc.); and thus they have only two forms (isolated and end). This property produces the piece of Arabic words (PAWs) where one word can be composed in one or more sub-words (PAWs). In addition, many groups of letters share the same body form and are different only by the diacritics. Figure 2 shows 3 examples of this case *'khaa'*, *'haa'* and *'Jiim'* (Fig. 2a), *'raa'* and *'zaay'* (Fig. 2b) and *'baa'* and *'taa'* (Fig. 2c). Other diacritics are used for the vocalization of the text, but they are not essential for the text understandin

4 Proposed Method

The current work was carried out as a part of an automatic system of identity document reading and verification. The full system can be seen as a pipeline composed of several steps. Given, an input image in the wild, the document is firstly localized and classified. Then, the obtained document image is corrected and the text is localized based on a well defined reference layout. Finally, comes the OCR step that takes as input the text bounding boxes extracted by the previous step.



(a) (b) (c)

Fig. 2. Example of 3 groups of letters having same structure and different diacritics.

In this section, the proposed OCR for Arabic language is detailed. We will present the model architecture as well as the training parameters. The data pre-processing which is a crucial step for such OCR will also be described.

4.1 Architecture Description

In this work, we have studied two architectures that will be referred, in the following, as basic and optimized respectively. The basic architecture is composed of 3 parts. A feature extraction part that computes the input image features and present them as a reduced space matrix. A sequence modeler part that takes the output of the previous part and outputs an embedding space. The last part is a Connectionist Temporal Classification (CTC) layer that classifies the input into the corresponding letter. More details about this architecture are in Table 1a.

Features Extractor. It is a convolutional neural network composed of three blocks. Each block is composed of a convolution layer with a 3×3 kernel size followed by a max-pool layer with a kernel size of 2×2. The two first max-pool layers have a stride $(2, 2)$ whereas the 3^{rd} max-pool layer has a stride $(2, 1)$ and a padding $(0, 1)$. The number of filters used in the convolution layers within the 3 blocks are respectively 64, 128 then 256. Only the 3rd max-pool layer have a s The input of this feature extractor is $[1 \times 864$ x $48]$ which is an image of the field containing the text. Its output is a matrix with a size of $[256 \times 6$ x $217]$. Even if the extraction part is fairly light weight, the experiments show that the extracted features allows the sequence modeler to easily decode the field image.

Sequence Modeler. We used two layers of bidirectional LSTM each of which contains 240 neurons. The input of this LSTM is the extracted feature vector that is flatten to a vector of size [256 × 1302]. The bidirectional property makes the network able to predict a certain letter based on the sequence from the beginning and from the end. This allows a better reading, exceptionally when the image is of low quality. The output of the sequence modeler is then mapped to the number of alphabets with a linear layer.

CTC. This layer allows the network to output the sequence into a classification soft-max layer. Then a beam search algorithm is performed to transform the output into its corresponding text.

Table 1. (a) Basic and (b) Optimized architecture details.

Layer	**Params**	**Output size**
CNN_1	64,3,3	64,48,864
Relu		64,48,864
maxpool_1		64,24,432
CNN_2	128,3,3	128,24,432
Relu_2		128,24,432
maxpool_2		128,12,216
CNN_3	256,3,3	256,12,216
Relu_3		256,12,216
maxpool_3		256,6,217
batch_nor_1		256,6,217
flatten layer		256,1302
blstm x 2	240	240,1302
out_embedding	n_classes	1302,n_classes

(a)

Layer	**Params**	**Output size**
CNN_1	64,3,3	64,48,864
Relu		64,48,864
maxpool_1		64,24,432
CNN_2	128,3,3	128,24,432
Relu_2		128,24,432
maxpool_2		128,12,216
CNN_3	256,3,3	256,12,216
Relu_3		256,12,216
maxpool_3		256,6,217
batch_nor_1		256,6,217
map_seq		217,64
blstm x 2	240	240,217
out_embedding	n_classes	217,n_classes

(b)

Optimized Model. In order to make the basic model lighter and faster, the initial architecture is used, while reducing the number of weights by replacing the flatten layer with a mapping linear layer. In the basic model we directly flatten the output of the convolutions to generate a vector of size [256 × 1302]. This flatten layer is replaced in the optimized model by a linear layer, which reduces the size to [217 × 64]. Then, the obtained matrices are reshaped to get the appropriate output vector. Using the same number of neurons in the LSTM, the number of weights is drastically reduced while preserving the accuracy of the models. More specifically, the number of weights for the basic model is around 1.6M while for the optimized model, it is around 800K. The optimized network is then 3 times faster than the basic model (the processing time is reduced from 0.23s to 0.07s). More details about the Optimized model are shown in Table 1b.

