Object-Based Scene Classification Modeled by Hidden Markov Models Architecture

Benrais Lamine, USTHB, Bab Ezzouar, Algeria*
Baha Nadia, USTHB, Bab Ezzouar, Algeria

ABSTRACT

Multiclass classification problems such as document classification, medical diagnosis, or scene classification are very challenging problems to address due to similarities between mutual classes. The use of strong and reliable tools is necessary in order to achieve good classification results. This paper addresses the scene classification problem using objects as attributes. The process of classification is modeled by a well-known mathematical tool: The hidden Markov models architecture. The authors introduce suitable relations that scale the parameters of the hidden Markov model into variables of the scene classification problem. The construction of hidden Markov chains is done with the support of proposed weight measures and sorting functions. Lastly, inference algorithms are proposed to extract the most suitable scene category from constructed discrete Markov chains. A parallelism approach is proposed where several discrete Markov chains are constructed at the same time in order to cover more possibilities and improve the accuracy of the classification process. To test the efficiency of the proposed method, this research provides numerous tests on different datasets (MIT Indoor, LabelMe, SUN397, and its refinef version SUN150). The authors also compare the obtained classification accuracies with some state of the art methods. This paper shows that the proposed approach distinguishes itself by outperforming the other methods while same datasets are tested.

KEYWORDS

Hidden Markov Models, Objects Based, Parallel Approach, Scene Classification, Weight Measures

1. INTRODUCTION

Having the knowledge of surrounding environments is a major advantage for any existing agent (human or robot) in taking decisions or achieving tasks. Therefore, the ability to classify the current scene into a specific label guides the agent into accomplishing a better work and meeting expectations. However, the state of the art of scene classification reveals huge and persistent difficulties in implementing a reliable classifier. The accuracy in classifying a scene and the number of scene categories (number of classes) remain major aspects in determining the classification consistency. Therefore, scene classification problem became an open and a challenging area of research. This paper addresses the scene classification problem (Li, 2010) (Sikirić, 2014) with objects as attributes shortened as (SC:O) modeled by hidden Markov models (HMM) architectures and algorithms (Ghahramani, 2001). This approach was chosen since the HMMs are well known to be strong and reliable mathematical tools for classification and prediction. Moreover, HMMs treated efficiently and with great success similar problems such as speech recognition (Gales, 2008) (Gautam, 2017), speech synthesis (Reddy, 2017),

DOI: 10.4018/IJCINI.20211001.oa6

*Corresponding Author

machine translation (Wang, 2017) (Vogel, 1996), handwriting recognition (Sangeetha, 2017), activity recognition (Ozawa, 2017) (Alp, 2017), sign language recognition (Khandelwal, 2017) etc. Following the same perspectives, scene classification problem is a very active and attractive area of research. It is found in several research domains such as traffic road (Sikirić, 2014) (Lin, 2011) where it helps in taking decisions and organizes traffic, streets and airports scene surveillance systems (Lin, 2007) (Foresti, 1998) (Besada, 2001) where the classification process suggests and points out anomalies and suspicious behaviors. The scene classification can as well be found in area of research treating navigations (Liu, 2019) (Chen, 2019) where robots benefit of the semantic information about the surrounding environment provided by the category of scene. Several other types of scenes can be addressed such as areal scenes (Zheng, 2019) (Devi, 2019), indoor scenes (Li, 2019) (Hayat, 2016), outdoor scenes (Payne, 2005), or even war scenes (Raja, 2012). It can also be benefit to prediction systems where, in some circumstances, actions are predicted from a given scene categories (Vu, 2014).

The first challenge of the proposed method consists on finding the right relations between the SC:O problem and the HMMs architecture and algorithms. Analogies between inputs and outputs parameters and prerequisites of both entities are analyzed. A perfect similarity was achieved which made us believe that the scene classification problem can be solved by a HMM architecture. On the other hand, experimentation made us realize that properly ordered input parameters presented to the HMM happen to be very critical for the construction of the discrete Markov chain (DMC) and thus affects the accuracy of classification process. This concern made us put in place weights and sorting functions to evaluate existing objects. Weight functions assign a proper measure to each object of the dataset in such a way it reflects its saliency. It can be calculated dependently and independently of any category of scene. Based on weight functions, we then introduce objects sorting functions in order to organize objects of the input scene. Two sorting functions are proposed, the static sorting function, which organizes the input scenes' objects before starting the construction of the DMC and the dynamic approach that organizes the objects of the input scene while the DMC is under construction. The way a sorting function organizes the scene's objects in order to present them to the DMC construction can be compared to the way a human eye perceives the scene and distinguishes the most salient objects in order to start the scene recognition process. i.e. comparison of human ability to recognize a scene with the proposed method. Once finished, the process generates one or many discrete Markov chains (DMC) containing scene categories that are most likely to represent the selected objects in the input scene. The process can be extended to compute several DMCs at the same time. A degree of parallelism is introduced in order to cover more scene categories. The next and final step consists on implementing an inference algorithm that retrieves the most suitable scene category from the DMC. This way of classification using HMMs is not common in the literature since HMMs, like SVMs (Weston, 1998), handle multiclass classification approach using the "one Vs all" architecture (Ghahramani, 2001), (Hinton, 2001). This paper introduces a novel classification approach that uses just one HMM and handles a multiclass scene classification problem.

The remainder of the paper is organized as follow. Section 2 presents an overview of existing approaches and methods treating the scene classification problem. Section 3 presents the formal definition of hidden Markov models (HMMs) and the construction of the discrete Markov chain (DMC). Section 4 introduces the proposed method and explains with details all the stated contributions which can be briefly summarized in the following:

- Formal definition of the scene classification problem that uses objects as attributes
- Analogy between the scene classification problem and the HMM architecture
- Weight and sorting functions
- Inference algorithms to extract the most suitable scene category

Finally, section 5 experiments the proposed method's accuracy and computation time. A comparison with some existing methods in the state of the art is also presented. We conclude by summarizing our results and outlining steps to improve the proposed scene classification accuracy using hidden Markov models.

2. RELATED WORKS

In recent years, several scene descriptors have been developed in order to increase the ability of computer vision systems on recognizing and classifying their surrounding environment. Arens et *al* (Arens, 2004) are one of the first to implement a scene classification experiment in concrete street traffic application retrieving textual description of videos sequences. Later on, their work has been improved by Pangercic et *al* (Pangercic, 2009) where a scene interpretation based a Description Logic (DL) and a top down guided 3D CAD model-based vision algorithm were implemented to bring more autonomous activity to robot on objects and scenes classification. Such as (Pangercic, 2009) logical languages for scene classifications have been widely studied (Sikos, 2017) (Baader, 2005) where predicates represent the different properties of the scene (objects, size, positions, etc.) while the logical inference system is used to identify the associated scene categories. The Description Logic (DL) was the most successful for representing a real-world state. Neumann et *al* (Neumann, 2008) introduced the DL as knowledge reasoning and representation system for scene classification with temporal and special relationships. Their proposed approach exploits relationship between objects, occurrences, events and episodes joining at the same time visual evidence and contextual information. A more specific contribution has been made by (Hummel, 2007) applying the DL for road scenes classification and intersections geometries. The formalism of the DL being similar to the problem of scene classification, the approach shows successful and promising results.

The complexity of scene classification increases relatively to the number and size of scenes. To face this issue, first proposed approaches were to reduce the choice of scene categories to a binary perception, for instance: Indoor/outdoor scene classification (Ghomsheh, 2012) (Szummer, 1998) and very satisfying results were obtained. Nevertheless, the approaches were not extendable to multiclass classifications. Another approaches consist on predicting the location of salient area in the scene (Itti, 1998), (Lin, 2014) where the classification process is isolated to a “Focus of attention” analogously to human vision activities (Hwang, 2012). Quattoni and Torbralba (Quattoni, 2009) have proposed a model of indoor scene classification where a comparison between scenes is made using a set of ROI to find the right scene category of the given image. Swadzba (Swadzba, 2010) proposed an indoor scene classification using a 3D approach mixed with Gist scene features, while (Torralba, 2003) recorded better results using Gist features in outdoor scene classification.

We can find in the literature several methods of scene classification using low-level approaches (Li, 2010) (Grauman, 2005) (Zuo, 2014). Even if (Zuo, 2014) was able to get quality results by adopting an approach that shares discriminative feature between the different scene categories, however, based on (Lazebnik, 2006) (Oliva, 2001) (Szummer, 1998), scene classification depending on low-level approaches works poorly. In contrast, high-level approaches of scene classification were developed where a scene is represented with semantic high-level information (Oliva, 2001) (Li, 2010) such as objects, actions, spatial information etc. The main idea consists on associating similarities between scenes containing same semantic properties. In the same perspective, (Li, 2011) (Pandey, 2011) proposed a deformable part-based models (DPM’s) using SVM’s as training models. The originality of (Li, 2011) (Pandey, 2011) is the introduction of an open-ended learning of latent structures for scene classification problems. (Zhu, 2010) Proposed an SVM classification model using maximization likelihood and margin, this approach is made possible by the fact that the optimization problem was efficiently solved. (Wu, 2011) Introduced a new visual descriptor for recognizing scene categories based on a holistic representation and has a strong overview for category recognition. It is mainly based on encoding the structural properties within an image and suppresses detailed textural information. Representing an image as a bag of objects has recently demonstrated impressive results (Lazebnik, 2006) (Herranz, 2016) (Nanni, 2013) (Zitnick, 2016). Song et al (Song, 2017) (Song, 2016) explored the path of scene classification using conventional neural networks (CNNs) exploring the way to combine effectively scene centric and object centric knowledge into a CNN architecture. Scene classification state of the art based on CNNs becomes very successful (Herranz, 2016) principally due to the impressive obtained results on the imagenet 2012 (Krizhevsky, 2012). However, CNNs are known for two main inconveniences: -The huge amount of data needed for the training part;-The high computational cost. In the same perspective of high-level scene classification, (Oliva, 2002) and (Oliva, 2001) introduced a spatial envelope for scene classification purposes providing a meaningful description of the real-world properties. The proposed characteristics of the spatial envelope are size, perspectives, mean depth and the nature of the general contents. On the other hand, Biederman et al (Biederman, 1973) assume that relations between an object and its environment can be reduced to five classes in order to characterize the organization of objects into real-world scenes. These classes have the ability to reduce the anomalies that can occur in scene classification problem. Further investigations have been introduced later on by (Sadeghi, 2011) integrating other classes of relationship. Fuzzy logic has also been widely used for scene classification (Song, 2016) (Elbaşi, 2013). Baiget et *al* (Baiget, 2007) were one of the first who computerized the geometrical construction of scenes studying human behavior, and the learning was done using a derivation of fuzzy logic called FMTHL (fuzzy metric temporal horn logic). Following the same idea, Zitnick et *al* (Zitnick, 2016) adopted a statistic approach to extract semantic information and identify the scene categories. Their approach and results are influenced by the assumption that abstract images can accurately represent real world scenes. While all the approaches reviewed in the literature differ in many features, they share the same aspect of using a learning part known as background knowledge to assist the identification of the scene category.

