IJACSA
WHERE WISDOM SHARES

International Journal of Advanced Computer Science and Applications

Special Issue

Special Issue on Selected Papers from Third international symposium on Automatic Amazigh processing (SITACAM' 13)

ISSN 2156-5570(Online)
ISSN 2158-107X(Print)

SAI

www.ijacsa.thesai.org

IJACSA

WHERE WISDOM SHARES

INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS

SAI

THE SCIENCE AND INFORMATION ORGANIZATION

www.thesai.org | info@thesai.org

Associate Editors

CONTENTS

Application of Data Mining Tools for Recognition of Tifinagh Characters

M. OUJAOURA, R. EL AYACHI
Computer Science Department
Faculty of Science and Technology
Sultan Moulay Slimane University
Béni Mellal, Morocco

O. BENCHAREF, Y. CHIHAB
Computer Science Department
Higher School of Technology
Cadi Ayyad University
Essaouira, Morocco

B. JARMOUNI
Computer Science Department
Faculty of Science
Mohamed V University
Rabat, Morocco.

***Abstract*—The majority of Tifinagh OCR presented in the literature does not exceed the scope of simulation software such as Matlab. In this work, the objective is to compare the classification data mining tool for Tifinagh character recognition. This comparison is performed in a working environment using an Oracle database and Oracle Data Mining tools (ODM) to determine the algorithms that gives the best Recognition rates (rate / time).**

***Keywords*—OCR; Data Mining; Classification; Recognition; Tifinagh; geodesic descriptors; Zernike Moments; CART; AdaBoost; KNN; SVM; RNA; ANFIS**

I. INTRODUCTION

The Optical Character Recognition (OCR) is a rapidly expanding field in several areas where the text is the working basis. In general, a character recognition system consists of several phases [1, 2, 3, 4, 8]. The extraction is a phase that focuses on the release of attributes from an image. In this article, the geodesic descriptors and Zernike moments are two approaches used to calculate the parameters. The effectiveness of the system is based on the results given by the classification phase using data mining tools, which is the purpose of this document.

The rest of the paper is organized as follows. The Section 2 discusses the first primordial task of any recognition system. It's the features extraction problems in addition to a brief formulation for geodesic descriptors and Zernike moments as features extraction methods. The Section 3 is reserved for the second important task which is the classification problems using some classifiers based on several algorithms like ANFIS, ANN, SVM, CART, KNN and AdaBoost. Finally, the Section 4 presents the experimental results for the recognition system.

II. FEATURES EXTRACTION

Tifinagh is the set of alphabets used by the Amazigh population. The Royal Institute of Amazigh Culture (IRCAM) has normalized the Tifinagh alphabet of thirty-three characters as shown in Fig. 1.

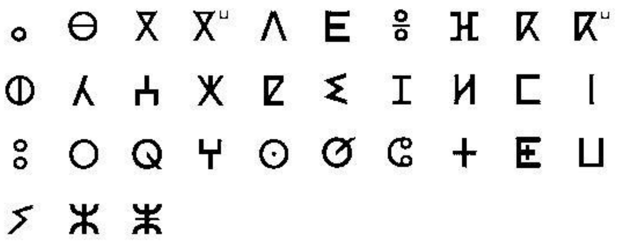


Fig. 1. Tifinagh characters – IRCAM.

The Tifinagh alphabet has several characters that can be obtained from others by a simple rotation, which makes invariant descriptors commonly used less effective. For this reason, we used a combination of Geodesic descriptors [5] and Zernike moments [6].

A. *Geodesic descriptors*

A geodesic descriptor is the shortest path between two points along the spatial deformation of the surface. In the case of binary images; we used a Shumfer simplification which comprises those operations:

- Calculate the number of pixels traveled between the two points;
- Penalize the transition between horizontal and vertical pixels by 1 and moving diagonally by 1.5;
- Choose the optimal path.

We consider the preliminary processing that consists of two standard processes: (i) the noise elimination and (ii) the extremities detection (Fig. 2).

1) Extremities detection

In order to identify the extremities, we use an algorithm that runs through the character contour and detects the nearest points to the corners of the image.

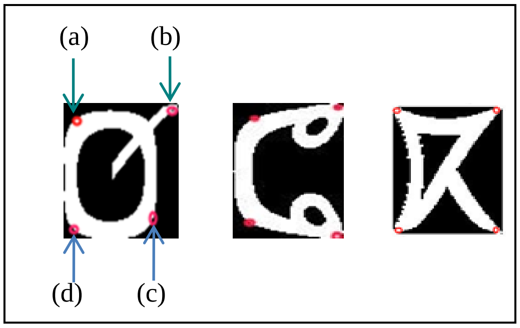


Fig. 2. Example of the extremities of three Tifinagh characters.

2) Geodesic descriptors calculation

We called "geodesic descriptors" the distances between the four detected extremities of the character divided by their Euclidean distances. We set:

- $Dl_M(xy)$: Geodesic distance between x and y;
- $d$xy: Euclidean distance between x and y;
- a, b, c and d: Detected extremities of each character.

And, we call:

- 1st metric descriptor: $D_1 = Dl_M(ab)/d_{ab}$
- 2nd metric descriptor: $D_2 = Dl_M(ac)/d_{ac}$
- 3rd metric descriptor: $D_3 = Dl_M(ad)/d_{ad}$
- 4th metric descriptor: $D_4 = Dl_M(bc)/d_{bc}$
- 5th metric descriptor: $D_5 = Dl_M(bd)/d_{bd}$
- 6th metric descriptor: $D_6 = Dl_M(cd)/d_{cd}$

To ensure resistance to scale change of the proposed descriptors, we divided each geodesic path by the corresponding Euclidean distances.

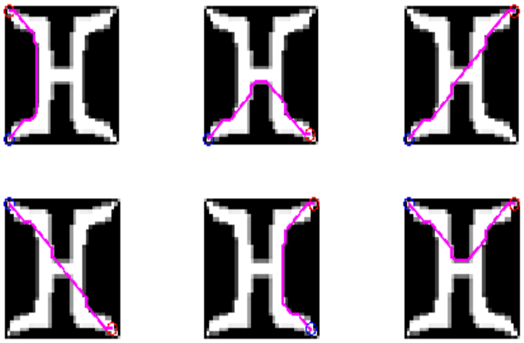


Fig. 3. Example of Geodesic descriptors calculation.

B. Zernike moments

The Zernike moments are a series of calculations that converts an image into vectors with real components representing moments A_{ij}.

By definition, the geometrical moments of a function f(x, y) is the projection of this function on the space of polynomials generated by $x^p y^q$ where $(p, q) \in N^2$. Zernike introduced a set of complex polynomials which form an orthonormal basis inside the unit circle as $x^2 + y^2 \leq 1$. The form of such polynomials is [7]:

$$V^*_{nm}(x,y)=V_{nm}(\rho,\theta)=R_{nm}(\rho).\exp(jm\theta) \qquad (1)$$

Where:

n: a positive or null integer;

m: an integer such that $|m| \leq n$;

r: length of the vector from the origin to the pixel (x, y);

θ: angle between the vector x and p;

R_{nm}: radial polynomial.

V * (x, y): complex polynomial projection of f (x, y) on the space of complex polynomials.

Such polynomials are orthogonal since:

$$\int_{x^2+y^2<=1}\int [V^*_{nm}(x,y)]V_{pq}(x,y)dxdy = 0 \; or \; 1 \qquad (2)$$

The geometrical Zernike moments are the projection of the function f (x, y) describing an image on a space of orthogonal polynomials generated by:

$$A_{nm}=\frac{n+1}{\pi}\sum_x\sum_y f(x,y).V^*_{nm}(\rho,\theta)dxdy \qquad (3)$$

For identification of the image, the Zernike moments modules are used:

$$|A_{nm}|=\sqrt{\mathrm{Re}^2(A_{nm})+\mathrm{Im}^2(A_{nm})} \qquad (4)$$

III. CLASSIFICATION

The choice of the classifier is primordial. It is the decision element in a pattern recognition system. In this context, we compared the performance of six Data Mining algorithms.

A. CART Algorithm

CART (Classification And Regression Tree) builds a strictly binary decision tree with exactly two branches for each decision node. The algorithm partitions or divides recursively the training set using the principle of "divide and conquer" [9].

B. KNN (k-nearest neighbor)

The k-nearest neighbors algorithm (kNN) [10] is a learning method based instances. To estimate the associated output with a new input x, the method of k nearest neighbors is taken into account (with the same way) the k training samples whose entrance is nearest to the new input x, according to distance measurement to be defined.

C. SVM (Support Vector Machines)

Support Vector Machines (SVM) [11] are a class of learning algorithms that can be applied to any problem that involves a phenomenon f that produces output y=f(x) from a set of input x and wherein the goal is to find f from the observation of a number of couples input/output. The problem is to find a boundary decision that separates the space into two regions, to find the hyper-plane that classifies the data correctly. This hyper-plane must be as far as possible of all the examples in order to maximize the margin which is the distance from the nearest point of the hyper-plane.

D. ADaBoost Algorithm

AdaBoost [12] is a meta-algorithm to boost the performance of a classifier based on neural network. The algorithm combines weak assumptions made by a classifier trained on a small training subset which the distribution of elements is reinforced, iteration after iteration, so that learning can be focused on examples that pose the greatest difficulties to the trained classifier.

E. ANFIS

The ANFIS (Adaptive Network Fuzzy Inference System based) [13] where the adaptive network fuzzy inference system uses a hybrid learning algorithm to identify the parameters of the association function of the single output type systems Fuzzy Inference of Sugeno (FIS) [13]. A combination of the square and back-propagation gradient descent methods are used for the parameters of the training of the FIS and functions to model a given set of input / output data. The program ANFIS is available Matlab fuzzy toolbox.

F. ANN (Artificial Neural Networks)

Multi-layers Artificial Neural Networks (ANN) are typically built according to a normalized model that includes 3 or 4 layers in total (i.e. 1 or 2 hidden layers).

The method of changing weight is easy with the algorithm of Rosenblatt, but it involves some learning limitations [14]. In the case of multilayer perceptrons, since the desired output of hidden layers are not known, only those of the last layer are known, it is necessary to propagate the errors responsibility of the last layer to the first layer in the opposite direction of network execution, hence the name back-propagation. Multilayer neural perceptrons use the sigmoid activation function; it allows the necessary nuances for proper use of back-propagation.

IV. RESULTS & DISCUSSIONS

In this paper, the recognition approach of Tifinagh character is tested on the database [15], it is composed of 2175 Tifinagh printed characters with different sizes and styles.

The feature vector of each character is the combination of geodesic descriptors and the seven first Zernike moments. Also, the recognition rates and the execution time have been calculated for the used algorithms: ANFIS, ANN, SVM, CART, KNN and AdaBoost.

The following table (Table 1) shows the obtained results for a test of 2000 isolated characters.

TABLE I. OBTAINED RESULTS: RCOGNITION RATES AND EXECUTION TIME
(PC WITH 2GHZ PROCESSOR AND 4GO OF RAM)

	Zernike	**Geodesic**	**Zernike + Geodesic**	**Time (s)**
CART	41	62	68	4.9
KNN	40	73	81	6.05
ANFIS	63	68	77	23.7
SVM	78	76	93	15
ANN	79	73	94	21
AdaBoost	73	67	93	16

An overview in Table 1 shows that the combination of the two methods (geodesic descriptors and Zernike Moments) used together to calculate the character image parameters in the extraction phase increases the recognition rate. One might also note that the ANN present the best recognition rate, but with a large computation time. The SVM and AdaBoost algorithms show similar results, but with a less hard learning time. The CART algorithm has the best computation time, but with a lower recognition rate.

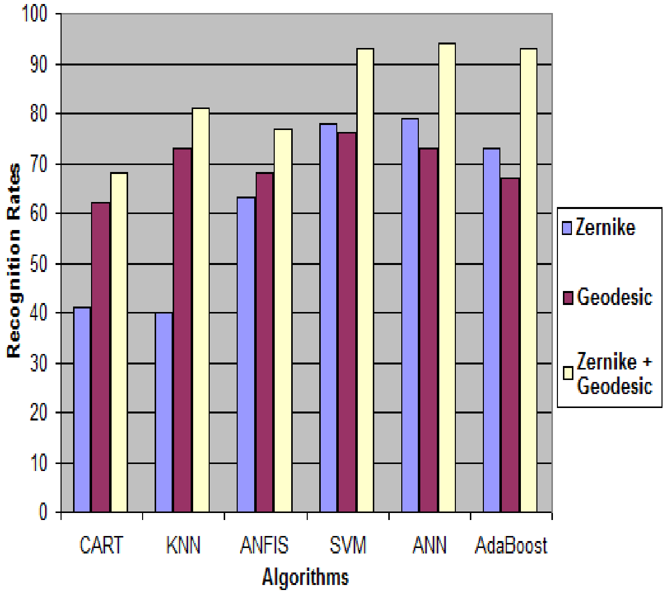


Fig. 4. Comparison of recognition rates for the used algorithms: ANFIS, ANN, SVM, CART, KNN and AdaBoost.

The Figure 4 and 5 gives respectively a comparison of recognition rates and execution time for the used algorithms that are ANFIS, ANN, SVM, CART, KNN and AdaBoost.

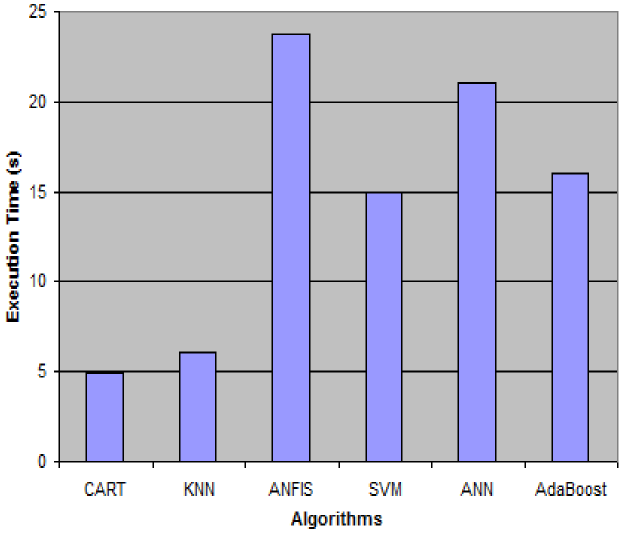


Fig. 5. Comparison of the execution time for the used algorithms: ANFIS, ANN, SVM, CART, KNN and AdaBoost.

All the tests are performed using a database containing a set of 2000 images of isolated characters [15]. The proposed system has been implemented and tested on a core 2 Duo personnel computer with 2 Ghz processor and 4Go of RAM using Matlab software.

V. CONCLUSION

This work presented a Tifinagh character recognition system that uses Data mining tools (ANFIS, ANN, SVM, CART, KNN and AdaBoost) in the classification phase. Also the geodesic descriptors and Zernike moments are adopted to extract the attributes in the features extraction phases. It can be concluded that each one of the tested algorithms has advantages and disadvantages regarding to the recognition rate or execution time. In the future work, the performance of the combination of these classification Data mining algorithms will be studied.

REFERENCES

[1] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass. (2010). Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata, ICGST-GVIP Journal, Volume 10, Issue 2, June 2010.

[2] O. Bencharef, M. Fakir, N. Idrissi, B. Bouikhalen et B. Minaoui. (2011). Application de la géométrie riemannienne à la reconnaissance des caractères Tifinaghe, Agadir-Maroc, 06-07 Mai 2011. pp : 179 :188.

[3] R. El Ayachi, M. Fakir et B. Bouikhalen. (2012). Transformation de Fourier et moments invariants appliqués à la reconnaissance des caractères Tifinaghe, Revue eTI, Numéro 6. 2012.

[4] .El Ayachi, K. Moro, M. Fakir and B. Bouikhalene. (2010). On The Recognition Of Tifinaghe Scripts, JATIT, vol. 20, No. 2, pp: 61-66, 2010.

[5] O. Bencharef, M. Fakir, B. Minaoui et B. Bouikhalene. (2011). Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence, 1-5.

[6] A. Prata, W.V.T. Rusche, Algorithm for computation of Zernike polynomials expansion coefficients, Appl. Opt. 28, pp. 749–754, 1989.

[7] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees Wadsworth, 1984.

[8] Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass. (2011). Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character. International Journal of Advanced Science and Technology Vol. 33, Août, 33-50.

[9] C.W. Chong, P. Raveendran, R. Mukundan, A comparative analysis of algorithms for fast computation of Zernike moments, Pattern Recognition 36 (3) , pp. 731–742, 2003.

[10] Oren Boiman, Eli Shechtman and Michal Irani, In Defense of Nearest-Neighbor Based Image Classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008.

[11] K.-B. Duan, S.S Keerthi, Which is the best multiclass SVM method? An empirical study. Technical Report CD-03-12, Control Division, Department of Mechanical Engineering, National University of Singapore, 2003.

[12] Paul Viola and Michael Jones. Robust real-time object detection. In International Journal of Computer Vision, 2001.

[13] J.-S.R. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference Systems, IEEE Trans. Systems, Man & Cybernetics 23 (1993), pp: 665—685.

[14] O. Lezoray. Segmentation d'images par morphologie mathématique et classification de données par réseaux de neurones : Application à la classification de cellules en cytologie des séreuses. Thèse de doctorat, Université de Caen (2000).

[15] Y. Ait Ouguengay, M. Taalabi, "Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage", Systèmes intelligents-Théories et applications, Paris: Europia, cop. 2009 (impr. au Maroc), ISBN-102909-285553, 2009.

Annotation and research of pedagogical documents in a platform of e-learning based on Semantic Web

S. BOUKIL, C. DAOUI, B. BOUIKHALENE, M.FAKIR
Information Processing and Telecommunication Teams
FST, Sultan MoulaySlimaneUniversity,
BeniMellal, Morocco.

***Abstract*—E-learning is considered as one of the areas in which the Semantic Web can make a real improvement whatsoever in finding information, or reusing of educational resources or even personalized learning paths. This paper aimsto develop an educational ontology that will be used to annotate learning materials and pedagogical documents.**

Keywords—Ontologie ; Web sémantique ; XML ; RDF ; RDFS ; OWL ; métadonnées ; enseignement à distance.

I. INTRODUCTION

At its creation by Tim Berners Lee in the early 90s, the goal of the Web was to allow any user to access and share large amounts of information on the net, and very quickly the Web has achieved its objectives. Therefore, we have a large volume of information, but no control of content. The Semantic Web is the solution proposed by the W3C (World Wide Web Consortium), this new vision of the Web aims to make Web resources understandable not only by humans but also by machines. To achieve this goal, W3C begins to develop new and better languages: XML (Extensible Markup Language), RDF (Resource Description Framework), OWL (Web Ontology Language) ... etc. The Semantic Web has used engineering knowledge to provide a tool for knowledge representation and it seems that ontologies were most suitable for such an environment. An ontology is a conceptual system that enables the sharing of knowledge between humans and computers and between computers. The work presented in this paper lies at the intersection of the Semantic Web with the field of distance education. This work aims to achieve a double goal: the first one is to design an application ontology that describes the educational materials used for university education, and the second one concerns the development of an application for annotation and retrieval of documents in educational exploitation of metadata that describe them.

II. SEMANTIC WEB

A. Semantic Web Hierarchy Model

The Semantic Web is a vision of the Web in which information is given explicit meaning to machines facilitating the processing and integration of information on the Web.

The Semantic Web is built on the ability of XML to define customized tagging schemas and the flexibility of RDF to represent the data. If the machines are supposed to do useful reasoning tasks on these documents, the language must go beyond basic semantics of RDF Schema. OWL has been designed to meet that need. OWL is part of a scalable stack of W3C recommendations with respect to the Semantic Web [1].

Fig. 1 shows the standardized technologies of the Semantic Web, which is cited below:

1) XML: provides a syntactical surface for structured documents [2]., achieves interoperability of data to different platforms or systems using different languages. XML has been designed for document exchange, and define the data structure [3].

2) XML Schema: is the substitute of the DTD (Document Type Definitions). Itself uses XML syntax, but more flexible than DTD, provides more data types, services for XML documents more efficiently. If theXML is standardized data formatthen the XML Schema defines the data types of an XML document [3].

3) RDF is a standard for describing Web resources proposed by the W3C, as its name implies, RDF (Resource Description Framework) is a metalanguage for resource description framework, to make the information necessary to search engines more "structured" and, more generally, to all necessary tool for automated analysis of web pages [6]. RDF uses a particular terminology to indicate the various parts of statements. Specifically, the part that identifies the thing mentioned in the statement is called the subject (subject), the part that identifies the property or characteristic of the subject is called the predicate (predicate), the part that identifies the value of this property is called the object (object). Thus RDF statements are in the form of a triplet <Subjet, Prédicat, Object>. Graph model is used to query and process RDF [3]. In this model, a statement is represented by:

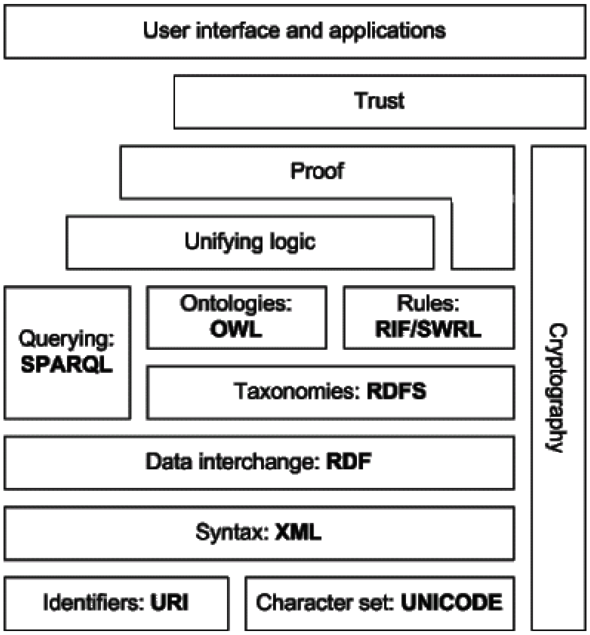


Fig.1. Semantic Web Layers

a) a node for the subject.

b) a node for the object.

c) an arc, directed from the node subject to the node object, for the predicate.

So any RDF statement could be represented by the graph shown in Fig.2.

4) RDF Schema: is a vocabulary for describing properties and classes of RDF resources [1]. It uses a kind of understandable system by the computer to define vocabularies resources described [3].

5) Ontology: RDF provides only basic relations of description, reasoning ability is limited. The context of the Semantic Web develops on the ontology layer and the layer of logical reasoning based on RDF to support knowledge representation and reasoning based on semantics. Ontology is a philosophical term introduced in the nineteenth century that characterizes the study of beings in our universe. In computer science, an ontology is a structured representation of domain knowledge in the form of a network of concepts linked by semantic links. The ultimate goal of the ontology is to show the implicit information accurately. W3C has defined OWL (Ontology Web Language) as the standard recommended language for describing ontologies.

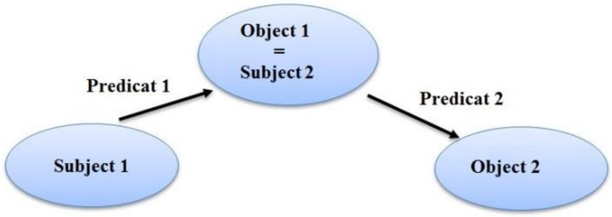


Fig.2. RDF statement in graph model.

6) OWL: like RDF, OWL is an XML language enjoying the universal syntax of XML. It adds the ability to make comparisons between properties and classes: identity, equivalence, contrast, symmetry, cardinality, transitivity, disjunction, etc. The W3C provides OWL with three sublanguages with increasing capacity of expression, and it is as necessary that we choose the appropriate language.

d) OWL Lite is the simplest sublanguage of OWL, it is intended to represent hierarchies of simple concepts.

e) OWL DL is more complex than the previous one, it is based on description logic, hence its name (OWL Description Logics). It is adapted to reasoning, and ensures the completeness of reasoning and its decidability.

f) OWL Full is the most complex version of OWL, designed for situations where it is important to have a high level of ability to describe, even if they cannot guarantee the completeness and decidability of calculations related to the ontology [7].

III. ONTOLOGIESAND E-LEARNING

A. Metadata :

We can define metadata as "data about data" treatable by a machine [8], in the case of pedagogical documents, document content is data and information on the authors, their interests and their pedagogical goals are metadata.

B. Metadata based on ontologies

In the context of the Semantic Web, ontologies provide specifically rich semantics, better than any other method of knowledge representation known. In a research topic of educational content on a platform of education, basing on the conceptual vocabulary defined in ontology may help to improve the accuracy of this research by avoiding ambiguities in terminology and allowing inferences decreasing noise and increasing relevance.

C. Ontologies for e-learning

In June 2000 the European Commission defines e-learning as "the use of new multimedia technologies and the Internet to improve the quality of learning by facilitating access to resources and services, as well as exchanges and remote collaboration". E-learning, and as other Web services, can benefit from the new vision of the Semantic Web while relying particularly on the potential of ontologies.

1) Need of e-learning systems

The different needs of e-learning system that ontologies play a role to fulfill, can be summarized in:

a) Need for archiving and information research: An e-learning application is put online through the use of Web. Given the diversity and the exponential growth of pedagogical resources used in an e-learningtype of education, it is increasingly difficult to find relevant pedagogical materials.

b) Need to share: problem with keywords to use to search for learning materials.

c) Need for reusingof pedagogical objects: Given the volume of increasingly growing pedagogical materials *available on the net, just a small number of pedagogical objects are reusable. The search and selection of relevant text fragments, figures, exercises, from a document with the aim of their reuse in a new document has become almost impossible.*

d) Need for customization and adaptation: A system of e-learning is for a community of users who do not have the same expectations, knowledge, skills, interests, etc. They are not able to understand or accept documents, except those of the organization, content and presentation are adapted to their needs.

IV. Conception And Implementation

A. Construction process

The process of building ontology exploitable within a computer system is based on two steps: ontologization and operationalization. The ontologization consists of building a conceptual ontology. This means that we intend to provide a description of the target world, faced with this task we take into account the various sources of knowledge: glossary of terms, other ontologies, texts, interviews with experts, etc. The operationalization consistsin encoding conceptual ontology obtained, using an operational language of knowledge representation (provided with mechanisms of inferences). It should be noted that this process is not linear and that many trips are a priori necessary to develop an ontology adapted to operational needs.