4.2 Training Parameters

Both architectures (Basic and Optimized) were trained using the PyTorch framework. The RMSprop optimizer was used with a triangular function to assign the learning rate with values between 0.01 to 0.001. The models were trained on a GTX 1080Ti (12G).

(a) (b)

Fig. 3. (a) Example of field image extracted from an identity document. (b) The horizontal mirroring of field image in (a).

4.3 Data Pre-processing

- **Field image mirroring.** It is worth remembering that a text reading system is generally composed of two main steps: text detection and text recognition. To cope with the complexity of the identity document background and the variability of the length of the text fields, the first step was designed to return bounding box with theoretical maximum width defined in the knowledge base (see Fig. 3a).

  Thus, the extracted images have generally empty regions at their left sides, which may causes some problems for training the OCR models. In fact, during the training, the OCR model tries to align the ground truth with the image starting from left to right which leads to aligning the first characters of the ground truth with the empty region and thus disturbing the training. To cope with this problem the image is horizontally mirrored before being used for the training. In addition to placing the text on the left side of the image, this mirroring reverse the order of the characters in the image which allows a best alignment of these characters to their correspondent in the text ground truth. Figure 3b shows an example of an horizontally mirrored image.
- **Ground truth string reversing.** It is known that Arabic text is right-aligned, i.e. each text field starts at the right side of its dedicated zone. Nevertheless, in some documents, mainly passports, some Arabic fields are left-aligned and they finished at the left side of their dedicated zones (see Fig. 4).

Fig. 4. Left aligned text field.

Unlike the right-aligned fields, we do not need to mirror the image for this kind of fields as the empty area is on its right part. But we need to reverse the ground truth text for the training in order to ensure that each character is correctly aligned with its pixels in the image. In fact, Arabic text is written from right to left and the image is traversed form left to right by the OCR model. Therefore, reversing ground truth is necessary.

5 Experiments

5.1 Dataset

A private data set composed of Algerian, Moroccan and Tunisian identity documents (ID cards and passports) is used for experimenting the proposed model. It is collected from a real production flow and used internally for research purposes. Some statistics on this data set are available in the table 2.

Table 2. Private data set description.

Document class	**Nb samples**
DZA_ID	6378
DZA_P	14402
MAR_ID	15596
MAR_P	13402
TUN_ID	2632
TUN_P	14924
Total	67334

For each document we extract field images. Each field contain one line of text (First Name, Last Name, ID Number, etc.). The Fig. 5 shows two examples of text fields extracted from a Moroccan identity card.

(a) (b)

Fig. 5. Examples of field images extracted from documents.

The extracted field images are then transformed to gray scale. To solve the problem of variable size on the network input, the images are padded to have size (1800 × 100) and then resized to (864 × 48). This resizing make the network lighter and faster during the inference, which is very important in an industrial context.

The data set is split into train, validation and test sets containing respectively 80%,10% and 10% of the total number of samples.

5.2 Evaluation Metrics

- **Character Error Rate (CER)**. This metric is based on Levenshtein distance that measures the similarity between two strings. This distance corresponds to the number of deletions, insertions, or substitutions at character-level required to transform the ground truth text (aka reference text) into the OCR output. For each field, the CER is computed using this formula:

$$CER = \frac{D + I + S}{N}$$

  where D, I, S correspond to the number of deletions, insertions and substitutions respectively and N corresponds to the total number of the characters in the reference text.
- **Accuracy.** This metric is defined as the number of the text fields that are correctly read by the OCR divided by the total number of text fields.

$$CER = \frac{\text{nb correctly read fields}}{\text{nb all fields}}$$

5.3 Model Training

Data Augmentation. As image labeling is a tedious task, we start training the model using a relatively small data set, then we increased progressively the number of annotated document. The behaviour of the model accuracy in function of the training data size is studied, in order to estimate the optimal data set size (Fig. 6). In addition to the use of manually annotated real data, artificially generated data is used. The artificial images were created using a randomly cropped background from real documents on which a text was written with a random blur and distortion to simulate more realistic data. It is worth noting that the model

