Use of Discrete Cosine Transformation and Histograms of Oriented Gradients for Optical Arabic Word Recognition of Different Font Styles and Sizes

Naser Telesi, Mohamed Hashmi and Prof. Dr. Aleksandar Jevremović

Faculty of Informatics and Computing,
University Singidunum
Danijelova 32, 11000 Belgrade

Abstract
For our proposal, we used discrete cosine transform descriptors (DCT) and the histogram of oriented gradients (HOG) for word-level recognition. Extracting and classifying features from word-images are features of all machine learning algorithms. We used the k-Nearest-Neighbour (KNN) as classifiers and evaluated the features and classifiers on seven fonts.
Four fonts, specifically Times New Roman, Arial, Arabic Transparent and Simplified Arabic are used for written books, magazines, and research papers. Three other fonts, namely Arial Unicode MS, Tahoma and K Traffic, are used for adverts and CAPTCHA. Each font type includes different sizes (20, 24, and 28). We obtained high accuracy in the proposed features using the KNN.
The results show that the features significantly outperformed by using the pixel density directly from the images.
Keywords: Optical Character Recognition (OCR); Features Descriptors; Features Extraction; DCT; HOG; Features Vectors, KNN.

1. Introduction

OCR is a method of recognizing optical patterns (characters, digits, and symbols) such as those in digital images. [1]. Once defined, OCR classifies the patterns in the digital images [2]. OCR is achieved in four steps in the following order: pre-processing, segmentation, feature extraction, and classification. It can be applied in two ways. The OCR system can either recognize text images at the character level, or it can recognize text images at the word level.
The second approach prevents the problem of character segmentation and can override errors in character recognition. In the Arabic language, the characters are cursive and mostly connected, thus making it difficult to determine the segmentation point between characters. Therefore, character segmentation is exceeded by recognizing text images of the complete word [2]. Other languages, such as Urdu and Persian, use Arabic letters, but Arabic text recognition does not have the same accuracy of sophistication as other languages, especially English. This is attributed to several issues, such as a lack of essential interaction between researchers in this field and a deficiency of infrastructure supporting utilities, including Arabic text databases, electronic language corpora, and supporting staff; thus, each researcher has their own system and database [3].
Therefore, it is difficult to compare results for the proposed methods due to the deficiency of benchmark databases. Despite this, word recognition remains a challenge. Depending on the language properties and the number of classes, it is difficult to easily find the suitable features [4].
In our proposal, we use discrete cosine transform (DCT) and histogram of oriented gradients (HOG) for word-level recognition. The features are extracted directly from the word-image using DCT and HOG as descriptors, which are then passed to the KNN algorithm to be classified. The system recognizes the word image by comparing it with the reference word images This vector is matched against a pre-estimated database of vectors from random Arabic words. Vectors from the database with the highest score (least error) are returned from the classifiers as the class for the unknown image.
The database has a feature vector for each word. Database feature vectors are generated using a single font as a reference (reference model) for other fonts. These feature descriptors and classifiers are evaluated on seven fonts. Four fonts, namely Times New Roman, Arial, Arabic Transparent and Simplified Arabic, are used for books, magazines, and research papers.
Three other fonts, namely Arial Unicode MS, Tahoma and K Traffic are used for adverts and CAPTCHA. Each font type includes different sizes (20, 24, and 28). Combining the proposed feature descriptors with the k-nearest neighbour (KNN) yields high accuracy. The results showed that descriptors significantly use the pixel density of the images directly.
This paper is divided into the following sections: Section 2 discusses the relevant literature, Section 3 discusses our proposed method, Section 4 presents the experimental results and analysis of our method and Section 5 Conclusion.