B. Conception of the application ontology

1) Choice of a construct methodology

To build the application ontology, there are different methods of construction, and the choice between these methods is performed according to our needs.The method developed by [Bernaras et al, 1996] was used in this work, it is based on three steps:

- Specify the application based on the ontology, in particular terms to collect and tasks to execute using this ontology.
- Organize terms using meta-categories: concepts, relations, attributes, etc.
- Refine and organize the ontology according to the principles of modularization and hierarchical organization.

This choice can be justified by two reasons:

- This method is suitable for the application ontologies rather than domain ontologies.
- It is structured around a set of terms that must be transformed into an ontology.

2) Construction principles

a) Clarity and objectivity [9]: all terms used in this ontology have been associated with definitions.

b) Completeness [9]: to respond to this principle definitions of concepts and relations of our ontology have been associated with conditions, others have been associated with necessary and sufficient conditions, but of course depending on the possibility to define these conditions.

c) Maximum ontological extensibility [9]: the definition of a term explains just the term itself, its definition cannot be the same except for a more general term, or a more specialized term.

d) Principle of ontological distinction [10]: the concepts in the ontology are sufficiently disjoint.

e) Minimum semantic distance [11]: there is a minimum distance between the concepts children of the same parents.

3) Representation of concepts :

Fig. 3 shows a hierarchical representation of the concepts used to model the pedagogical universe in our ontology.

4) UML class diagram

The class diagram shown in Fig. 4 illustrates concepts, attributes and relations linking concepts together:

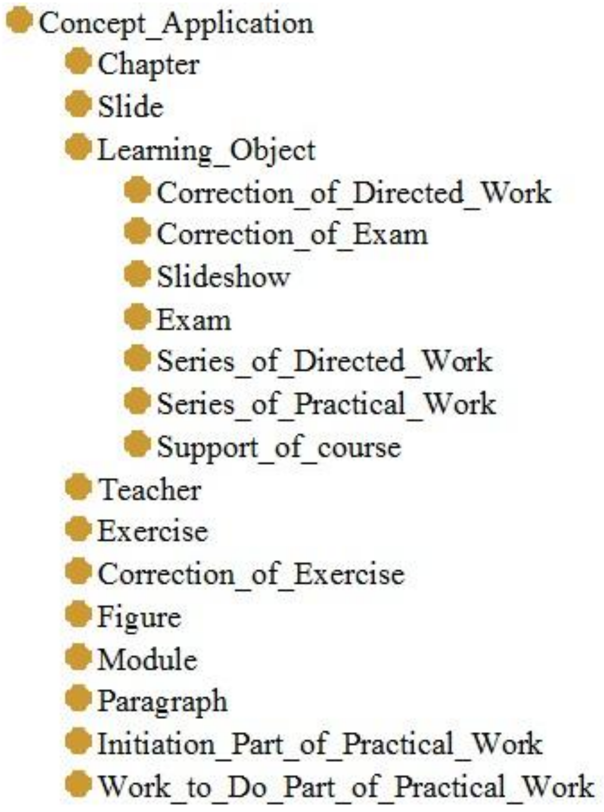


Fig.3. Hierarchical representation of the ontology

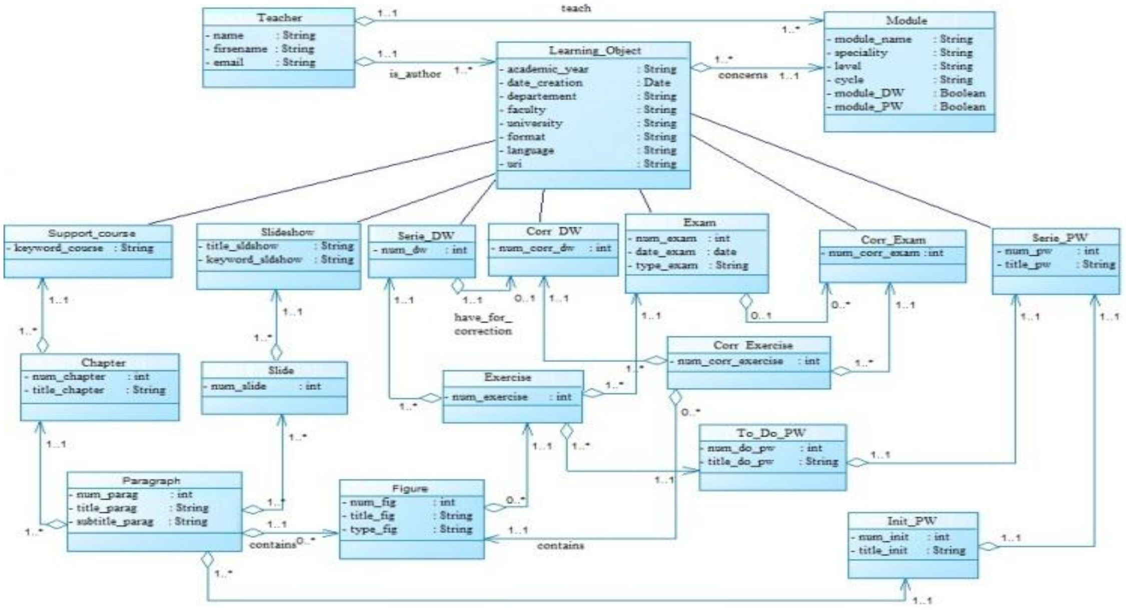


Fig.4. UML class diagram of the ontology

C. *Implementation and use of the ontology:*

The ontology editor « Protégé version 3.1.1 », was used to edit our ontology with the aim to automatically generate the OWL code corresponding as well as to generate the HTML documentation. A fragment of OWL code generated is illustrated in Fig. 5.

The process of ontology building can be integrated into the life cycle of an ontology as shown in Figure 6 [12].

```
<owl:onProperty>
<owl:ObjectPropertyrdf:ID="inverse_of_est_compose_chapitr
e"/></owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction><owl:onProperty>
<owl:DatatypePropertyrdf:ID="num_chapitre"/>
</owl:onProperty>
<owl:cardinality
rdf:datatype=" http://www.w3.org/2001/XMLSchema#int">1<
/owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:cardinalityrdf:datatype=" http://www.w3.org/2001/XML
Schema#int" >1</owl:cardinality>
```

Fig.5. Fragment of OWL code generated

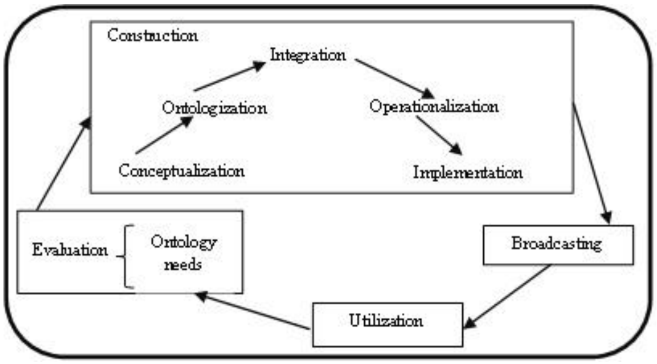


Fig.6. Life cycle of an ontology

D. *Application platform:*

The platform was developed in JSP (Java Server Pages). The JSP API is part of J2EE (Java 2 Enterprise Edition), it gives to developers the means to develop Web applications in a simple and powerful way. As it allows to separate the programmatic logic (java code) from presentation (HTML tags). It is also essential to use the framework Jena which is a Java framework oriented toward the development of Semantic Web applications, it presents a set of tools (API) open source developed by HP Labs Semantic Web Program to read and manipulate ontologies described in RDFS or OWL, and to apply some inference mechanisms. There are three types of users of this platform: administrator, teacher and student. Different tasks that can perform each type of user are shown in the following use case diagram:

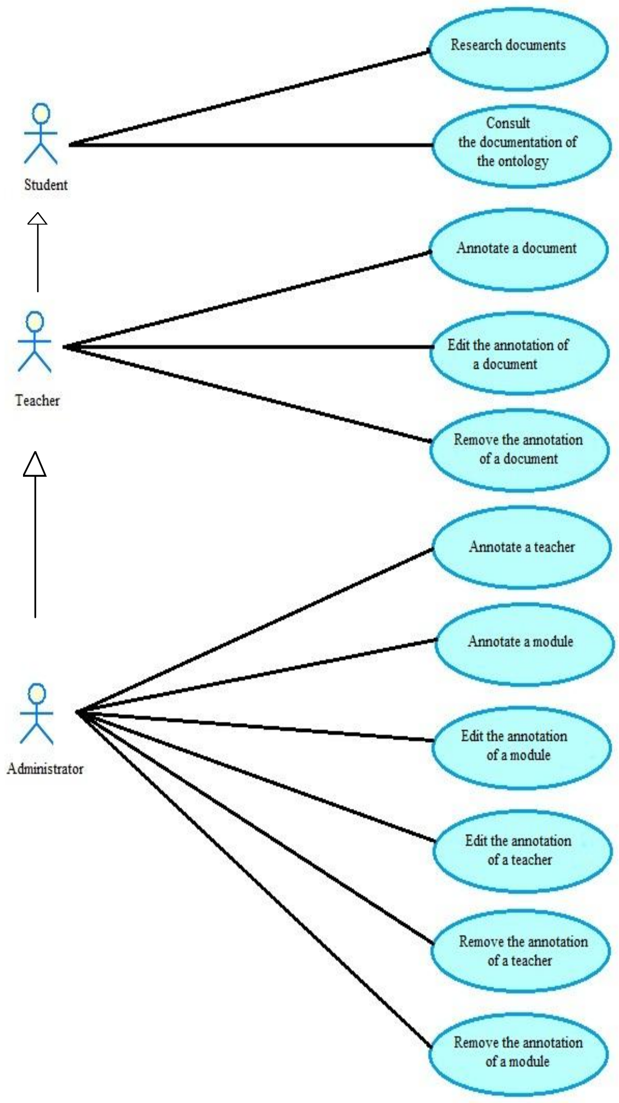
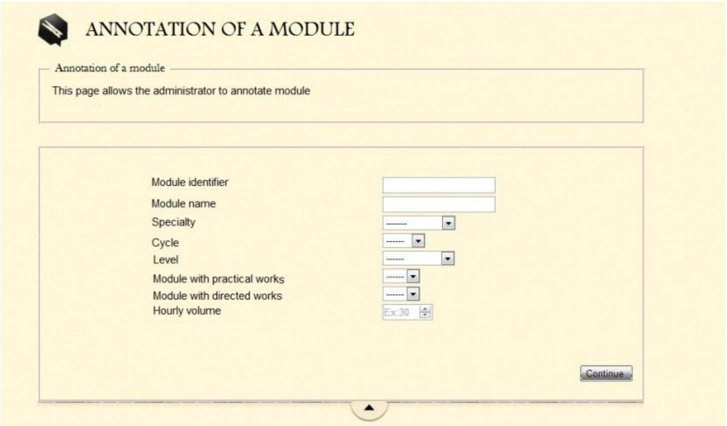

Student
Teacher
Administrator
Research documents
Consult the documentation of the ontology
Annotate a document
Edit the annotation of a document
Remove the annotation of a document
Annotate a teacher
Annotate a module
Edit the annotation of a module
Edit the annotation of a teacher
Remove the annotation of a teacher
Remove the annotation of a module

Fig.7. Use case diagram

It should be noted that the approach to annotate the documents consisted of:

- Add the metadata that describe a document to the file OWL that encodes the application ontology.
- Store documents in a specific location on the server.
- Manage access to documents with metadata "URI".

Fig. 8, shows an example of annotation (annotation of a module).



ANNOTATION OF A MODULE

Annotation of a module

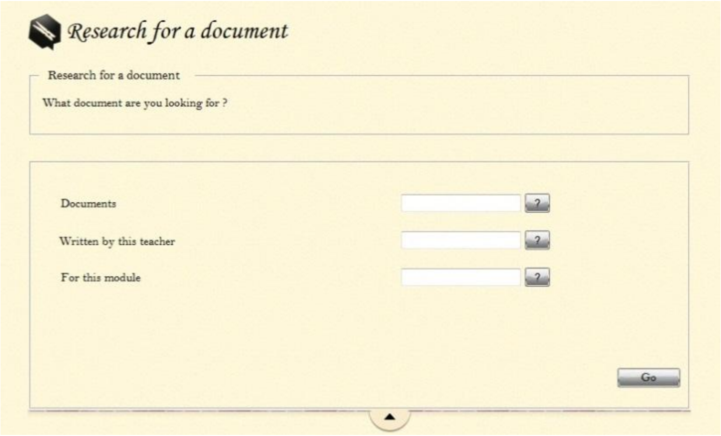
This page allows the administrator to annotate module

Module identifier
Module name
Specialty -------
Cycle ------
Level ------
Module with practical works ------
Module with directed works ------
Hourly volume Ex:30

Fig.8. Annotation of a module

Regarding the search for a document, it takes three options, one option depending on document metadata, an option according to the author of the document and the last option as the taught module (Figure 9).

To exploit the ontology in the application, a dedicated query language is essential; SPARQL (his name is an acronym for SPARQL Protocol And RDF Query Language). SPARQL Conducts research on RDF graphs [4]. Thus, SPARQL is a programming interface between applications, an API as a Web services standard, and it opens the way to a universal API for querying structured data, in which the semantics of the query is no longer situated in the API, which limits the possibilities, but in the query itself. [5]



Research for a document

Research for a document

What document are you looking for ?

Documents ?
Written by this teacher ?
For this module ?

Fig.9. Research for a document

V. Conclusion And Percpectives

This paper shows what is the Semantic Web is, and on what it is based in terms of standards and languages, and also covers the notion of ontology and the contribution of ontologies in Semantic Web context.

The objective of this paper was the conception of ontology to provide a vocabulary for the annotation and research of documents in a platform for distance education. However, this work is not perfect and can be improved in several areas, such as:

- Develop other ontologies and combine them with those made here to enrich the vocabulary used for annotating and research.
- Reuse this ontology in other platform based on semantic web techniques.
- Add the intelligent agent technologies at the application to provide reactivity with users

ACKNOWLEDGMENT

The authors of this article are grateful to all their colleagues who have contributed to the achievement of this article.

REFERENCES

[1] S.Boutemdjit, "Web sémantique et e-learning.", cours ITF 6162, University of Montréal, Winter 2004.

[2] P.Laublet, C.Reynaud and J.Charlet, "Sur quelques aspects du web sémantique", Universityof Paris-Sorbonne - CNRS (LaLLIC), University of Paris-Sud - CNRS (L.R.I.) & INRIA, University of Paris-X Nanterre, 2002.

[3] R.Liu andM.Deng, "A research on e - learning resources construction based on semantic web", College of Educational Science,China West Normal University,Nanchong, Sichuan 637000, China. 2012.

[4] http://web-semantique.developpez.com/tutoriels/jena/arq/introduction-sparql/

[5] http://www.lespetitescases.net/les-technologies-du-web-semantique

[6] http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/

[7] X.Lacot,"Introduction à OWL, un langage XML d'ontologies web", 2005.

[8] Y.Goita, "Les applications de XML à la production d'objets pédagogiques interactifs " ,2001.

[9] T.Gruber, "A translation approach to portable ontology specifications". Knowledge Acquisition, 5(2), 199-220, 1993.

[10] S.Borgo, N.Guarino and C.Masolo, " Stratified ontologies: the case of physical objects". Paper presented at the ECAI96. Workshop on Ontological Engineering, Budapest,1996.

[11] J. Arpirez, A. Gómez-Perez, A. Lozano andS. Pinto, "(ONTO)2Agent: An ontology-based WWW broker to select ontologies". Paper presented at the Workshop on Applications of Ontologies and PSMs, Brighton, England, 1998.

[12] F. Gandon,"Ontology engeneering: a survey and a return on experience", rapport de recherche 4396, INRIA, 2002.

Hierarchical Algorithm for Hidden Markov Model

SANAA CHAFIK
Laboratory of modelisation and calcul.
University Sultan Moulay Slimane
Béni Mellal, Moroco

DAOUI CHERKI
Laboratory of modelisation and calcul.
University Sultan Moulay Slimane
Béni Mellal, Moroco

***Abstract*—The Forward algorithm is an inference algorithm for hidden Markov models, which often leads to a very large hidden state space. The objective of this work is to reduce the task of solving the Forward algorithm, by offering faster improved algorithm which is based on divide and conquer technique.**

***Keywords* —Hidden Markov Model; Forward; Divide and Conquer; Decomposition; Communicating Class.**

I. INTRODUCTION

A Hidden Markov Model (HMM) is a doubly stochastic process, one of whose components is an unobservable Markov chain; it is used extensively in pattern recognition, speech recognition [1, 2], Handwriting recognition [3, 4, 5], computational biology [6], Machine translation [7]. During the use of HMMs we are led to treat three fundamental problems: Evaluation, decoding and learning [8].

The HMMs fall most often on a large dimension state space that makes interesting use of the Divide and Conquer technique. The principle is based on dividing a large problem into several similar problems which avoids the curse of dimensionality. In this direction, we will propose a decomposition method and improved algorithm to solve large HMMs.

This paper is organized as follows: We briefly present in the second section a general introduction to Hidden Markov Models and their fundamental problems. In the third section the Forward algorithm is described. The problematic and the solution are given in the next section. Finally, we propose an improved version and the complexity of the Forward algorithm in the fourth section.

II. HIDDEN MARKOV MODEL

A. Definition

The HMM is defined by a tuple [9, 10] {N, M, A, B, Π }:

The Model is formed by N states $S=\{S_1,S_2,...,S_N\}$.

The M observations $O=\{O_1,O_2,...,O_M\}$.

The matrix of transition probabilities is denoted by A=$\left[a_{ij}\right]$,where:

$$a_{ij}=P(s_{t+1}=S_j \mid s_t=S_i) \text{ where } \sum_{j\in S} a_{i,j}=1 \text{ , } \forall i\in S \quad (1)$$

a_{ij} Specifies the probability of transitioning from state i to state j.

The observation probability matrix or emission probability, denoted by $B=\left[b_j(m)\right]$:

$$b_j(m)=P(o_t=O_m \mid s_t=S_j) \quad (2)$$

$b_j(m)$ Represents the probability of emitting symbol O_m at the instant t by the state S_j.

The probability distribution of the initial state is denoted by $\Pi=\left[\pi_i\right]$:

$$\pi_i=P(s_1=S_i) \text{ where } \sum_{i\in S}\pi(i)=1 \quad (3)$$

π_i specifies the probability of being in state i at time zero.

B. Fundamental problems of HMM

There are three basic HMM problems that must be solved:

Evaluation: Given an observation sequence $O=O_1,O_2,...,O_T$ and a model $\lambda=\{\Pi,A,B\}$, what is the probability of the model generating that observation sequence?

Decoding: Given the observation $O=O_1,O_2,...,O_T$ and an HMM model $\lambda=\{\Pi,A,B\}$, how do we find the state sequences that best explain the observation?

Learning: How do we adjust the model parameters $\lambda=\{\Pi,A,B\}$, to maximize $P(O\mid\lambda)$?

III. FORWARD ALGORITHM

For each pair (state, time) we associate the Forward variable $\alpha_t(i)$ given in equation (5) which represents the probability of the partial observation sequence $\{O_1,...,O_t\}$ (until time t) and state S_i at time t, given the model λ.

$$\alpha_t(i)=P(o_1=O_1,...,o_t=O_t,s_t=S_i \mid \lambda) \quad (5)$$

Algorithm 3.1.

Step1: Initialization, let $i\in S$

$$\alpha_1(i)=\pi_i\times b_i(o_1) \quad (6)$$

Step2: Induction, let $t\in[1,T-1], j\in[1,N]$

$$\alpha_{t+1}(j)=\left[\sum_{i\in S}\alpha_t(i)\times a_{ij}\right]b_j(o_{t+1}) \qquad (7)$$

Step3: Termination

$$P(O\mid\lambda)=\sum_{i\in S}\alpha_T(i) \qquad (8)$$

IV. PROBLEMATIC AND SOLUTION

A. The curse of dimensionality

The statistical learning algorithms such as those dedicated to hidden Markov chains they are suffering from the exponentially increase of the cost when the volume of data grows, which is known as the curse of dimensionality [11].

B. Divide and Conquer

The term Divide and Conquer algorithmic technique [12, 13] yields elegant, simple and very efficient algorithms, their principle is based on dividing a large problem into several similar problems which avoids the curse of dimensionality.

C. Principe of decomposition

Decomposition technique [14] consists of the following steps. First, the algorithm of decomposition to levels is applied, thereafter the restricted HMMs are constructed, eventually, we combine all the partial solutions in order to construct the global solution of the HMM.

In this section, we consider HMM, Let G=(S, U) be the graph associated with the HMM, that is, the state space S represents the set of nodes and $U=\{(i,j)\in S^2 : a_{ij}>0\}$ the set of directed arcs.

1) Decomposition into levels

The state space can be partitioned into strongly connected classes $C_1, C_2, ..., C_H$. Note that the strongly connected classes are defined to be the classes with respect to the relation on G defined by: i is strongly connected to j if and only if i=j or there exist a directed path from i to j and directed path from j to i. There are many good algorithms in graph theory for the computations of such partition, see [15]. Now we construct by induction the levels of the graph G. The level L_0 is formed by all classes C_i such that C_i is closed, that is, any arc emanating from C_i has both nodes in C_i. The path level L_p is formed by all classes C_i such that the end of any arc emanating from C_i is in some level $L_{p-1}, L_{p-2}, ..., L_0$.

Remark 4.1. Let C_i be strongly connected class in the level L_p then C_i is closed with respect to the restricted HMM to the state space $S-(L_{p-1}, L_{p-2}, ..., L_0)$.

It is clear that, from Remark 4.1, the following algorithm finds the levels

Algorithm 4.1.

```
Beginning
$\Omega \leftarrow S$
$p \leftarrow 0$
$L_p = \{C_i : C_i$ closed at $\Omega\}$
Repeat
    $\Omega \leftarrow \Omega \setminus L_p$
    If $\Omega \neq \varnothing$ then
        $L_{p+1} = \{C_i : C_i$ closed for HMM restricted to $\Omega\}$
        $p \leftarrow p+1$
    End if
Until $\Omega=\varnothing$
```

Example: The classes $C_i; i=1,...,5$ shown in Fig. 1, construct three levels L_0, L_1 and L_2, the class C_1 which is situated in the level L_1, is closed by way of contribution of the level L_2.

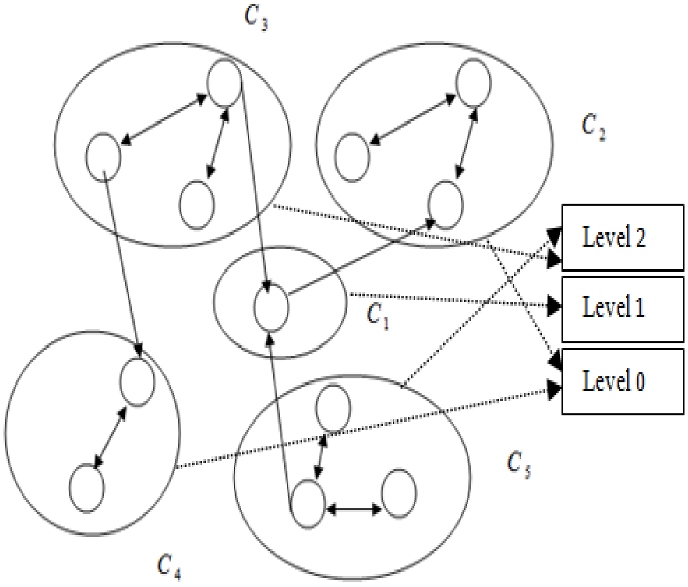


Fig. 1. Construction of levels

2) Restricted HMM for decoding problem using Forward algorithm

In what follows, we construct, by induction, the restricted HMMs corresponding to each level $L_p; p=n,...,0$ .Let (C_{pk}), $k\in\{1,2,...,K(p)\}$ be the K^{th} strongly connected class corresponding to the nodes in level p, where K(p) represent the maximum of the classes included in the level p.

a) Construction of the restricted HMM in level L_n

For each $k = 1,2,\ldots K(n)$, we denote by HMM_{nk} the restricted HMM corresponding to the class C_{nk} that is the restricted HMM in which the state space restricted is $S_{nk} = C_{nk}$; the same M symbols $O_1, O_2, ..., O_M$; the matrix of transition probabilities and the matrix of observation probability are restricted to S_{nk}.

b) Construction of the restricted HMM in level $L_p, p = n-1,...,0$

For each $k = 1,2,\ldots K(p)$ and $p = n-1,...,0$ we denote by HMM_{pk} the restricted HMM corresponding to the class C_{pk}. Let $\Gamma^{-1}(C_{pk})$ be the set of predecessors for each state $i \in C_{pk}$ .The restricted HMM_{pk} defined by:

The state space: $S_{pk} = C_{pk} \cup \Gamma^{-1}(C_{pk})$.

The matrix of transition probabilities A: for each $\mathrm{j, i} \in \mathrm{S}_{pk}$, $\mathrm{A}=\left[a_{ij}\right]_{\mathrm{i}\in \mathrm{S}_{pk}}$ where $a_{ij} = P_{pk}(j \mid i)$ if $j \in C_{pk}$.

The same symbols $O = \{O_1, O_2, ..., O_M\}$.

The probability distribution of the initial state $\Pi = [\pi_i]_{\mathrm{i}\in \mathrm{C_{pk}}}$, where $\pi(i) = P(s_1 = S_i)$ if $\mathrm{i} \in \mathrm{C_{pk}}$.

The matrix of observation probability $B = \left[b_j(m)\right]_{\mathrm{j}\in C_{pk}}$, where $b_j(m) = P(o_t = O_m \mid s_t = S_j)$ if $\mathrm{j} \in C_{pk}$.

V. Improved Forward Algorithm

We denote by $\alpha_{t,pk}(i)$, $\mathrm{t} \in \{1,...,\mathrm{T}\}$, p=n,...,0 and k={1,..,K(p)} the Forward variable in state $i \in C_{pk}$.

Lemma 5.1. Let $\mathrm{j} \in \mathrm{C}_{pk}$, the Forward variable for j at time t+1 is defined by:

$$\alpha_{t+1,pk}(j) = \left[\sum_{\mathrm{i} \in \mathrm{S}_{pk}} \alpha_{t,pk}(i) \times a_{ij}\right] b_j(o_{t+1}) \qquad (9)$$

Proof. From equation (7) to calculate the Forward variable $\alpha_{t+1}(j)$ we need only the states i such as $P(x_t = j \mid x_{t-1} = i) \neq 0$, $\forall$ $\mathrm{j} \in C_{pk}$, these states belongs to the original set states of the class C_{pk} or $i \in \Gamma^{-1}(C_{pk})$.