3. HIDDEN MARKOV MODELS (HMMS)

3.1. Definition of HMMs

The hidden Markov model is a probabilistic signal processing approach that aims to extract the maximum likelihood model from a sequence of observable events (Ghahramani, 2001). It has been known to mathematicians since a long time but has only been applied recently on numerous modern applications such as speech recognition (Gales, 2008), (Gautam, 2017), synthesis (Reddy, 2017), machine translation (Vogel, 1996) (Sangeetha, 2017), handwriting (Khandelwal, 2017), activity recognition (Alp, 2017), sign language recognition (Ozawa, 2017) and many other areas of artificial intelligence and pattern recognition (Ghahramani, 2001).

In theory, the HMMs are presented as a finite number N of states and M of observations symbols. Each state is assigned to a clock time t and possesses a measurable property. Every change of state is based on a transition probability conditioned by the previous state. This condition is called the Markovian property (Ghahramani, 2001). After each transition made, an observation output symbol is yield based on an emission probability specific to the current state. There are thus N emission probabilities for each of the M observations. Formally, the HMMs are defined as follow (Ghahramani, 2001):

- T: Observation sequence length (total number of clock times t)
- N: Number of hidden states $\{S_1 \ldots S_n\}$
- M: Number of observations symbols $\{o_1 \ldots o_n\}$
- A: state transition probability $\{a_{ij}\}$ where $a_{ij} = P\ [q_{t+1} = S_i \mid q_t = S_j)]\ j, i$ in $[1, N]$
- B: observation emission probability $\{b_i(o_k)\}$ where $b_i(o_k) = P\ [\mathrm{o}_\mathrm{k}$ at $t[q_t = S_i]\ i$ in $[1, N]$, k in $[1, M]]$
- π: The initial state distribution$\{\pi_\mathrm{i}\}$ where $\pi_\mathrm{i} = P[q_1 = S_i]\ i$ in $[1, N]$

Having the appropriate values of *N, M, A, B* and π an observation sequence $O= \{o_1, o_2, o_{3\ldots}\ o_T\}$ is generated following Algorithm *1*:

Algorithm 1: **Native DMC construction**

```
Inputs: N; M; A; B; π; O
Output: DMC
1- Choose an initial state $q_1$ according to the initial state distribution π
2- Set t=1.
3- Choose $o_t$ according to $B_i(o_t)$ the symbol probability distribution in state t
4- Choose t+1 according to $a_{i,i+1}$. The state transition probability distribution for state t;
5- Set t=t+1
6- If t <T go to 3 else terminate the process
```

A compact notation λ is used to represent in (1) for a given HMM.

$$\lambda = (A, B, \pi) \tag{1}$$

3.2. Inference of Hidden Markov Model and Dynamic Programming

Given a model $\lambda = (A, B, \pi)$ and an observation sequence $O = \{o_1, o_2 \ldots o_n\}$ the most basic approach to estimate the probability of *O* knowing λ i.e. *P(O|λ)* is by computing the probabilities of all possible sequences of hidden states having a length of *T* (*T = Card(O)*) that are eligible to emit *O*. The probability of such a sequence can be computed as follow: First we compute the probability of a fixed set of hidden states *I* knowing a model λ using (2) is made.

$$P\left(I|\right) = \grave{\mathrm{A}}_{1\mathrm{i}} a_{i1i2} a_{i2i3} a_{i3i4} \ldots a_{iT-1iT} \tag{2}$$

Next, we compute the probability of a given observation *O* knowing the hidden states *I* and the model λ using (3).

$$P\left(O|I,\right) = b_{i1}\left(O_1\right) b_{i2}\left(O_2\right) b_{i3}\left(O_3\right) \ldots b_{iT}\left(O_T\right) \tag{3}$$

The probability where *O* and *I* occur at the same time (i.e. is emitted by *I)* is simply the product of (3) and (4) as illustrated in (4).

$$P\left(O,I|\,T\right) = P\left(O|I,\right)(P\left(I|\right) \tag{4}$$

Finally, the probability of *O* knowing λ is obtain by summing the probability computed in (4) over all the possible hidden states *I* as represented in (5).

$$P\left(O|\right) = \sum_{i=1}^{T} P\left(O_i|I_{i},\right)*P(I_i|) \tag{5}$$

An explanation of (5) can be seen as the following: At time $t=1$, we are in the hidden state i_1 with an initial probability of π_i and emit the symbol o_1 with the probability $b_{i1}(o_1)$. At time $t=2$, we will make a transition to the hidden state i_2 with the transition probability a_{i1i2} (note that the transition can be reflexive) and emitting the symbol o_2 with probability $b_{i2}(o_2)$ and so on until $t=T$.

The reader can easily notice that computing the probability (5) requires a lot of computation time, exactly up to $\left(2T-1\right)N^T$ multiplications and N^T-1 additions. As a solution, another approach is proposed with dynamic programming.

In our case, we are more interested in finding the most likely sequence of hidden states that can emit a sequence of given observations. A dynamic programming algorithm for finding such a sequence is widely known as the Viterbi algorithm (Yamato, 1992). The key idea of the Viterbi algorithm is to keep only the max probability path of the hidden states -not all the paths- that can emit the current sequence of observation.

Algorithm 2: **Viterbi, optimized DMC construction**

```
Input: λ,O
output: DMC
1- Creates a path probability matrix VITEBI[N+2,T]
2- For each state I do
3- VITERBI[S,1]:=pi1*bi(O1)
4- BackPointer[s,1]:=0
5- End for
6- For each time step t from 2 to T do
7- For each state I
8- viterbi[S,t]:=MAX{s'=1:N} viterbi[s',t-1]*as's*bi(Ot)
9- Backbpointer[s,t]:=argmax{s'=1,N} viterbi[s',t-1]*as',s
10- End for
11- End for
12- ZT= argmax{s'=1,N} viterbi[s',T]*as',s
```

Given a model $\lambda = (A, B, \pi)$ a set of observation $O = \{o_1, o_2 ... o_T\}$. The Viterbi algorithm, presented in algorithm 2, introduces the dynamic programming method (Yamato, 1992).

The complexity of the Viterbi algorithm is on the order of $o(MN)$ where M is the number of observations symbols and N is the number of hidden states (Ghahramani, 2001). This complexity is significantly better than the previous method.

4. PROPOSED METHOD

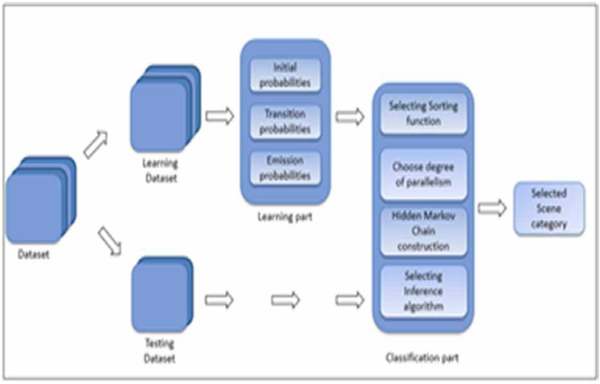
In this section, all contributions of this paper are explained. First, we propose a formal definition of the scene classification problem with objects as attributes, then, we continue by presenting the investigation made to solve the scene classification problem using an HMM architecture. The aim is to ensure a perfect analogy between their formal two definitions and explain how they can perfectly match. Then, weights measures functions that evaluates the saliency of objects and similarities between categories of scenes are introduced along with the sorting functions. Finally, the novel multiclass classification approach is presented and an inference algorithm is put into place in order to extract the most suitable scene category from the generated discrete Markov chains (DMC).

First, we formally define a scene classification problem based on objects (SC:O) as the following: Given a set of finite scene categories $SC = \left\{SC_1,\ SC_2,\ldots,SC_n\right\}$ and an input scene S containing a set of finite properties $P = \left\{P_1,\ P_2,\ \ldots,\ P_n\right\}$; we are not going to define rigorously what a property is, but we can simply say that it contains semantic information about S .e.g. objects, actions, sizes, relationships, etc. We wish to assign the most suitable scene category SC_i to S knowing P. For convenience, the compact notation (6) is used in the remaining of this paper.

$$\mu = \left(SC, P\right) \tag{6}$$

To illustrate the rest of contributions, Figure 1 summarizes the complete workflow of the proposed method. First, the dataset is divided into a learning part (80%) and a test part (20%) as recommended by (Quattoni, 2009). The learning dataset is used to construct the necessary entities to the proposed classification process. Initial, distribution, transition and emission probabilities are computed. This part of the workflow is called the *"learning part"*. The test dataset is used to certify the reliability of the proposed method's classification. A sorting function orders the objects of the input scene in such a way the most salient objects are put forward. Next, a degree of parallelism is defined to designate the number of DMCs constructed at the same time. The next step consists on starting the construction of the DMCs while going through the selected objects. Finally, an inference algorithm is developed in order to extract the most suitable scene category from discrete Markov chains. Each of these steps will be deeply explained in the upcoming subsections.

Figure 1. Workflow of the proposed classification process based on hidden Markov model



4.1. Analogy Between the Scene Classification Problem and the HMMs Architecture

In this subsection, we are going to demonstrate how the scene classification problem μ can be modeled using an HMM model λ. According to their formal definitions, let $=\left(A,B,\text{À}\right)$ and ¼$=\left(SC,P\right)$. In order to demonstrate the analogy between λ and ¼, each component of µ will get its correspondent in λ. After investigations and analyzes, Table 1 regroups the obtained correspondences.

Table 1. Analogy between the scene classification problem using objects and the Hidden Markov Model formal definitions.

Hidden Markov Model Architecture λ		High level object-based scene classification problem µ
T: Observation sequence length (total number of clock times t)	→	T': Cardinality of the set of properties P in a given the input scene S
N: set of hidden states {S_1, S_2 …S_n}	→	SC: Set of scene categories {SC_1, SC_2, … SC_n}
M: set of observation symbols {o_1, o_2, …, o_n}	→	P: Set of properties {p_1, p_2 …p_n}
Transition probabilities	→	Similarity between two SC as in (11)
Emission probabilities	→	Weight measure $\overset{\prime}{W}$ based on a given Scene Category SC_i as in (8)
Initial probabilities distribution π	→	Absolute weight measure $\tilde{W}$ independent from any scene category SC_i as in (10)

Based on the resulting correspondences in Table 1, we can easily see that the analogy between λ and µ is possible and state that, indeed, a problem µ can be represented by an HMM architecture λ. The same steps and algorithms defined in section 3 are used to construct the hidden Markov model for the purpose of µ problem.

