Universitat de Barcelona
Universitat Rovira i Virgili
Universitat Politècnica de Catalunya
Màster en Intel·ligència Artificial

Tesi de Màster

Contextual Bag-of-Visual-Words and ECOC-Rank for Retrieval and Multi-class Object Recognition

Estudiant: Mehdi Mirza-Mohammadi
Director(s): Dr. Sergio Escalera and Dr. Petia Radeva
Ponent: Mehdi Mirza-Mohammadi

Data: 01/09/2009

Index

Abstract

Multi-class object categorization is an important line of research in Computer Vision and Pattern Recognition fields. An artificial intelligent system is able to interact with its environment if it is able to distinguish among a set of cases, instances, situations, objects, etc. The World is inherently multi-class, and thus, the efficiency of a system can be determined by its accuracy discriminating among a set of cases. A recently applied procedure in the literature is the Bag-Of-Visual-Words (BOVW). This methodology is based on the natural language processing theory, where a set of sentences are defined based on word frequencies. Analogy, in the pattern recognition domain, an object is described based on the frequency of its parts appearance. However, a general drawback of this method is that the dictionary construction does not take into account geometrical information about object parts. In order to include parts relations in the BOVW model, we propose the Contextual BOVW (C-BOVW), where the dictionary construction is guided by a geometricaly-based merging procedure. As a result, objects are described as sentences where geometrical information is implicitly considered.

In order to extend the proposed system to the multi-class case, we used the Error-Correcting Output Codes framework (ECOC). State-of-the-art multi-class techniques are frequently defined as an ensemble of binary classifiers. In this sense, the ECOC framework, based on error-correcting principles, showed to be a powerful tool, being able to classify a huge number of classes at the same time that corrects classification errors produced by the individual learners.

In our case, the C-BOVW sentences are learnt by means of an ECOC configuration, obtaining high discriminative power. Moreover, we used the ECOC outputs obtained by the new methodology to rank classes. In some situations, more than one label is required to work with multiple hypothesis and find similar cases, such as in the well-known retrieval problems. In this sense, we also included contextual and semantic information to modify the ECOC outputs and defined an ECOC-rank methodology. Altering the ECOC output values by means of the adjacency of classes based on features and classes relations based on ontologies, we also reported a significant improvement in class-retrieval problems.

Resum

La multi-classificació d'objectes és una important línia de investigació a les àrees de Visió Artificial i Reconeixement de Patrons. Un sistema intel·ligent és capaç de interactuar amb el seu entorn si pot distingir entre un conjunt de casos, instàncies, situacions, objectes, etc. El món és inherentment multi-classe. i per tant, la eficàcia d'un sistema pot estar determinada por la seva robustesa discriminant entre un conjunt de casos. Un mètode recent en aquest àmbit és el "Bag-of-Visual-Words" (BOVW). Aquesta metodologia es basa en el processament del llenguatge natural, on un conjunt de sentències es defineixen en funció de la freqüència de les seves paraules. De forma anàloga, en el domini del Reconeixement de Patrons, un objecte és descrit a partir de la freqüència d'aparició de les parts que el composen. No obstant, un dels principals problemes d'aquesta metodologia és que la construcció del diccionari no té en compte la informació geomètrica entre les parts dels objectes. Amb l'objectiu d'incloure informació relacional entre les parts dels objectes, en aquest treball proposem el BOVW contextual (C-BOVW), on la construcció del diccionari bé guiada per un procés de fusió. Com a resultat, els objectes són descrits com sentències a on la informació geomètrica està implícitament definida.

Amb l'objectiu d'estendre el sistema proposat al cas multi-classe, hem utilitzat el marc dels "Error-Correcting Output Codes" (ECOC). Les tècniques de multi-classificació de l'estat de l'art freqüentment estan definides a partir de l'assemblatge de classificadors binaris. En aquest sentit, el marc dels ECOC, basats en els principis de la correcció d'errors, han demostrar ser una potent eina, essent capaços de classificar grans conjunts de classes a la vegada que corregeixen errors de classificació produïts pels classificadors individuals.