Remark 5.1. To calculate the Forward variable $\alpha_{t+1,pk}(j)$ we need the Forward variable $\alpha_{t,pk}(i)$ for each $i \in \Gamma^{-1}(C_{pk})$, therefore, we always need some values that have been already calculated in the upper levels.

Algorithm 5.1.

Step1 : Initialization

For p=n,...,0 and $k = 1,2,\ldots K(p)$; let $\mathrm{i} \in S_{pk}$

$$\alpha_{1,pk}(i) = \pi_i \times b_i(o_1) \text{ if } \mathrm{i} \in \mathrm{C}_{pk} \qquad (10)$$
$$\alpha_{1,pk}(i) = \alpha_{1,mk'}(i) \text{ if } \mathrm{i} \in \Gamma^{-1}(\mathrm{C}_{pk}),\ (\mathrm{i} \in \mathrm{C}_{mk'}, m > p)$$

Step2 : Iteration

For $t \in [1,\ T-1]$, p=n,...,0 and $k = 1,2,\ldots K(p)$; let $j \in S_{pk}$

$$\alpha_{t+1,pk}(j) = \left[\sum_{\mathrm{i} \in \mathrm{S}_{pk}} \alpha_{t,pk}(i) \times a_{ij}\right] b_j(o_{t+1}) \text{ if } \mathrm{j} \in \mathrm{C}_{pk}$$
$$\alpha_{t+1,pk}(j) = \alpha_{t+1,mk'}(j) \text{ if } \mathrm{j} \in \Gamma^{-1}(\mathrm{C}_{pk}),\ (j \in \mathrm{C}_{mk'}, m > p) \qquad (11)$$

Step2 : Termination

$$P(O \mid \lambda) = \sum_{p=0}^{n} \sum_{k=1}^{K(p)} \sum_{\mathrm{i} \in \mathrm{C}_{pk}} \alpha_{T,pk}(i) \qquad (12)$$

VI. Complexity

The classical Forward algorithm generate N (N +1) (T-1) + N multiplication and N (N-1) (T-1) addition, it takes on the order of N^2T computations, which represents the quadratic complexity in the number of state. Whereas, the complexity of improved Forward algorithm can be calculated by using Akra Bazzi theorem [16] (A generalization to the well known Master Theorem [17]) which allows calculating the complexity for this type of problem. Therefore, it will be quasi-linear, equal to $N\log(N)$.

VII. Conclusion

The Forward algorithm progressively calculate the probability of an observation sequence, it is used in the recognition and learning because it represents the basis for revaluation in the Baum-Welch algorithm. We benefited from the method divide and conquer to reduce the charge of calculation of the Forward algorithm.

References

[1] Pellegrini T., Duée R., "Suivi de Voix Parlée grace aux Modèles de Markov Cachés," IRCAM, PARIS, 2003.

[2] Juang B. H., Rabiner L. R., "Hidden Markov Models for Speech Recognition," Technometrics, Vol. 33, No. 3. 1991, pp. 251-272.

[3] Ramy Al-Hajj M., Chafic Mokbel, Laurence Likforman-Sulem, "Reconnaissance de l'écriture arabe cursive: combinaison de classifieurs MMCs à fenêtres orientées," Université de Balamand, Faculté de Génie, 2006, pp 1-6.

[4] Ben Amara N., Belaïd A., Ellouze N., "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe état de l'art," Colloque International Francophone sur l'Ecrit et le Document (CIFED'00), Lyon, France.

[5] Nakai M., Akira N., Shimodaira H., Sagayama S., "Substroke Approach to HMM-based On-line Kanji Handwriting Recognition," Sixth International Conference on Document Analysis and Recognition (ICDAR 2001), pp.491-495.

[6] Krogh A., I.Saira Mian, Haussler D., "A hidden Markov model that finds genes in E.coli DNA," Nucleic Acids Research, Vol. 22, No. 22.

[7] Morwal S., Jahan N., Chopra D., "Named Entity Recognition using Hidden Markov Model (HMM)," International Journal on Natural Language Computing (IJNLC), 2012, Vol. 1, No.4.

[8] Rabiner R., "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceding of the IEEE, Vol. 77, No. 2. 2010.

[9] Dequier J., "Chaînes de Markov et applications," Centre D'enseignement de Grenoble, 2005, pp.9-35.

[10] Khreich W., Granger E., Miri A., Sabourin R., "On the memory complexity of the forward–backward algorithm," Pattern Recognition Letters 31. 2010.

[11] Rust J., "Using Randomization to Break the Curse of Dimensionality," Econometrica, Vol. 65, No. 3, May, 1997, pp. 487-516.

[12] Van Caneghem M., "Algorithmes recursifs Diviser pour régner,", 2003.

[13] Canet L., "Algorithmique, graphes et programmation dynamique," 2003, pp 23-25.

[14] Cherki D., "decomposition des problemes de decision markoviens," Thesis of doctorat in Faculty of science Rabat, 2002, pp 64-69.

[15] Gondran M., Minoux M., "Graphes et Algorithmes" 2nd edition, 1990.

[16] Drmota M., Szpankowski W., "A Master Theorem for Discrete Divide and Conquer Recurrences," Austrian Science Foundation FWF Grant No. S9604.

[17] Roura S. (March 2001), "Improved Master Theorems for Divide-and-Conquer Recurrences," Journal of the ACM, Vol. 48, No. 2, pp. 170–205.

Review of Color Image Segmentation

Abderrahmane ELBALAOUI[1] M.FAKIR[2]
Faculty of Science and Technology /Informatics,
Beni-Mellal, Morocco.

N.IDRISSI[3], A.MARBOHA[4]
Faculty of Science and Technology /Informatics,
Beni-Mellal, Morocco.

***Abstract*—This paper provides a review of methods advanced in the past few years for segmentation of color images. After a brief definition of the segmentation, we outline the various existing techniques, classified according to their approaches. We have identified five that are based approaches contours, those relying on notion of region, structural approaches, those based on the form and then using those notions of graphs. For each of these approaches, we then explained and illustrated their most important methods. This review is not intended to be exhaustive and the classification of certain methods may be discussed since at the boundary between different approaches.**

***Keywords*—Image segmentation; k-means; region and Boundary.**

I. Introduction

Image segmentation is one of the fundamental and most studied problems in computer vision. Segmentation is a commonly used term for identifying differences between particularly interesting and uninteresting objects as well as distinguishing foreground from background content. Segmentation is to partition an image into homogeneous regions that is to say, a connected set of points in the image with common properties. This important step in the process and analysis of color images refers to the feature extraction using two approaches commonly known as: region approach and contour approach. We briefly give the differences between these two concepts:

1) The notion of "contour" is associated with the search for local discontinuities, transitions between different areas where certain parameters such as color, intensity, texture are not identical. But this notion of contour does not directly yield a segmentation of the image. Indeed, it is quite rare to get directly after treatment, outlines fully connected and it is then necessary to close the contours in order to then assimilate regions to areas within a closed line.

2) The concept of "region" is associated with clusters of connected pixels with common attributes (color, texture). These methods lead directly to a partition of the image where each pixel is assigned to a single region.

The paper is organized as following. Section 2 we focus on different methods of region, those based on the form, those based on a paradigm of regions, those using a structural approach, those using graph theory and finally those based on classification.

Section 3 we describe the methods that we classified into five major themes that are operating on the outline approaches.

II. Region approach

Region approach seeks to combine pixel homogeneous regions. It is characterized by the extent of uniformity of the built-up areas in the image. These regions are constructed by evaluating the similarity between the pixels or between a pixel and those of the same region. Methods are distinguished by region growing division-fusion and classification.

A. Segmentation analysis of spatial properties

This first part is devoted to methods that consider a region as a set of connected pixels whose colors are close to each other. These methods operate by scanning the image in order to build and involve regions the color information for decision making. We can distinguish four approaches:

- Regions dividing [1].
- Region Growing [2].
- Region Merging [4].
- Split-and-merge [5].

1) Regions dividing

The segmentation approach by dividing regions is to divide the original image into homogeneous regions in terms of a given criterion. This process is recursive and considers that the initial region corresponds to the image to be analyzed. If a region does not meet homogeneity predicate, it is divided into four sub-regions of equal size. Each sub-region is analyzed. The recursive algorithm stops when all regions meet the homogeneity predicate. Because of the division into four regions, this method is more adapted to square images having a number of rows and columns equal to a power of two, and wherein the regions are rectangular in shape. On the other hand, this method tends to show the effects of blocks.

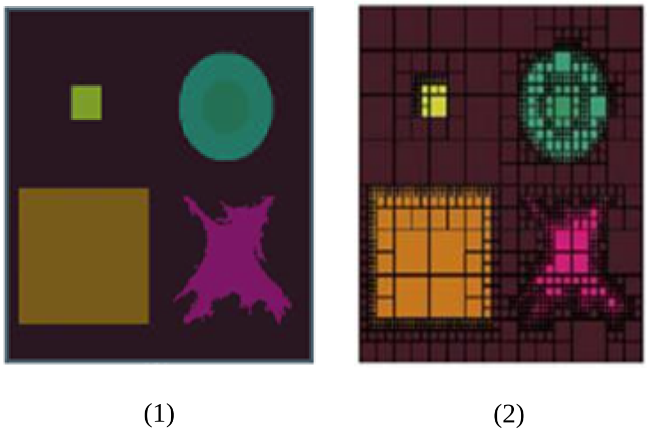


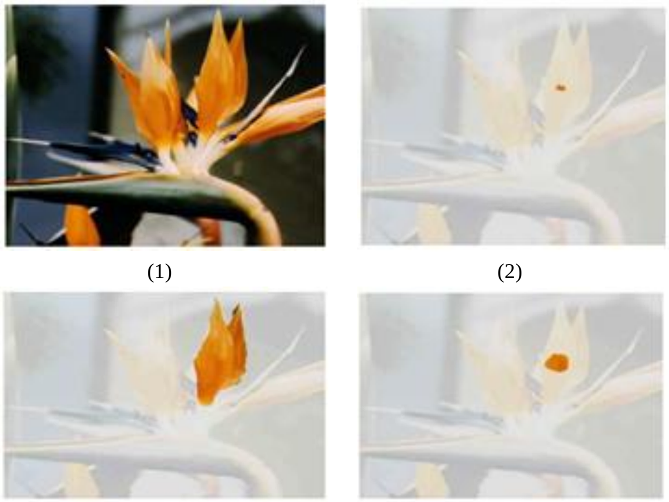
Fig. 1. (1) Original image, (2) Image segmented[1].

2) *Region Growing*

This type of segmentation is to grow regions by adding successively adjacent pixels that meet a criterion of homogeneity. Growth stops when all pixels have been processed.

The initial step is to select seed regions that generally correspond to a pixel. Then, regions are constructed by successively adding pixels that are related and that satisfy a similarity criterion color.

Most algorithms [2] growth regions are covered by a path of the image scanning from left to right and from top to bottom (fig.2).



(1) (2)

(4) (3)

Fig. 2. (1) Original image, (2) Seed, (3) Growth, (4) region final [2].

The advantage of the growth areas is to preserve the shape of each region of the image.

However, poor selection of seeds or a choice of similarity criterion unsuitable phenomena can cause sub-segmentation or over-segmentation.

3) *Region Merging*

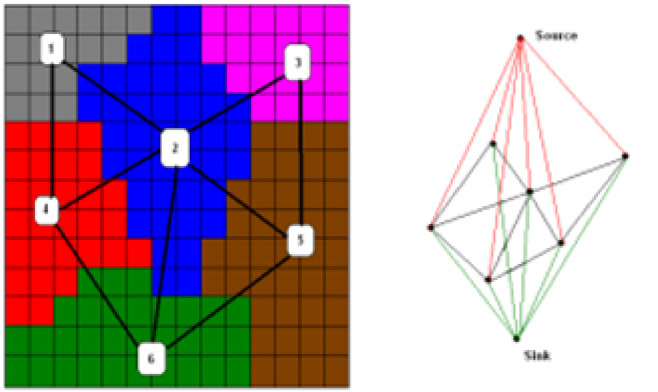
The region merging approach is mainly based on the analysis of an adjacency graph regions which analyzes a segmented image consists of a set of regions.

This is a data structure consisting of a non-oriented graph of which each node represents a region and each edge represents an adjacency between two regions.

Fig.3 (2) shows, as an example, the adjacency graph of regions obtained from the image in Fig.3 (1) .

The process of merging two nodes connected by an edge provided they meet a criterion of fusion.

R. Schettini [3] proposes the use of a criterion taking into account the similarity of two color regions and spatial proximity. It introduces a similarity function based on a color distance weighted by the relative length of the common border of the two regions to merge.



(1) (2)

Fig. 3. (1) Original image, (2) Graph adjacency[4]

J. Stawiaski [4] proposes a weighting each edge adjacency graph of the regions by half the distance between the two color regions corresponding to the nodes of the edge.

At each iteration, regions connected by the bridge which carries the minimum weight are merged. The edge weights are updated according to the number of pixels belonging to edges associated with their color and distance regions. With the weighting chosen, the algorithm tends to favor the adjacent regions of low surface. The fusion algorithm stops when previously determined number of iterations is reached or when the weights of the edges reach a limit value.

4) *Split-and-Merge*

Segmentation Split-and-Merge combines the previous two approaches. Firstly, the image is divided into homogeneous regions that meet global criteria and adjacent areas that meet local criteria are merged. We describe two data structures to perform this approach: the quadtree and the Voronoi diagram.

a) quadtree

V. Coutance [5] segmenting color images into regions by analyzing a structure called quadtree proposed by S. L. Horowitz and S. Pavlidis [6] in the context of image segmentation grayscale.

The quadtree is a tetra-tree in which each node has exactly four son nodes, except the terminal nodes. Each node corresponds to a block that is an area of the image of square shape, the number of rows and columns which is a power of two. The root of the tree corresponds to the entire image must be square and the number of rows and columns is also a power of two (fig.4).

Each block with a quadtree node of the initial partition is recursively analyzed to determine whether it should be divided into four sub-blocks. The recursive analysis stops when each sub-block follows a predicate homogeneity.

Each block with a quadtree node of the initial partition is recursively analyzed to determine whether it should be divided into four sub-blocks. The recursive analysis stops when each sub-block follows a predicate homogeneity.

At this stage of the analysis, some adjacent blocks in the image are the same color characteristics. This is why these pairs of blocks are merged. The analysis stops when there is more torque that respects the predicate fusion.

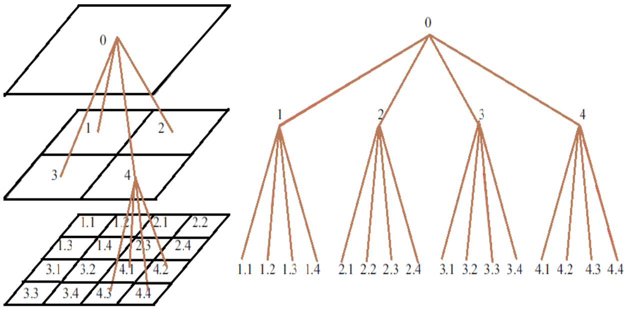


Fig. 4. Structure pyramidal quadtree

The major drawback of the quadtree is the rigidity of the square cut. It leads to a partitioning of the global image that does not always form regions present in the image. In addition, the consolidation phase of the blocks is sensitive to the order of the path quadtree.

b) Voronoi diagram [7]

The Voronoi diagram approach is part of an iterative process of divisions, mergers and can be seen as an improvement of the segmentation analysis of a quadtree. Phase division is not carried out by cutting square-shaped regions, but by cutting Voronoi polygons that fit the shapes of these regions in the image.

The Voronoi diagram generates a partition of the image from germs. Each seed is assigned a Voronoi region consists of all of the closest seed pixels.

Fig.4 shows an example of a Voronoi diagram. In this figure, the seeds are the red dots. The closest a seed pixels are those listed in the blue centered on the seed polygon.

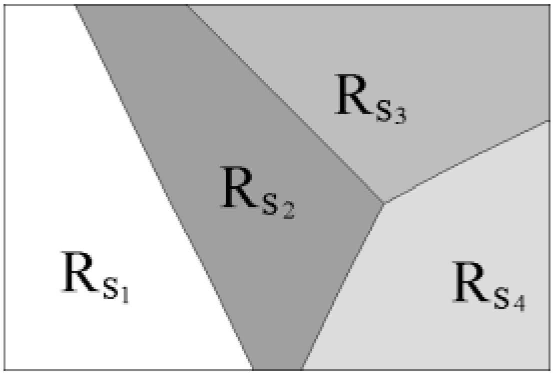


Fig. 5. Example of Voronoi diagram [7].

This approach is presented for grayscale images, but can be generalized to color images.

The given segmentation method comprises an initialization step, a division step followed by a melting step:

- Initialization: Germs are positioned and uniformly distributed in the image using a Poisson process. Each seed is associated with a region whose boundaries are established using a Voronoi diagram.
- Division: A predicate homogeneity is calculated for each region and the non-homogeneous regions are divided by introduction of new seeds. The Voronoi diagram is updated and the division process is repeated until all the Voronoi regions meet a predicate homogeneity.
- Melting: Germs that do not correspond to an object in the image are eliminated in this final step. Thus, adjacent regions whose average colors are close and where the length of the border divided by the sum of their perimeters is less than a threshold are merged.

Segmentation analysis of pixels

This second part is mainly devoted to methods from the field of classification of multidimensional data that are applied to the segmentation of color images.

These methods consider a region as a set of connected pixels belonging to the same class. They therefore assume that the pixels that belong to the same region have similar colors and form a cloud of points clearly identifiable in the representation space 3D color. The classification consists in finding these clouds of points that correspond to the classes of pixels present in the image.

Supervised classification is called when a priori information is introduced into the process of building classes as a learning sample. Classification is called unsupervised when no a priori knowledge is available. However, we can question the validity of the so-called unsupervised segmentation in the sense that the methods generally involve thresholds that introduce certain way information on the classes of pixels to build.

In this section, we present two families of methods:

- Unsupervised pixel classification
- unsupervised methods

1) Unsupervised pixel classification

a) K-means algorithm

The k-means clustering algorithm [8] is the best known and most widely used because of its simplicity of implementation. It partitions the image data into K clusters.

Unlike other methods known hierarchical structure that create a "cluster tree" to describe groups, k-means creates only one level of clusters.

The algorithm returns a data partition in which objects within each cluster are as close as possible to each other and as far as possible objects of other clusters. Each cluster in the partition is defined by its objects and its centroid.

The k-means is an iterative algorithm that minimizes the sum of distances between each object and the centroid of the cluster.

The initial position of the centroid determines the final result, so that the centroid should be initially placed as far as possible from each other so as to optimize the algorithm. K-means cluster objects exchange until the sum can no longer decreasing. The result is a set of compact and clearly separated clusters, provided that they have chosen the correct value of the number of clusters K.

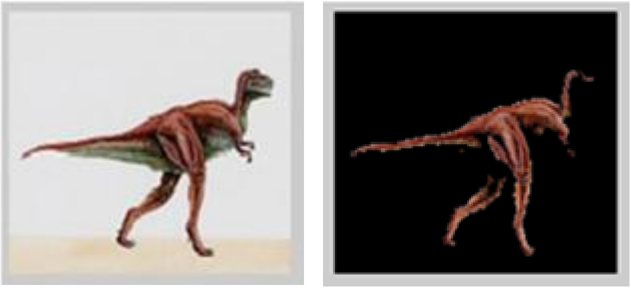
The K-means algorithm is an iterative technique that is used to partition an image into K clusters. The basic algorithm is:

1) Pick K cluster centers, either randomly or based on some heuristic

2) Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center

3) Re-compute the cluster centers by averaging all of the pixels in the cluster

4) Repeat steps 2 and 3 until convergence is attained



(1) (2)

Fig. 6. (1) Original image, (2) Image segmented.

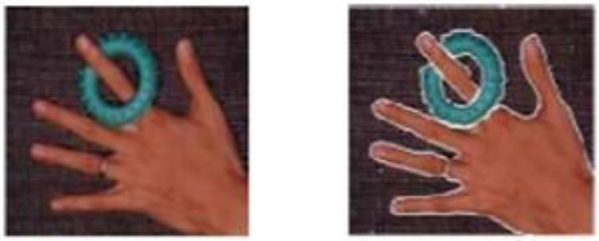
a) Fuzzy c-means

The algorithm of fuzzy c-means [9] is a fuzzy clustering algorithm based on optimization of a quadratic criterion of classification where each class is represented by its center of gravity. The algorithm requires knowing the number of classes in advance and generates classes so that the sum of squared deviations interclass and intraclass are respectively maximum and minimum.

Y. W. Lim and S. U. Lee [10] proposed to segment color images in a coarse to fine decomposition by performing a first segmentation multithresholding, followed by a more detailed segmentation by applying an algorithm of fuzzy c-means.

They analyze marginally each color component by the scale space filtering approach to determine the thresholds that separate modes detected. This technique is proposed by Witkin in [11] and to determine the most prominent peaks dimensional histograms. Thus, the color space is partitioned into subspaces defined by different thresholds determined for each component. These subspaces correspond to each class of pixels. Among these classes, some of which are composed of a number of pixels too low to be meaningful. They are then removed. The fine segmentation is to classify the unlabeled pixels by the coarse segmentation. Thus, the pixels that do not belong to any class are assigned to the class for which their degree is maximum.

R. Krishna Priya, C. Thangaraj and C. Kesavadas [12] propose to improve the classical algorithm of fuzzy c-means taking into account the local spatial information. Thus, the criterion to be minimized is modified by inserting a weighting which depends on the color difference between the central pixel and its neighboring pixels. Taking into account the local spatial information improves the classification results and provides satisfactory segmentation results.



(a) (b)

Fig. 7. **(a)** Test color image, **(2)** Image segmented [12].

b) Fisher algorithm

Fisher algorithm [13] is to perform a classification of pixels of the image into k classes, using the partitioning of a histogram grayscale k disjoint classes such that the sum of the variances of classes is minimal.

Under the color should be applied this algorithm separately the three components of the color space representation.

2) Supervised pixel classification

a) Mono-dimensional histogram

Many authors define classes of pixels per recursive analysis of one-dimensional histograms of different color components to segment images by techniques multithresholding [14].

The histograms dimensional recursive analysis can highlight, as and when the iterations, the modes corresponding to the classes of pixels. It partitions the set of pixels of the image into different classes according to the following algorithm:

```
Stacking all the pixels of the image in the stack of pixels to be processed.
While the stack of the pixels to be processed is non-empty do
Unstack the pixels,
Calculate dimensional histogram of these pixels,
Search the histogram shows how the most important
If the mode is large enough then
Constitute a new class with the pixels that belong to this mode,
Stacking the pixels not belonging to this user in the stack of the pixels to be processed,
End If
End while
```

The methods proposed in literature differ in:

- The choice of the color components used
- The method and criteria for extracting modes.

- The determination of the most representative component.
- The criteria for stopping the recursive analysis of histograms

b) Nearest neighbor algorithm

The method of k-nearest neighbors (k-NN)[15] is a method for estimating non-parametric density [15]. It is to seek from a learning base and a distance set to "data space" K-nearest neighbors of an element. That is to calculate the estimate of density ri of the class C_i at the point x defined by the following relationship:

$$ri(x) = \frac{Ki(x)}{ni \times V(x)} \quad (1)$$

Where:

$Ki(x)$: is the number of points C_i belonging to k-nearest neighbor the x.

ni : is the cardinality of the class $C_{i.}$

$V(x)$: Volume of the smaller ball containing k-nearest neighbor the x.

The probability of belonging to a class of the element x is proportional to the number of examples of this class among its k-nearest neighbors. The class of x is then the element having the highest probability.

III. Edge detection approach

Early approaches to contour detection aim at quantifying the presence of a boundary at a given image location through local measurements. The Roberts [16], Sobel [17], and Prewitt [18] operators detect edges by convolving a grayscale image with local derivative filters. Marr and Hildreth [19] use zero crossings of the Laplacian of Gaussian operator. The Canny detector [20] also models edges as sharp discontinuities in the brightness channel, adding non-maximum suppression and hysteresis thresholding steps. A richer description can be obtained by considering the response of the image to a family of filters of different scales and orientations. An example is the Oriented Energy approach, which uses quadrature pairs of even and odd symmetric filters. Lindeberg [21] proposes a filter-based method with an automatic scale selection mechanism.

More recent local approaches take into account color and texture information and make use of learning techniques for cue combination. Martin et al. [22] define gradient operators for brightness, color, and texture channels, and use them as input to a logistic regression classifier for predicting edge strength. Rather than rely on such hand-crafted features, Dollar et al. [23] propose a Boosted Edge Learning (BEL) algorithm which attempts to learn an edge classifier in the form of a probabilistic boosting tree from thousands of simple features computed on image patches. An advantage of this approach is that it may be possible to handle cues such as parallelism and completion in the initial classification stage. Mairal et al. [24] create both generic and class-specific edge detectors by learning discriminative sparse representations of local image patches. For each class, they learn a discriminative dictionary and use the reconstruction error obtained with each dictionary as feature input to a final classifier.

IV. Advantages and Disadvantages of Segmentation Technique

Segmentation Technique	Advantages	Disadvantages
Regions dividing	• Guaranteed to produce coherent regions. • Works from the inside out instead of the outside inwhich object a pixel belongs to is immediate.	• Cutting areas unfaithful to the original image
Region Growing	• Simple and fast. • It allows the object segmentation complex topology. • It preserves the shape of each region of the image.	• The computation is consuming, no matter the time or power. • Noise or variation of intensity may result in holes or oversegmentation. • This method may not distinguish the shading of the real images.
Region Merging	• The image could be split progressively according to our demanded resolution because the number of splitting level is determined by us. • We could split the image using the criteria we decide, such as mean or variance of segment pixel value. In addition, the merging criteria could be different to the splitting criteria	• This method depends on the merging criterion that can influence the final result segmentation. • It can introduce the effect of sub-segmentation.
Quadtree	• The quadtree structure is very common due to its simplicity and data secondly its low computation time.	• The rigidity of the square cut • It leads to a partitioning of the overall image does not always respect the form regions present in the image. • The division quadtree generally provides an over-segmentation
Voronoi diagram	• The based on the Voronoi diagram approach can be considered a improved segmentation analysis of quad-tree. • Voronoi fit the shapes of regions present in the image.	• Initialization germs can lead to different segmentation results.
K-means	• K-means algorithm is easy to implement. • Its time complexity is O(n), where n is the number of patterns. It is faster than the hierarchical clustering.	• The result is sensitive to the selection of the initial random centroids. • We cannot show the clustering details as hierarchical clustering does.
Histogram Thresholding	• It does not need a prior information of the image. And it has less computational complexity	• Does not work well for an image without any obvious peaks or with broad and flat valleys • Does not consider the spatial details, so cannot guarantee that the segmented regions are contiguous

V. Conclusions

In this study, the overview of various segmentation methodologies applied for digital image processing is explained briefly. The study also reviews the research on various research methodologies applied for image segmentation and various research issues in this field of study. This study aims to provide a simple guide to the researcher for those carried out their research study in the image segmentation.