4.2. Object's Weight Measure and Scene Categories Similarity

In this subsection, we introduce the concept of weight measure. Preferred over probabilities due to a lack of data, weight measure formulas fully meet the aspect of object saliency evaluation and similarity measure between two different scene categories. Moreover, experimentation shows that probabilities tend to extremely small values, which can twist the HMM process's calculations and results. The weight of an object can be extrapolated to the whole scene. Therefore, a similarity between two scenes categories can be computed. In the following, the two aspects of weight measure and scene categories similarities are introduced.

4.2.1. Object's Weight Measure Computation

In order to determine if an object is important in a scene category and has an impact on classification process, it is necessary to develop a weight function to quantify its saliency. We introduce the following definitions to clarify the content of upcoming equations.

Let,

- FO (SC_i, o_i): A function that returns the frequency of appearance (counting doubles) of object o_i in all the different dataset's scenes that are labeled as scene category SC_i

- *NO* (SC_i, o_i): A function that returns the number of times (without counting doubles) object o_i appears in all the dataset's scenes labeled with scene category SC_i
- FO_{all} (o_i): A function that returns the frequency of appearance (counting doubles) of object o_i in all the dataset
- NO_{all} (o_i): A function that returns the number of times (without counting doubles) object o_i appears in all the dataset.

Note: The *FO* (and FO_{all}) returns all the occurrences of appearance of the object o_i in current scene, conversely, the *NO* (and NO_{all}) returns the number of time an object o_i exists in the current scene. The correlation between *FO* (respectively FO_{all}) and *NO* (respectively NO_{all}) is defined in (7)

Let O_{all} be the set of all objects in the dataset

$$\forall\ o_i \in O_{all},\ FO\left(o_i\right) \geq NO\left(o_i\right) \tag{7}$$

After introducing basic concepts and definitions, now, we calculate the weight measure W' of a given object o_i knowing its scene category SC_i. Equation (8) demonstrates how the weight measure W' is calculated.

$$W(SC_i,o_i) = \begin{cases} 0 if NO(SC_i,o_i) = 0 \\ (\dfrac{NO(SC_i,o_i)^{FO(SC_i,o_i)}}{NO_{all(o_i)} * FO_{all(o_i)}}) else \end{cases} \tag{8}$$

We generalized (8) to get an equation independent of any scene category as presented in (9).

$$W(o_i) = \begin{cases} 0 if \dfrac{\max}{SC_i}(NO(SC_i,o_i)) = 0 \\ else \\ \dfrac{\dfrac{\max(NO(SC_i,o_i))^{(\max(FO(SC_i,o_i))_{sc_i}}}{SC_i}}{NO_{all}(SC_i,o_i) * FO_{all}(SC_i,o_i)} \\ where \\ FO(SC_i,o_i) : SC_i = SC_1,\ldots,SC_N \\ and \\ NO(SC_i,o_i) : SC_i = SC_1,\ldots,SC_N \end{cases} \tag{9}$$

Nevertheless, in order to generate appropriate calculations processes, a normalized version of (9) is developed in order to ensure that the weight measures values of objects o_i are held between 0 and 1. Equation (10) demonstrates the normalized version of (9).

$$W(O_i)\begin{cases} 0if\ \max(NO(SC_i,o_i)) = 0 \\ \hfill else \\ \dfrac{\dfrac{\max(NO(SC_i,o_i))^{\max(FO(SC_i,o_i))}_{SC_i}}{SC_i}}{NO_{all}(SC_i,o_i) * FO_{all}(SC_i,o_i)} \;/\; MaxValue \\ where \\ FO(SC_i,o_i)(SC_i = SC_1,\ldots,SC_N \\ and \\ NO(SC_i,o_i)SC_i = SC_1,\ldots,SC_N \end{cases} \tag{10}$$

Since (10) has no upper bound, its value expends as the occurrences of the object raises, we associate the value of the variable "*MaxValue*" according to the current dataset. Experimentation led to assume that the weight functions W and $\tilde{W}$ represent more faithfully the saliency of a given object o_i than simple probability measures. Nevertheless, the results are biased by the experimented datasets.

4.2.2. Scene Categories Similarity Computation

The aim of quantifying the similarity measure between two scene categories SC_i and SC_j is to grant the classification process the possibility to switch to the most suitable scene category in a given clock time "*t*". Equation (11) shows how to calculate the similarity measure α between two scene categories SC_i and SC_j (*i* can be equal to *j*).

Let SC_i and SC_j be two scene categories from the given dataset and $SC_iO_i=\{o_{i1}, oi_2...o_{ik}\}$ the set of objects belonging to all the scenes in the dataset labeled as SC_i and $SC_jO_j=\{o_{j1},o_{j2}..o_{jk'}\}$ be the set of objects belonging to all the scenes in the dataset labeled as SC_j.

We introduce the function α (SC_iO_i, SC_jO_j) which returns the similarity of SC_i toward SC_j as in equation (11)

$$\pm\left(SC_iO_i,\ SC_jO_j\right) = \frac{Card\left(SC_iO_i \cap SC_jO_j\right)}{Card\left(SC_iO_i\right)} \tag{11}$$

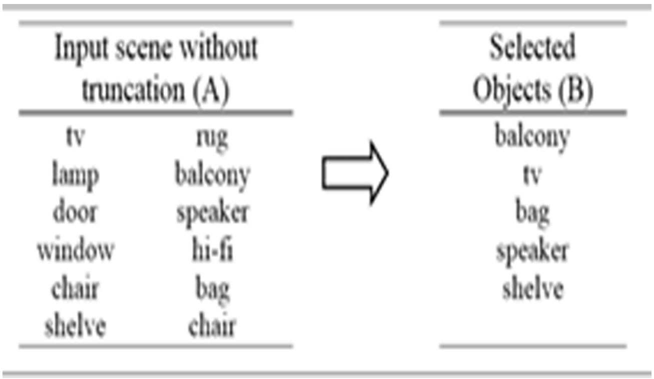
The similarity function is non-commutative: $\pm\left(SC_io_i, SC_jo_j\right) \neq \pm\left(SC_io_i, SC_jo_j\right)$.

4.3. Object's Sorting Functions

After assigning different kind of weight measures to given objects, sorting functions were developed to exploit the weight measure and sorts the input scene set of objects O for the discrete Markov chain construction. It is very important and crucial to have the most significant and finest sorting since the promoted scene categories depend deeply on the sorting functions and thus the scene classification process accuracy.

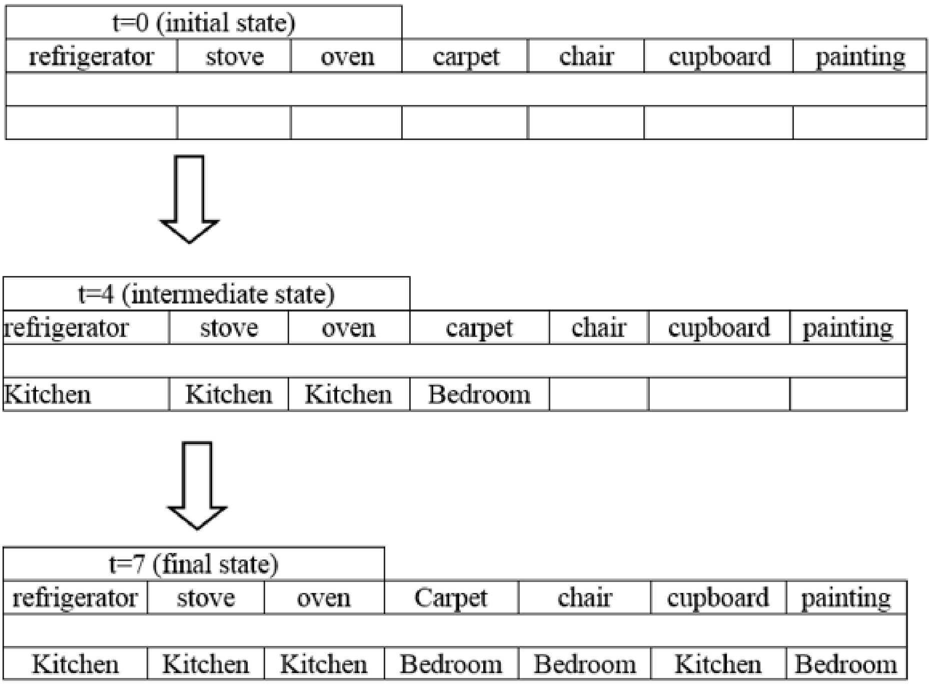
However, before starting, we need to filter out objects that are judged non-salient or considered as noise in the input scene. A truncation of insignificant (less salient) objects is made in order to reduce the length of the DMC as results a significant reduction of the combinatory computation. Also, to protect the classification process to get lost and diverge to insignificant scene categories. The truncation function relies exclusively on the weight measure presented in equation (10). Figure 2

Figure 2. Exemple of trancated objects from a real given input scene taken from MIT Indoor (Quattoni, 2009)



Input scene without truncation (A)		Selected Objects (B)
tv	rug	balcony
lamp	balcony	tv
door	speaker	bag
window	hi-fi	speaker
chair	bag	shelve
shelve	chair	

Figure 3.



t=0 (initial state)						
refrigerator	stove	oven	carpet	chair	cupboard	painting

t=4 (intermediate state)						
refrigerator	stove	oven	carpet	chair	cupboard	painting
Kitchen	Kitchen	Kitchen	Bedroom			

t=7 (final state)						
refrigerator	stove	oven	Carpet	chair	cupboard	painting
Kitchen	Kitchen	Kitchen	Bedroom	Bedroom	Kitchen	Bedroom

shows an example of a real input scene taken from MIT Indoor dataset (Quattoni, 2009) to illustrate the truncation process.

From Figure 2, we can see that some objects of the input scene in state (A) were removed in state (B) e.g.: “lamp”, “door”, “window”, etc. while the most salient objects according to the weight measure calculated in (10) remain present in state (B) such as: “balcony”, “tv”, “speaker”, etc. The purpose of the truncation is to take out the less salient objects existing in the input scene such as the HMM process will directly guide the DMC construction toward the most suitable scene category. After the truncation process, we proceed to a sorting phase that will determine how the selected objects will be presented in the HMM process for the DMC construction. We developed three different kinds of sorting functions. The descending and ascending sorting which are considered as static sorting and a dynamic sorting function.