2. Related Work

In Arabic text recognition, little research has been published compared with research on Latin text recognition on recognizing human users and computers for the Arabic language.
In the first work, proposed by Aziz and Farah [5], three multilayer perceptrons were combined for Arabic word recognition, with a recognition accuracy of 94%. El-Hajj [6] combined three homogeneous Hidden Markov Model (HMM)-based classifiers using Neural Networks with various features as input, yielding a recognition accuracy of 94.44%. Alma'adeed [7] concerted a rule-based recognizer with a set of HMMs to recognize words from a bank of 47 words. More than 4000 words tested in this system achieved 60% recognition accuracy. The recognition system used a hybrid approach by Souici-Meslati [8] that presented a word-level recognition approach, which was a multi-classifier run in parallel (Neural Networks, KNN, and Fuzzy KNN), achieving 96% recognition accuracy. Burrow [9] applied KNN as a classifier on each sub-word.
The correct classes of sub-words were 74%. Kadhm and Mustafa S. proposed a handwritten word recognition system using Support Vector Machine (SVM) as a classifier, achieving the best recognition accuracy of 96.32%. Abdulwahab G. Krayem [10] proposed a fully integrated system (baseline system) as a word-level recognizer. The presented baseline system was a holistic word-based recognition approach characterised as the probabilistic ranked task.
The obtained average accuracy by the recognizer was 67.24%, which was improved to 78.35%. Hassiba Nemmour and Youcef Chibani [11] introduced a handwritten Arabic-word recognition system using a classifier proposed for lexicons. The recognition rate increased from 72.3% to 84.8%. 3. Tables, Figures and Equations

3 Arabic Characters

Several methods for recognizing Latin and Chinese characters have been proposed, while the recognition of Arabic characters has been relatively sparse [12], [13]. Due to the variation in character structure with other languages, the methods for characters recognition in other languages cannot be used for recognizing Arabic characters. Arabic characters are joined and cursive in general. Therefore, the recognition accuracy of Arabic characters is less than that of disjointed characters, such as printed English.

3.1 Main Characteristics of Arabic Characters:

- Arabic Script constituted of 28 characters. Fifteen of them have dots and 13 are without dots. Each character may appear in two or four different shapes or forms depending on the position of the character, Beginning, Middle, Isolated, and End Form (BF, MF, IF, and EF respectively), 100 in total [14]. Table 1.1 shows a sample of characters and their four possible shapes.

Table 1.1: Sample of Arabic characters and their four possible forms.

IF	BF	MF	EF	IF	BF	MF	EF
ا	-	-	ـا	ف	فـ	ـفـ	ـف
ب	بـ	ـبـ	ـب	ق	-	-	ـق
ح	حـ	ـحـ	ـح	ك	كـ	ـكـ	ـك
د	-	-	ـد	ل	لـ	ـلـ	ـل
ر	-	-	ـر	م	مـ	ـمـ	ـم
س	سـ	ـسـ	ـس	ن	-	-	ـن
ص	صـ	ـصـ	ـص	ه	هـ	ـهـ	ـه
ط	طـ	ـطـ	ـط	و	-	-	ـو
ع	عـ	ـعـ	ـع	ى	-	-	ـى

- Arabic characters are cursive and connected along a baseline, in general. as exemplified below.

Baseline ← الحمد لله رب العالمين

- Some characters have dots, which are placed above or under the character, such as:

ب ق ي

- Some characters have a similar form, but can be distinguished by the dots that can either be above or below the character that takes a different meaning, as shown below.

ج خ ح س ش

- The problem of overlapping and ligatures makes it difficult to determine the segmentation point between characters as shown by the following words.

مجموع الجمال نموذج

- The Arabic language has short vowels referred to as "Diacritics". A diacritic is placed above or under of the Arabic character.

شدٍ شدِّ شدٌّ باً

- In the Arabic language, there are some characters can be joined from one side only. Out of the 28 basic Arabic characters, six can be joined from the right side only while the others can be joined from both sides. These characters are**:**

ا د ذ و ز ر

These characters appearing in two forms, Isolated form, and End form. Whereas the rest characters can appear in four forms.

- In the Arabic language and Urdu, some words may consist of more sub-words exemplified:

طرابلس القاهرة

- There are four characters which may take the secondary character "ء". Those are:

أ ك ؤ ئ

3. Proposed method

In word recognition systems, feature extraction is an important step. Suitable feature extraction leads to higher recognition rates. This is one of the basic decisive and challenging steps in many pattern recognition problems and especially in text recognition applications. Images contain extra information that is ambiguous and unnecessary for classification.