En el nostre cas, les sentències C-BOVW són apreses per mitjà d'un disseny ECOC. obtenint un alt poder discriminador. A més, hem considerat les sortides dels ECOC amb l'objectiu d'ordenar i obtenir un ranking de les classes. En algunes situacions, més d'una etiqueta és necessària amb l'objectiu de treballar amb múltiples hipòtesis i trobar casos similars, com és el cas dels ben coneguts problemes de "retrieval". En aquest sentit, hem inclòs informació contextual i semàntica per modificar les sortides dels ECOC i definir la metodologia ECOC-Rank. Alterant les sortides dels ECOC a partir de l'adjacència entre classes basada en valors de característiques i les relacions entre classes per mitjà d'ontologies, també hem obtingut millores significatives en els problemes de "retrieval" de classes.

Resumen

La multi-clasificación de objetos en una importante línea de investigación en las áreas de Visión Artificial y Reconocimiento de Patrones. Un sistema inteligente es capaz de interactuar con su entorno en la medida que pueda distinguir entre un conjunto de casos, instancias, situaciones, objetos, etc. El mundo es inherentemente multi-clase, y por lo tanto, la eficacia de un sistema puede venir determinada por su robustez discriminando entre un conjunto de casos. Un método reciente en este ámbito ampliamente considerado en la literatura es el ”Bag-of-Visual-Words” (BOVW). Esta metodología se basa en el procesamiento del lenguaje natural, donde un conjunto de frases se definen en función de la frecuencia de aparición de sus palabras. De forma análoga, en el dominio del Reconocimiento de Patrones, un objeto es descrito a partir de la frecuencia de aparición de las partes que lo componen. No obstante, uno de los problemas principales de esta metodología es que la construcción del diccionario no tiene en cuenta la información geométrica entre las partes de los objetos. Con el objetivo de incluir información relacional entre las partes de los objetos, en este trabajo proponemos el BOVW contextual (C-BOVW), donde la construcción del diccionario viene determinada por un proceso guiado de mezcla. Como resultado, los objetos son descritos como sentencias donde la información geométrica está implícitamente definida.

Con el objetivo de extender el sistema propuesto al caso multi-clase, hemos usado el marco de los ”Error-Correcting Output Codes” (ECOC). Las técnicas de multi-clasificación del estado del arte frecuentemente vienen definidas a través del ensamblaje de clasificadores binarios. En este sentido, el marco de los ECOC, basados en los principios de corrección de errores, han demostrado ser una potente herramienta, siendo capaces de clasificar grandes conjuntos de clases a la vez que corrijen errores de clasificación producidos por los clasificadores individuales.

En nuestro caso, las sentencias C-BOVW son aprendidas a través de un diseño ECOC, obteniendo un alto poder de discriminación. Además, hemos considerado las salidas de los ECOC con el objetivo de ordenar y obtener un ranking de las clases. En algunas situaciones, más de una etiqueta es necesaria con el objetivo de trabajar con múltiples hipótesis y encontrar casos similares o candidatos, como en los bien conocidos problemas de ”retrieval”. En este sentido, hemos incluido información contextual y semántica para modificar las salidas de los ECOC y definir la metodología ECOC-Rank. Alterando las salidas de los ECOC a través de la adyacencia de clases basada en valores de características y las relaciones entre clases basándonos en ontologías, también hemos obtenido mejoras significativas en los problemas de ”retrieval” de clases.

Key words: Multi-class classification, Bag-Of-Visual Words, Error-Correcting Output Codes, Retrieval, Ranking.

Acknowledgements

I like to give my special thanks to my advisor Sergio Escalera, who taught me in the research and helped me through each step of my study. Also I like to thanks Oriol Pujol and Petia Padeva for their help and supervision through my study at UB and all other colleagues and friends there.

1 Introduction

We, humans from our first days of life learn to group our observations into meaningful and abstract categories. We can learn new categories from just few instances, and have the ability to recognize any new instance easily. As a matter of fact, categorization is the most fundamental cognitive ability, which helps us get through daily life.

For computers, images are just array of data, and machine has no knowledge about its semantic meaning. Categorization is one of the most challenging tasks in computer vision, which could have plenty of applications.