References

[1] H. D. Cheng, X. H. Jiang, Y. Sun, et J. Wang. Color image segmentation : advances and projects. Pattern Recognition, 34 :2259–2281, 2001

[2] Slawo Wesolkowski ; Paul W. Fieguth. Color image segmentation using vector angle-based region growing. SPIE.4421..910W, 2002.

[3] R. Schettini. A segmentation algorithm for color images. Pattern Recognition Letters, 14 :499–506, 1993.

[4] J. Stawiaski and E. Decenciere. Region Merging via Graph-Cuts. ICS XII - 2007: International Congress of the International Society of Stereology, Saint-Etienne, France.

[5] V. Coutance. La couleur en vision par ordinateur. Application à la robotique. Thèse de doctorat, Université Paul Sabatier de Toulouse, Janvier 2001.

[6] S. L. Horowitz et S. Pavlidis. Picture segmentation by a directed split and merge procedure.In 2nd Int. Joint Conf. on Pattern Recognition, pages 424–433, 1991.

[7] Andres Arbelaez, P.; Cohen, L.D. Color Image Segmentation by Voronoi Partitions. 2004.

[8] Anil Z Chitade, Dr. S.K. Katiyar, "Color Based Image Segmentation using K-Means Clustering", International Journal of Engineering Science and Technology Vol. 2(10), 2010, 5319-5325

[9] J. C. Bezdeck. Pattern recognition with fuzzy objective function algorithms. Plenum Press Ed., New York, 1997.

[10] Y. W. Lim et S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. Pattern Recognition, 23(9) :935–952, 1995.

[11] A.W.C. Liew, S.H. Leung, et W.H. Lau. Fuzzy image clustering incorporating spatial continuity. IEE Proceedings on Vision Image Signal Processing, 147(2) :185–192, April 2000.

[12] R. Krishna Priya1, C. Thangaraj and C. Kesavadas. Fuzzy C-Means method for Colour Image Segmentation. ISSN (Online): 1694-0814. 2011

[13] W. FISHER. « On grouping for maximum homogeneity ». *JASA*, vol. 53, pages 789–798, 1998.

[14] R. DUDA, P.HART & D.STORK. « Pattern Classification ». Xiley Interscience- 2e édition, 2001.

[15] Wojciech Bieniecki and Szymon Grabowski, "Nearest Neighbor
Classifiers for Color Image Segmentation", TCSET' 2004 Feb 24-28

[16] L. G. Roberts, "Machine perception of three-dimensional solids," In Optical and Electro-Optical Information Processing, J. T. Tippett et al. Eds. Cambridge, MA: MIT Press, 1965.

[17] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[18] J. M. S. Prewitt, "Object enhancement and extraction," In Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld. Eds.

[19] D. C. Marr and E. Hildreth, "Theory of edge detection," Proceedings of the Royal Society of London, 1980.

[20] J. Canny, "A computational approach to edge detection," PAMI, 1996.

[21] T. Lindeberg, "Edge detection and ridge detection with automaticscale selection," IJCV, 1998.

[22] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues,"

[23] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," CVPR, 2006.

[24] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," ECCV, 2008.

Invariant Descriptors and Classifiers Combination for Recognition of Isolated Printed Tifinagh Characters

M. OUJAOURA, R. EL AYACHI, B. MINAOUI,
M. FAKIR and B. BOUIKHALENE
Computer Science Department
Faculty of Science and Technology
Sultan Moulay Slimane University
Béni Mellal, Morocco

O. BENCHAREF
Computer Science Department
Higher School of Technology
Cadi Ayyad University
Essaouira, Morocco

***Abstract*—In order to improve the recognition rate, this document proposes an automatic system to recognize isolated printed Tifinagh characters by using a fusion of 3 classifiers and a combination of some features extraction methods. The Legendre moments, Zernike moments and Hu moments are used as descriptors in the features extraction phase due to their invariance to translation, rotation and scaling changes. In the classification phase, the neural network, the multiclass SVM (Support Vector Machine) and the nearest neighbour classifiers are combined together. The experimental results of each single features extraction method and each single classification method are compared with our approach to show its robustness.**

***Keywords*— Recognition system; Legendre moments; Zernike moment; Hu moments; Neural Networks; Multiclass SVM; nearest neighbour classifier**

I. INTRODUCTION

The Optical Character Recognition (OCR) is a field of pattern recognition that focuses on character shapes.

In the recent years, some research works of Tifinagh characters are published, such as:

- Es Saady and all [1] are treated in their paper printed characters Amazigh isolated using an automatic character recognition based on the formalism of finite automata.
- Bencharef and all [2] are used the metric descriptors based on the calculation of the Riemannian metric. These descriptors are known for their reliability towards the change of scale, the existence of noise and geometric distortions. For the classification of Tifinagh characters, SVM and neural networks were used.
- El Ayachi and all [3] are developed a system Tifinagh characters recognition using a multilayer neural networks in classification step. In the extraction phase, Walsh transform is adopted.

In this work, the proposed system (Fig.1) contains three phases to recognize isolated printed Tifinagh characters: Pre-processing, Extraction and Classification phases.

In the pre-processing phase, normalization is applied to remove unwanted areas using the method of histogram; In this phase, we first calculate the horizontal and vertical histograms, then the histogram is scanned horizontally in both directions: respectively from the top to the bottom and from the bottom to the top until finding the first black pixels, thereafter, the vertical histogram is traversed in both directions: respectively from the left to the right and from the right to the left until finding the first black pixels. Finally, after determining the positions of the first black pixels, we eliminate the unwanted areas.

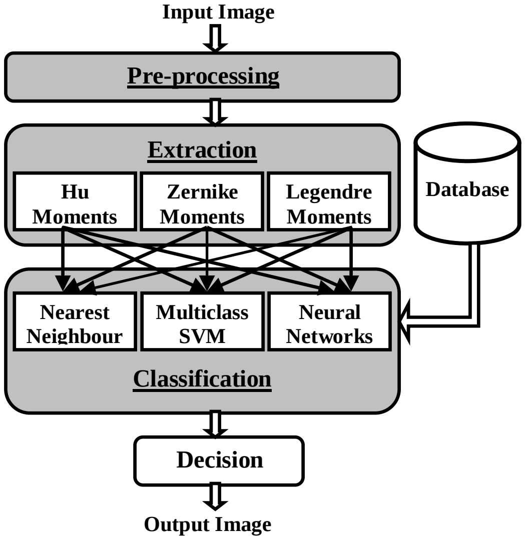


Fig. 1. Bloc diagram of the proposed recognition system

To calculate the feature of image, Legendre moments, Hu moments and Zernike moments are used in the second phase which is the extraction phase.

The classification phase adopts the neural networks, the SVM (Support Vector Machine) and the nearest neighbour classifier to recognize the correct character.

Most of the reported works for recognition systems uses a single method for calculating the parameters of an image and a single approach in the classification phase. The objective of this paper is to combine three methods in the extraction phase and three approaches to classify the characters in order to improve the recognition rate and reduce the error rate.

The rest of the paper is organized as follows. The Section 2 discusses the primordial task of any recognition system. It's the classification problems using some classifiers based on a combination of several algorithms like neural network, SVM and nearest neighbour classifiers. The Section 3 is reserved for another important task which is the features extraction problems in addition to a brief formulation for Hu moments, Legendre moments and Zernike moments as features extraction methods. The Section 4 presents the experimental results for the recognition system. Finally, the conclusion is given in the section 5.

II. Classification

The robustness of the recognition system is based on the decision given by the classification phase. As a result, this document presents some approaches (Neuronal Network, SVM, nearest neighbor classifiers) known in the recognition field [4].

A. *Neural network*

Neural networks (or artificial neural networks) learn by experience, generalize from previous experiences to new ones, and can make decisions [5, 6].

A multilayer neural network consists of an input layer including a set of input nodes, one or more hidden layers of nodes, and an output layer of nodes. Fig. 2 shows an example of a three layer network used in this paper, having input layer formed by M nodes, one hidden layer formed by L nodes, and output layer formed by N nodes. This neural network is trained to classify inputs according to target classes. The training input data are loaded from the reference database while the target data should consist of vectors of all zero values except for a one element, where its index is the class they are to represent. The transfer function used in this tree layer neural network is hyperbolic tangent sigmoid transfer function defined by:

$$tsig(x)=2/\big(1+\exp(-2x)\big)-1 \qquad (1)$$

According to authors in [7], the number of neurons in the hidden layer is approximately equal to:

$$L=E\Big(1+\sqrt{M(N+2)}\Big) \qquad (2)$$

Where:

E(x) denotes the integer part of x.

M and N are respectively the number of neurons in the input and output layers.

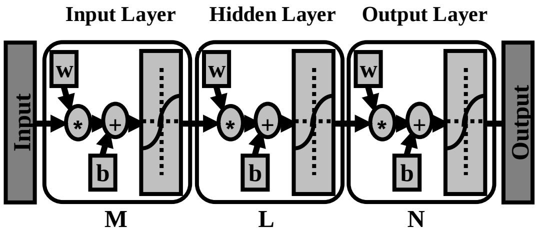


Fig. 2. The three layer neural network

B. *Support vector machine (SVM)*

Support vector machine (SVM) were originally designed for binary classification. As shown in Fig. 3, SVM is a classification method which is based on finding a hyper-plan that separates data sets into two classes. Several methods have been proposed to construct a multi-class classifier [9] by combining one-against-one binary classifiers or one-against-all binary classifiers. The data sets can be linearly separable or nonlinearly separable. The nonlinearly separable cases require the use of kernel function in order to obtain linearly separable data sets [8] [9].

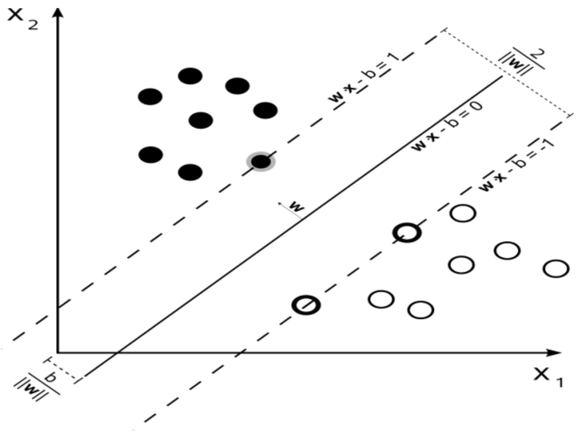


Fig. 3. Binary SVM with samples from two classes

In this paper, the one-against-one and the one-against-all binary classifier are used. Those classifiers are based on the Gaussian kernel function defined by:

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (3)$$

Where $\sigma \succ 0$ equal to 1 in our case.

Many other kernel functions can be used for each binary classifier.

1) *One-against-one binary classifier*

From N class in data sets, the one-against-one multiclass SVM method constructs N(N-1)/2 binary classifier where each one is trained on data from two classes. The Structure the one-against-one multiclass SVM classifier can be represented by the Fig. 4.

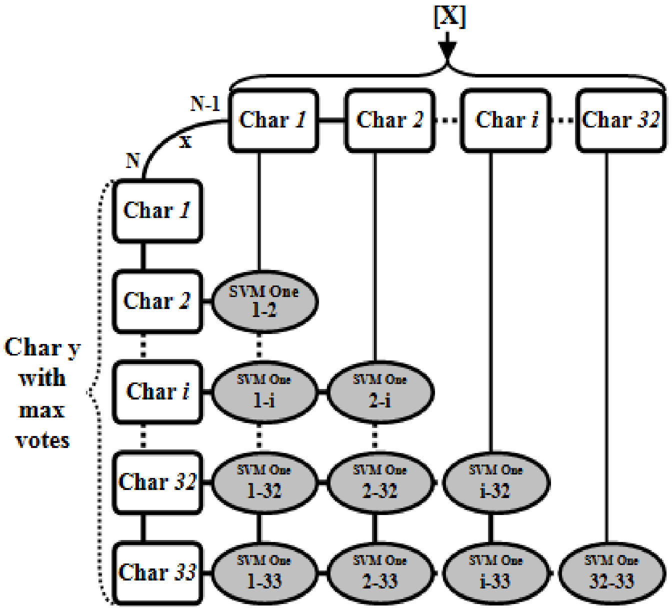


Fig. 4. Structure the one-against-one multiclass SVM classifier

To design and extend SVM binary classifier into a one-against-one multiclass SVM, two groups of data examples are constructed from two classes. The obtained SVM binary classifier is trained to decide if the class is from the first class or it belongs to the second class. This process is repeated for another couple of classes until finishing all the possible couples of the classes from data sets. So, by following this way, multiclass SVM is transformed to a multiple N(N-1)/2 SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to the features extracted as an observation from a class. When classifying an object with an input features vector, each binary classifier from the multiclass SVM one-against-one model decides and votes for only one class. The class with the majority votes is the correct class which the object belongs to.

2) *One-against-all binary classifier*

The one-against-all multiclass SVM classifier contains N binary classifier, where N is the number of class in data sets. The ith binary SVM is trained with all of the data examples in the ith class with positive labels, and all other data examples with negative labels.

To construct a one-against-all multiclass SVM model from binary classifier, the classes are divided into two groups: the first group is formed by one class, and the second group is all the other classes. The obtained SVM binary classifier is trained to decide if the class is from the first group or it belongs to the second groups of classes. This process is repeated for the second group that contains more than two classes until having only one class for each group. The process must stop there. So, by following this way, multiclass SVM is transformed to a multiple SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to features extracted as an observation from a class. After training phase, the multiclass SVM model is able to decide the correct class for an input features vector. To classify an object, its input features vector is presented iteratively to the ith against all binary classifier from the first to the Nth classifier while the result is negative. When the ith binary classifier gives a positive result, the process is stoped. This means that the object belongs to the ith class. The Structure of the one-against-all multiclass SVM classifier is given by the Fig. 5.

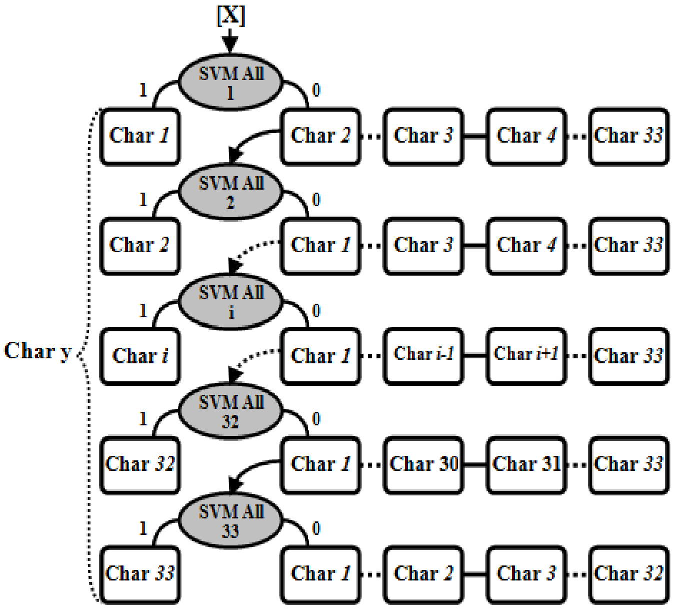


Fig. 5. Structure of the one-against-all multiclass SVM classifier

C. *Nearest neighbour*

The nearest neighbour classifier is used to compare the feature vector of the input image and the feature vectors stored in the database. It is obtained by finding the distance between the prototype image and the database. The class is found by measuring the distance between a feature vector of input image and feature vectors of images in reference database. The Euclidean distance measurement is used in this paper, but other distance measurement can be also used [10].

Let X_1, X_2, …, X_k be the k class features vectors in the database and X_q the feature vector of the query image. The feature vector with the minimum distance is found to be the closest matching vector. It is given by:

$$d(X_q, X_j) = \min_{j\in\{1,2,\dots,k\}} \left\{ \sqrt{\sum_i \left(x_q(i) - x_j(i)\right)^2} \right\} \qquad (4)$$

The nearest neighbour classifier doesn't need any training phase. But, if the database is very large, it takes a considerable time to calculate all the distances between the query image and database classes.

III. EXTRACTION

Extraction is the second phase to apply in the recognition system. It is an important step, because the computed parameters will be used in the classification stage.

In this paper, the used approaches are: Hu moments [11], Legendre moments [12, 13, 14] and Zernike moments [15, 16].

A. Hu moments

For a discrete image of M x N pixels with intensity function f(x, y), Hu [11] defined the following seven moments that are invariant to the change of scale, translation and rotation:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{5}$$

$$\phi_2 = \left(\mu_{20} + \mu_{02}\right)^2 + 4\mu_{11}^2 \tag{6}$$

$$\phi_3 = \left(\mu_{30} - 3\mu_{12}\right)^2 + \left(3\mu_{21} - \mu_{03}\right)^2 \tag{7}$$

$$\phi_4 = \left(\mu_{30} - \mu_{12}\right)^2 + \left(\mu_{21} - \mu_{03}\right)^2 \tag{8}$$

$$\begin{aligned}\phi_5 = {} & \left(\mu_{30} - 3\mu_{12}\right)\left(\mu_{30} + \mu_{12}\right)\times \\ & \left[\left(\mu_{30} + \mu_{12}\right)^2 - 3\left(\mu_{21} + \mu_{03}\right)^2\right] + \\ & \left(3\mu_{21} - \mu_{03}\right)\left(\mu_{21} + \mu_{03}\right)\times \\ & \left[3\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right]\end{aligned} \tag{9}$$

$$\begin{aligned}\phi_6 = {} & \left(\mu_{20} - \mu_{02}\right)\left[\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right] \\ & + 4\mu_{11}\left(\mu_{30} + \mu_{12}\right)\left(\mu_{21} + \mu_{03}\right)\end{aligned} \tag{10}$$

$$\begin{aligned}\phi_7 = {} & \left(3\mu_{21} - \mu_{03}\right)\left(\mu_{30} + \mu_{12}\right)\times \\ & \left[\left(\mu_{30} + \mu_{12}\right)^2 - 3\left(\mu_{21} + \mu_{03}\right)^2\right] - \\ & \left(\mu_{30} - 3\mu_{12}\right)\left(\mu_{21} + \mu_{03}\right)\times \\ & \left[3\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right]\end{aligned} \tag{11}$$

Where

$\mu_{pq} = \dfrac{\alpha_{pq}}{\alpha_{00}^{\frac{p+q}{2}+1}}$ are the Normalized central moments

$\alpha_{pq} = \int\limits_{a_1}^{a_2}\int\limits_{b_1}^{b_2}\left(x - \dfrac{M_{10}}{M_{00}}\right)^p\left(y - \dfrac{M_{01}}{M_{00}}\right)^q f(x,y)\,dxdy$ are the central moments, and

$M_{pq} = \int\limits_{a_1}^{a_2}\int\limits_{b_1}^{b_2} x^p y^q f(x,y)\,dxdy$ are the two-dimensional geometric moment of order (p + q) of an image f(x, y).

B. Legendre moments

The Legendre moments were first introduced by Teague [12]. They were used in several pattern recognition applications [13]. The orthogonal property of Legendre polynomials implies no redundancy or overlap of information between the moments with different orders. This property enables the contribution of each moment to be unique and independent of the information in an image [14].

The Legendre moments for a discrete image of M x N pixels with intensity function f(x, y) is the following:

$$L_{pq} = \lambda_{pq}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} P_p(x_i)\,P_q(y_j)\ f(x,y) \tag{12}$$

Where

$\lambda_{pq} = \dfrac{(2p+1)(2q+1)}{M \times N}$, x_i and y_j denote the normalized pixel coordinates in the range of [-1, +1], which are given by:

$$\begin{cases} x_i = \dfrac{2x - (M-1)}{M-1} \\ y_j = \dfrac{2y - (N-1)}{N-1} \end{cases} \tag{13}$$

$P_p(x)$ is the pth-order Legendre polynomial defined by:

$$P_p(x) = \sum_{k=0}^{p}\left\{\frac{(-1)^{\frac{p-k}{2}}(p+k)!\,x^k}{2^p k!\left(\dfrac{p-k}{2}\right)!\left(\dfrac{p+k}{2}\right)!}\right\}_{p-k=even} \tag{14}$$

In order to increase the computation speed for calculating Legendre polynomials, we used the recurrent formula of the Legendre polynomials defined by:

$$\begin{cases} P_p(x) = \dfrac{(2p-1)x}{p}\ P_{p-1}(x) - \dfrac{(p-1)}{p}P_{p-2}(x) \\ P_1(x) = x \quad , \quad P_0(x) = 1 \end{cases} \tag{15}$$

C. Zernike moments

Zernike moments are the mapping of an image onto a set of complex Zernike polynomials. As these Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlap of information between the moments [15]. Due to these characteristics, Zernike moments have been utilized as feature sets in many applications [16].

The discrete form of the Zernike moments of an image size M × N represented by f(x, y) is expressed, in the unit disk $x^2 + y^2 = 1$, as follows:

$$Z_{pq} = \frac{p+1}{\lambda}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} R_{p,q}\left(r_{xy}\right)e^{-jq\theta_{xy}}\ f(x,y) \tag{16}$$

Where

$$\begin{cases} R_{p,q}(r) = \displaystyle\sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)!\, r^{p-2s}}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!} \\ p-|q| = (even),\ |q| \le p,\ p \ge 0 \end{cases} \quad (17)$$

λ is the number of pixels located in the unit circle, and the transformed phase θ_{xy} and the distance r_{xy} at the pixel of coordinates (x, y) are [17]:

$$\begin{cases} \theta_{xy} = \tan^{-1}\left(\frac{(2y-(N-1))/(N-1)}{(2x-(M-1))/(M-1)}\right) \\ r_{xy} = \sqrt{\left(\frac{2x-(M-1)}{M-1}\right)^2 + \left(\frac{2y-(N-1)}{N-1}\right)^2} \end{cases} \quad (18)$$

Most of the time taken for the computation of Zernike moments is due to the computation of radial polynomials. Therefore, researchers have proposed faster methods that reduce the factorial terms by utilizing the recurrence's relations on the radial polynomials [17, 18]. In this paper, we obtained Zernike moments using the direct method and other method based on the recurrence's relations, defined by:

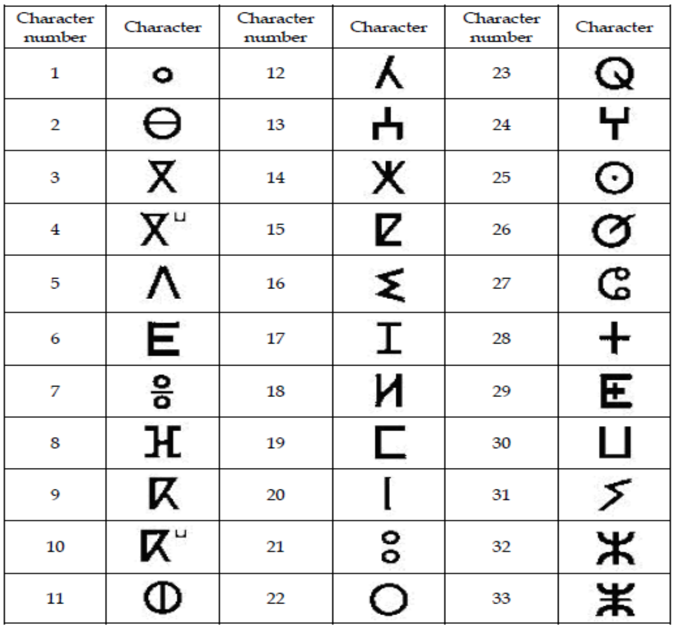
$$\begin{cases} \begin{cases} R_{p,q}(r) = \displaystyle\sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)!\, r^{p-2s}}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!} \\ if\ \ q=0,\ p-|q| = (even),\ |q| \le p,\ p \ge 0 \end{cases} \\ \begin{cases} R_{p,q}(r) = r^p \\ if\quad p = q \end{cases} \\ \begin{cases} R_{p,q}(r) = \dfrac{2rp\, R_{p-1,\,q-1}(r) - (p-q)R_{p-2,\,q}(r)}{p+q} \\ if\ \ p \ne q,\ q \ne 0 \end{cases} \end{cases} \quad (19)$$

The usage of direct method to compute radial polynomials in the case of q = 0 will considerably increase the computation time especially when p is very large.

IV. Results

The Tifinagh alphabet adopted by IRCAM [19] is composed from thirty-three characters representing consonants and vowels as shown in Fig. 6.

To test the accuracy of the combined classifiers and descriptors, we calculated the recognition rate of each classifiers and descriptors separately. In our experiment, we used voting rule classifier combination schemes. Then, each one of the 3 combined classifiers, for each descriptor, votes for the appropriate character. The character with maximum votes is selected and considered to be the suited character.