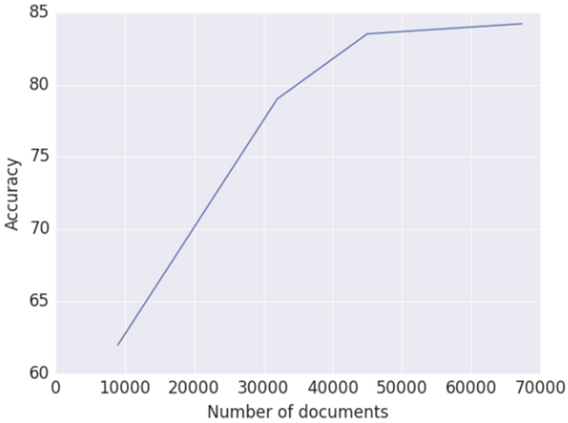


Fig. 6. The effect of the data set size on the accuracy of the model.

generalizes less well when it is trained mostly on generated data than when it is trained using real data. Therefore, we favor using real data, and we serve of artificial data mainly to cope with the unbalanced data problem by adding more samples of the less represented characters in the initial real data set.

Generic vs Specific Model. As we deal with multiple identity document classes (different versions of ID cards, passports, driving licences, etc. from various countries), two different strategies to train the proposed model arise:

- Generic training. This strategy aims to make the model generic, i.e. applicable for several identity document classes, by training it with labelled data extracted from various classes. This is particularly important to reduce the cost of data labelling since a unique data set is required. It is also adapted to deal with the cases of documents for which there is not enough data to use for a specific training.
- Specific training. This strategy consist of training specific model for each document class using labelled data extracted only from this particular class. The main advantage is that such a model will be more accurate since it will encounter less variability in terms of text background, font and styles. Nevertheless, a lot of data from each document class should be labeled, which is tedious and time consuming. Moreover, even if the training is generally straightforward, some specific tuning could be required.

We start our experimentation by trying the generic strategy. During the training, the model achieves an accuracy of about 73% on the validation data set. However, this accuracy has decreased on the test data set, mainly on the document classes that were not present in the training set and only 61% of accuracy is obtained. For the document classes that were present in the training data set, the accuracy on the test set remains comparable to the training accuracy and it varies between 66% and 73% depending on the document class complexity and its frequency in the training set. Based on these results, the generic model is kept for the document classes that do not have enough training data. For other classes, specific models are built.

5.4 Model Evaluation

Once trained, the proposed models are evaluated on two test data sets:

- Private data set: all the documents in this data set belong to classes that were represented in the training data set.
- Public MIDV data set: contrary to the private data set, all the documents in this data set belong to classes that were not seen during the training.

Results on the Private Data Set. Both versions of the proposed model are tested on the private test set and the obtained results are summarized in the Table 3. The optimized model is quite similar in term of performance to the basic model, but it is much faster in inference time. The results show that the generic model achieves comparable accuracy to the specific models for the document classes that have few training samples (TUN_ID and DZA_ID) and the model that presents a more complex background (TUN_PA) than the other classes.

Table 3. Performance of the basic and optimized model on the private test set.

	Basic model		Optim model	
	Acc.(%)	CER. (%)	Acc. (%)	CER. (%)
DZA_ID	**72**	**6.33**	71	7.02
DZA_P	86	4.15	**87**	**3.98**
MAR_ID	**78**	**5.9**	77	6.14
MAR_P	87	3.72	**87**	**3.42**
TUN_ID	**71**	**8.1**	69	8.3
TUN_P	**70**	**7.8**	70	7.9
GENERIC	**69**	**8.3**	68	9.4

These results are compared to those obtained using easyOCR which is a ready-to-use tool based on the work published in [10], as well as two commercial solutions: Nuance and Google OCR. We also evaluate a private OCR developed by a company working on KYC solutions in Arabic countries (we will use the name *"OCR for Arabic KYC"* to designate this OCR). The table 4 shows that the best OCR among those we evaluated achieves an accuracy of about 43%, which proves that reading Arabic text within identity documents is very challenging. The model we proposed is significantly more efficient and exceeds this best OCR by 26 percentage points. This can be explained by the fact that our OCR is specialized on identity documents and has learned how to cope with their specificity, mainly their background complexity.

Table 4. Our OCR outperforms 4 OCRs that we evaluated during our study.