Therefore, the first step in image classification is to simplify the image by extracting the important information and ignoring the rest. Many feature extraction techniques have been proposed in the fields of computer vision (CV) and recognition purpose [15]. Here, we propose to use DCT and HOG descriptors as feature extractors for word-level recognition in different font styles and sizes.

3.1 Discrete Cosine Transform Features (DCT)

The DCT method converts an image's data from its domain into its frequency domain [16]. Highly correlated data yield good energy compression [10]. The efficacy of a transformation scheme relates directly to its ability to input data into as few coefficients as possible. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT (Figure 1). The lower right corner represents higher frequencies and is small enough to ignore with fewer visible distortions. The DC coefficient describes the average illumination level of the input image, and the AC coefficients correspond to different frequencies. Figure 1: Illustration of the DCT's energy compaction.

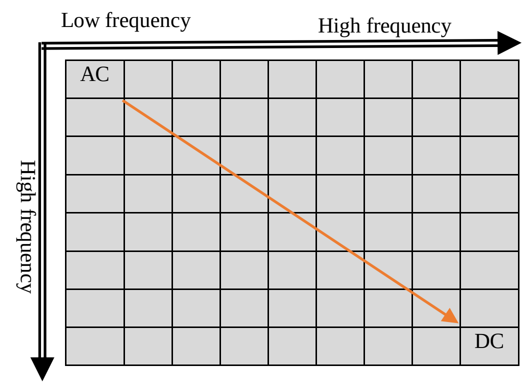

Figure 1: Illustration of the energy compaction of the DCT.

The 2D DCT, C (u,v) of an N×N image Im (x,y) is defined by:

$$c(u,v) = \frac{2}{N} a(u).a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Im(x,y).cos\left[\frac{(2x+1)u\pi}{2N}\right].cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad ...(1)$$

Where:

$$\alpha(u), a(v) = \begin{cases} \sqrt{\frac{2}{N}} & \text{For u, v = 0} \\ \sqrt{\frac{1}{N}} & \text{Otherwise} \end{cases}$$

This is useful for pattern recognition due to its robust energy compression efficiency.

The DCT can contribute to the OCR system with classification techniques such as KNN [11]. DCT features have demonstrated to be efficient for several recognition problems such as face, character, and fingerprint recognition. Words are recognised using features directly extracted from the image. The required steps for this method are shown in Figure 2.

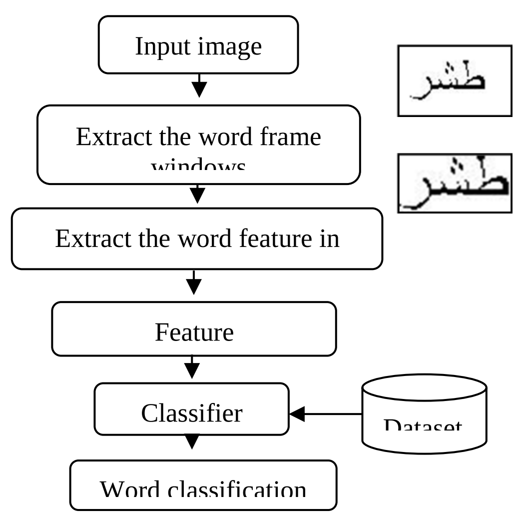

Figure 2: Extracting the word feature by DCT

3.2 Histogram of Oriented Gradient (HOG)

HOG was first used by Dalal and Triggs [17] for human body detection, but it is now highly efficient and commonly described in pattern recognition (PR) and computer vision (CV). HOG counts orientation occurrences of the gradient in a portion of an image, thus describing that appearance. Before applying HOG, the binary images are converted to grayscale, then, for improved accuracy, the local histograms are normalised based on the contrast to make it stable upon illumination variation.
The first step of the HOG algorithm extracts the word frame and scales it to a fixed size, maintaining its original aspect ratio. The size should be suitable for the words to remain readable, but not too small, to eliminate most of the noise and other details that are unnecessary for improving accuracy. For print-style Arabic words, we obtained the best results with a word image size of 45*90. (Results are shown in the tables.)
Before extracting HOG features, the input image is converted to grayscale. A proposed edge detection mask filter is then used to find the image magnitude and direction.