Character number	Character	Character number	Character	Character number	Character
1	ⴰ	12	ⵃ	23	ⵕ
2	ⴱ	13	ⵄ	24	ⵖ
3	ⴳ	14	ⵅ	25	ⵙ
4	ⴳⵯ	15	ⵇ	26	ⵚ
5	ⴷ	16	ⵉ	27	ⵛ
6	ⴹ	17	ⵊ	28	ⵜ
7	ⴻ	18	ⵍ	29	ⵟ
8	ⴼ	19	ⵎ	30	ⵡ
9	ⴽ	20	ⵏ	31	ⵢ
10	ⴽⵯ	21	ⵓ	32	ⵣ
11	ⵀ	22	ⵔ	33	ⵥ

Fig. 6. Tifinagh characters adopted by IRCAM

Finally, the efficacy of the recognition system is based on the decision given in the classification phase. The computation of recognition rate and error rate is presented in the following table (table 1):

TABLE I. RECOGNITION RATE AND ERROR RATE OF SINGLE CLASSIFIER AND DESCRIPTOR

	Descriptors	**Classification Approaches**			
		Neural Network	***Nearest Neighbour***	***SVM One***	***SVM All***
Recognition rate (%)	Hu Moments	72.73	83.64	56.97	41.82
	Zernike Moments	80.61	95.76	69.70	87.88
	Legendre Moments	81.21	97.58	49.70	78.79
Error rate (%)	Hu Moments	17.86	16.36	43.03	58.18
	Zernike Moments	19.39	4.24	30.30	12.12
	Legendre Moments	18.79	2.42	50.30	21.21
Execution time (s)	Hu Moments	8.34	1.42	94.01	3.36
	Zernike Moments	859.92	943.90	1152.36	754.65
	Legendre Moments	853.16	805.33	892.41	806.00

The recognition rate and error rate of the proposed method are presented in the following table (table 2):

TABLE II. RECOGNITION RATE AND ERROR RATE OF THE PROPOSED METHOD

	Recognition rate (%)	Error rate (%)	Execution time (s)
Proposed method	98.79	1.21	2248.09

From the results in Table 1 and Table 2, the recognition rate of the proposed system is improved in the recognition field of the isolated printed Tifinagh characters as shown in Fig. 7, despite of the processing time which is significantly increased.

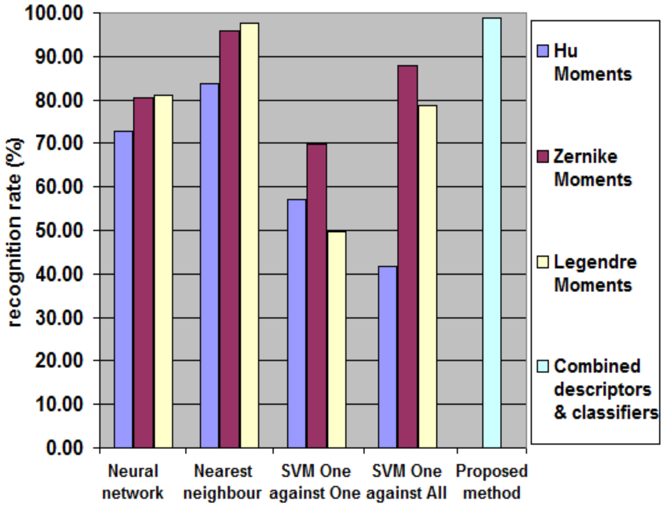
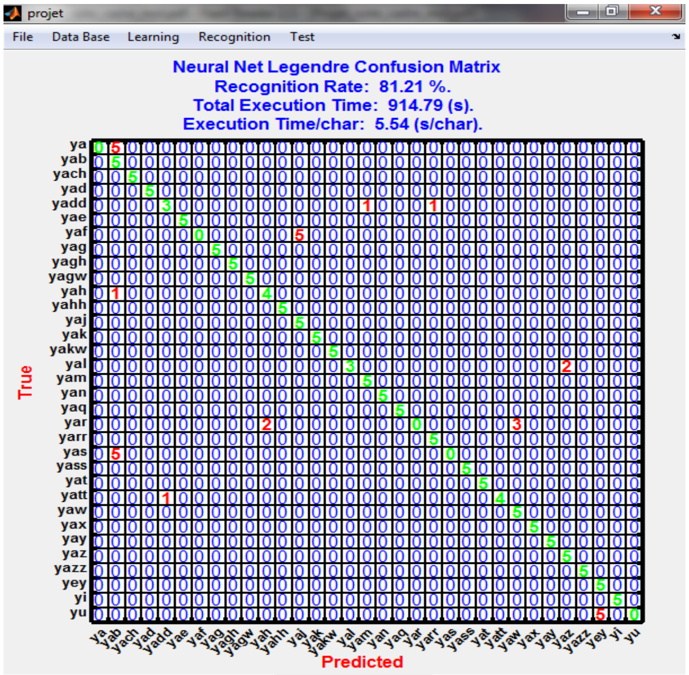


Fig. 7. Comparison of recognition rates

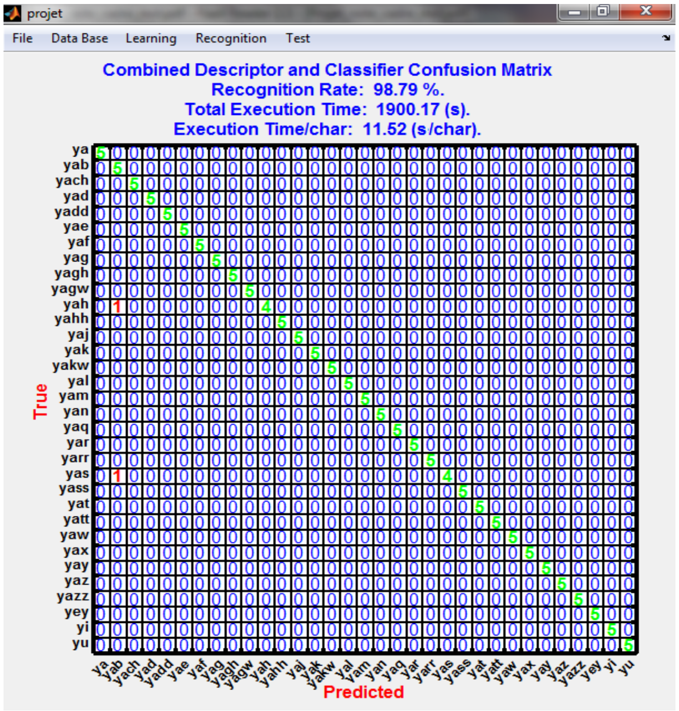
In order to show the robustness of the proposed approach, we give more details by calculating the confusion matrix of some method as presented in Fig. 8 and Fig. 9.



Neural Net Legendre Confusion Matrix
Recognition Rate: 81.21 %.
Total Execution Time: 914.79 (s).
Execution Time/char: 5.54 (s/char).

True \ Predicted	ya	yab	yach	yad	yadd	yae	yaf	yag	yagh	yagw	yah	yahh	yaj	yak	yakw	yal	yam	yan	yaq	yar	yarr	yas	yass	yat	yatt	yaw	yax	yay	yaz	yazz	yey	yi	yu
ya	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yab	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yach	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yad	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yadd	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
yae	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaf	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yag	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yagh	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yagw	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yah	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yahh	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaj	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yak	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yakw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
yam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaq	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yar	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
yarr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
yas	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yass	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
yat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
yatt	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
yaw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
yax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
yay	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
yaz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
yazz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
yey	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
yi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
yu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0

Fig. 8. Confusion matrix of Legendre Moments descriptor and Neural network classifier.



Combined Descriptor and Classifier Confusion Matrix
Recognition Rate: 98.79 %.
Total Execution Time: 1900.17 (s).
Execution Time/char: 11.52 (s/char).

True \ Predicted	ya	yab	yach	yad	yadd	yae	yaf	yag	yagh	yagw	yah	yahh	yaj	yak	yakw	yal	yam	yan	yaq	yar	yarr	yas	yass	yat	yatt	yaw	yax	yay	yaz	yazz	yey	yi	yu
ya	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yab	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yach	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yad	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yadd	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yae	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaf	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yag	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yagh	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yagw	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yah	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yahh	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaj	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yak	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yakw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yaq	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
yarr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
yas	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
yass	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
yat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
yatt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
yaw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
yax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
yay	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
yaz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
yazz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
yey	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
yi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
yu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5

Fig. 9. Confusion matrix of the proposed method.

We can see from the 2 Figures of confusion matrix (Fig. 8 and Fig.9) that the misclassified characters (indicated by red color in the confusion matrix) in the case of using Legendre moment as single descriptor and neural network as single classifier are reduced in the case of using our proposed approach based on combination of many descriptors and classifiers.

All the tests are performed using a database containing a set of 33 reference characters and 165 image characters [20]. The proposed system has been implemented and tested on a core 2 Duo personnel computer using Matlab software.

V. CONCLUSION

The objective of the recognition system is to choose one image of the references images which is similar to the input image. The recognition system adopted in this paper contains three phases: Preprocessing, Extraction and Classification.

In the extraction phase, Hu moments, Legendre moments and Zernike moments are used to compute the attributes of the input image. Neural network, Multiclass SVM and nearest neighbour are three methods used in the classification phase.

Generally, the recognition system uses one method to calculate the parameters of the input image and one approach in the classification. In order to increase the recognition rate, three methods are combined in the extraction phase and three approaches are combined to make classification.

The obtained results show the robustness of the system based on the proposed approach in the recognition field of the isolated printed Tifinagh characters. The use of other fast descriptors and classifiers may increase the recognition rate of Tifinagh characters.

References

[1] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, « Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata », ICGST-GVIP Journal, Volume 10, Issue 2, June 2010.

[2] O. Bencharef, M. Fakir, N. Idrissi, B. Bouikhalen et B. Minaoui, « Application de la géométrie riemannienne à la reconnaissance des caractères Tifinaghe », Agadir-Maroc, 06-07 Mai 2011. pp : 179 :188.

[3] R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts ». JATIT, vol. 20, No. 2, pp: 61-66, 2010.

[4] Mustapha Oujaoura, Brahim Minaoui and Mohammed Fakir, Multilayer Neural Networks and Nearest Neighbor Classifier Performances for Image Annotation, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 11, pp.165–171, 2012. Published by The Science and Information Organization, New York, USA.

[5] Yue Cao, Xiabi Liu, Jie Bing and Li Song, Using Neural Network to Combine Measures of Word Semantic Similarity for Image Annotation, IEEE International Conference on Information and Automation (ICIA), pp. 833 – 837, 2011.

[6] P.Simard, D. Steinkraus, J. C. Platt, Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, ICDAR, 2003, pp. 958-962.

[7] R. Lepage, & B. Solaiman. Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur, Ecole de technologie supérieure, 2003.

[8] R. Rifkin, A. Klautau. In defence of one-versus-all classificaiton. Journal of Machine Learning Research, Vol. 5, pp. 101–141, 2004.

[9] K.-B. Duan, S.S Keerthi, Which is the best multiclass SVM method? An empirical study. Technical Report CD-03-12, Control Division, Department of Mechanical Engineering, National University of Singapore, 2003.

[10] Oren Boiman, Eli Shechtman and Michal Irani, In Defense of Nearest-Neighbor Based Image Classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008.

[11] R. El Ayachi, B. Bouikhalene et M. Fakir, « Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe », SITACAM'11, Agadir-Maroc, 06-07 Mai 2011. pp : 41-54.

[12] F. L. Alt, Digital Pattern Recognition by Moments, J. Assoc. Computing Machinery, Vol. 9, pp. 240-258, 1962.

[13] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Amer. 70, pp. 920–930, 1980.

[14] Chee-Way Chonga, P. Raveendranb and R. Mukundan, Translation and scale invariants of Legendre moments, Pattern Recognition 37, pp. 119 – 129, 2004.

[15] Sun-Kyoo Hwang, Whoi-Yul Kim, A novel approach to the fast computation of Zernike moments, Pattern Recognition 39, pp. 2065 – 2076, 2006.

[16] A. Prata, W.V.T. Rusche, Algorithm for computation of Zernike polynomials expansion coefficients, Appl. Opt. 28, pp. 749–754, 1989.

[17] Mustapha Oujaoura, Brahim Minaoui and Mohammed Fakir. Article: Image Annotation using Moments and Multilayer Neural Networks. IJCA Special Issue on Software Engineering, Databases and Expert Systems SEDEX (1): pp. 46-55, September 2012. Published by Foundation of Computer Science, New York, USA.

[18] C.W. Chong, P. Raveendran, R. Mukundan, A comparative analysis of algorithms for fast computation of Zernike moments, Pattern Recognition 36 (3) , pp. 731–742, 2003.

[19] R. El Ayachi, M. Fakir and B. Bouikhalene « Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network », chapter in the book "Document Recognition and Understanding", ISBN 978-953-307-995-0, INTECH, 2011.

[20] Y. Ait ouguengay , M. Taalabi,. Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe : Phase d'apprentissage". 5e Conférence internationale sur les "Systèmes Intelligents : Théories et Applications", Paris : Europia, cop. 2009 (impr. au Maroc), ISBN 978-2-909285-55-3. (2009)

Handwritten Tifinagh Text Recognition Using Fuzzy K-NN and Bi-gram Language Model

Said Gounane and Mohammad Fakir
Faculty of sciences and technology
University Sultan Moulay Slimane
Beni Mellal, Morocco

Belaid Bouikhalen
Multidisciplinary Faculty
University Sultan Moulay Slimane
Beni Mellal, Morocco

***Abstract*—In this paper we present a new approach in Tifinagh character recognition using a combination of, k-nearest neighbor algorithm and the bigram language model. After the preprocessing of the text image, and the word segmentation, for each image character, the k-NN algorithm proposes candidates weighted of their membership degree. Then we use the bigram language model to choose the most appropriate sequence of characters. Results show that our method increases the recognition rate.**

***Keywords*—Tifinagh character recognition; fuzzy k-nearest neighbor; features extraction; bigram language model.**

I. Introduction

The task of handwritten text recognition is challenging and has been a focus of research for several decades [1][3][7]. A human always use his language knowledge when reading a text. Thus the input image is not enough to estimate the most likely characters (or words) sequence, but also the target language [2]. Thus, modern recognizers use a statistical approach in which the returned characters (words) sequence is the one that maximizes the probability with respect to the input image according to a model trained on handwritten text as well as the language according to a model trained on a language corpus.

This paper is organized as follows: Section 2 presents tow techniques used for features extraction, the gravity center distance method and the pixel density method. The fuzzy k-Nearest neighbor and bi-gram language model are presented in section 3. Our complete handwriting recognition process is outlined in section 4. Experimental results and discussions are in section 5, then the conclusion in section 6.

II. Featurs extraction

Pr-processing techniques like word segmentation, thinning, slant correction and smoothening are applied. To extract the image features, we use a linear combination of the gravity center distance method and the pixel density method. Those two methods require the spatial division of the character image into W × H boxes so that the portions of character will be in some of these boxes. There could be boxes that are empty. The choice of number of boxes is arrived at by experimentation.

A. Gravity center destance

For each box B, the gravity center ***G*** of black pixels is computed. The feature is then obtained by dividing the distance ***d*** between ***G*** and bottom left corner of the box by the length of the diagonal of that box D "Fig 1".

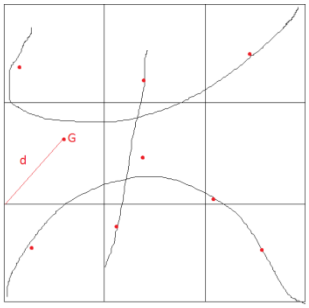


Fig. 1. Gravity center destance method

One can easily see that this characterization is invariant of the character image dimensions. Hence an image of whatever size gets transformed into a vector of W × H predetermined dimensions.

B. Pixel dencity

As in the first method, the character Image is divided into W × H boxes. The feature for each box is the number of black pixels divided by the total number of pixels in that box.

The same as the pixel density method, this characterization is invariant of the character image dimensions.

III. Fuzzy k-Nearest Neighbor and Bi-gram language model

In this section, we give a brief presentation of both techniques used in this work, fuzzy k-nearest neighbor and bigram language model.

A. Fuzzy k-nearest neighbo

Fuzzy k-Nearest neighbor is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition, image processing and many others[6][8]. Some successful applications are including recognition of handwriting.

It's a very simple technique. It works based on an unsupervised clustering algorithm like fuzzy k-means algorithm to determine prototypes representing each cluster. Then the minimum distance from the query instance to these prototypes are used to determine the K-nearest neighbors. After and basing on membership function, we take the neighbor with the maximum membership degree to be the prediction of the query instance.

From a given labeled examples, and using fuzzy k-means algorithm, we generate a number of prototypes P_{ij} representing each class C_i . In this step, the number of training samples is reduced and the outliers are excluded.

For a given image character X, the k-Nearest neighbor prototypes $N_l \in \{P_{ij}\}, l = 1 \dots k$ are generated (using distance similarity), then the membership degree of X to the class C_i , is calculated as follows [5] :

$$\mu_i(X) = \frac{\sum_{j=1}^{k} \mu_{ij}/d(X,N_j)^{2/(m-1)}}{\sum_{j=1}^{k} 1/d(X,N_j)^{2/(m-1)}} \qquad (1)$$

The term μ_{ij} is the membership degree of N_j to the class C_i . This term is equal to 1 if $N_j \in C_i$, 0 otherwise.

This algorithm has better performances versus the classical k-nearest neighbor algorithm. In the classic approach, the vector X is affected to the class with major prototypes in $\{N_{j=1..k}\}$ with no consideration of how close these prototypes are to X. However in the fuzzy approach, the vector X could be affected to a class that have only one prototype in $\{N_{j=1..k}\}$ just because it's closer enough to X and the other k-1 prototypes are not (fig).

A. *Bigram language model*

An N-gram model is a type of probabilistic language model for predicting the next item in a sequence. N-gram models are now widely used in probability, communication theory, computational biology, data compression and in statistical natural language processing.

Let $P(w)$ be the probability that a word w (sequence of characters $w = \{c_1, c_2, \dots, c_n\}$ appears in a dictionary. By using the chain rule, this probability can be computed as follows:

$$P(w) = P(c_1) \times \prod_{i=2}^{n} P(c_i/c_1, c_2, \dots, c_{i-1}) \qquad (2)$$

By using the Markov assumption, this probability is approximated by:

$$P(w) \approx P(c_1) \times \prod_{i=2}^{n} P(c_i/c_{i-1}) \qquad (3)$$

Each character depends only on the previous one rather than all the previous characters. This assumption is important because it massively simplifies the problem of learning the language model from data.

$$P(c_i/c_{i-1}) = \frac{count(c_{i-1},c_i)}{count(c_{i-1})} \qquad (4)$$

In practice, however, one should smooth the probability distributions by also assigning non-zero probabilities to unseen bigrams [4]. The technique used is called the add-one smoothing, and it's given by the following expression:

$$P(c_i/c_{i-1}) = \frac{count(c_{i-1},c_i)+1}{count(c_{i-1})+v} \qquad (5)$$

Where, v is the number of characters in the used language alphabet.

In order to use the bigram context of the first and the last character in a word, we need to augment this word with a special symbols <s> and </s> at the beginning and the end of the word respectively.

For example, to compute P("□□□□"), we first need to augment this word with <s>and </s>. Then we have:

P ("□□□□") =P (□/<s>).P (□/□).P (□/□).P (□/□).P (□/</s >)

IV. Recognition System

Our recognition system is made of three blocks "Fig. 2": the pre-processing block, the isolated character recognition block and the word recognition block.

First, the word image is pre-processed by thinning; slant correction and smoothening. The word image is then segmented to a sequence of character images.

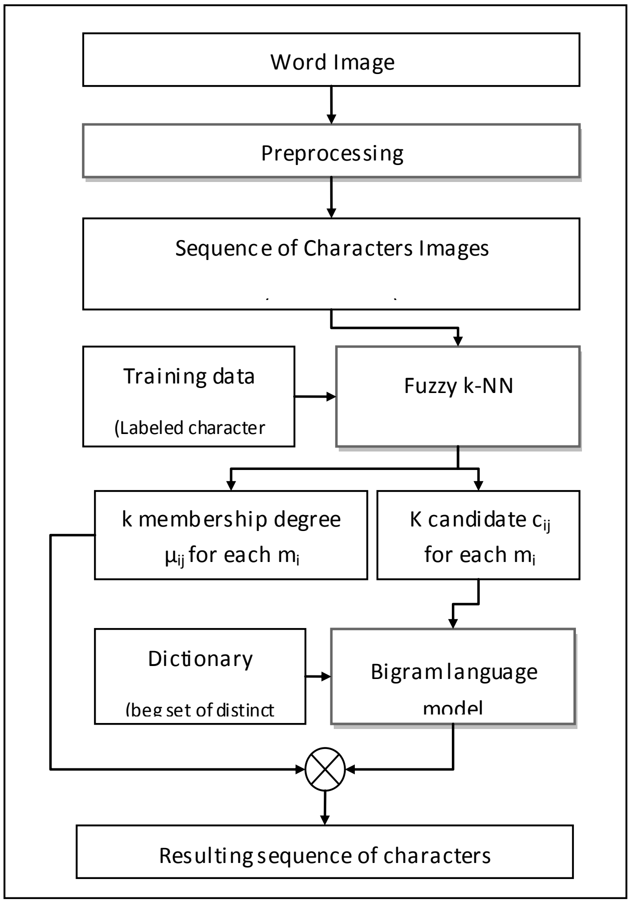


Fig. 2. Recognition system

For each character image the fuzzy k-nearest neighbor proposes candidates, each one with a membership degree $\mu(c_{ij})$ (the j^th candidate proposed for the i^th character image). From these candidates we generate all m possible combinations $w_{i=1..m}$. The membership degree of each word is given by:

$$\mu(w_i) = \prod_{c \in w_i} \mu(c), \; i = 1 \dots m \qquad (6)$$

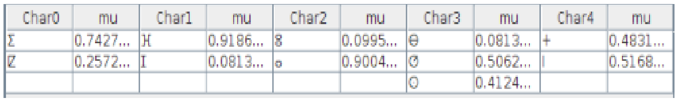
For example, in the word image "□□□□□" the k-NN algorithm propose a set of candidates for each character in the word image "Fig. 3". For the first character "□" tow candidates are proposed, "□" with membership degree 0.742 and "□" with a membership degree 0.258, the decision is clear with no ambiguity "□" because "□" has the higher membership degree. The same case for the second and third character. For the forth character "□" it proposes three candidates "□": 0.081, "□":0.506 and "□":0.412 the decision become ambiguous. The same problem with the last character, it proposes "□": 0.484 and "□":0.516 for the character "□". Using these membership degrees, the fuzzy k-nearest neighbor algorithm gives the character sequence "□□□□□" as result, instead of "□□□□□".

In the previews stage of our recognition system, the k-nearest neighbor algorithm misrecognizes some ambiguous characters. To avoid this problem we use the bigrams to select the most appropriate word as follows:

$$\widetilde{w} = argmax_{w \epsilon \{w_{i=1..m}\}}(P(w).\mu(w)) \qquad (7)$$

A. Algorithm

```
For each word image w
        For each c in w
                Using fuzzy k-NN algorithm, Generate
                candidates and their membership degrees
        End for
        Generate all m possible combinations $w_{i=1..m}$.
        For each word w$_i$
                Compute $\mu(w_i)$ using (1) and (6)
                Compute P(w$_i$) using (3)
        End for
        The selected sequence of characters
        $\widetilde{w}$ is given by (7)
End for
```



Char0	mu	Char1	mu	Char2	mu	Char3	mu	Char4	mu
Σ	0.7427...	H	0.9186...	8	0.0995...	Θ	0.0813...	+	0.4831...
Z	0.2572...	I	0.0813...	ο	0.9004...	Ø	0.5062...	|	0.5168...
						⊙	0.4124...		

Fig. 3. Proposed candidates for each character image and thier membershep degrees.

V. Experimental Results

In order to test our approach, we made up a java desktop application. The data base used to train the fuzzy k-nearest neighbor was made up of 1940 handwritten characters image. The dictionary used in the bigram model was a set of 1059 different Amazigh words written in Tifinagh. In addition to the character images database, and the dictionary size, Results obtained depends on many other parameters: the features extraction method, the size of the grid used in the features extraction $W \times H$, the number of neighbors k and the fuzziness parameter m.

A. Features extraction

In the features extraction stage we divide the character image into $W \times H$ boxes. Since most Tifinagh characters are symmetric with respect to x or y axes or both (□, □), it's more appropriate to use an odd numbers for both W and H. In addition, if W and H are too high we lose generality and we do more computation, and if they are too low we lose precision and many deferent characters will be seen as one. Table 1 shows the recognition rate for deferent choice of W and H.

TABLE I. Reconition rate for diferent gride sizes

	Recognition rate					
	3x3	*3x5*	*5x5*	*5x7*	*7x5*	*9x11*
FKNN	80.87	85.24	88.73	86.99	88.26	81.88
FKNN & Bigrams	84.20	86.64	91.05	89.31	89.66	85.48

The gravity center distance method gives information about the position of the black pixels in the box. Therefore, tow boxes with deferent amount of black pixels but with the same gravity center will have the same feature value. In the other hand, the pixel density method gives information about the amount of the black pixels in a box. The weakness of this method is that it gives no information about the position of black pixels in that box. So, tow boxes with same amount of black pixels will have the same feature value, even if the black pixel distribution is not the same.

In order to take advantage of both methods we used a linear combination of these two methods. So if v_d is the feature given by the pixel density method and $v_g$the one given by the gravity center distance, the used feature is given by the following expression:

$$v = \gamma v_g + (1-\gamma)v_d \quad 0 \leq \gamma \leq 1 \qquad (8)$$

"Fig. 4" shows that the combination of pixel density and gravity center distance method ($\gamma = 0.02$) gives better results than using only one method ($\gamma = 1 or\ 0$). We can also see that, for all values of γ, the k-nearest neighbor combined to the bigram model (solid curve), produces higher performances than the only use of the k-nearest neighbors algorithm (dashed curve).

B. Neighbors number k and the fuzziness factom m

The k-nearest neighbor algorithm is based on the equation (1). This equation contains tow parameters that affects the accuracy of the classifier, these parameters are the number of nearest neighbor k and the fuzziness factor m. High values of k will produce large number of candidates with small membership degrees. This big number of candidates leads to a huge number of combinations, and then, some of these combinations could have a higher probability even if they are far away from the real word image "Fig. 5". In the other hand if k is too low, the k-nearest neighbor will not propose enough candidates to be used by the bigram model "Fig. 5".

"Fig. 6" shows that, if the m parameter is too large all proposed candidate will have approximately the same membership degree and the decision will be made by the bigrams.

An example that illustrates an inappropriate choice of k and m is shown in “Fig. 7”. Where k=11, m=3 and the word image “□□□□□□”, for each character image, the membership degrees, of proposed candidates are all between 0.08 and 0.38, the number of possible words is 72576.