The task of object categorization in computer vision usually is split in two main stages: object description, where discriminative features are extracted from the object to represent, and object classification, where a set of extracted features are labeled as a particular object given the output of a trained classifier. There are plenty of methods on feature extraction, which vary based on the application, like localizing and classifying single objects, texture, or scenes. Recognition of real world scenes may be initiated from the global configuration, ignoring most of the details and object information (Bidderman, 1988; Potter, 1976) [1], meanwhile in object recognition the main attention is focused on the characteristic of individual instances. An example of an object categorization task is shown in Figure 1. In the image, the Wall-E robot of the Disney's film categorizes new objects into a set of categories.

A general tendency in object recognition to deal with the object description stage is to define a bottom-up procedure where initial features are obtained by means of region detection techniques. These techniques are based on determining relevant image key-points (i.e. using edge-based information [2]), and then defining a support region around the key-point (i.e. looking for extrema over scale-space [2]). Several alternatives for region detection have been proposed in the literature [2]. Once a set of regions is defined, it should be described using some kind of descriptor (i.e. SIFT descriptor [3]), and the region-descriptions are related in some way to define a model of the object of interest. Very few methods take into account relations of object parts when defining the feature space, such as in the Shape Context descriptor [4], and relations use to take place in the learning step, such as in graphical models as Conditional Random Fields or Hidden Markov Models [5,6].

1.1 Contextual Bag-of-Visual-Words motivation

Based on the previous tendency, a recent technique to model visual objects is by means of a Bag-Of-Visual-Words. The BOVW model is inspired by the text

Fig. 1. Scene of the Disney's Wall-e film. From top to down and left to right: The robot classifies objects into different categories. For a new object, the robot considers two classification options, fork and spoon. Finally, he classifies the new object in a new Fork-Spoon category.

classification problem using a Bag-Of-Words. In Bag-Of-Words model each document is represented by an unsorted set of contained words, regardless of the grammar and word order, and assumes order of words has no significance. Analogously, in object categorization problems, an image is represented by an unsorted set of discrete visual words, which are obtained by the object local descriptions.

Many promising results have been achieved with the BOVW systems in natural language processing, texture recognition, Hierarchical Bayesian models for documents, object classification [7], object retrieval problems [8,9], or natural scene categorization [10], just to mention a few. However, one of the main drawbacks of the BOVW model is that dictionary construction does not take into account the geometrical information among visual instances. Although this issue can be beneficial in natural language analysis, its adaptation to visual word description needs special attention. Note that based on the description strategy used to describe visual words, very close regions can have far descriptors in the feature space, being grouped as different visual words. This effect occurs for most of the state-of-the-art descriptors, even when coping with different invariance, and thus, a grouping based on spatial information of regions could be beneficial for the construction of the visual dictionary.

The first contribution of this work is a method for considering spatial information of visual words in the dictionary construction step, which we call Contextual Bag-Of-Visual-Words model. Objects interest regions are obtained by means of the Harris-Affine detector and then described using the SIFT descriptor. Afterward, a contextual-space and a feature-space are defined. The first space codifies the contextual properties of regions meanwhile the second space contains the region descriptions. A merging process is then used to fuse feature words based on their proximity in the contextual-space. Moreover, the new dictionary is learned using the Error Correcting Output Codes framework [11] in order to perform multi-class object categorization. We compared our approach to the standard BOVW design and validated over public multi-class categorization data sets, considering different state-of-the-art classifiers in the ECOC multi-classification procedure. Results show significant classification improvements when spatial information is taken into account in the dictionary construction step.

On the other hand, we extended our methodology to deal with another challenging computer vision task. In many situations only one predicted label of a multi-class problem is not enough, and we need to retrieve several possible and ordered case. This is the case of the retrieval problem.

1.2 ECOC-Rank motivation

Information Retrieval deals with uncertainty and vagueness in information systems (IR Specialist Group of German Informatics Society, 1991). This information could be in forms such as text, audio, or image. The science field which deals with information retrieval in images which is called Content-based image retrieval (CBIR). CBIR corresponds to any technology that for example helps to organize digital picture archives by their visual content. By this definition, anything ranging from an image similarity function to a robust image annotation engine falls under the purview of CBIR [12]. An example of a real image retrieval system is shown in Figure 2, where a set of Tour Eiffel samples are retrieved given an input sample using the Google retrieval engine.

In last decade, many research and work have been performed to describe color, shape, and texture features, without considering image semantics. Eakins [13] defines three levels of queries in CBIR.