OCR	**Accuracy** (%)
Our OCR	**69**
easyOCR	19.7
Google OCR	20.4
Nuance	5.0
OCR for Arabic KYC	43.1

Results on the Public MIDV Dataset. Two examples of fields extracted from this data set are shown in Fig. 7. As we already said, this data set contains a small number of documents with slightly different angles. In order to have more field images for our test, an augmentation procedure is applied on the existing fields by generating random noise, blur and geometrical distortion. The generic model achieves an accuracy of 61% and 60% using the basic and the optimized version respectively. This low performance can be explained firstly by the fact that these document classes are not seen during the training. Secondly, the quality degradation by adding blur and noise made some cases hard to read.

(a) (b)

Fig. 7. Two Examples of field images extracted from MIDV dataset.

6 Conclusion

In this paper, we show that a CNN-BLSTM model, originally designed for Latin text recognition, can be adapted to Arabic script which has challenging characteristics. This adaptation aims to align an input text line image with its ground truth text, during the training of the CNN-BLSTM. Two main operations are proposed, image mirroring and ground truth string reversing, in order to align the image pixels to their correspondent characters. The presented model was experimented on a large private data set and a small public one. The obtained results show that it outperforms several famous commercial OCRs. This can be explained by the fact that our model is dedicated to the identity documents contrary to the other OCRs which are generic and more adapted to full page text that are less challenging. We plan, for the next step, to explore other deep learning models based on transformer architectures. Such models are well adapted for both image understanding and character level text generation, and they have shown very good results on several text recognition problems.

References

1. Bulatovich, B.K., et al. "MIDV-2020: A comprehensive benchmark dataset for identity document analysis". In: 46.2 (2022), pp. 252–270
2. Al-Hashim A.G., Mahmoud, S.A.: Benchmark database and GUI environment for printed Arabic text recognition research. In: WSEAS Trans. Inf. Sci. Appl. 7.4 (2010), pp. 587–597
3. Jaiem, F.K., et al. "Database for Arabic printed text recognition research". In: ICIAP. 2013, pp. 251–259
4. Maalej, R., Tagougui, N., Kherallah, M.:Online Arabic handwriting recognition with dropout applied in deep recurrent neural networks. In: 12th IAPR DAS. IEEE. 2016, pp. 417–421
5. El-Mahallawy, M. S. M.: A Large scale HMM-based omni front-written OCR system for cursive scripts. In: Ph.D. thesis, Cairo University. 2008

6. Mahmoud, S. A., et al.: ”KHATT: an open Arabic offline handwritten text database”. In: Pattern Recognition 47.3 (2014), pp. 1096–1112
7. Ngoc, M., Fabrizio, J., éraud, T.G.,: Saliency-based detection of identy documents captured by smartphones. In: 13th DAS. 2018, pp. 387–392
8. Ramdan, J., Omar, K., Faidzul. M.: A Novel method to detect segmentation points of Arabic words using peaks and neural network. In: IJASEIT 7.2 (2017), pp. 625–631
9. de Sá Soares, A., das Neves Junior, R. B., Bezerra, B.L.D.:BID Dataset: a challenge dataset for document processing tasks. In: 18th Conference on Graphics, Patterns and Images. SBC. 2020, pp. 143–146
10. Shi, B., Bai, X., Yao. C.: An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. In: CoRR abs/1507.05717 (2015). arXiv: 1507.05717
11. Shivakumara, P., et al.:”CNN-RNN based method for license plate recognition”. In: CAAI Transactions on Intelligence Technology 3.3 (2018), pp. 169–175
12. Shu, Y., Xu, Y.: End-to-End Captcha Recognition Using Deep CNNRNN Network. In: IEEE 3rd IMCEC. IEEE. 2019, pp. 54–58
13. Slimane, F., et al.: A new Arabic printed text image database and evaluation protocols. In: 2009 10th International Conference on Document Analysis and Recognition. IEEE. 2009, pp. 946–950
14. Suvarnam, B., Ch, V.S.: Combination of CNN-GRU model to recognize characters of a license plate number without segmentation. In: 5th ICACCS. IEEE. 2019, pp. 317–322
15. Yousef, M., Hussain, K.F., Mohammed, U.S.: Accurate, Data-Efficient, unconstrained text recognition with convolutional neural networks. In: CoRR abs/1812.11894 (2018). arXiv: 1812.11894
16. Yousfi, S.: Embedded Arabic text detection and recognition in videos. PhD thesis. Université de Lyon, 2016
17. Zeki, A., Zakaria, M., Liong, C.: The Use of Area-Voronoi diagram in separating Arabic text connected components. In: 3rd ACEA. 2007, pp. 251–288