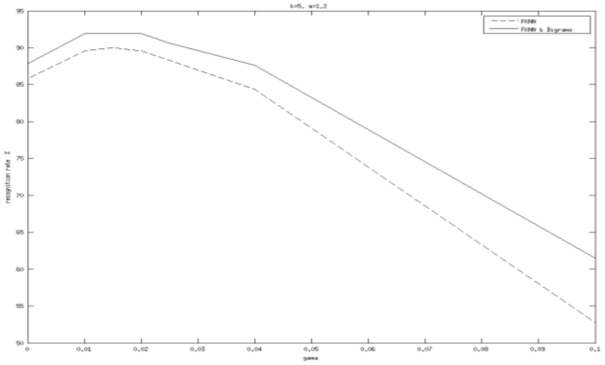


Fig. 4. Recognition rate function of γ, Fuzzy KNN & bigrams in solide line, FKNN in dashed line

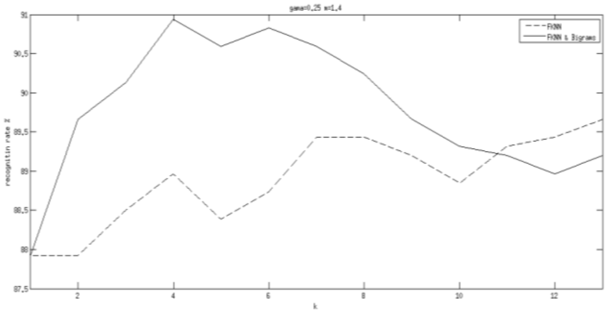


Fig. 5. Recognition rate function of k, Fuzzy KNN & bigrams in solide line, FKNN in dashed line

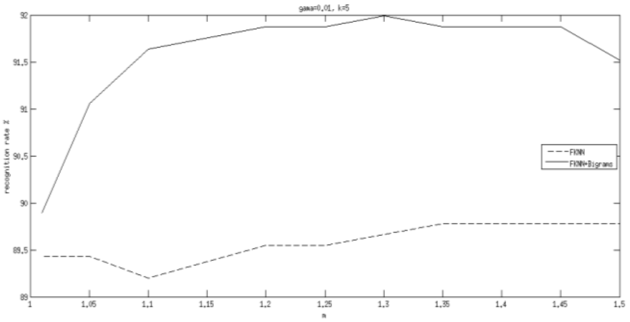
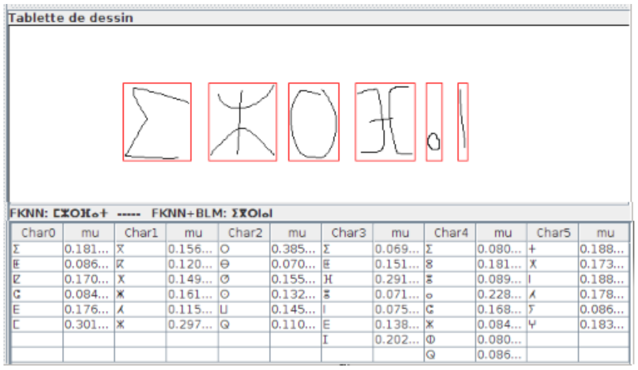


Fig. 6. Recognition rate function of m, Fuzzy KNN & bigrams in solide line, FKNN in dashed line



FKNN: ⊏ẊOЖₒ+ ----- FKNN+BLM: ΣẊOǀₒǀ

Char0	mu	Char1	mu	Char2	mu	Char3	mu	Char4	mu	Char5	mu
Σ	0.181...	Ẋ	0.156...	O	0.385...	Σ	0.069...	Σ	0.080...	+	0.188...
Ɇ	0.086...	Ʀ	0.120...	Θ	0.070...	Ɇ	0.151...	8	0.181...	X	0.173...
Ꝁ	0.170...	X	0.149...	Ø	0.155...	H	0.291...	Ξ	0.089...	ǀ	0.188...
G	0.084...	Ж	0.161...	⊙	0.132...	Ξ	0.071...	ₒ	0.228...	ʎ	0.178...
E	0.176...	ʎ	0.115...	U	0.145...	ǀ	0.075...	G	0.168...	Ƹ	0.086...
⊏	0.301...	Ж	0.297...	Q	0.110...	E	0.138...	Ж	0.084...	Ψ	0.183...
						I	0.202...	Φ	0.080...		
								Q	0.086...		

Fig. 7. Proposed candidate with an inapropriat values of k=11 and m=3.

The result given by the fuzzy k-nearest neighbor algorithm was “□□□□□□” and “□□□□□□” by the combination of fuzzy k-NN and bigram model. In both cases tow characters were misrecognized, the difference is that the second one “sounds” to be a real Amazigh word “□□□□□□”.

VI. Conclusion

In this work we presented a new technique for handwritten Tifinagh text recognition. In the features extraction stage we used a linear combination of the gravity center distance and pixel density. Then the fuzzy k nearest neighbor proposes candidates for each character image with its membership degree, the bigram model assigns a probability for each characters sequence. The selected characters sequence is the one that has the higher product. With a good choice of the number of neighbors k, and the fuzziness factor m, bigram model can improved significantly the recognition rate.

References

[1] A. L. Koerich, R. S. (2003). Large Vocabulary Of f-Line Handwriting Recognition: A Survey.Pattern Analysis and Applications. 97–121.

[2] Bunke, U.-V. M. (2001). Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. Int'l Journal of Pattern Recognition and Artificial Intelligence , 15:65–90.

[3] D.Slezak, J. B. Image processing and Pattern recognition. New York: Springer.

[4] Martin H, Jurafsky. D. (2000). Speech and language processing. Prentice Hall.

[5] J.Sagau. (n.d.). Logique flou en classification. Techniques de l'Ingénieur , H3638.

[6] M.Hanmandlu, A. A. Fuzzy Model Based Recognition of Handwritten Hindi Numerals using bacterial foraging. 6th IEEE/ACIS ICIS.

[7] S.Theodoridis, &. K. (2009). Pattern recognation (fort edition). Elsevier.

[8] Zheng, F. (n.d.). K-Means-based fuzzy classifier.

Performance evaluation of ad hoc routing protocols in VANETs

Mohammed ERRITALI, Bouabid El Ouahidi
laboratory of computer science research
Department of Computer Science
Mohamed V University – Faculty of Sciences Rabat, Morocco

Abstract—**The objective of this work is to compare with simulating, using OPNET the performance of five Ad hoc routing protocols: DSR, AODV, OLSR ,TORA and GRP , and to examine the impact of mobility and the density of nodes on the behavior of these protocols in a Vehicular Ad hoc NETwork (VANET). The results show that there is not a protocol that is favorite for all evaluation criteria. Indeed, each protocol has different behavior in relation to performance metrics considered, including the rate of routing packets sent, delay, and the debit.**

Keywords—Routing protocols; OPNET; Performance evaluation; DSR; AODV; OLSR; TORA; GRP.

I. INTRODUCTION

The VANETs (Vehicular Ad hoc NETworks) [1,2] are a new form of mobile ad hoc networks used to establish communications between vehicles or with infrastructure located on roadsides. They are characterized by a dynamic topology and are generally ad hoc networks, that is to say without management and control infrastructure of communication.

These networks are used to meet the needs of communication applied to transportation networks [1,2] to improve the driving and road safety for road users.

In a VANET network routing is an important mechanism for finding and maintaining a communication path between a pair of remote nodes.

In recent years, the improved performance of routing mechanisms in ad hoc networks was one of the main issues. This is reflected by the appearance of hundreds of protocols in the literature [3,4,5,6,7,8,9,10,11,12,13] of which only a few have been subjected to standardization.

For the evaluation of these protocols simulation offers a vision on the performance of the protocol itself before adopting and commercializing by a organization of standardization and brings us closer to the real use of the protocol, that helps us to select the best protocols that have a good behavior in different scenario, the simulation also allows the developer to test their protocols routing algorithms to improve them.

The performance metrics of ad hoc protocols have been discussed in the first RFC published by the MANET working group [15], by separating these metric to external metrics measurement (how the performance of the protocol is perceived by mechanisms using routing) and internal measurement (for the same level of external performance, two algorithms will deploy different volumes overhead).

External measures the effectiveness of an ad hoc protocol includes:

- The delay
- The debit
- The acquisition time of a road that is only valid for reactive protocols.
- The percentage of segments received out of sequence

To measure the internal efficiency of the protocol, the following metrics are recommended:

- The average number of bits of data transmitted / number of data bits received
- The average number of control bits transmitted / number of data bits received
- The average number of control packets and data transmitted / number of data packets received.

The purpose of this paper is to study the performance of ad hoc routing protocols to meet the following three questions:

1) What are the main differences between the ad hoc routing protocols?

2) What routing protocol provides better performance in vehicular ad-hoc networks?

3) What are the factors that influence the performance of these routing protocols?

In attempting to answer the above questions, we used OPNET [14] to compare the performance of the five protocols AODV, DSR, TORA, OLSR and GRP to examine the impact of mobility and density of nodes on the behavior these protocols regarding the speed, the traffic of routing packet sent, the debit and delay.

II. AD HOC ROUTING PROTOCOLS

Routing is the mechanism that allows the nodes of a vehicular network to communicate continually with each other information about the topology and link state in order to determine the best routes.

Figure 1 illustrates the taxonomy of these routing protocols which can be classified as topology-based and geographic (position-based) in VANET.

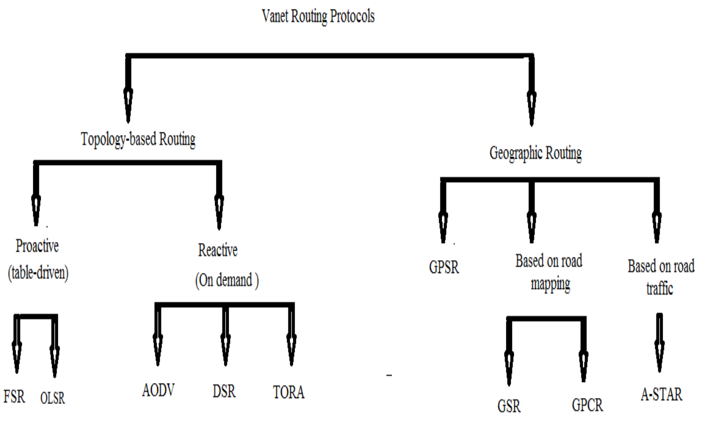


Fig.1. VANET routing Protocols

A. Topology-based Protocols:

These routing protocols use the link information which exists in the network to send the data packets from source to destination. They can also be classified as active (table-driven) and reactive (on demand) routing.

1) Proactive (table-driven):

Proactive protocols are used to establish the routes in advance on the basis of the periodic exchange of routing tables.

a) Fisheye State Routing (FSR)

FSR [3] is a proactive protocol link state. It assumes that a faraway topology change has not a significant influence on the calculation of the route locally. Therefore, the exchange of routing table updates is a function of the distance. More a node is faraway less it receives local topology updates.

FSR does not flood the network with updates, but all nodes periodically exchange with its neighbors partial routing update information. Indeed, all links propagates hop by hop in each sends.

FSR has a complete map of the network but cannot guarantee accuracy of all connections between farther nodes.

b) The protocol Optimized Link State Routing Protocol (OLSR)

OLSR [4] is a link-state protocol, which optimizes the way of broadcast of control messages to save bandwidth consumption through the use of the concept of "multipoint relays" (MPRs) shown Figure 2 in which each node selects a subset of its neighbors to forward its packets when broadcast.

Based on the distribution using the MPRs, all nodes in the network are achieved with fewer repetitions. A MPR set of node N is the minimal set of its 1-hop neighbors that cover (in the direction of the communication range) two-hop neighbors.

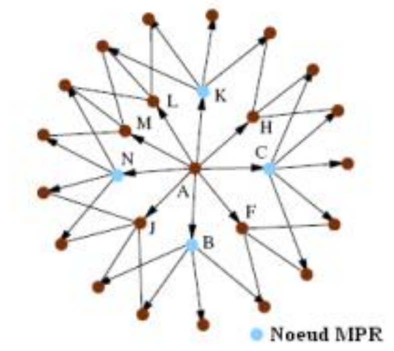


Fig.2. Multipoint relays in OLSR

2) Reactive (On Demand)

Reactive protocols, only keep the roads in use for routing. On demand at the time of packet routing, the protocol will search through the network a route to the destination. The conventional method of route lookup is to flood the network with a query, in order to find the target station, which responds with the reverse path.

a) Dynamic Source Routing (DSR)

The DSR protocol [5] is based on the "source routing", which means that the path traveled by the packet is included in the header of the data packet from the source to be read by routers.

The DSR protocol is composed of two distinct mechanisms: the first is used to find routes on demand, and the second is responsible for the maintenance of the communication route in progress.

The limitation of this protocol is that the route maintenance process does not locally repair a broken link and the performance of the protocol decreases with increasing mobility.

b) Ad Hoc On Demand Distance Vector (AODV)

AODV [6] is a reactive protocol based on the principles of distance vector routing protocols. It borrows its mechanisms in DSR discovery and maintenance of routes and uses a hop by hop routing, and sequence numbers.

AODV builds routes by using a series of queries "route request / route reply".

When a source node wants to establish a route to a destination for which it does not already have a route, it broadcasts a packet "route request" (RREQ) throughout the network. Nodes receiving this packet update their information for the source and set pointers of return to the source in the routing tables. In addition to the IP of the source, the current sequence number and a broadcast identifier, the RREQ also contains the sequence number of the destination most recent known by the source.

A node receiving a RREQ packet will then emit a "route reply" (RREP) if it is the destination or if it has a route to the destination with a sequence number greater than or equal to that included in the RREQ. If this is the case, it sends a RREP packet to the source. Otherwise, it re-broadcasts the RREQ. Nodes keep track of each source IP and identifiers of RREQ broadcast packets. If they receive a RREQ which they have already dealt with, they remove it.

c) Temporally Ordered Routing Protocol (TORA)

TORA [7] algorithm belongs to the class of algorithms (Link Reversal). It is based on the principle algorithms that try to maintain the property called "destination orientation" directed acyclic graphs (DAG). A directed acyclic graph is oriented destination if there is still a possible path to a specified destination. The graph is undirected destination if at least one link becomes faulty. In this case, the algorithms use the concept of reverse links. This concept ensures the transformation of the previous paragraph in a directed graph destination in a finite time. To maintain the DAG oriented destination, the TORA algorithm uses the concept of node size. each node has a size that exchange it with all of its immediate neighbors. This new concept is used in the orientation of the network links. A link is always directed from the node to the larger to the smaller node size.

The protocol is based on three phases: the creation of routes, maintenance of routes and clearing of route using three separate packets QRY, UPD and CLR.

TORA adapts to the mobility of vehicular networks by storing multiple paths to the same destination, so that many topology changes will not affect the routing of data, unless all paths that lead to destination are lost.

B. Position based protocols:

It is a routing taking into account the geographical position [8] of the nodes. To perform a geographic routing in an ad hoc network, it is essential that: all nodes have location means: via native system such as GPS, and that the source node knows the position of the destination node.

Various solutions have been proposed. Among these:

a) Greedy Perimeter Stateless Routing (GPSR) :

The GPSR [2,9,10] algorithm uses the position of the nodes in addition to the destination to determine the next node to which retransmit the packet. By default, a node GPSR uses greedy mode and forwards the packet to the neighbor closest to the destination. The Neighborhood detection is done by periodic diffusion of Beacon messages containing the address of the node and its position (x, y).

b) Geographic Source Routing (GSR)

GSR [2,11] is based on the geographical position and the information about the road topology in order to build a knowledge adapted to the urban environment. According to the protocol GSR, a source vehicle wishing to send a data packet to a target vehicle, calculates the shortest routing path to reach this target vehicle using geographical information of road map. We note that the routing path in question is calculated in its entirety, using for example the algorithm Djikstra. From the calculated routing path, the vehicle source selects a sequence of intersections through which the data packet must travel to reach the target vehicle. This sequence of intersections is formed by a set of fixe geographical points of data packet passage. And to send messages from one intersection to another, the authors propose to use a greedy approach.

c) Greedy, Perimeter Coordinator Routing (GPCR) :

GPCR [2,12] is a combination of protocol GPSR and the use of road mapping. The authors assume that each node can know whether if it is in intersection which case it acquires the status of a coordinator node. Thus, the messages are transmitted along the road using a greedy approach with preference given to coordinators nodes.This means that choosing of next relay node, a node coordinator (a node at an intersection), is preferred to a non-coordinator node, even if it is not the closest to the destination, and this to avoid obstacles radios.

d) Anchor-based Street and Traffic Aware Routing (A-STAR)

A-STAR [2,13] is a routing protocol based on the geographical position of Metropolitan vehicular environment. It uses special information on city bus routes to identify an anchor path with high connectivity for packet forwarding. A-STAR protocol is similar to the GSR by adopting an approach based routing anchor that reflects the characteristics of the streets. However, unlike GSR it calculates the "anchor paths" depending on the traffic. A weight is assigned to each street according to its capacity (large or Small Street).

III. THE METRICS INCLUDED IN OUR SIMULATION

A. The routing traffic

Routing traffic sent by each node is an important measure of scalability of routing protocol, and therefore the network. It is defined as the total number of routing packets transmitted over the network, and is expressed in bits per second or packets per second, this number is recalculated in each transmitting or receiving packets. This test is done to see if the network is loaded or not. Routing traffic is expressed by the following equation:

Traffic routing = number of packets-(number of received packets +number of packets lost) (1)

This metric is used to select the protocol that encumbers the least network.

B. The debit

The debit is the transfer rate at a given time, it is expressed in bits per second. The factors that affect the debit in VANET are frequent topology changes and limited bandwidth.

C. delay

The delay is the time required for a packet to arrive at the destination, ie: the time that elapses between the generation of packet source and its receipt by the application layer of the destination. For each packet in the network this period is calculated as follows:

delay = time of receipt of package - transmission time of the packet (2)

IV. MODELING OF SIMULATION AND SCENARIOS

Our goal in this section is to model the behavior of routing protocols under different conditions of load and speed mobility in vehicles to infrastructure communications (V2I) [2] to examine the statistical averages of debit, delay, and the routing traffic generated in the entire VANET network .

We modeled the network with a determined surface area. Nodes (vehicles) and the server were randomly distributed in the geographic area. In the simulations, we used the TCP traffic to study the effects of ad hoc protocols on TCP-based applications such as web and file transfer.

We have configured an FTP application profile in our study where the nodes are a WLAN mobile clients with a debit fixed at 11 Mbps, while the destination is a server of a wireless network also with a debit of 11 Mbps.

We also used the mobility model " random waypoint " widely accepted to describe mobility behavior of VANET nodes.

We also selected two-speed mobility, the first is 10m / s (36Km / h), which reflects a low mobility, and the second is 28ms / s (100.8 km / h), which reflects the high mobility of nodes.

For the density of the network we have chosen the number of nodes: 5 for a reliable density, 20 for a medium density and 40 for a high density.

Table 1 below shows the 30 scenarios used in our simulation. For each routing protocol we proceed to 6 scenarios, each scenario is characterized either by a change in the mobility speed or the number of nodes:

TABLE I. Scenarios Used

Scenarios	Nombre des nœuds	Vitesse de mobilité (m/S)	Protocole utilisé
Scenario 1	5	10	DSR
Scenario 2	5	28	DSR
Scenario 3	20	10	DSR
Scenario 4	20	28	DSR
Scenario 5	40	10	DSR
Scenario 6	40	28	DSR
Scenario 7	5	10	AODV
Scenario 8	5	28	AODV
Scenario 9	20	10	AODV
Scenario 10	20	28	AODV
Scenario 11	40	10	AODV
Scenario 12	40	28	AODV
Scenario 13	5	10	TORA
Scenario 14	5	28	TORA
Scenario 15	20	10	TORA
Scenario 16	20	28	TORA
Scenario 17	40	10	TORA
Scenario 18	40	28	TORA
Scenario 19	5	10	OLSR
Scenario 20	5	28	OLSR
Scenario 21	20	10	OLSR
Scenario 22	20	28	OLSR
Scenario 23	40	10	OLSR
Scenario 24	40	28	OLSR
Scenario 25	5	10	GRP
Scenario 26	5	28	GRP
Scenario 27	20	10	GRP
Scenario 28	20	28	GRP
Scenario 29	40	10	GRP
Scenario 30	40	28	GRP

V. RESULTS AND INTERPRETATION

In this section, we will discuss and analyze the results of our simulations of routing protocols used, starting with an analysis of routing traffic sent in the network, then the delay and finally the network debit.

a) Routing traffic

According to Figures 3, 4, 5, 6, 7, 8 we note for the DSR protocol, in all scenarios that mobility does not affect the traffic in the case of 5 nodes it is estimated by 10 bits / sec in the case of nodes 20 it is 200 bits / sec and in the case of nodes 40 it is 500 bits / s.

Regarding the protocol AODV we notice the same thing, mobility does not have any influence on the traffic.

The TORA protocol knows a decrease in routing traffic in the case of 5 and 20 nodes with different mobility speed and stabilizes with a traffic of 425 bits / sec to 5 nodes and 6,000 bits / sec to 20 nodes. With more nodes and high speed mobility: TORA generates a big traffic.

For OLSR mobility does not affect routing traffic but increases with the network load and is almost at the same level in all scenarios, with a 1150 bit / sec in the case of a network of 5 nodes, 9000 bits / sec for the network of 20 nodes and 30000 bits / sec for high network load estimated withe 40 nodes. For the routing protocol GRP we note that traffic routing rapidly decreases and remains almost stable and weak the entire duration of the simulation for all scenarios, traffic is estimated by 550 bits / sec in the case of 5 nodes and 2000 bits / sec in the case of 20 and 40 nodes. Our simulation shows that the geographic routing GRP does not generate a large traffic routing in comparison with TORA and OLSR protocols that use control messages to detect topology changes in their neighborhood.

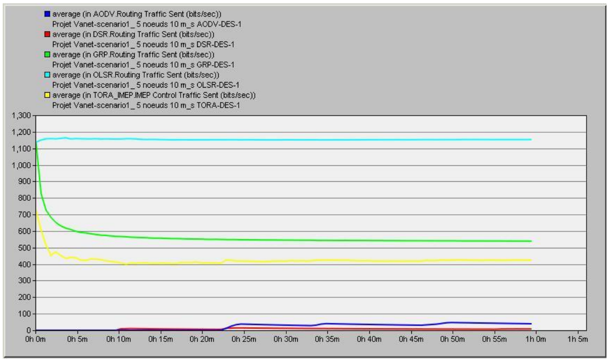


Fig.3. Routing traffic of 5 nodes and mobility speed of 10 m/s.

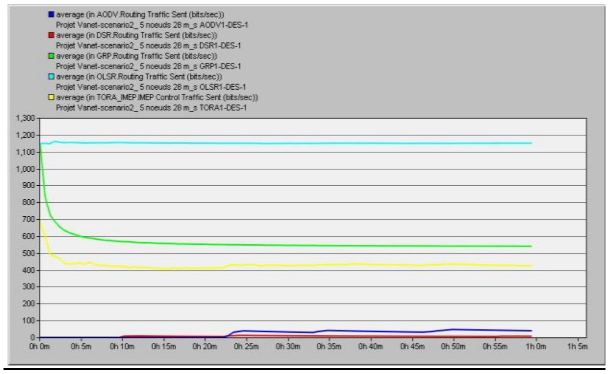


Fig.4. Routing traffic of 5 nodes and mobility speed of 28 m/s.

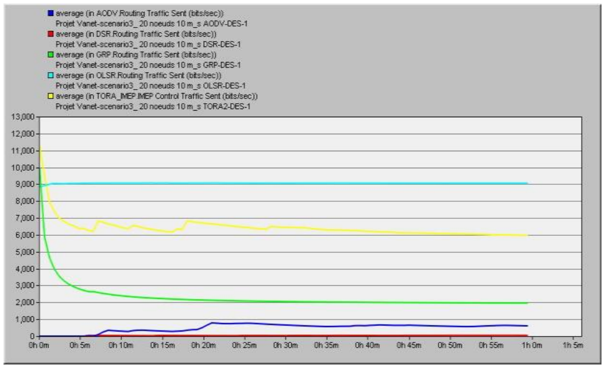


Fig.5. Routing traffic of 20 nodes and mobility speed of 10 m/s.

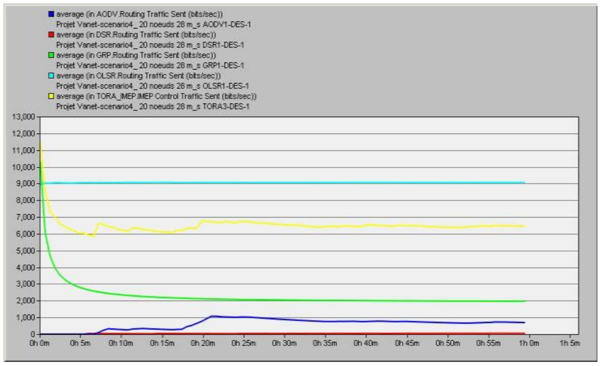


Fig.6. Routing traffic of 20 nodes and mobility speed of 28 m/s.

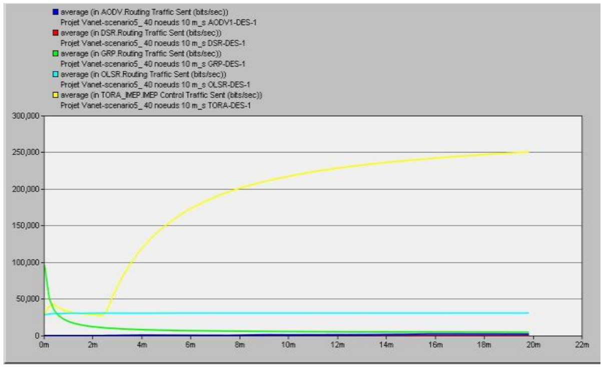


Fig.7. Routing traffic of 40 nodes and mobility speed of 10 m/s.

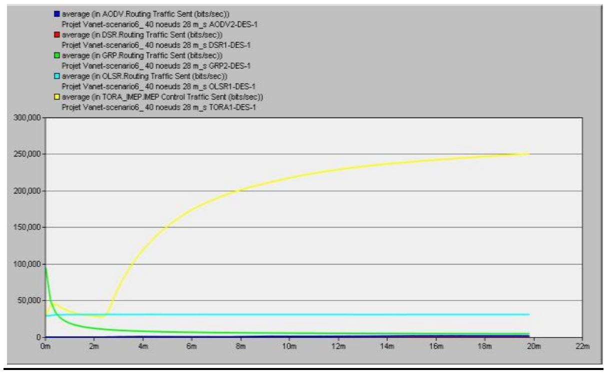


Fig.8. Routing traffic of 40 nodes and mobility speed of 28 m/s.

b) The delay

According to figures 9, 10, 11, 12, 13, 14 in all scenarios, we observe that OLSR has the lowest delay that varies between 0.0003 sec and 0.0004 sec. OLSR is a proactive routing protocol, which means that the network connections are always ready every time when the application layer has a traffic to send. The periodic updates keep the routing paths available for use and the absence of a high latency induced by the discovery process routes in OLSR explains its relatively low delay. With a larger number of mobile nodes, the performances of this protocol are in competition with those of AODV protocol with a delay of 0.0005 sec.