Level 1: Retrieval by primitive features such as color, texture, shape, or the spatial location of image elements. A typical query is for example 'find pictures like this'.

Level 2: Retrieval of objects of a given type identified by derived features, considering some degree of logical inference. For example 'find a picture of a flower'.

Level 3: Retrieval by abstract attributes, involving a significant amount of high-level reasoning about the purpose of the objects or scenes depicted. This includes retrieval of named events, pictures with emotional or religious significance, etc. A query example could be 'find pictures of a joyful crowd'.

Levels 2 and 3 together are referred to as semantic image retrieval, and the gap between Levels 1 and 2 as the semantic gap [13]. Recently, some research has focused to fill this gap. Some state-of-the-art techniques use object ontology to define high-level concepts [14–17]. Other works use supervised or unsupervised learning methods to associate low-level features with query concepts [18–22], introduce Relevance feedback into retrieval loop for continuous learning of users intention [23–25], generate Semantic Template to support high-level image retrieval [26–28], or make use of both textual information obtained from the Web and visual content of images for Web image retrieval [29,27,30]. Many systems exploit one or more of the above techniques to implement high-level semantic-based image retrieval [23,24,27,30,20,29,27,31].

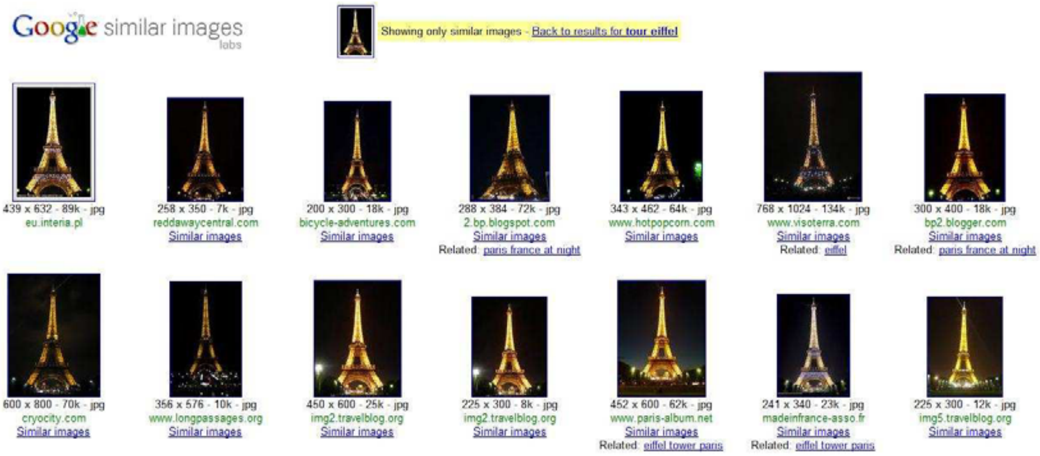


Fig. 2. Example of the retrieval of similar images for a Eiffel tower sample using the Google Similar Images engine (http://similar-images.googlelabs.com/). In this case, the retrieval process is based on a graph of relations designed by means of image features and web links.

Most of the works in the literature related to retrieval processes are based on the retrieving of samples from a same category. However, in our case we based on the class retrieval problem. Here, we define the class retrieval problem as the problem of retrieving classes similar in some way (i.e. given a semantic class distance) to the original class label of a given sample. Suppose we have an image from a cat animal category. In that case, we want to retrieve similar categories and not just samples from the same animal (i.e. tiger could be a possible solution). In order to deal with this problem, we focus on the output of multi-class classifiers to rank classes and then perform class retrieval.

Theoretical studies of machine learning have focused almost entirely on learning binary functions [32,33], but in many real-world learning tasks, such as in object classification, we need to deal with a discrete set of classes. Some of the state-of-the-art binary classification methods have been extended to deal with multi-class problems, such as decision trees, Adaboost, and Support Vector Machines. However, this kind of extensions is not always trivial. In such cases, the usual way to proceed is to reduce the complexity of the problem to a set of simpler classifiers and to combine them. In this sense, Error-Correcting output code (ECOC) is a general framework that combines binary problems to address the multi-class problem. The ECOC technique can be broken into two stages, encoding and decoding. Given a set of classes, the coding stage designs a codeword for each class based on different binary problems. The decoding stage makes a classification decision for a given test sample based on the value of the output code. Based on error-correcting properties, the ECOC framework has shown to be a powerful tool for combining binary classifiers, outperforming classical voting procedures.