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

In pattern recognition and computer vision, HOG is used as a feature descriptor for detecting objects. This technique counts gradient orientation appearances in a localized part of an image. This procedure is identical to that of edge orientation histograms and scale-invariant feature transform descriptors (SIFT), but differs in that it computes on an intense cell grid and uses normalisation for improved accuracy. At each point, the approximations of the horizontal and vertical gradients and direction are combined as shown in the equations to obtain the gradient norm [13].

$$g = \sqrt{g_x^2 + g_y^2} \qquad \theta = tan^{-1}\left(\frac{g_y}{g_x}\right)$$

The required steps of this method are shown in Figure 3.

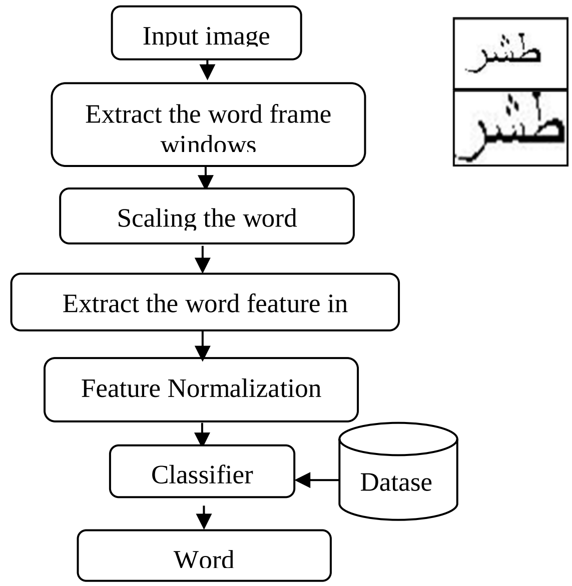

Figure 3: Extracting the word feature by HOG

4. Experiments and Results

4.1 Experiments

This section analyses the proposed methods for word level recognition in HOG and DCT descriptors. Two main experiments were conducted.
Features were extracted directly from the word-image in four font styles and three sizes in Times New Roman, Arial, Arabic Transparent and Simplified Arabic. These fonts are commonly used in books, magazines and research papers as described in the first experiment. In the second experiment, the features were extracted directly from the word-image in three fonts and three sizes written in Arial Unicode MS, Tahoma, and TK Traffic. These fonts are commonly used for adverts and CAPTCHA.

These words consist of three characters, including all character types at the beginning, middle, and the end of the character. The database contains feature vectors for each word. Database feature vectors were generated using a single font as a reference (training model) font for multi fonts. A reference model for font size and type, which received the highest recognition, was used as a model in the database vectors for both font type and font size for the others. The reference model was selected experimentally.

4.2 Results

The proposed method was implemented using MATLAB, version R2015a, a Windows 10 pro-64-bit operating system, with 4GB of RAM, and CPU 1.7 GHz core i3, achieving fast and effective results. The proposed dataset has 8400 word images; each word has 21 images in different font styles and sizes. In the word classification system, 70% of the dataset was used for training (5880) images, and 30% was used for testing (2520) images. It achieved 97.14% using DCT and 89.47% using HOG in the first experiment and 57.32% using DCT and 53.95% using HOG in the second experiment (Tables 1-7). The font type and size was selected as a reference for the others. The first test was to select the font size, and size 28 yielded the best results, although the others were similar; thus, size 28 was adopted. The second step was to select the experimental font type, and the most accurate font was approved as a reference model in the database vectors.