For small networks with 5 nodes, we observe that TORA has a delay of 0.0013 sec and outperforms DSR, which has a delay of 0.0033 sec in the two case of mobility. On the other hand, TORA is in concurrence with AODV in the case of low speed where they have almost the same delay estimated by 0.0013 sec but the TORA protocol outperforms AODV in the case of high speed with a delay of 0, 0070 sec.

When the number of nodes is growing, we note that TORA suffers from a significant deterioration in his delay. One reason for this deterioration is the process of route discovery. The AODV protocol also has a very low delay estimated by 0, 0013 sec in the case of 5 nodes and 0.0005 sec in the case of 20 nodes, and it happens the second after OLSR. This is observed in all scenarios, except in the case of reducing the number of nodes and high speed where it outperforms TORA. However, we observe that the performance of AODV protocol improves with increasing the number of nodes.

DSR shows a uniform delay in the case of the two speed mobility with 5 and 20 nodes where it is 0.0033 sec and increases in the case of 40 nodes has 0/0070 sec.

The DSR protocol uses cached routes and often it sends traffic on obsolete routes, which can cause retransmissions and cause excessive delays.

The GRP protocol outlines a more coherent delay than other protocols ranging between 0.0004 and 0.0005 in all scenarios.

In conclusion, we observed that OLSR exhibits a very low delay in all scenarios. While for the TORA protocol we note that it has a big delay in a network with a big traffic, and that the mobility did not affect his delay.

Regarding the AODV protocol, we notice an improvement in delay when the network size increases and that the speed does not affect the delay, and finally the DSR protocol adopts a coherent approach and undergoes further delay when the network grows but the speed of mobility does not have profound effects on its performance.

The three reactive protocols had high delays due to the increasing demands of route discovery, unlike the geographic routing, which exposes an important delay only in the case of a less loaded network.

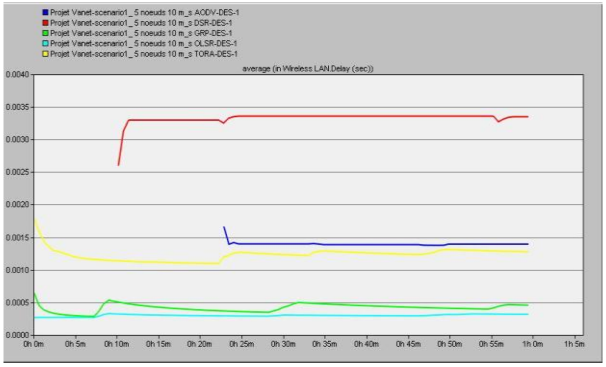


Fig.9. Delay of 5 nodes and mobility speed of 10 m/s.

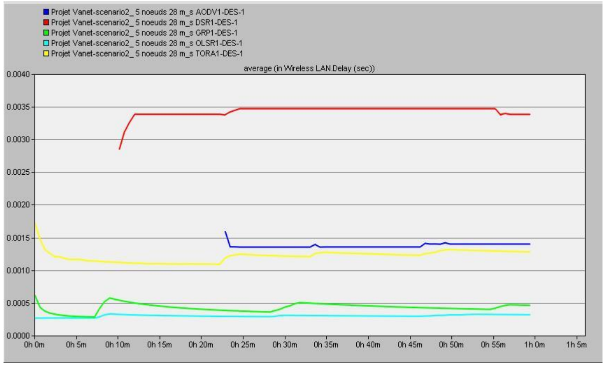


Fig.10. Delay of 5 nodes and mobility speed of 28 m/s.

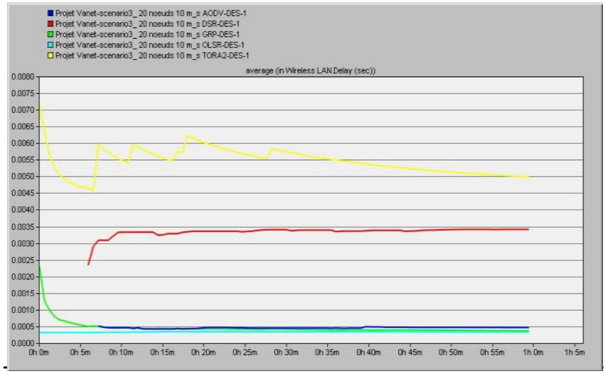


Fig.11. Delay of 20 nodes and mobility speed of 10 m/s.

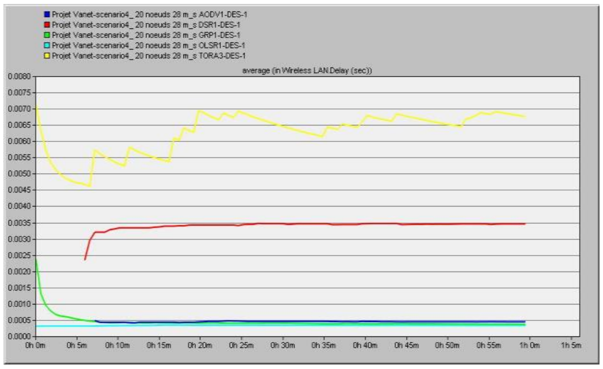


Fig.12. Delay of 20 nodes and mobility speed of 28 m/s.

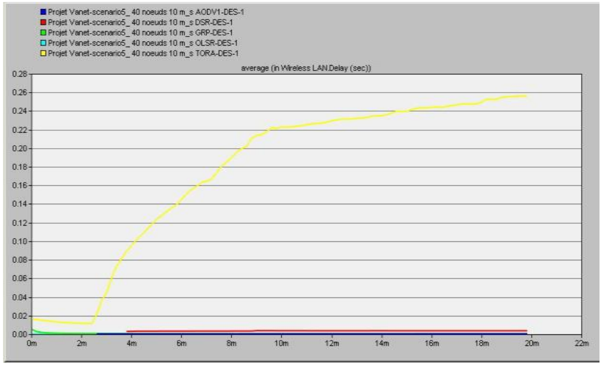


Fig.13. Delay of 40 nodes and mobility speed of 10 m/s.

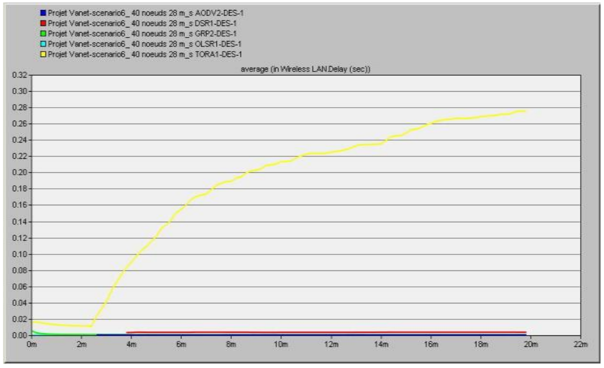


Fig.14. Delay of 40 nodes and mobility speed of 28 m/s.

c) The debit

According to Figures 15, 16, 17, 18, 19, 20 we observe that the OLSR protocol has a very significant debit where it exceeds 12,000 bits / sec.

In the network with 5 nodes, DSR has a debit of 4700 bits / sec and outperforms TORA which has a debit of 3500 bits / sec and AODV which has a debit of 3000 bits / sec at the two case of mobility speed.

The AODV protocol has a lowest debit about 20,000 bits / s When the network are composed with a small number of nodes that move with low speed , and It has a big debit approximately 100,000 bits / sec in the case of a higher speed mobility in larger networks.

OLSR has a constant debit in the case of two mobility speed where it exceeds 12,000 bits / sec.

The debit of TORA protocol Increases in the case of high mobility to reach 400,000 bits / sec.

For the GRP protocol we notice that de debit decreases on the Beginning of the simulation and then remains almost stable but for a network of 5 nodes with the two mobility speed the debit varies According to a form closer to a sine function around 6800 bits / sec, and remains significant in all phases of the simulation.

According to our simulation we notice that the geographic routing has better debit compared to DSR, AODV and TORA.

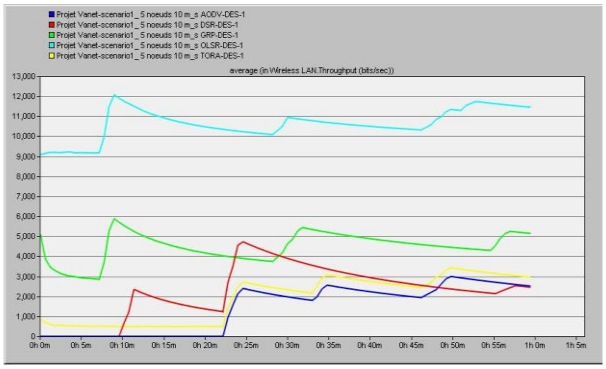


Fig.15. Debit of 5 nodes and mobility speed of 10 m/s

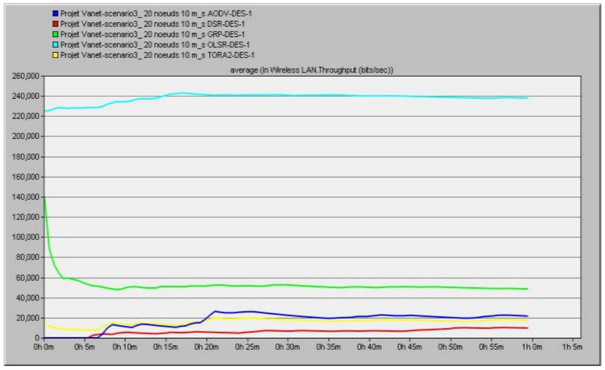


Fig.16. Debit of 5 nodes and mobility speed of 28 m/s

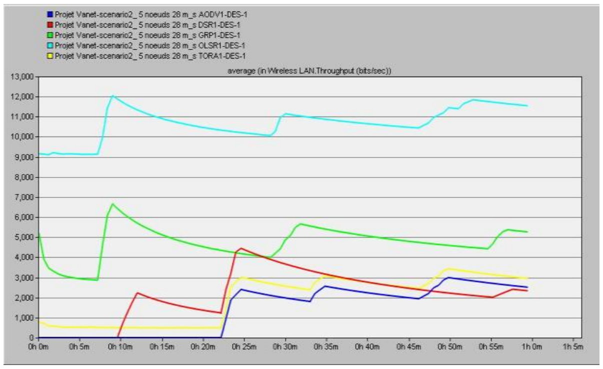


Fig.17. Debit of 20 nodes and mobility speed of 10 m/s

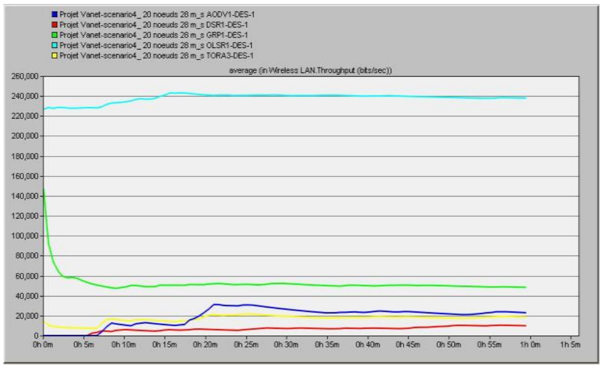


Fig.18. Debit of 20 nodes and mobility speed of 28 m/s

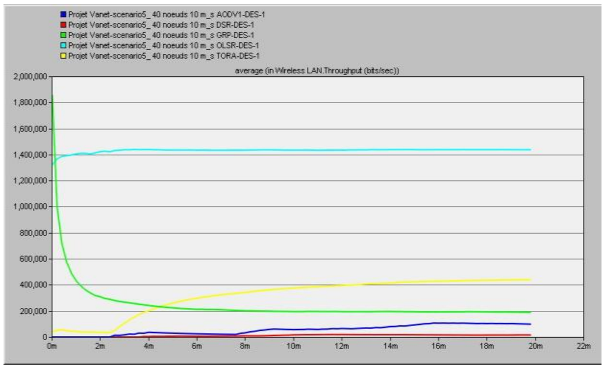


Fig.19. Debit of 40 nodes and mobility speed of 10 m/s

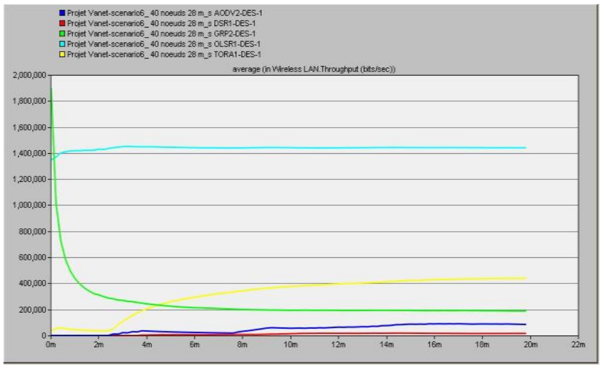


Fig.20. Debit of 40 nodes and mobility speed of 28 m/s

VI. Conclusion

In this paper we have presented a comparative study of five protocols AODV, DSR, TORA, OLSR and GRP regarding the debit, the routing traffic sent and delay, the table below summarizes the characteristics noted in our simulation.

TABLE II. Comparison Of Routing Protocols

The protocol used	Conclusions
OLSR	OLSR outperforms AODV, DSR and TORA in debit and delay. OLSR has the worst performance in routing traffic generated. It is therefore well suited for high debit networks. High traffic routing shows that OLSR is not suitable for low debit networks.
DSR	DSR is the best candidate in networks of low debit and medium in size. However, it has a large traffic routing in the case of long paths (large networks). It inserts in the packet IP addresses of all nodes in the route it uses.
AODV	AODV is suitable for networks of low and medium load with low speed mobility.
TORA	TORA is suitable for small networks. It loses its performance under rapid expansion.
GRP	GRP significantly reduces the signaling (control packets), especially in large and dynamic networks. Simplicity and low memory requirements. More suited to large networks.

We found in this study that geographic routing is best suited for highly dynamic vehicular networks because of its low latency and low amount of routing traffic.

The problem remains in the ability to choose metrics that provide appropriate results to both vehicular environments (cities, highways), which have different characteristics.

References

[1] S. Khalfallah, M. Jerbi, M. O. Cherif , S.M. Senouci , Bertrand Ducourthial, Expérimentations des communications inter-véhicules, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP), Les Arcs : France (2008).

[2] M.JERBI , Protocoles pour les communications dans les réseaux de véhicules en environnement urbain : Routage et GeoCast basés sur les intersections . Thèse , France (2008).

[3] Bijan Paul, Md. Ibrahim, Md. Abu Naser Bikas, VANET Routing Protocols: Pros and Cons, International Journal of Computer Applications (0975 – 8887),Volume 20– No.3, April 2011.

[4] Thomas Clausen and Philippe Jacquet. IETF RFC3626 : Optimized link state routing protocol (OLSR). October 2003.

[5] D.B. Johnson, D.A. Maltz, Y-C. Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) , Internet Draft : <draft-ietf-manetdsr-10.txt>, 19 Juillet 2004.

[6] C.E. Perkins, E.M. Belding-Royer, and S. Das, Ad Hoc On Demand Distance Vector (AODV) Routing , IETF RFC 3561, 2003.

[7] V. Park, S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", IETF Internet draft, work in progress, draft-ietf-manet-tora-spec-04.txt, 2001.

[8] Giordano S., et al., "Position based routing algorithms for ad hoc networks: A taxonomy," in Ad Hoc Wireless Networking, X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, December 2003.

[9] Karp, B. and Kung, H.T., Greedy Perimeter Stateless Routing for Wireless Networks, inproceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, August, 2000.

[10] Young Jin Kim and Ramesh Govidan ,Greedy Perimeter Stateless Routing (GPSR) Detailed-level Design Specification (DDS) Version 0.1 , Department of Computer Science University of Southern California, (July 6, 2003).

[11] C. LOCHERT, H. HARTENSTEIN, J. TIAN, D. HERRMANN, H. Füßler, M. MAUVE, "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments", IEEE Intelligent Vehicles Symposium (IV2003), pp. 156-161, Columbus, OH, USA,(June 2003).

[12] C. LOCHERT, M. MAUVE, H. Füßler, H. HARTENSTEIN, "Geographic Routing in City Scenarios", MobiCom (2004).

[13] B.-C. SEET, G. LIU, B.-S. LEE, C. H. FOH, K. J. WONG, K.-K. Lee, "A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications". NETWORKING 2004, 989-999.

[14] Opnet, http://www.opnet.com/

[15] Corson (S.) et Macker (J.). – Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations. – RFC n2501, IETF,(janvier 1999).

Recognition of Amazigh characters using SURF & GIST descriptors

H. Moudni, M. Er-rouidi, M. Oujaoura
Computer Science Department
Faculty of Science and Technology
Sultan Moulay Slimane University
Béni Mellal, Morocco.

O. Bencharef
Computer Science Department
Higher School of Technology
Cadi Ayyad University
Essaouira, Morocco.

Abstract— **In this article, we describe the recognition system of Amazigh handwritten letters. The SURF descriptor, specifically the SURF-36, and the GIST descriptor are used for extracting feature vectors of each letter from our database which consists of 25740 manuscripts isolated Amazigh characters. All the feature vectors of each letter form a training set which is used to train the neural network so that it can calculate a single output on the information it receives. Finally, we made a comparative study between the SURF-36 descriptor and GIST descriptor.**

Keywords—SURF; GIST; Principal Component Analysis; Neural Network; Amazigh Characters.

I. Introduction

Today in Morocco, pattern recognition, especially the recognition of Amazigh characters has become a growing field in which several researchers work. In order to find algorithms that can solve the problems of computer pattern recognition, which are intuitively solved by humans, several research efforts have been made and some research works of Tifinagh characters are published [1, 2, 3, 4, 5]. In this context, we proposed a system that allows for the recognition of Amazigh handwritten characters using the simplest descriptors like SURF-36 and GIST Today in Morocco, pattern recognition, especially the recognition of Amazigh characters has become a growing field in which several researchers work. In order to find algorithms that can solve the problems of computer pattern recognition, which are intuitively solved by humans, several research efforts have been made. In this context, we proposed a system that allows for the recognition of Amazigh handwritten characters using the simplest descriptor, SURF-36 and GIST as descriptors, in addition to the neural network as a classifier.

For the rest of the paper, in Section 2 and Section 3, we present the SURF-36 and GIST descriptors in addition to their calculation steps. Section 4 and section 5 have been reserved respectively for the neural networks and the database used in this paper. In Section 6, we presented the principal component analysis technique and studied the possibility of its application in this case. Section 7 presents the discussion of results. Finally, in Section 8, we concluded our work.

II. SURF Descriptor

Speeded Up Robust Features (SURF) is a visual feature extraction algorithm from an image to describe it based on the detection of interest points. We worked with the reduced SURF descriptor (SURF-36) which is slightly worse compared to usual descriptor SURF. Nevertheless, it allows a very rapid adaptation, the performance remain acceptable in comparison with other descriptors in the literature.

The SURF descriptor is mainly known for its fast computation. Its algorithm consists of two main steps. The first one is to detect the interest points in the image and the second one is to describe these interest points using a vector of 36 features.

A. Detection of Interest Points

To decrease computation time, the image to be analyzed is transformed into an integral picture. The integral images allow fast calculation of convolution and rectangular areas. Let I our initial image, I (x, y) represents the pixel value of the image at coordinates x and y.

The integral image denoted IΣ (x, y) is an image of the same size as the original image, it is calculated from this image. Each pixel of the integral image contains the sum of pixels located above and left of the pixel in the original image.

The value of a pixel of the integral image IΣ (x, y) is defined on the basis of the image I by the following equation:

$$I_{\Sigma}(x,y)=\sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} I(x,y) \quad (1)$$

The pixels Areas in the image with high change of intensity are searched. The Hessian matrix, based on the calculation of partial derivatives of order two, is used for this. For a function of two variables f (x, y), the Hessian matrix is defined as follows:

$$H(f(x,y))=\begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial y \partial x} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix} \quad (2)$$

If the determinant of the Hessian matrix is positive, then the eigenvalues of the matrix are both positive or both negative, which means that an extremum is present. Points of interest will therefore be located where the determinant of the Hessian matrix is maximal. Specifically, the partial derivatives of the signal are calculated by convolution with a Gaussian. To gain speed calculation, these are approximated by a Gaussian step function called box filter.

The representation at lower levels of scale is achieved by increasing the size of the Gaussian filter. In the end, the interest points for which the determinant of the Hessian matrix is positive and which are local maximum in a neighborhood 3 * 3 * 3 (x-axis * y- axis * scale-axis) are retained.

B. *Description of Interest Points*

Once the interest points are extracted, the second step is to calculate the corresponding descriptor. The SURF descriptor describes the intensity of the pixels in a neighborhood around each interest point. The x and y Haar wavelets response is calculated in a neighborhood of 6s, where s is the scale at which the interest point was found. From these values, the dominant orientation of each point of interest is calculated by dragging a window orientation.

To calculate the descriptor, a square of size 20s oriented along the dominant orientation is extracted. This area is divided into 3 x 3 squares. For each of the sub-regions, Haar wavelets are calculated on 15 x 15 points.

Let dx and dy be the response to the Haar wavelet, four values are calculated for each sub-regions:

$$v_{subregion} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right] \qquad (3)$$

Finally, each of the extracted points in the previous step is described by a vector composed of 3*3*4 values that is 36 dimensions [6].

III. GIST DESCRIPTOR

In computer vision, GIST descriptors are a representation of an image in low dimension that contains enough information to identify the scene. Actually, any global descriptor must approach the GIST to be useful.

GIST descriptor was proposed by Oliva and more precisely by Torralba. They tried to capture the GIST descriptor of the image by analyzing the spatial frequencies and orientations. The global descriptor is built by combining the amplitudes obtained in the output of K Gabor filters at different scales and orientations. For reducing the size, each image in filter output is resized to a size N*N (N between 2 and 16), which gives a vector of dimension N*N*K. This dimension is further reduced through a principal component analysis (PCA), which also gives the weights applied to different filters [7].

IV. NEURAL NETWORKS

Neural networks are composed of simple elements (or neurons) working in parallel. These were strongly inspired from biological nervous system. As in nature, the functioning of the neural network is strongly influenced by the connections between the elements. It can lead a neural network to a specific task (eg OCR) by adjusting the values of connections (or weight) between the elements (neurons).

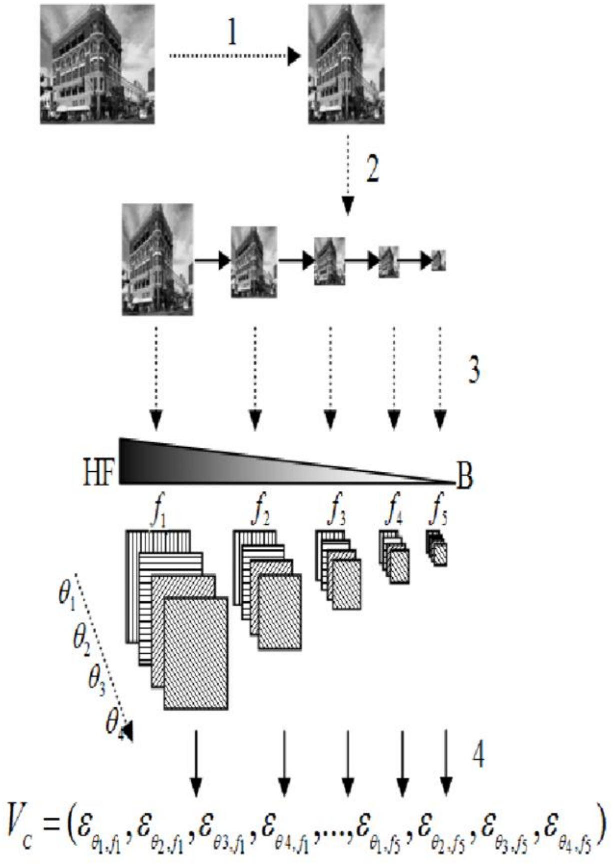

1
2
3
HF
B
f_1
f_2
f_3
f_4
f_5
θ_1
θ_2
θ_3
θ_4
4
$V_c = (\varepsilon_{\theta_1,f_1}, \varepsilon_{\theta_2,f_1}, \varepsilon_{\theta 3,f_1}, \varepsilon_{\theta 4,f_1}, \ldots, \varepsilon_{\theta_1,f_5}, \varepsilon_{\theta_2,f_5}, \varepsilon_{\theta_3,f_5}, \varepsilon_{\theta_4,f_5})$

Fig. 1. GIST Descriptor

In general, the neural networks learning tasks is done and performed so that for a particular entry, the neural network give a specific target. The weight adjustment is carried out by comparison of the network response (or output) and the target, until the output corresponds at best to the target [8].

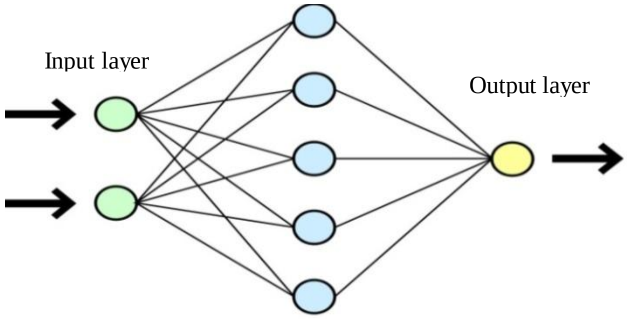


Fig. 2. Simplified diagram of a neural network

V. USED DATABASE

The database contains 33 handwritten characters Amazigh. Each character is represented by 780 ways and sizes which gives 25740 handwritten characters Amazigh. This database was developed at the Laboratory of IRF-SIC Ibn Zuhr University in Agadir, Morocco.