Up to now, the ECOC framework has been just applied to the multi-class object recognition problem, where just one label was required. Then, based on the ECOC framework and our C-BOVW proposal, our second contribution consists on the extension of the ECOC framework to work on class retrieval problems. Altering the ECOC output values by means of the adjacency of classes based on features and classes relations based on ontology, we alter the ECOC output to improve class ranking. An adjacency matrix is computed by computing the mean distance among a set of class representant obtained by the k-means clustering, and changing the distance value to a measure of likelihood. The ontology matrix is computed by defining a new taxonomy distance using a semantic tree structure. The new ranking is used to look at the first retrieved classes to perform class retrieval based on semantic relations of classes. The results of the new ECOC-Rank approach show that performances improvements are obtained when including contextual and semantic information in the ranking class process. Moreover, using the proposed C-BOVW feature space also shown to outperform the classical BOVW approach in the class-retrieval problem.

Summarizing, the contributions of this work are:

a) Contextual Bag-of-Visual-Words: by defining a **feature and a contextual space** we merge words and include geometrical information in the design of the feature dictionary, improving posterior classification.

b) ECOC-Rank: We alter the output rank of the Error-Correcting Output Codes technique to improve results in **class retrieval** problem. We define an adjacency matrix by looking the feature space and a **ontology matrix** defining a tree of class **taxonomies**. Both matrices are used to alter the ECOC output rank and perform class retrieval.

The rest of this work is organized as follows: Section 2 presents our Contextual Bag-Of-Visual-Words. Section 3 overviews the ECOC framework used for multi-class object recognition and ranking. Section 4 describes the ECOC-Rank methodology and section 5 presents the results of the C-BOVW and ECOC-Rank methodologies over different real and public multi-class data sets. Finally, section 6 concludes the paper with a discussion of the presented approaches. At the end of the document, we show the publications regarding this work and summarize the nomenclature of the whole paper.

2 Contextual Bag-Of-Visual-Words

A simple approach for information retrieval in Natural-Language-Processing is to consider each document as a bag of words. It assumes that the order of words has no significance and considers a document as an unordered collection of words. Using such a representation, methods such as probabilistic latent semantic analysis (pLSA) [34] and latent dirichlet allocation (LDA) [35] are able to extract coherent topics within document collections in an unsupervised manner.

The same approach has been applied to the computer vision domain by treating images as a collection of regions, describing only their appearance and ignoring their spatial structure [36,37]. In the previous approach, for each image, a codebook is generated based on features. In this way, each patch in an image is mapped to a certain word in the codebook. Then, an histogram can be computed by counting the number of local feature vectors that fall within each word, and each image is represented as an histogram. This histogram is then used as a feature vector in the learning process [38].

One of the disadvantages of BOVW model is that it ignores spatial relation between words. Some works tried to deal with this problem. S. Lazebnik and her colleagues [39] split image into increasingly fine sub-regions and compute histograms of local features found inside each sub-region. J.C. Niebles and L.Fei-Fei [40] combine spatial information in a hierarchal way with features in the BOVW model.

In this section, we reformulate the BOVW model so that geometrical information can be taken into account in conjunction with the key-point descriptions in the dictionary construction step.

The algorithm is split into four main stages: contextual and feature space definition, merging, represent computation, and sentence construction.

Space definition: Given a set of samples for a n-multi-class problem, a set of regions of interest are computed and described for each sample in the training set. Then, k-means is applied over the descriptions and the spatial locations of each region to obtain a K-cluster feature-space and a K-cluster contextual-space, respectively. In our case, we use the Harris-affine region detector and SIFT descriptor. The x and y coordinates of each region normalized by the height and width of the image in conjunction with the ellipse parameters that define the region are considered to design the contextual-space.