4.3.1. Ascending and Descending Static Sorting

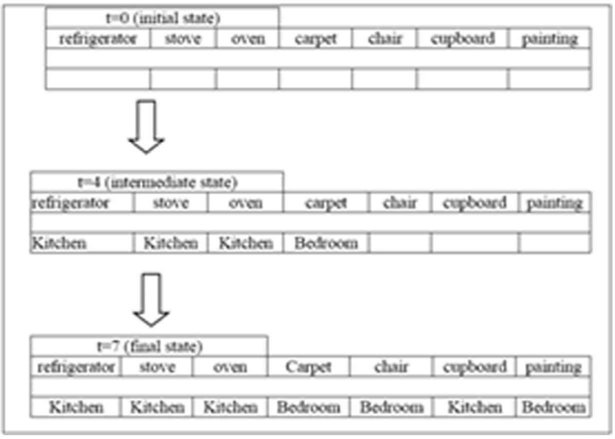
The ascending and descending sorting are considered as static sorting since they provide the order of the objects before the DMC construction starts. The descending and ascending sorting organize objects of the input scenes from the most salient to the less salient, respectively from the less salient to most salient, based on the weight measure introduced in (10) noted $\tilde{W}\left(O_i\right)$. The descending sorting approach assumes that the DMC construction should be provided immediately with the most salient objects in order to start with the right path and begins the emission process with the most representative scene category. Then, as the process goes on, it tries to construct the DMC using less salient objects while avoiding to diverge to other irrelevant scene categories.

Figure 3 illustrates how the DMC is constructed using a static descending sorting. The objects are taken from a real scene category labeled “Kitchen” belonging to the MIT Indoor dataset (Quattoni, 2009).

When the process starts (*t=0*), the scene categories generated has the correct value (Kitchen). However, as the process goes on (*t=4*), the DMC starts to diverge as less salient objects are handed. The final state (*t=7)* shows how the DMC looks like when all the scene categories are generated.

While the descending static sorting adopts the approach of “avoid divergence”, the ascending sorting approach, on the other hand, supports the idea of “convergence”. First are introduced general (less salient) objects and gradually with the progression of the DMC construction, the process converges to the most appropriate scene category as it is handed progressively more salient objects.

Figure 4. Example of a constructed Markov chain using the static Ascending sorting



t=0 (initial state)

refrigerator	stove	oven	carpet	chair	cupboard	painting

t=4 (intermediate state)

refrigerator	stove	oven	carpet	chair	cupboard	painting
Kitchen	Kitchen	Kitchen	Bedroom			

t=7 (final state)

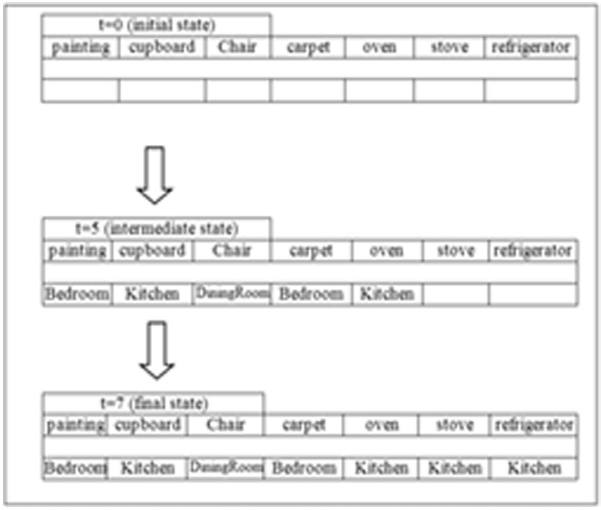
refrigerator	stove	oven	Carpet	chair	cupboard	painting
Kitchen	Kitchen	Kitchen	Bedroom	Bedroom	Kitchen	Bedroom

Figure 4 shows how the DMC is constructed when the objects are already on an ascending sorting as shown in the initial state (*t=0)*. When the process starts, the scene categories generated get erroneous values (Bedroom, DiningRoom). However, as the process goes on, the DMC starts to converge to the correct scene category (*t=2*, *t=5*) as more salient objects are handed. The final states (*t=5, t=6, t=7)* shows how the DMC was able to totally converge to the right scene category "Kitchen".

4.3.2. Dynamic Sorting

The dynamic sorting is revised and adapted to the current state of the DMC. As the DMC is constructed, it positions itself on a certain scene category at a specific time clock *t*. The next handed object for the DMC construction at time clock *t+1* is then chosen such as it suits the current scene category (at time clock *t*). Figure 5 shows how the dynamic sorting influences the construction of the DMC.

Figure 5. Example of a constructed Markov chain using the dynamic sorting



t=0 (initial state)

painting	cupboard	Chair	carpet	oven	stove	refrigerator

t=5 (intermediate state)

painting	cupboard	Chair	carpet	oven	stove	refrigerator
Bedroom	Kitchen	DiningRoom	Bedroom	Kitchen		

t=7 (final state)

painting	cupboard	Chair	carpet	oven	stove	refrigerator
Bedroom	Kitchen	DiningRoom	Bedroom	Kitchen	Kitchen	Kitchen

We notice from Figure 5 that the objects are not initially sorted in state *t=0* unlike in Figure 3 and Figure 4, the dynamic approach consists in the way objects are chosen alongside the construction of the DMC. The dynamic sorting chooses from the remaining objects of the input scene, the most suitable next object according to the current scene category.

The same object "chair" emitted the scene category "Bedroom" in Figure 4 and Figure 6 while the scene category "Dining Room" was emitted in Figure 5. This difference means that the emitted scene category is indeed influenced by the way objects are sorted. This ascertainment made us conclude that the order of the objects is very critical and crucial to the DMC construction and will directly affect the accuracy of the classification process. Once the DMC is constructed, inferences algorithms try to extract from it the most suitable scene category.

4.4. Inference Algorithm

The common way to handle multiclass classification problems using HMMs is by adopting the "one Vs all" method (Ghahramani, 2001), (Hinton, 2001). In this paper, we introduce a novel approach of multiclass classification that models the scene classification problem represented by the HMM where a single DMC is used to classify all scene categories SC_i. Inference algorithms are developed with the purpose of extracting the most suitable scene category among the set of scene categories emitted by the DMC. We proposed two inferences algorithms: “HMM inference” and “Frequency inference” both explained in the following.

4.4.1. HMM Inference Algorithm

The first proposed method to extract the most suitable scene category SC_i among the different ones emitted in the DMC is by using the associated weight measures calculated when the DMC was constructed. The DMC construction is based on a weight measure in order to emit a certain scene category SC_i. The HMM inference algorithm use those weights measures to extract the most suitable scene category from the DMC. Algorithm 3 shows how the “HMM inference” method is executed.

Algorithm 3 **HMM inference**

```
Input: discrete Markov chain DMC
Output: Chosen scene category: S
1- Extract the priorities between scene categories existing in the DMC. (first to appear gets the highest priority)
2- Extract all the emitted scene categories SC in the DMC
3- Assign to each scene category a weight equivalent to the sum of all the associated probabilities
4- Return the scene category getting the higher weight. If Two (or more) scene categories get the same weight, split the conflict with the priority calculated in step 1
```

However, this approach can be seen as inconsistent since the multiplication of weight measures at best remain the same (1x1) but in all the other cases decrease. This approach prioritizes the scene categories appearing first in the DMC which is not equitable. This inequality led us develop another inference algorithm that provides equality between the emitted scene categories called “Frequency inference”.

4.4.2. Frequency Inference Algorithm

The developed “frequency inference” method extracts the most suitable scene category based on a simple and equitable approach. The method counts the frequency of appearances of each scene category in the DMC, if two scene categories get the same frequency of appearance, the priority goes to the scene category appearing first in the DMC in the case where static descending or dynamic sorting were used. However, if ascending sorting was used, the priority goes to the last one. Algorithm 4 shows how the method is executed

Algorithm 4 **Frequency inference**

```
Input: discrete Markov chain DMC
Output: Chosen scene category: S
1- Extract the priorities between scene categories existing in the DMC according to the sorting algorithm chosen. (first to appear gets the highest priority)
2- Count the number of occurrences of each scene category mentioned in the DMC
3- If Two (or more) scene categories get the same number of occurrence in the DMC, split the conflict with the priority calculated in step 1
```

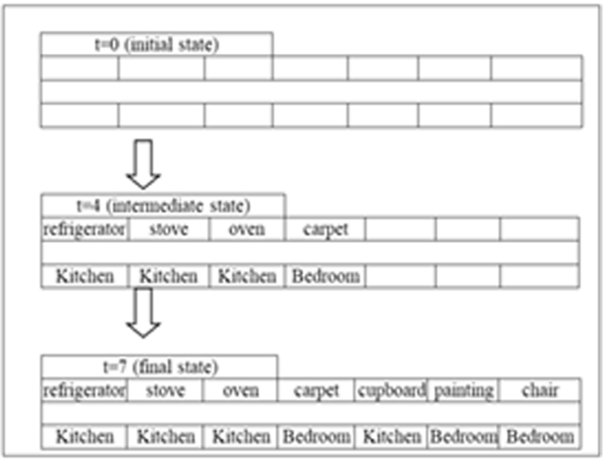
4.5. Degree of Parallelism

The construction of the DMC is based on the emission of the most suitable scene category knowing the given object at a particular time clock *t* and the previous state of the DMC at time clock *t-1*. This approach can be generalized by constructing at the same time more than one DMC where the second constructed DMC returns the second most suitable scene category and so on. This approach can be seen as a parallelism in the construction of DMCs. The benefit of constructing more than one DMC at the same time resides in a creation of a larger spectrum of scene categories providing more options to the inferences algorithms, which positively influences the accuracy of the classification process and accentuates the difference between correct and incorrect scene categories. Algorithm 5 shows how the global process of constructing several DMC at the same time is executed.

Algorithm 5 **Parallelism degree**

```
Input: set of input scene's objects: O, degree of parallelism: dp
Output: HMC with parallelism degree
1- Sort SC according to one of the sorting Algorithms
2- Initialize a set of SelectedScenes to NULL having size of dp
3- Initialize a set of Previous Scene to NULL having size of dp
4- For i=1 to size of O
5- For j=1 todp
6- following the HMM Algorithm, knowing $O_i$and the $j^{th}$ value of PreviousScenes, select the most suitable Scene Category SC that does not exist in Selected Scenes
7- Add SC to Selected Scenes
8- Add SC to the $j^{th}$ row of the DMC
9- End For
10- Previous Scene = Selected Scenes
11- Selected Scenes = NULL
12- End For
13- Return DMC
```

Figure 6. Constructed discrete Markov chain with a parallelism degree set to 3



t=0 (initial state)

t=4 (intermediate state)

refrigerator	stove	oven	carpet			
Kitchen	Kitchen	Kitchen	Bedroom			

t=7 (final state)

refrigerator	stove	oven	carpet	cupboard	painting	chair
Kitchen	Kitchen	Kitchen	Bedroom	Kitchen	Bedroom	Bedroom

Figure 6 presents a real example taken from a constructed DMC having a degree of parallelism equals to *3* which means three most suitable scene categories are taken into consideration when the HMM algorithms processes the current object.