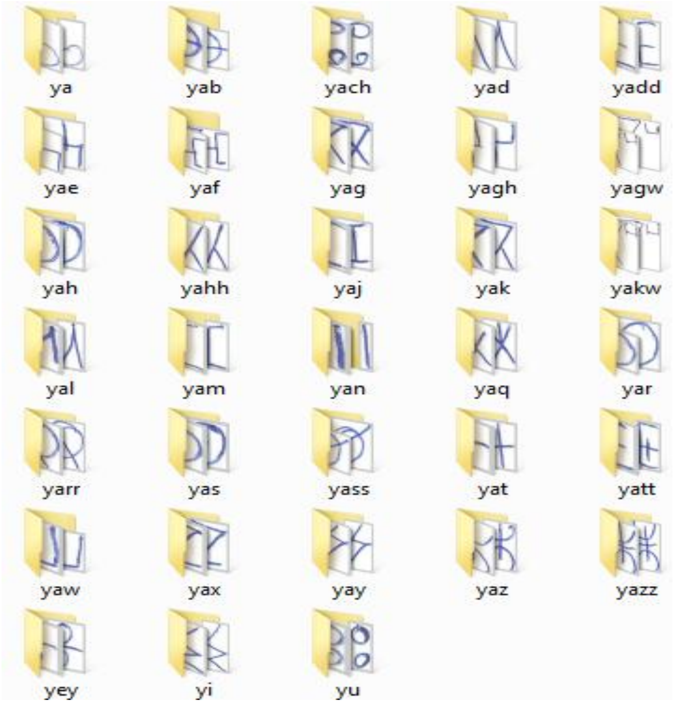


Fig. 3. The database files.

The basic representation of Amazigh characters is given by the following figure [9]:

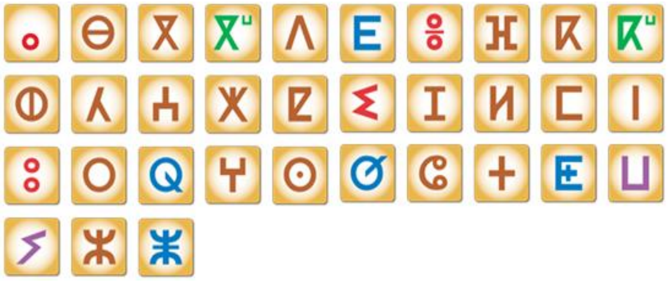


Fig. 4. The basic representation of Amazigh characters.

VI. Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is one of multivariate descriptive analyzes. The purpose of this analysis is to summarize the maximum possible information losing only the least possible information in order to:

- Facilitate the interpretation of a large number of initial data.
- Give more meaning to the reduced data.

Therefore, the PCA reduces data tables of large sizes in a small number of variables (usually 2 or 3) while keeping a maximum of information. Baseline variables are called 'metric'.

To analyze the results of PCA, it would be very helpful and useful to answer three questions:

- Are the data factorisable?
- How many factors should be retained?
- How to interpret the results?

To answer the first question, firstly, the correlation matrix should be observed. If several variables are correlated (> 0. 5), the factorization is possible. If not, the factorization has no sense or meaning and is therefore not recommended [10].

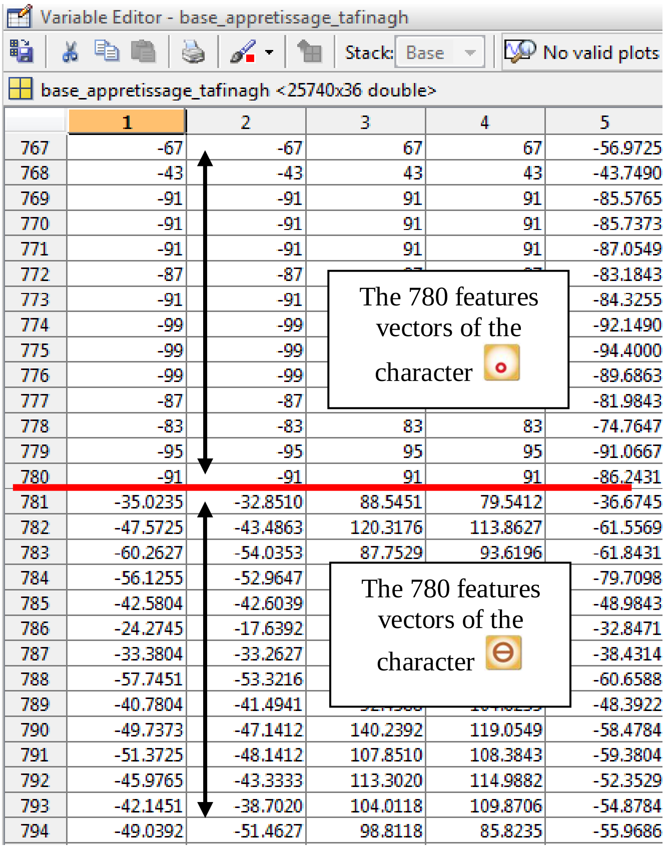
In our example we used the function corrcoef (base) from the MATLAB code to determine the correlation matrix. By examining this matrix, we find that several variables are not correlated (<0. 5).

So we cannot use the CPA for our database Amazigh characters.

VII. Results Interpretation

Our learning database, sized of (25740x36), contains all the characteristic vectors of 33 Amazigh characters (each character is represented by 780 different handwriting ways).

According to Figure 5, the feature vectors are different for each character (the 780 representations of the first character are different from the 780 representations of the second character, and so on).



Variable Editor - base_appretissage_tafinagh

base_appretissage_tafinagh <25740x36 double>

	1	2	3	4	5
767	-67	-67	67	67	-56.9725
768	-43	-43	43	43	-43.7490
769	-91	-91	91	91	-85.5765
770	-91	-91	91	91	-85.7373
771	-91	-91	91	91	-87.0549
772	-87	-87			-83.1843
773	-91	-91			-84.3255
774	-99	-99			-92.1490
775	-99	-99			-94.4000
776	-99	-99			-89.6863
777	-87	-87			-81.9843
778	-83	-83	83	83	-74.7647
779	-95	-95	95	95	-91.0667
780	-91	-91	91	91	-86.2431
781	-35.0235	-32.8510	88.5451	79.5412	-36.6745
782	-47.5725	-43.4863	120.3176	113.8627	-61.5569
783	-60.2627	-54.0353	87.7529	93.6196	-61.8431
784	-56.1255	-52.9647			-79.7098
785	-42.5804	-42.6039			-48.9843
786	-24.2745	-17.6392			-32.8471
787	-33.3804	-33.2627			-38.4314
788	-57.7451	-53.3216			-60.6588
789	-40.7804	-41.4941			-48.3922
790	-49.7373	-47.1412	140.2392	119.0549	-58.4784
791	-51.3725	-48.1412	107.8510	108.3843	-59.3804
792	-45.9765	-43.3333	113.3020	114.9882	-52.3529
793	-42.1451	-38.7020	104.0118	109.8706	-54.8784
794	-49.0392	-51.4627	98.8118	85.8235	-55.9686

The 780 features vectors of the character ⴰ

The 780 features vectors of the character ⴱ

Fig. 5. Overview of the learning database.

The produced learning database will be used as an input for the neural network. When the Neural network established its identification and final decision, we did a test on all vectors of the training database with a simple MATLAB simulation function sim(neuron, vector). To simplify this task, we created a program that simulates the entire database and giving as a result the error rate for each used descriptor.

We presented the summary of results in Table 1. The error rate is 25% for the SURF-36 descriptor and 17% for the GIST descriptor.

TABLE I. ERROR RATES AND EXECUTION TIME OF SURF-36 & GIST DESCRIPTORS

Descriptor	**SURF-36**	**GIST**
Error rate	25%	17%
Execution Time in Min	8	180

From Figures 6 and 7, we observe that the GIST descriptor is more powerful than the SURF-36 descriptor. But SURF-36 is faster than GIST. Indeed, the performance of SURF-36 took 8 minutes to process our database of Amazigh characters while GIST took over 3 hours.

VIII. CONCLUSION

We note that the SURF-36 descriptor is a little less efficient than the GIST descriptor, but it allows very rapid adaptation in recognition of Amazigh handwriting characters. The combination with the neural network, both for the SURF-36 descriptor or GIST descriptor gives insufficient results. In future work, other descriptors and other classifiers will be used to improve the recognition rate of the Amazigh characters recognition system.

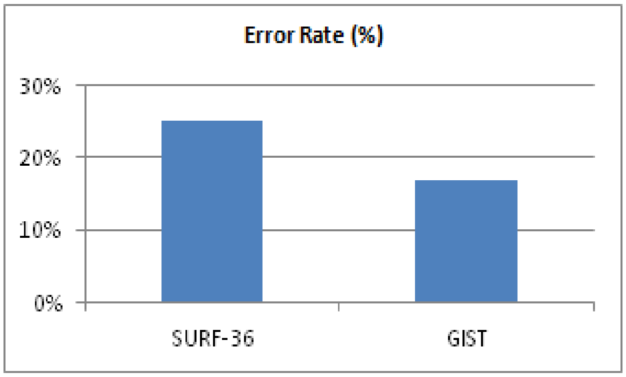


Fig. 6. SURF-36 and GIST descriptors error rate comparison.

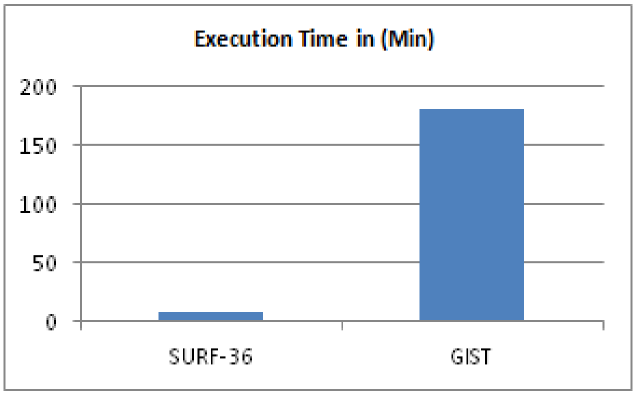


Fig. 7. Execution time Comparison of SURF-36 and GIST descriptors.

REFERENCES

[1] O. Bencharef, M. Fakir, B. Minaoui et B. Bouikhalene. (2011). Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence, 1-5.

[2] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, « Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata », ICGST-GVIP Journal, Volume 10, Issue 2, June 2010.

[3] O. Bencharef, M. Fakir, N. Idrissi, B. Bouikhalen et B. Minaoui, « Application de la géométrie riemannienne à la reconnaissance des caractères Tifinaghe », Agadir-Maroc, 06-07 Mai 2011. pp : 179 :188.

[4] R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts ». JATIT, vol. 20, No. 2, pp: 61-66, 2010.

[5] R. El Ayachi, M. Fakir et B. Bouikhalen. (2012). Transformation de Fourier et moments invariants appliqués à la reconnaissance des caractères Tifinaghe, Revue eTI, Numéro 6. 2012.

[6] Olivier Augereau, Nicholas Journet, Jean-Philippe Domenger, "Reconnaissance et Extraction de Pièces d'identité : Une application industrielle à la détection de cartes d'identité et de passeports," 1re soumission à CIFED 2012, le 8 décembre 2011.

[7] TOUSCH A.M. « Hiérarchies sémantiques pour l'annotation multifacette d'images ». Thèse de doctorat (Mathématiques, Informatique) soutenue le 15 février 2010 à l'Ecole des Ponts - Paris Tech.

[8] http://www.obs.univbpclermont.fr/atmos/enseignement/cours-Master-2A/cours_RN_2006.pdf

[9] http://upload.wikimedia.org/wikipedia/commons/a/a5/Alphtifingh.jpg

[10] http://www.lemoal.org/download/spss/ACP.

Printed Arabic Character Classification Using Cadre of Level Feature Extraction Technique

S.Nouri *, M.Fakir **
Department of Mathematics and Computer Science, Beni Mellal, Morocco

***Abstract*—a Feature extraction technique is important in character recognition, because they can enhance the efficacy of recognition in comparison to other approaches. This study aims to investigate the novel feature extraction technique called the Cadre of Level technique in order to represent printed characters and digits. This technique gives statistic and morphologic information, i.e. the calculation is based on a statistical approach but in the positions which can give some information about the morphologic of character. The image of a character is divided into 100 zones, then for each zone we average 5 extracted values (one value for each level) to 1 value for each zone, which gives 100 features for each character. This technique was applied to 105 different characters and 10 different digits of Arabic printed script. K-Nearest Neighbor algorithm was used to classify the printed characters and digits.**

Keywords—Arabic Character; Cadre of Level; Recognition; K-Nearest Neighbor; Digits.

I. Introduction

Recognition of Arabic characters has been famously researched since of the 1970s [1] and this branch of research based in general on the analytic approach which proposes the systems for recognizing all type of Arabic characters has greatly contributed to the field. In actuality, there is not an ideal system to use for recognition of Arabic script. In our contribution we propose a system to recognize one type of Arabic printed script with new features to show if we can ameliorate the level of recognition.

The systems propose three stages, first for processing and preparing the images for a second stage. In this stage the image is scanned and transformed to a binary image, and then the sides of the character are localized for cutting additional parts from the image in order to resize it to 100×100 pixels. Next, two morphological operations are applied, first to find a skeleton of the character, and second, to close the holes in the connected component. In this second stage, in order to extract the features from each character or digit, we use an algorithm, namely Cadre of Level.

In the Cadre of Level technique, the matrix of the image is divided into 100 zones with each zone representing a matrix of 10×10. For each zone, a density of pixels is calculated for each level to get 5 values (1 for each level).Then averaging the values to obtain one value for each zone, we obtain finally a vector of 100 features. In the third stage, the Cadre of Level technique is applied to data sets including printed characters (105 classes) and Arabic printed digits (10 classes), then the data sets are classified by the k-Nearest Neighbors algorithm using the Spearman distance as a function for computing distances between data points.

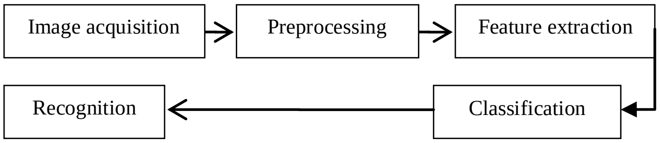
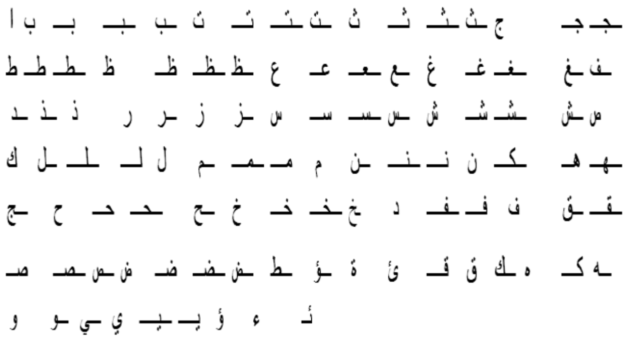


Fig.1. Stages of recognition system

In this study, the classification rates obtained using Cadre of Level algorithms are compared.

II. Characteristics And Problems Of Printed Arabic Characters [2]

- The Arabic script is written from right to left and it is cursive,
- Arabic script consists of 28 characters, but in cursive script each character can take 2, 3, or 4 different forms, and some isolated characters can form distinct ligatures (for example ؤ,لا,ئ), so in total there exists more than 105 different characters.
- No concept of upper or lowercase characters exists.
- The same Arabic printed script can be used in more than one font.
- Similarity of some characters like, ح,خ ,غ ,ع ,ف ,ق ب , ت…
- The noise, background of the image, overlapping of script, and the shape of the characters give us a small quantity of information to distinguish all characters from each other: therefore we experienced some difficulty in the classification stage.



(a)

0 1 2 3 4 5 6 7 8 9

(b)

Fig.2. (a) Arabic printed characters in different forms
(b) Digits used in Arabic printed script.

III. Preprocessing

To extract symbolic information from millions of pixels in document images, each component in the character recognition system is designed to reduce the amount of data .As the first important step, image and data preprocessing serve the purpose of extracting regions of interest, enhancing and cleaning up the images, so that they can be directly and efficiently processed by the feature extraction component. [10]

A. *Thresholding [9]*

Thresholding is a technique that aims to transform a matrix of three dimensions representing an R.G.B. image into a matrix of two dimensions representing a gray image where values are between 0 and 1. Then this matrix is transformed into a binary matrix where values are between 0 or 1, to obtain a binary image to use in the next step of processing.

B. *Noise reduction. [7] [8]*

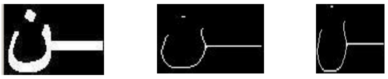
Various techniques, such as morphological operations, are used to connect unconnected pixels, to remove isolated pixels, and to smooth boundaries. those techniques are also used to minimize the effects of noise on image quality or quantity of information that we use to extract features. In this approach we used thresholding to eliminate noise and unnecessary information and keep most significant information. After extracting the skeleton we used the techniques of mathematical morphology to eliminate the negative effects such as distortion and unconnected pixels, subsequently keeping the general shape of characters and digits.

C. *Normalization*

To process images of the same size we have resized the matrices representing the characters or digits to fixed dimensions, the choice of dimension depends on the nature of the object (character, digit,) and quantity of information that we need to extract features.
We set a size of 100 × 100 to represent characters and 80 × 80 to represent digits.

D. *Skeletonization*

To minimize the influence from thick script, the image was normalized and thinned using a thin [3] algorithm to define the skeleton shape for localizing a character and using it to extract Cadre of Level features.



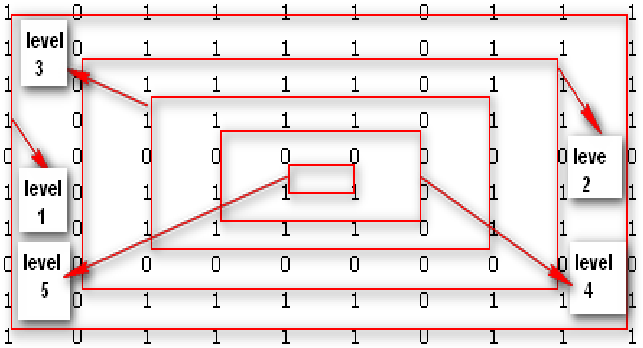
(a) (b) (c)

Fig.3. (a) Image of character before
(b) Thinned image
(c) Cropped image

IV. Features Extraction

Features extraction is a sensitive stage which directly influences the recognition rate. To provide a symbolic of information and to maximize the possibility of distinguishing each character or digit from each other, we have opted for a novel method of extracting 100 features which were obtained from the shape and distribution of pixels in the image. For that we have proposed a technique named Cadre of Level. In the Cadre of Level technique, the image of a skeleton character is used to divide it into 100 zones, each representing a matrix of 10×10 pixels. Each zone is used to statistically calculate values into 1 and 0 as following:

- The matrix is divided into 5 cadres, each cadre representing one level (Figure 3.).



1	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1

Fig.4. Example of cadre of level of one zone

For each level the following calculation was performed.

- r_1= densities of pixels in the upper row
- r_2= densities of pixels in the lower row
- cl =densities of pixels in the left column
- cr= densities of pixels in the right column

$$r = (r_1 + r_2)/2 \quad (1)$$

$$c = (cl + cr)/2 \quad (2)$$

$$l_i = (r + c)/2 \quad (3)$$

- l_i Represents features of the level number *i*.

The features of the five levels were divided by 5 and the result represents a feature for one zone.

Finally we obtained 100 features to represent one vector of (1×100) for each digit or character**.**

V. Classification

In the classification, to classify the data sets which represent Arabic characters and digits, we have used the k-Nearest Neighbor algorithm with the Spearman distance as a function for computing distances between data points.

A. *K-Nearest Neighbor algorithm*

The k-Nearest Neighbor algorithm (*k*-NN) [4], [5] is a method for classifying objects based on closest training examples in the feature space. Conceder of X and Y respectively test and training data. X is a vector of the size N features. Y is a matrix of $N_s \times N$ with N_s the number of samples in training data.

The first step of k-Nearest Neighbor algorithm (*k*-NN) involves calculating the distances between X and the N_s samples of the training data Y. In this paper, three distances were studied: City block D1, Spearman D2 and Correlation distance D3. All calculated features were positive. The second step involved sorting the distances obtained. Conceder D the vector contained the K-nearest neighbor distances. The final step is to determine the frequency of each 3 classes contained in the vector D. The minimum distance was adopted to classify the test data. The parameter K, the type of distance used, and features extracted from those three parameters can be influenced as a result of using k-Nearest Neighbor algorithm.

1) Distance Metrics [6]

Given an m-by-n data matrix X, which is treated as mx (1-by-n) row vectors x_1, x_2... x_m, and my-by-n data matrix Y, which is treated as my (1-by-n) row vectors y_1, y_2, ...,y_m, the various distances between the vector x_s and y_t are defined as follows:

a) Spearman distance

$$d_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'}\sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \qquad (4)$$

Where

- r_{sj} is the rank of x_{sj} taken over $x_{1j}, x_{2j}, ...x_{mx,j}$,
- r_{tj} is the rank of y_{tj} taken over $y_{1j}, y_{2j}, ...y_{my,j}$,
- r_s and r_t are the coordinate-wise rank vectors of x_s and y_t, i.e., $r_s = (r_{s1}, r_{s2}, ... r_{sn})$ and
- $r_t = (r_{t1}, r_{t2}, ... r_{tn})$

$$\bar{r}_s = \frac{1}{n}\sum_j r_{sj} = \frac{(n+1)}{2} \qquad (5)$$

$$\bar{r}_t = \frac{1}{n}\sum_j r_{tj} = \frac{(n+1)}{2} \qquad (6)$$

b) City block metric

$$d_{sj} = \sum_{j=1}^{n} \left| x_{sj} - y_{tj} \right| \qquad (7)$$

Notice that the city block distance is a special case of the Minkowski metric, where p=1.

c) Correlation distance

$$d_{st} = 1 - \frac{(x_s - \bar{x}_s)(y_t - \bar{y}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(y_t - \bar{y}_t)(y_t - \bar{y}_t)'}} \qquad (8)$$

Where

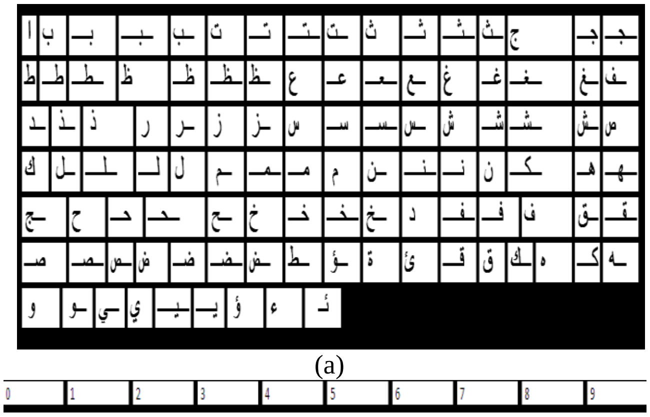
$$\bar{x}_s = \frac{1}{n}\sum_j x_{sj} \qquad (9)$$

$$\bar{y}_t = \frac{1}{n}\sum_j y_{tj} \qquad (10)$$

VI. Experimental Results

In our study we have adopted 'Simplified Arabic Fixed' as a printed font for digits and characters. The Cadre of Level was applied on the binary skeleton images as a method to extract 100 features, then 64 features, and k-Nearest Neighbor algorithm was applied to classify the data of characters and digits. We have analyzed 17,850 different images of characters using 16,590 images for training and 1,260 for testing. And we analyzed 480 images of numerals using 360 images for training and 120 digits for testing.

To compare the results obtained, we have used three different distances, City block, Spearman and Correlation distance.



(a)

(b)

Fig.5. (a) Sample of Arabic printed characters database used
(b) Sample of printed digits database used

The following tables illustrate the results obtained.

TABLE I. Results Of Character Recognition Using Cadre Of Level With The K-Nearest Neighbor Algorithm.

Distance	64 features	100 features
Spearman	96.82%	98.65%
Correlation	87.69%	96.03%
City block	91.58%	96.98 %

TABLE II. RESULTS OF DIGITS RECOGNITION USING CADRE OF LEVEL WITH K-NEAREST NEIGHBOR ALGORITHM.

Distance	100 Features	64 Features
Spearman	95%	99.16%
Correlation	92.5%	88.33%
City Block	90%	90.83%

In the first table (see Table 1), we have obtained 98.65% as the highest rate percentage recognized from printed characters using Spearman distance, then 96.98 % using City block distance, and 96.03% using Correlation distance. These statistics represent analysis on 100 features. But when we use 64 features, we obtain 96.82% as the highest rate percentage using Spearman distance.

In the second Table (see Table 2), we have obtained 99.16% as the highest rate percentage recognized from printed digits using Spearman distance for 64 features.

Preprocessing and distance metrics chosen with k-Nearest Neighbor algorithm *k*-N.N. were influenced directly by the performance of recognition system.

VII. CONCLUSION

In this paper, a new type of feature extraction, namely Cadre of Level extraction, is proposed to elaborate a new offline system for printed Arabic digits and characters. Two approaches using 64 features and 100 features have been chosen to elaborate data for training and testing using k-Nearest Neighbor algorithm. To compare the recognition efficiency of the proposed Cadre of Level method of feature extraction, the k-Nearest Neighbor is used with three different distances, City block, Spearman, and Correlation distance. Experimental results reveal that 100 features gives better recognition accuracy than 64 features for all distances using Cadre of Level feature extraction. From the test results it has been identified that the Spearman distance yields the highest recognition of character accuracy of 98.65% for 100 features and 96.82% for 64 features, and the highest recognition of digits accuracy of 99.16% for 64 features and 95% for 100 features. This study has proved that the field of recognition of Arabic printed characters needs more precision and specificity in all stages to obtain a stronger system of recognition. To improve it is a necessity to continue research with an alternate vision.

References

[1] M. Eden and M. Hall, ‘’ The characterisation of cursives writing ‘’ , proc.4th symp.Informatics Theory, London 1961, pp:287-299

[2] M. Fakir, C. Sodeyama, ” Machine recognition of Arabic printed scripts by dynamic programming matching, “ Transaction on informatics Systems, vol. 76, No 2, pp. 235-242, 1993.

[3] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column.

[4] Kartar Singh Siddharth , Mahesh Jangid, Renu Dhir, Rajneesh Rani, “Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features”, International Journal on Computer Science and Engineering (0975-3397), Vol. 3 No. 6 June 2011.

[5] Puneet Jhajj, D. Sharma, “Recognition of Isolated Handwritten Characters in Gurmukhi Script”, International Journal of Computer Applications (0975-8887), Vol. 4, No. 8, 2010.

[6] http://www.mathworks.com/help/stats/pdist.html

[7] Kong, T. Yung and Azriel Rosenfeld, Topological Algorithms for Digital Image Processing, Elsevier Science, Inc., 1996.

[8] Pratt, William K., Digital Image Processing, John Wiley & Sons, Inc., 1991.

[9] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[10] Book : ”Character recognition systems a guide for students and practioners”