Merging: Lets define a contextual-feature relational matrix M, where the position (i, j) of this matrix represents the percentage of points from the jth visual word of the feature-space that match with the points of the ith visual word of the contextual-space. Then, from each row of M, the two maximums are selected. These maximums correspond to the two words of the feature-space which share more percentage of elements for a same contextual word. In order to fuse relevant feature words, we select the contextual word which maximizes the minimums of all pairs of selected maximums. It prevents unbalanced feature words to be merged. Finally, the two feature words with maximum percentage in M for that contextual word (which have not been previously considered together) are labeled to be merged at the end of the procedure, and the process is iterated while an evaluation metric is satisfied or a maximum number of merging iterations is reached. Once the merging loop finishes, the pairs of feature words labeled during the previous strategy are merged and define the new visual words.

Representant computation: When the new C-BOVW dictionary is obtained, a set of represent for each final word is computed. In order to obtain a stratified number of represent related to the word densities, only one represent is assigned to the word with the minimum number of elements. Then, a proportional number of represent is computed for the rest of words by applying k-means and computing the mean vector for each of the word sub-clusters. With the final set of represent feature vectors, a normalized sentence of word occurrences is computed for each sample in the training set, defining its probability density function of C-BOVW visual words. The whole C-BOVW procedure is formally described in Algorithm 1. An example of a two-iteration C-BOVW definition for a motorbike sample is shown in Figure 3. At the top of the figure, the initial spaces are shown. In the second row, the shared elements from the two spaces which maximize the percentage of matches for a given contextual word are shown. The contextual-space just considers the x and y coordinates, and the 128 SIFT feature-space is projected into a two-dimensional feature-space using the two principal components. Note that the feature descriptions for the two considered words are very close in the feature space though they belong to different visual words before merging. On the right of the figure the new merged feature cluster is shown within a dashed rectangle. The same procedure is applied for the second iteration of the merging procedure in the bottom row of the figure.

Sentence construction: After the definition of the new dictionary, a new test sample can be simply described using the Bag-Of-Visual-Words without the need of including geometrical information since it is implicitly considered in the new visual words. The sentence for the new sample is then computed by matching its descriptors to the visual words with the nearest represent. Finally, the test sentence can be learned and classified using any kind of classification strategy. In the next chapter, we explain the framework used to learn the CBOVW feature space.

Algorithm 1 Contextual Bag-Of-Visual-Words algorithm.

```
Require: $D = \{(\mathbf{x}_1, l_1), .., (\mathbf{x}_m, l_m)\}$, where $\mathbf{x}_i$ is an object sample of label $l_i \in [1, .., n]$ for a $n$-class problem, $K$ clusters, and $I$ merging steps.
Ensure: Representant $R = \{(r_1, w_1), .., (r_v, w_b)\}$, where $r_v$ is a representant for word $w_i$, $i \in [1, ..b]$ for $b$ words. Sentences $S = \{(s_1, l_1), .., (s_m, l_m)\}$, where $s_i$ is the sentence of sample $\mathbf{x}_i$.
for each sample $\mathbf{x}_i \in D$ do
    Detect regions of interest for sample $\mathbf{x}_i$:
    $X_i = \{(x_1, y_1, \rho_1^1, \rho_1^2, \rho_1^3), .., (x_j, y_j, \rho_j^1, \rho_j^2, \rho_j^3)\}$, where $x$ and $y$ are spatial coordinates normalized by the height and width of the image, and $\rho^1$, $\rho^2$, and $\rho^3$ are ellipse parameters for affine region detectors.
    Compute region descriptors: $X_i^r = \{r_1, .., r_j\}$, where $r_j$ is the description of the $j$th detected region of sample $\mathbf{x}_i$.
end for
Define a contextual-space $C = \{(c_1, w_1^C), .., (c_q, w_q^C)\}$ using $k$-means to define $K$ contextual clusters, where $w_i^C$ is the $i$th word of the contextual-space.
Define a feature-space $F = \{(f_1, w_1^F), .., (f_q, w_q^F)\}$ using $k$-means to define $K$ feature clusters, where $w_i^F$ is the $i$th word of the feature-space.
Initialize a contextual-feature relational matrix $M$: $M(i, j) = 0$, $i, j \in [1, .., K]$
Initialize $W = \emptyset$ the list of feature words to be merged
for $I$ merging steps do
    update $M$ based on the contextual clusters and new feature clusters so that $M(i, j) = \frac{d(C,F,i,j)}{|w_j^F|}$, where $d(C, F, i, j)$ returns the number of points from contextual-space of word $w_i^C$ that belong to the feature-space $j$th word $w_j^F$, and $|w_j^F|$ is the number of regions of the $j$th feature word.
    Select the pair of positions with the maximum value for each row of $M$:
    $\max_{j,k} M(i, \_)$, $j \neq k$, $\forall i$, where $'\_'$ stands for all row positions.
    $W = W \cup (w_j^F, w_k^F)$: Select the contextual word $w_i^C$ and words $w_j^F$ and $w_k^F$ from the feature-space based on $\max_i \left(\min(M(i, j), M(i, k))\right), \forall j, k$
end for
for each pair $(w_j^F, w_k^F)$ in $W$ do
    update $F$ so that $w_j^F \leftarrow w_k^F$, and rename feature words so that $w_i^F, i \in [1, .., p]$ becomes $w_i^F, i \in [1, .., p - 1]$
end for
Compute representant $R = \{(r_1, w_1), .., (r_v, w_b)\}$ for the new $F$, where:
    $z_i = \text{round}\left(\frac{w_i}{\min |w_j| \forall j}\right)$
    is the number of representant for word $w_i$, computed using $z_i$-means, and $\{r_1, .., r_{z_i}\}$ representant are computed as the mean value for each sub-cluster of $w_i$, obtaining a stratified number of representant respect the words densities.
Compute sentences $S = \{(s_1, l_1), .., (s_m, l_m)\}$ for all training samples of all categories comparing with word representant of $R$.
```