We notice in Figure 6 some missing values in the construction of the DMC represented by the symbol "/", this is explained by the fact that no scene category can be reached having the combination of duplet (Refrigerator, Living-room). The rest of the line is then declared void.

A value of absorption set to *1.00e-10* is used in order to avoid the DMC process to get withdrawn when a weight measure of transition (or emission) is set to *null*.

After presenting and explaining all contributions of the proposed approach, the following section presents the tests conducted in order to determine the best tuning of the proposed approach and comparison with the state of the art's methods.

5. TEST AND RESULTS

In this section, we perform experiments of the proposed objects based scene classification (SC:O) method over several datasets: LabelMe(Russell, 2008), MIT INDOOR(Quattoni, 2009), SUN150(Xiao, 2010) and SUN397(Xiao, 2010). First, we evaluate the accuracy of the proposed method varying its own input parameters, then, we perform comparisons with the existing state of art of the scene classification methods which use the same dataset. In the following subsections, an overall summary of the used datasets is presented alongside some interesting statistics.

5.1. Datasets Presentation

Table 2. Summary and statistics extracted from the datasets used

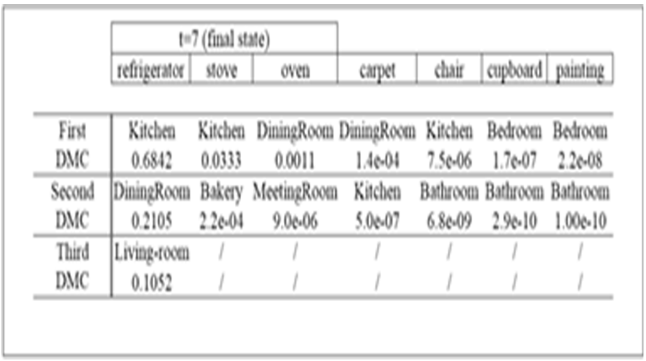
Dataset Statistics	LabelMe (Russell, 2008)	MIT INDOOR (Quattoni, 2009)	SUN150 (Xiao, 2010)	SUN397 (Xiao, 2010)
Number of scene category	16	67	150	395
Number of all scenes	1306	2742	12950	15723
Number of scene in learning part	1042	2140	10453	12811
Number of scene in test part	266	574	2499	2914
Number of objects	1620	2240	2294	2726
Smallest scene (number of objects)	1	1	2	1
Biggest scene (number of objects)	288	111	269	291
Average scene (number of objects)	24.93	21.86	28.1510	26.8084

We can see from Table 2 that presented datasets vary in all statistics attributes. This variation is important and highlights the strength of the proposed method when it comes to handle both small and large datasets. All the datasets were divided into two parts: The learning part (contains 80% of the initial dataset) and the test part (contains 20% of the initial dataset). We measure the good classification accuracy using a ratio between good classified scene instances and all scene instances provided by the test part. The statistic: "number of objects" demonstrates the diversity and the complexity of the proposed datasets where LabelMe (Russell, 2008) dataset containing 1620 different types of objects while SUN397 (Xiao, 2010) contains almost two times more with 2726 different types of objects. The statistics specific to the length of scenes shows the homogeneity of the different datasets, presenting an average size of almost identical scenes length. The scene categories in the presented datasets include a wild variety with all kind indoor, outdoor, small, large, public and private scenes. The SUN150 (Xiao, 2010) dataset is a subset of SUN397 (Xiao, 2010) that regroups scene categories that have at least more than 50 scenes (10 scenes for the test and 40 scenes for learning). This subset is more consistent and provides all the conditions to test the proposed method in a large and persistent dataset. In the upcoming subsections, several parameters of the proposed method are tested and put under extreme conditions in order to demonstrate its flexibility and robustness.

5.2. Varying the Objects Taken

In order to avoid a combinatory explanation and get an exploitable discrete Markov chain, the number of objects taken into consideration are varied in each input scene from *3,5,7* to *9* objects for all scenes categories. If a scene happens to have more than the number of allowed objects, a truncation is elaborated based on the weight measure $\tilde{W}$ calculated in (10). Figure 7 shows the different obtained results for all the datasets.

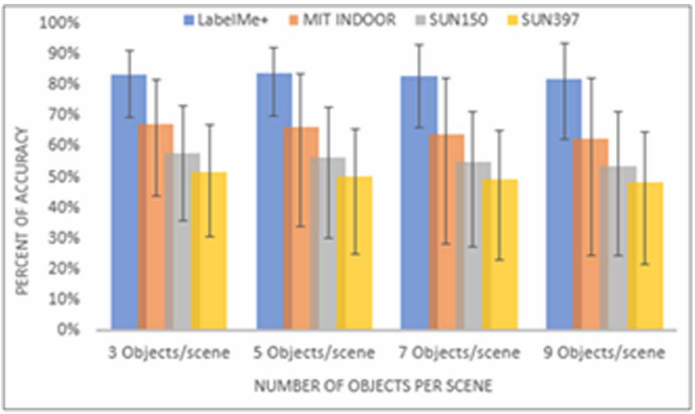
Figure 7. Proposed method accuracies while changing the number of objects taken into consideration for each scene



	t=7 (final state)						
	refrigerator	stove	oven	carpet	chair	cupboard	painting
First	Kitchen	Kitchen	DiningRoom	DiningRoom	Kitchen	Bedroom	Bedroom
DMC	0.6842	0.0333	0.0011	1.4e-04	7.5e-06	1.7e-07	2.2e-08
Second	DiningRoom	Bakery	MeetingRoom	Kitchen	Bathroom	Bathroom	Bathroom
DMC	0.2105	2.2e-04	9.0e-06	5.0e-07	6.8e-09	2.9e-10	1.00e-10
Third	Living-room	/	/	/	/	/	/
DMC	0.1052	/	/	/	/	/	/

Figure 7 shows a rise in accuracy in the classification process when objects are added (from *3* to *7* objects). In this part, the classification process is gaining practical information: above *7* objects, the accuracy drops and the classification process is misled for getting useless information (less salient

Figure 8. Proposed method accuracies while changing the number of scene categories suggested



objects). As result of the small size of LabelMe dataset, the variation of objects per scene does not affect the obtained results.

5.3. Varying the Number of Suggestions

In the next experiment, the same experience is provided while highlighting the number of suggestions made. Figure 8 illustrates the rate of well classified scenes when 1, 2 and 3 scene categories are suggested by the classifier

We notice from Figure 8 that the accuracy increases when the suggested scene categories do. This result claims that the DMC holds, for most of the time, the right scene category but the inference algorithm fails to extract it. This discordance is sanctioned with a gap of approximately 20% for all the datasets. Nevertheless, this gap is contained in just *3* scene categories and proves that the extraction made by the inference algorithm is still reliable. Next are tested the different inference algorithms.

5.4. Varying the Inference Algorithms

In order to see which inference algorithm performs the better on the provided datasets, we run experiments comparing the obtained accuracies. The results are plotted in Figure 9.

Figure 9. Accuracy of the proposed method by changing the inference algorithms

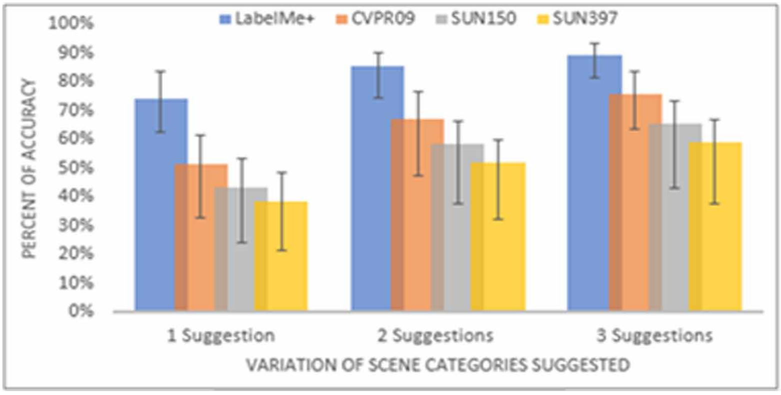


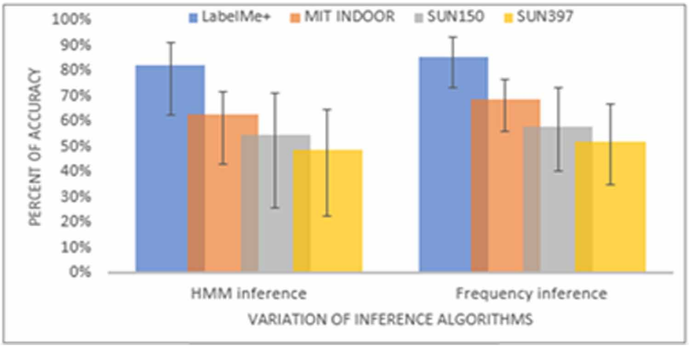
Figure 9 presents the accuracy of the two inference algorithms: "HMM inference" and "Frequency inference". In most of the time, the frequency inference algorithm returns slightly better results (around 3% better for all the datasets) than the HMM inference algorithm. This can be explained by the fact that the frequency inference algorithm depends on a very simple and intuitive approach while the HMM inference algorithm relies mainly on generated weight measures computed while the DMC is constructed. Moreover, when multiplying weight measures to combine scene categories appearances, the value drops while it needs to increase. That is why using weight measures based on probabilities are found to be less adequate for data fusion.

5.5. Varying the Sorting Algorithms

In the proposed method, we introduced three different manners of sorting the input scene objects. Two sorting algorithms are static and calculate the weight measure before starting the construction of the DMC while the dynamic algorithm calculates the weight measure taking into consideration the current state of the DMC. Figure 10 presents the obtained results on the different datasets

In all the datasets, Figure 10 reveals that the dynamic and descending static sorting functions return the best results with the dynamic sorting function getting a slight, yet noticeable, advantage.

Figure 10. Accuracy of the proposed method bychanging the sorting algorithms



On the other hand, the ascending static sorting function turns out to give very poor results dropping down to an average of *10.5%* compared to the dynamic sorting approach.

5.6. Varying the Parallelism Degree

In this subsection, Figure 11 shows the tests while changing the parallelism degree of the HMM process. This way of approach generates more scene categories since more than one DMC is constructed. Generating more scene categories provides useful information for the inferences algorithms.