Table 1: Results for Arabic Transparent as a reference

Font type	DCT	HOG	HOG Resize
Arabic Transparent	97.92%	80.25%	92.67%
Arial	96.00%	70.92%	86.08%
Times New Roman	95.50%	75.38%	86.50%
Simplified Arabic	97.75%	80.00%	92.00%
Average	96.79%	76.64%	89.31%

Table 2: Results for Arial as a reference

Font type	DCT	HOG	HOG Resize
Arabic Transparent	96.17%	74.92%	87.58%
Arial	98.83%	84.67%	91.75%
Times New Roman	98.17%	78.42%	92.17%
Simplified Arabic	96.50%	75.92%	89.00%
Average	97.42%	78.48%	90.13%

Table 3: Results for Times New Roman as a reference

Font type	DCT	HOG	HOG Resize
Arabic Transparent	96.17%	74.92%	86.08%
Arial	99.08%	79.42%	90.25%
Times New Roman	98.75%	85.42%	92.67%
Simplified Arabic	96.08%	75.58%	86.92%
Average	97.52%	78.83%	88.98%

Table 4: Results for Simplified Arabic as a reference

Font type	DCT	HOG	HOG Resize
Arabic Transparent	97.42%	78.42%	92.00%
Arial	96.25%	70.08%	86.67%
Times New Roman	95.92%	71.92%	86.67%
Simplified Arabic	97.67%	80.00%	92.42%
Average	96.81%	75.10%	89.44%

Table 5: Results for Arial Unicode as a reference

Font type	DCT	HOG Resize
Arial Unicode	99.17%	92.42%
Tahoma	63.17%	58.92%
K_Traffic	25.25%	24.25%
Average	62.53%	58.53%

Table 6: Results for Tahoma as a reference

Font type	DCT	HOG Resize
Arial Unicode	58.25%	54.42%
Tahoma	98.08%	94.08%
K_Traffic	19.33%	22.50%
Average	58.56%	57.00%

Table 7: Results for K_Traffic as a reference

Font type	DCT	HOG Resize
Arial Unicode	31.42%	26.33%
Tahoma	23.67%	25.75%
K_Traffic	97.50%	86.83%
Average	50.86%	46.31%

Table 8: Classification ratio in the confusion matrices for the first experiment

Ref. Font (Training Model)	DCT	HOG	HOG_Resize
Arabic Transparent	96.79%	76.64%	89.31%
Arial	97.42%	78.48%	90.13%
Times New Roman	97.52%	78.83%	88.98%
Simplified Arabic	96.81%	75.10%	89.44%

Table 9: Classification ratio in the confusion matrices for second experiment

Ref. Font (Training Model)	DCT	HOG_Resize
Arial Unicode_MS	62.53%	58.53%
Tahoma	58.56%	57.00%
K_Traffic	50.86%	46.31%

5. Conclusion

In our paper, we propose using DCT and HOG for word-level recognition. Extracting and classifying features from the word-images using these two descriptors was performed by machine learning algorithms. KNN was used as a classifier. The

algorithm used 70% of the training set and 30% of the testing set, yielding high accuracy.
A KNN classifier was used in the recognition phase that yielded better results for k=1. We evaluated these feature descriptors and classifiers on seven fonts. Four fonts are used for written books, magazines, and research papers, and the other three fonts are used for adverts and text-based CAPTCHA. Each font type included different sizes (20, 24, and 28). Using the proposed features and descriptors by KNN yielded high accuracy in the word datasets.
The results showed that the feature descriptors were powerful for the different fonts and sizes. The fonts used in the first experiment had similar characteristics. In the second experiment, two fonts had similar characteristics, while the third differed completely. As shown in Table 8, Times New Roman was the most accurate font when using DCT, while Arial was the most accurate font using HOG.
Table 9 shows that the highest accuracy was obtained using the Arial Unicode_MS font as a reference for both DCT and HOG. Some ambiguous results often occur in Arabic word classification systems due to their characteristics, where dots are used to distinguish between the meanings of words that have similar primary parts such as طسر - طشر - ظسر - ظشز. Thus, all the previously described features only described the primary part of the word.

However, for an enhanced classification system to correctly classify Arabic words with similar primary parts, the dot specifications (place and number of dots) distinguish between these similar words in cases of ambiguity. To reduce the number of classes, character dots (the secondary parts) were identified and removed.

This reduces the number of classes (words) and increases the accuracy ratio. The dots used in the final classification differentiate between similar words.

6. References

[1] Bunke, H., Wang, P. S. P. (Editors), Handbook of Character Recognition and Document Image Analysis, World Scientific, 1997.