Figure 11. Accuracy of the proposed method by changing the parallelism degree in the HMM process

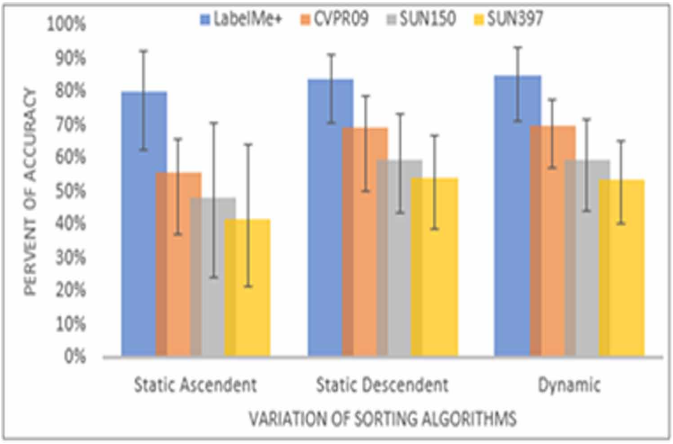


Figure 11 shows a change of parallelism degree from *1* to *3* DMCs constructed at the same time. We can notice that the accuracy of the classification process increases when the parallelism degree is incremented. The improvement is estimated at almost 2% for all the datasets. This can be explained by the fact that the inferences algorithms perform better when a considerable amount of scene categories are presented for selection. Moreover, the distinction between the correct and false scene categories is emphasized when more than one DMC point to the same results. A *2%* gain can be considered as a small improvement; nevertheless, this ability to construct several DMCs at the same time is not limited to *3* and can be expended in order to get better results. However, the cost of constructing several

DMCs at the same time comes with the expense of a significant time complexity. In the following subsection is presented the five best and worst results of the proposed method tested on the datasets.

5.7. Time Execution of the Proposed Method

In this subsection, Table 3, 4 and 5 present the average processing time necessary to the proposed method to classify *one* input scene. The results are obtained over the test part of the previously presented datasets (LabelMe (Russell, 2008), MIT Indoor (Quattoni, 2009), SUN150 (Xiao, 2010) and SUN397 (Xiao, 2010)). We choose to vary three different parameters (Number of objects taken; Number of suggestions and degree of parallelism) that affect the most the accuracy of the classification process (see subsections 5.6, 5.3 and 5.2). The aim of this experiment is to determine if these essential parameters are greedy in terms of time execution. The times are given in seconds and tests are run on a personal computer i3 processor and 4Gb of memory.

Table 3. Proposed method's average elapsed time to classify one scene while changing number of objects taken (second)

Datasets: Parameters	LabelMe (Russell, 2008)	MIT Indoor (Quattoni, 2009)	SUN150 (Xiao, 2010)	SUN397 (Xiao, 2010)
3 Objects taken	0,15 (s)	0,59 (s)	1,04 (s)	3,39 (s)
5 Objects taken	0,21 (s)	0,73 (s)	1,55 (s)	4,48 (s)
7 Objects taken	0,24 (s)	0,84 (s)	2,15 (s)	5,24 (s)
9 Objects taken	0,26 (s)	0,90 (s)	1,78 (s)	6,48 (s)

Table 3 presents the average time expressed in seconds to classify *one* input scene while varying the number of objects taken into consideration from *3, 5, 7* to *9*. The average times to classify *one* scene of small datasets LabelMe (Russell, 2008) and MIT indoor (Quattoni, 2009) are all less than *1*s. Also, on LabelMe (Russell, 2008) and MIT indoor (Quattoni, 2009), the increment of the "Object taken" parameter does not increase that much the computed times recorded. However, the large datasets SUN150 (Xiao, 2010) and SUN397 (Xiao, 2010) record more significant average time in order to process *one* input scene. This is due to the large amount of comparisons that the classifier must accomplish to find the most appropriate scene category. The increase in the number of objects taken into consideration does not help the combinatory approach. The needed time increases until the highest value recorded of *6.48s* for *9* objects taken on the SUN397 (Xiao, 2010) dataset.

Table 4. Proposed method's average elapsed time to classify one scene while changing number of results suggested (second)

Datasets: Parameters	LabelMe (Russell, 2008)	MIT INDOOR (Quattoni, 2009)	SUN150 (Xiao, 2010)	SUN397 (Xiao, 2010)
1 scene suggested	0,21 (s)	0,76 (s)	1,73 (s)	4,76 (s)
2 scenes suggested	0,21 (s)	0,76 (s)	1,54 (s)	4,73 (s)
3 scenes suggested	0,21 (s)	0,79 (s)	2,56 (s)	5,36 (s)

Table 4 introduces the average necessary time to classify *one* input scene while changing the number of scene suggested. On all the datasets, the average execution times recorded are almost constant (~*0.21s* for LabelMe (Russell, 2008), ~*0.76s* for MIT indoor (Quattoni, 2009), around *2s* for

SUN150(Xiao, 2010) and *5s* for SUN397(Xiao, 2010)). The increase in number of scene suggested slightly affects the average processing time for *one* input scene.

Table 5. Proposed method's average elapsed time to classify one scene while changing the degree of parallelism (second)

Datasets: Parameters	LabelMe (Russell, 2008)	MIT INDOOR (Quattoni, 2009)	SUN150 (Xiao, 2010)	SUN397 (Xiao, 2010)
Parallelism 1	0,21 (s)	0,73 (s)	1,52 (s)	4,64 (s)
Parallelism 2	0,17 (s)	0,60 (s)	1,28 (s)	3,97 (s)
Parallelism 3	0,27 (s)	0,98 (s)	2,05 (s)	6,19 (s)

Table 5 presents the average needed time to classify *one* input scene when increasing the parallelism degree. In all the datasets, the average time remains stable when raising the parallelism from *1* to *2*. This result is due to the lack of new scene categories proposed between the first and second best choice. However, a noticeable increase occurs when the parallelism degree is set to *3* (from *~0.20s* to *0.27s* on LabelMe (Russell, 2008), from *~0.65s* to *0.98s* on MIT indoor (Quattoni, 2009),from *~1.30s* to *~2s* on SUN150 (Xiao, 2010), from *~4.30s* to *6.19s* on SUN397(Xiao, 2010)). This gap is explained by the fact that the third best choice brings new scene categories that increases the combinatory and need more processing.

5.8. Discussion

In order to support the previous experiments, Table 6 introduces five best and worst results run with each dataset. The purpose of this experiment is to track the emerging parameters when considering extreme results (best and worst) of the proposed method. Table 6 is divided into four sections; each section represents one dataset. In turn, each section is divided into two parts; the first part presents the five best results whereas the second part represents the five worst results. The column of Table 6 shows the parameters of the proposed method: "Number of objects taken", "Scene categories suggested", "Degree of parallelism", "Type of sorting function" and "Inference algorithm" that were used in order to obtain the given classification accuracies.

We can see from the results of Table 6 some pattern homogeneity in the parameters that lead to the best results, respectively to the worst results in all datasets. We notice that the worst results, for most of the time, occur when the number of objects taken is high, the number of suggestions and the degree of parallelism are low, the ascending sorting is used while both, Frequency or HMM inferences algorithms appear. On the other hand, the best results are registered when the number of objects taken is relatively low, the number of suggestions and the degree of parallelism are both high, the dynamic or static descending sorting functions are used while still both frequency and HMM inference algorithms are used. However, we notice a fair advantage to the frequency inference algorithm (appearing in all top 3 for all datasets).

The generated results of previous experiments lead to state that some methods are always better than others ("static Ascending" sorting Vs "static Descending" and "Dynamic sorting"). Some parameters always perform better when set to specific values (degree of parallelism, number of suggested results). Whereas some other parameters remain unclear and appear in both good and bad results (inferences algorithms).

5.9. Comparison to the STATE of the ART'S METHODS

In this subsection, we compare the proposed method's best result with some reported methods in the literature that uses the same datasets. In this experiment, three datasets are tested: LabelMe (Russell,

Table 6. Presentation of the five best and worst results obtained by the proposed method over all the datasets (Fq): Frequency inference algorithm, (H): HMM inference algorithm, (Dyn): Dynamic sorting, (Des): Descending sorting, (Asc): Ascending Better scene in color.

Dataset Parameters	**LabelMe**									
	Best Five Results					**Worst Five Results**				
Number of objects taken	5	7	9	7	9	9	9	9	9	9
Scene categories suggested	3	3	3	3	3	1	1	1	1	1
Degree of parallelism	3	2	2	3	3	1	1	2	2	3
Type of sorting function	Asc	Dyn	Dyn	Dyn	Dyn	Asc	Asc	Asc	Asc	Asc
Inference algorithm	Fq	Fq	Fq	Fq	Fq	H	H	H	H	H
Obtained Accuracy[%]	**92**	**92**	**92.4**	**92.8**	**93.6**	**62.5**	**62.5**	**62.5**	**62.5**	**62.5**
Dataset Parameters	**MIT Indoor**									
	Best Five Results					Worst Five Results				
Number of objects taken	5	7	9	5	5	9	9	9	7	7
Scene categories suggested	3	3	3	3	3	1	1	1	1	1
Degree of parallelism	2	3	3	3	3	1	2	3	1	2
Type of sorting function	Dyn	Dyn	Dyn	Dyn	Des	Asc	Asc	Asc	Asc	Asc
Inference algorithm	Fq	H	H	Fq	Fq	H	H	H	H	H
Obtained Accuracy [%]	**82**	**82**	**82**	**82.8**	**83.4**	**24.5**	**25.3**	**25.3**	**28.7**	**29.6**
Dataset Parameters	**SUN150**									
	Best Five Results					Worst Five Results				
Number of objects taken	5	5	3	5	3	9	9	9	9	9
Scene categories suggested	3	3	3	3	3	1	1	1	1	1
Degree of parallelism	3	3	3	3	3	1	2	3	1	2
Type of sorting function	Des	Dyn	Dyn	Des	Des	Asc	Asc	Asc	Asc	Asc
Inference algorithm	H	Fq	Fq	Fq	Fq	H	H	H	H	H
Obtained Accuracy [%]	**71.5**	**71.5**	**71.8**	**72.5**	**73.1**	**24.5**	**24.5**	**24.5**	**25.7**	**27.1**
Dataset Parameters	**SUN397**									
	Best Five Results					Worst Five Results				
Number of objects taken	5	3	3	5	3	9	9	9	9	7
Scene categories suggested	3	3	3	3	3	1	1	1	1	1
Degree of parallelism	3	3	3	3	3	1	2	3	1	1
Type of sorting function	Des	Dyn	Des	Des	Des	Asc	Asc	Asc	Asc	Asc
Inference algorithm	H	H	Fq	Fq	Fq	H	H	H	H	H
Obtained Accuracy [%]	**65.1**	**65.4**	**65.5**	**65.8**	**67.1**	**21.5**	**21.5**	**21.5**	**22.5**	**23**

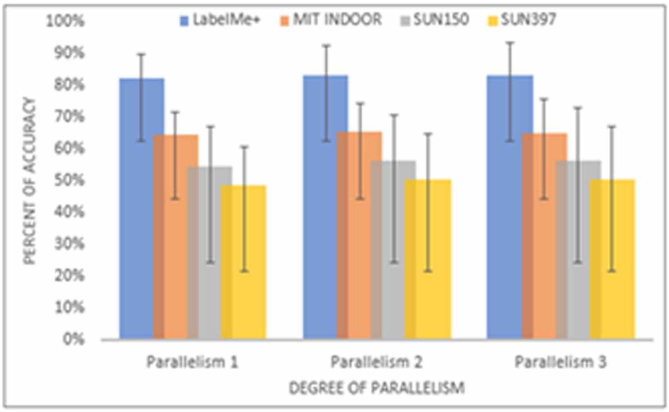
2008), MIT Indoor (Quattoni, 2009) and SUN397(Xiao, 2010). The comparisons are presented, for each dataset respectively, in Table 7 and Figure 12, Table 8 and Figure 13, Table 9 and Figure 14.

To perform the comparison with reported methods in the literature, we took the best-obtained results of the proposed method on the tested datasets. We can easily see from Table 7 and Figure 12, Table 8 and Figure 13, Table 9 and Figure 14 that the proposed method performs better in terms of scene classification accuracy compared to other existing methods in the literature. The best accuracy registered in our proposed method gets a rate of 93.56% (249 scenes) of well classified scenes on the

Table 7. Comparison between the proposed method's best results and reported methods in the literature using the LabelMe (Russell, 2008) dataset

Methods	Accuracy [%]
GIST (Csurka, 2004)	54.0
BoW (Csurka, 2004)	55.0
SPM (Csurka, 2004)	59.0
Object Bank-SVM (Li, 2010)	68.0
Object Bank-LR (Li, 2010)	76.0
Chong et *al* (Chong, 2009)	76.0
Method in (Maji, 2009)	83.0
Dixit et *al* (Dixit, 2011)	86.9
SPMSM (Kwitt, 2012)	87.5
Kernel Descriptor (Song, 2017)	87.3
BoW+SPM (Kwitt, 2012)	88.6
MFS (Song, 2017)	88.9
Object Bank (Li, 2014)	89.8
KCNF (Song, 2016)	89.8
Extended MFS (Song, 2017)	89.9
Proposed method (LabelMe)	93.56

Figure 12. Comparing the proposed method's best results and methods in the literature using the LabelMe (Russell, 2008)



LabelMe dataset (Russell, 2008), 83.40%(479 scenes) of well classified scenes on the MIT Indoor (Quattoni, 2009) dataset and 67.10% (1955 scenes) of well classified scenes on the SUN397(Xiao, 2010) dataset. We did not present any comparison using the SUN150 (Xiao, 2010) dataset with other method since the literature does not offer comparison using this particular subset of SUN397. We generated the SUN150 subset only to provide a consistent and large dataset to test our method on. As previously discussed in the related work section, the comparison with the state of the art's methods does

Table 8. Comparison between the proposed method's best results and reported methods in the literature on the MIT Indoor (Quattoni, 2009) dataset

Methods	Accuracy [%]
ROI+GIST (Quattoni, 2009)	26.50
MM-SCENE (Zhu, 2010)	28.00
DPM (Pandey, 2011)	30.40
CENTRIST (Wu, 2011)	36.90
Object Bank (Li, 2010)	37.60
DPM+GIST-Color (Pandey, 2011)	39.00
DPM+SP (Pandey, 2011)	40.50
DPM+SP+GIST-Color(Pandey, 2011)	43.10
Zuo et *al* (Zuo, 2014)	52.24
Method in (Donahue, 2014)	59.50
Juneja et *al* (Juneja, 2013)	63.18
Doersch et *al* (Doersch, 2013)	66.87
Method in (Liu, 2014)	68.20
Fc8-FV (Dixit, 2015)	72.86
MPP (Yoo, 2015)	75.67
Fc8-FV+fc7 (Dixit, 2015)	79.00
Proposed method (MIT INDOOR)	**83.40**

Table 9. Comparison between the proposed method's best results and reported methods in the literature using the SUN397(Xiao, 2010) dataset

Methods	Accuracy [%]
SUN(HOG) (Xiao, 2010)	27.2
SPMSM (Kwitt, 2012)	28.2
OTC (Grauman, 2005)	34.56
Meta-classes (Bergamo, 2014)	36.8
SUN(MKL) (Xiao, 2010)	38.00
Margolin et *al* (Xiao, 2010)	38.00
KCNF (Song, 2016)	40.8
DeCAF (Donahue, 2014)	40.94
Method in (Sánchez, 2013)	47.20
OTC+HOG2x2 (Krizhevsky, 2017)	49.60
fc7-VLAD (Weston, 1998)	51.98
fc7-FV (Weston, 1998)	53.0
fc8-FV (Dixit, 2015)	54.4
DSP (Bergamo, 2014)	59.78
fc8+fc7+places (Dixit, 2015)	61.72
Proposed method (SUN397)	**67.10**

Figure 13. Comparing the proposed method’s best result and methods in the literature using MIT INDOOR(Quattoni, 2009)

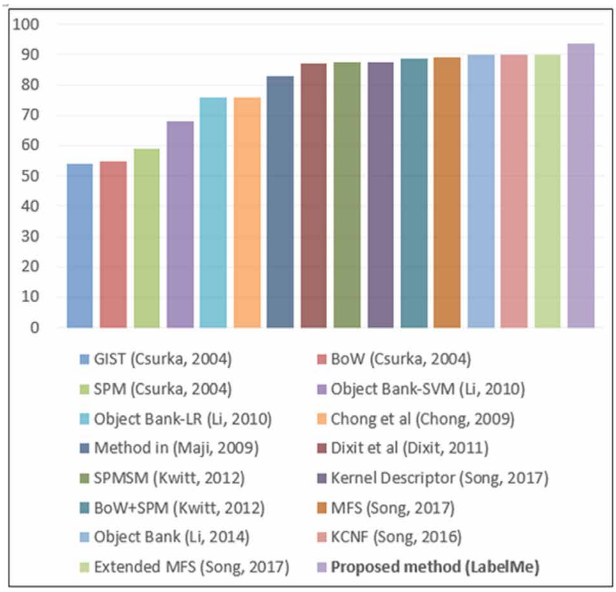
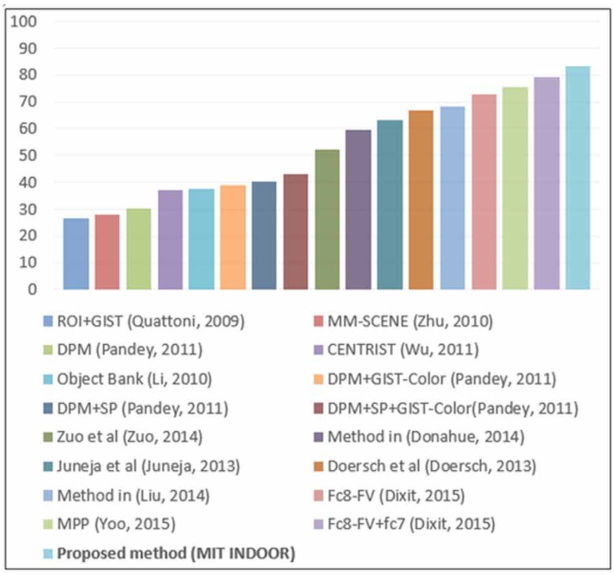


Figure 14. Comparison between the proposed method’s best results and methods in the literature using the SUN397 (Xiao, 2010)



not include the convolutional neural network (CNN) (Herranz, 2016) results since the developments and test environments are very different.

6. CONCLUSION AND PERSPECTIVES

This paper introduces a novel approach of high-level scene classification with objects as attributes using the hidden Markov model (HMM) architecture. The two main difficulties that encounter the scene classification problem are the enormous amount of scene categories and the lack of discriminant semantic properties that can define properly a scene. Although those inconveniences, we proposed a novel method based on a strong and reliable mathematic tool that operates as follow: After going through the learning process, which computes all entities of the hidden Markov model, the classification process starts by sorting the input objects, called observations, with the aim of putting the most salient ahead. Several sorting functions were developed and tested since this step is very critical to the whole classification process. The construction of the discrete Markov chain (DMC) starts by initializing the degree of parallelism, which determines the number of discrete Markov chains constructed. After that, the process generates scene categories, called hidden states, while going through the input scene's objects one by one. At the end, the discrete Markov chain contains a set of scene categories. The final step consists on extracting the most suitable scene category from the discrete Markov chain using developed inference algorithms.

The obtained results are promising and satisfying since very challenging datasets were used to run our tests. The obtained results are: LabelMe: 93.56%, MIT indoor: 83.40%, SUN150: 73.12% and SUN397: 67.10%. Some improvements can be possible by exploring more deeply the parallelism degree and see how far this parameter can improve the proposed method. The Markovian hypothesis, adopted in this approach, asserts that the future state can only be predicted knowing the current state, extending it and go further in the past can be relevant and improve the classification accuracy since previous steps hold important and useful information.

REFERENCES

Alp, E. C., & Keles, H. Y. (2017). Action recognition using MHI based Hu moments with HMMs. *International Conference on Smart Technologies IEEE EUROCON*, 212-216. doi:10.1109/EUROCON.2017.8011107

Arens, M., Ottlik, A., & Nagel, H. H. (2004). Using Behavioral Knowledge for Situated Prediction of Movements. *Proceedings of the 27th German Conference on Artiðcial Intelligence*, 141-155. doi:10.1007/978-3-540-30221-6_12

Baader, F., Horrocks, I., & Sattler, U. (2005). Description logics as ontology languages for the semantic web. Mechanizing Mathematical Reasoning, 228-248. doi:10.1007/978-3-540-32254-2_14

Baiget, P., Fernández, C., Roca, X., & Gonzalez, J. (2007). Automatic learning of conceptual knowledge in image sequences for human behavior interpretation. *Iberian Conference on Pattern Recognition and Image Analysis*, 507-514. doi:10.1007/978-3-540-72847-4_65

Bergamo, A., & Torresani, L. (2014). Classemes and other classifier-based features for efficient object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(10), 1988–2001. doi:10.1109/TPAMI.2014.2313111 PMID:26352630

Biederman, I., Glass, A. L., & Stacy, E. W. (1973). Searching for objects in real-world scenes. *Journal of Experimental Psychology*, *97*(1), 22–27. doi:10.1037/h0033776 PMID:4704195

Chong, W., Blei, D., & Li, F. F. (2009). Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition* (pp. 1903–1910). CVPR.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. *Workshop on statistical learning in computer vision (ECCV)*, 1-22.