[2] Erik J. Erlandson; John M. Trenkle; Robert C. Vogt;" Word-level recognition of multifont Arabic text using a feature vector matching approach"; COPYRIGHT Society of Photo-Optical Instrumentation Engineers (SPIE), 1996.

[3] Maad Shatnawi,*"Off-line Handwritten Arabic Character Recognition: A Survey"*, *Int'l Conf. IP, Comp. Vision, and Pattern Recognition | IPCV'15* |R. Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982. (book style).

[4] Nemmour H. and Chibani Y., "*Artificial Immune Algorithm for Handwritten Arabic Word Recognition*," *The International Arab Journal of Information Technology, Vol. 14, No. 2, 2017.*

[5] Aziz N., Farah N., Sellami N., and Ennaji A., " *Using Diversity in Classifier set Selection for Arsbic handwrittien recognition*" in proceedings of multi classifiers system , lecture note in CS, Berlin,pp.235-244,2010.

[6] El-Hajj R., Mokbel C., and Likforman-Sulem L., "*Combination of Arabic Handwritten words*". In proceedings of the 19th International Conferemnce on Document Analysis and Recognition,Parana,pp. 959-963,2007.

[7] S.Alma'adeed,C.Higgens,and D.Elliman,"*offline recognition of handwritten Arabic words using multiple hidden markov models*",knowledge based systems ,vol.17,pp. 75-79,2004.

[8] L. Souici-Meslati, M. Sellami, " *A Hybrid NeuroSymbolic Approach for Arabic Handwritten Word Recognition* ", JACIII, Journal of Advanced Computational Intelligence and Intelligent Informatics, FujiPress, Vol. 10, No. 1, pp. 17-25, 2006.

[9] Peter Burrow. "*Arabic handwriting recognition*". Technical report, School of Informatics, University of Edinburgh, 2004.

[10] Abdulwahab Gebril Krayem," *A High-Level Approach To Arabic Sentence Recognition"* A thesis submitted in partial fulfillment of the requirements of Nottingham Trent University for the degree of Doctor of Philosophy September 2013.

[11] Nemmour H. and Chibani Y., "*Off-Line Signature Verification Using Artificial Immune Recognition System*," *in Proceeding of the 10t.*

[12] Bushofa B. M. F. , M. Spann, "Segmentation and recognition of Arabic characters by structural classification", Image and Vision Computing15 (1997) 167-179, 1995.

[13] Sarfaz M. , Nawaz S. N. and Al-Kuraidly A., "Offline Arabic Text Recognition System", International Conference of Geometric Modeling and Graphics (GMAG'03), IEEE Trans. , 0-7695-1985-7/03, 2003.

[14] Hamami L. and Berkani D. , "Recognition System for Printed Multi-Font and Multi-size Arabic Characters", Arabian Journal for Science and Engineering, Vol 27,Number 1B, 57-72, 2002.

[15] Mustafa S. Kadhmand Asst. Prof. Dr. Alia Karim Abdul Hassan "*Handwriting Word Recognition Based on SVM Classifier*" *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 11, 2015*

[16] JIANG, J., WENG, Y. & LI, P. (2006) "*Dominant color extraction in DCT domain*". Image and Vision Computing, 24, 1269-1277.

[17] N. Dalal and B. Triggs. "*Histograms of oriented gradients for human detection*". In CVPR, 2005.

Naser Telesi[1] Has received the B.Sc. at the Faculty of Computer Science, Tripoli University, Tripoli-Libya. MSc at Libyan Academia form 2007. Now, Ph.D. student. major research focuses on the Pattern Recognition, Arabic character recognition.

Mohamed Ahmid[2] Has received his B.Sc. at the Faculty of the Computer Science, Tripoli University, Tripoli-Libya, 1990. M.Sc, Electronic Design Automation, Beijing university of Aeronautics & Astronautics, China, 2001, major research focuses on Software Engineering and internet security

A. Jevremovic[3] PhD, is a full professor at the Singidunum University. So far, he has authored/co-authored number of research papers and made contributions to three books about computer networks, computer networks security and Web development. He is recognized as an Expert Level Instructor at Cisco Networking Academy program.