Dixit, M., Chen, S., Gao, D., Rasiwasia, N., & Vasconcelos, N. (2015). Scene classification with semantic fisher vectors. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2974-2983.

Dixit, M., Rasiwasia, N., & Vasconcelos, N. (2011). Adapted gaussian mixtures for image classification. In *Computer Vision and Pattern Recognition* (pp. 937–943). CVPR.

Doersch, C., Gupta, A., & Efros, A. A. (2013). Mid-level visual element discovery as discriminative mode seeking. Advances in neural information processing systems, 494-502.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. *International conference on machine learning*, 647-655.

Elbaşi, E. (2013). Fuzzy logic-based scenario recognition from video sequences. *Journal of Applied Research and Technology*, *11*(5), 702–707. doi:10.1016/S1665-6423(13)71578-5

Gales, M., & Young, S. (2008). The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing, 1*(3), 195-304.

Gautam, P., & Soni, S. (2017). Efficient Speech Recognition with Hidden Markov Models. *International Journal of Advanced Research in Computer Science*, *8*(5).

Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *15*(01), 9–42. doi:10.1142/S0218001401000836

Ghomsheh, A. N., & Talebpour, A. (2012). A new method for indoor-outdoor image classification using color correlated temperature. *International Journal of Image Processing*, *6*(3), 167–181.

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. *The meeting of the ICCV Tenth IEEE International Conference on, 2*, 1458-1465.

Herranz, L., Jiang, S., & Li, X. (2016). Scene recognition with CNNs: objects, scales and dataset bias. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 571-579. doi:10.1109/CVPR.2016.68

Hinton, G. E., Brown, A. D., & London, Q. S. (2001). Training many small hidden markov models. *Proc. of the Workshop on Innovation in Speech Processing*.

Hummel, B., Thiemann, W., & Lulcheva, I. (2007). *Description logic for vision-based intersection understanding. In Proc. Cognitive Systems with Interactive Sensors*. COGIS.

Hwang, S. J., & Grauman, K. (2012). Learning the relative importance of objects from tagged images for retrieval and cross-modal search. *International Journal of Computer Vision*, *100*(2), 134–15. doi:10.1007/s11263-011-0494-3

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259. doi:10.1109/34.730558

Juneja, M., Vedaldi, A., Jawahar, C. V., & Zisserman, A. (2013). Blocks that shout: Distinctive parts for scene classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 923-930. doi:10.1109/CVPR.2013.124

Khandelwal, H., Gupta, S., & Jain, A. K. (2017). Review of Offline Handwriting Recognition Techniques in the fields of HCR and OCR. *International Journal of Computer Trends and Technology*, *47*(3), 161–164. doi:10.14445/22312803/IJCTT-V47P123

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 1097-1105.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. doi:10.1145/3065386

Kwitt, R., Vasconcelos, N., & Rasiwasia, N. (2012). Scene recognition on the semantic manifold. *European Conference on Computer Vision (ECCV)*, 359-372.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceeding CVPR '06 Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2169-2178. doi:10.1109/CVPR.2006.68

Li, L., & Sumanaphan, S. (2011). *Indoor Scene Recognition.* Unpublished. Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.374.9006&rep=rep1&type=pdf

Li, L. J., Su, H., Fei-Fei, L., & Xing, E. P. (2010). Object bank: A high-level image representation for scene classification and semantic feature sparsification. Advances in neural information processing systems, 1378-1386.

Li, L. J., Su, H., Lim, Y., & Fei-Fei, L. (2014). Object bank: An object-level image representation for high-level visual recognition. *International Journal of Computer Vision*, *107*(1), 20–39. doi:10.1007/s11263-013-0660-x

Li, L. J., Su, H., Lim, Y., & Li, F. (2010). Objects as Attributes for Scene Classification. ECCV Workshops, (1), 57-69

Lin, D., Lu, C., Liao, R., & Jia, J. (2014). Learning Important Spatial Pooling Regions for Scene Classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 3726-3733. doi:10.1109/CVPR.2014.476

Liu, L., Shen, C., Wang, L., Van Den Hengel, A., & Wang, C. (2014). Encoding high dimensional local features by sparse coding based fisher vectors. Advances in neural information processing systems, 1143-1151.

Maji, S., & Berg, A. C. (2009). Max-margin additive classifiers for detection. *IEEE 12th International Conference on Computer Vision (ICCV)*, 40-47.

Margolin, R., Zelnik-Manor, L., & Tal, A. (2014). Otc: A novel local descriptor for scene classification. *European Conference on Computer Vision*, 377-391. doi:10.1007/978-3-319-10584-0_25

Nanni, L., & Lumini, A. (2013). Heterogeneous bag-of-features for object/scene recognition. *Applied Soft Computing*, *13*(4), 2171–2178. doi:10.1016/j.asoc.2012.12.013

Neumann, B., & Möller, R. (2008). On scene interpretation with description logics. *Image and Vision Computing*, *26*(1), 82–101. doi:10.1016/j.imavis.2007.08.013

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145–175. doi:10.1023/A:1011139631724

Oliva, A., & Torralba, A. (2002). Scene-centered description from spatial envelope properties. Biologically motivated computer vision, 263-272. doi:10.1007/3-540-36181-2_26

Ozawa, T., Shibata, H., Nishimura, H., & Tanaka, H. (2017). Investigation of Feature Elements and Performance Improvement for Sign Language Recognition by Hidden Markov Model. *International Conference on Universal Access in Human-Computer Interaction*, 76-88 doi:10.1007/978-3-319-58703-5_6

Pandey, M., & Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with deformable part-based models. *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1307-1314. doi:10.1109/ICCV.2011.6126383

Pangercic, D., Tavcar, R., Tenorth, M., & Beetz, M. (2009). Visual scene detection and interpretation using encyclopedic knowledge and formal description logic. *Proceedings of the International Conference on Advanced Robotics (ICAR)*, *11*, 605-610.

Quattoni, A., & Torralba, A. (2009). Recognizing Indoor Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 413-420.

Reddy, M. K., & Rao, K. S. (2017). Robust pitch extraction method for the hmm-based speech synthesis system. *IEEE Signal Processing Letters*, *24*(8), 1133–1137. doi:10.1109/LSP.2017.2712646

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1), 157–173. doi:10.1007/s11263-007-0090-8

Sadeghi, M. A., & Farhadi, A. (2011). Recognition using visual phrases. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1745-1752.

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, *105*(3), 222–245. doi:10.1007/s11263-013-0636-x

Sangeetha, J., & Jothilakshmi, S. (2017). Speech translation system for english to dravidian languages. *Applied Intelligence*, *46*(3), 534–550. doi:10.1007/s10489-016-0846-3

Sikos, L. F. (2017). *Description logics in multimedia reasoning*. Springer. doi:10.1007/978-3-319-54066-5

Song, W., & Hagras, H. (2016). A Big-Bang Big-Crunch fuzzy logic based system for sports video scene classification. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 642-649. doi:10.1109/FUZZ-IEEE.2016.7737747

Song, X., Jiang, S., & Herranz, L. (2017). Multi-Scale Multi-Feature Context Modeling for Scene Recognition in the Semantic Manifold. *IEEE Transactions on Image Processing*, *26*(6), 2721–2735. doi:10.1109/TIP.2017.2686017 PMID:28333637

Song, X., Jiang, S., Herranz, L., Kong, Y., & Zheng, K. (2016). Category co-occurrence modeling for large scale scene recognition. *Pattern Recognition*, *59*, 98–111. doi:10.1016/j.patcog.2016.01.019

Swadzba, A., & Wachsmuth, S. (2010). Indoor scene classification using combined 3D and gist features. *Asian conference on computer vision*, 201-215.

Szummer, M., & Picard, R. W. (1998). Indoor-outdoor image classification. *IEEE International Workshop on Content-Based Access of Image and Video Database Proceedings*, 42-51. doi:10.1109/CAIVD.1998.646032

Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003). Context-based vision system for place and object recognition. *Ninth IEEE International Conference on computer vision*, 273-280. doi:10.1109/ICCV.2003.1238354

Vogel, S., Ney, H., & Tillmann, C. (1996). HMM-based word alignment in statistical translation. *Proceedings of the 16th conference on Computational linguistics*, 836-841 doi:10.3115/993268.993313

Wang, W., Alkhouli, T., Zhu, D., & Ney, H. (2017). Hybrid neural network alignment and lexicon model in direct hmm for statistical machine translation. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 125-131. doi:10.18653/v1/P17-2020

Weston, J., & Watkins, C. (1998). *Multi-class support vector machines*. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London.

Wu, J., & Rehg, J. M. (2011). CENTRIST: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1489–1501. doi:10.1109/TPAMI.2010.224 PMID:21173449

Xiao, J., Hayes, J., Ehringer, K., Olivia, A., & Torralba, A. (2010). SUN database: Largescale scene recognition from abbey to zoo. In *Computer vision and pattern recognition* (pp. 3485–3492). CVPR. doi:10.1109/CVPR.2010.5539970

Yamato, J., Ohya, J., & Ishii, K. (1992). Recognizing human action in time-sequential images using hidden markov model. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 379-385. doi:10.1109/CVPR.1992.223161

Yoo, D., Park, S., Lee, J. Y., & So Kweon, I. (2015). Multi-scale pyramid pooling for deep convolutional representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 71-80. doi:10.1109/CVPRW.2015.7301274

Zhu, J., Li, L. J., Fei-Fei, L., & Xing, E. P. (2010). Large margin learning of upstream scene understanding models. Advances in Neural Information Processing Systems, 2586-2594.

Zitnick, C. L., Vedantam, R., & Parikh, D. (2016). Adopting abstract images for semantic scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(4), 627–638. doi:10.1109/TPAMI.2014.2366143 PMID:26959669

Zuo, Z., Wang, G., Shuai, B., Zhao, L., Yang, Q., & Jiang, X. (2014). Learning discriminative and shareable features for scene classification. European Conference on Computer Vision, 552-568.

Lamine Benrais received his master degree at the University of Science and Technology Houari Boumedien in 2013. He has defended his PhD in the field of Artificial Vision and Artificial Intelligence in January 2021 in the same university. He is an author of numerous publications for conferences, proceedings and journals. His research interests include computer vision, artificial Intelligence, image understanding and scene interpretation.