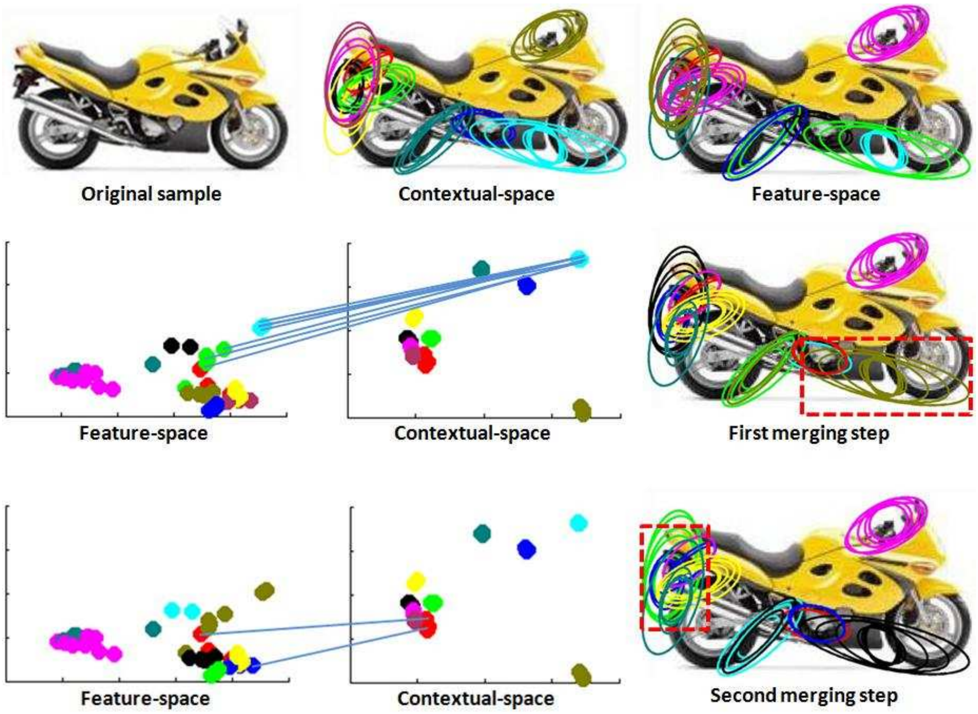


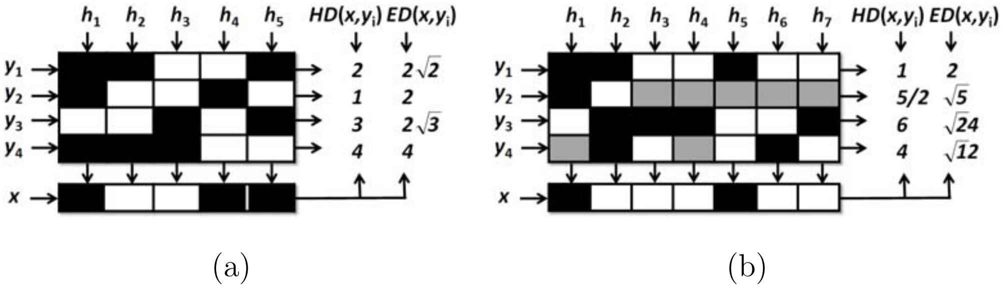
Fig. 3. Two iterations of C-BOVW algorithm over a motorbike sample.

3 Error-Correcting Output Codes

In this section, we review the Error-Correcting Output Codes framework, which is used to learn the previous C-BOVW sentences performing multi-class categorization as well as to rank the retrieval process that will be explained in the next section.

Given a set of N classes to be learnt in an ECOC framework, n different bi-partitions (groups of classes) are formed, and n binary problems (dichotomizers) over the partitions are trained. As a result, a codeword of length n is obtained for each class, where each position (bit) of the code corresponds to a response of a given dichotomizer (coded by +1 or -1 according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1,+1\}^{N \times n}$ in the binary case. In fig. 4(a) we show an example of a binary coding matrix M. The matrix is coded using 5 dichotomizers $\{h_1, ..., h_5\}$ for a 4-class problem $\{c_1, ..., c_4\}$ of respective codewords $\{y_1, ..., y_4\}$. The hypotheses are trained by considering the labeled training data samples $\{(\rho_1, l(\rho_1)), ..., (\rho_m, l(\rho_m))\}$ for a set of m data samples. The white regions of the coding matrix M are coded by +1 (considered as one class for its respective dichotomizer h_j), and the dark regions are coded by -1 (considered as the other one). For example, the first classifier is trained to discriminate c_3 against c_1, c_2, and c_4; the second one classifies c_2 and c_3 against c_1 and c_4, etc., as follows:

$$h_1(x) = \begin{cases} 1 & \text{if } x \in \{c_3\} \\ -1 & \text{if } x \in \{c_1, c_2, c_4\} \end{cases}, \dots, \quad h_5(x) = \begin{cases} 1 & \text{if } x \in \{c_2, c_4\} \\ -1 & \text{if } x \in \{c_1, c_3\} \end{cases} \tag{1}$$


h₁ h₂ h₃ h₄ h₅ HD(x,yᵢ) ED(x,yᵢ)
y₁ y₂ y₃ y₄ x
2 2√2
1 2
3 2√3
4 4
h₁ h₂ h₃ h₄ h₅ h₆ h₇ HD(x,yᵢ) ED(x,yᵢ)
y₁ y₂ y₃ y₄ x
1 2
5/2 √5
6 √24
4 √12

(a) (b)

Fig. 4. (a) Binary ECOC design for a 4-class problem. An input test codeword x is classified by class c_2 using the Hamming or the Euclidean Decoding. (b) Example of a ternary matrix M for a 4-class problem. A new test codeword x is classified by class c_1 using the Hamming and the Euclidean Decoding.

During the decoding process, applying the n binary classifiers, a code x is obtained for each data sample ρ in the test set. This code is compared to the base codewords ($y_i, i \in [1, .., N]$) of each class defined in the matrix M. And the data sample is assigned to the class with the *closest* codeword. In fig. 4(a), the new code x is compared to the class codewords $\{y_1, ..., y_4\}$ using the Hamming [41] and the Euclidean Decoding [42]. The test sample is classified by class c_2 in both cases, correcting one bit error.

In the ternary symbol-based ECOC, the coding matrix becomes $M \in \{-1, 0, +1\}^{N \times n}$. In this case, the symbol zero means that a particular class is not considered for a given classifier. A ternary coding design is shown in fig. 4(b). The matrix is coded using 7 dichotomizers $\{h_1, ..., h_7\}$ for a 4-class problem $\{c_1, ..., c_4\}$ of respective codewords $\{y_1, ..., y_4\}$. The white regions are coded by 1 (considered as one class by the respective dichotomizer h_j), the dark regions by -1 (considered as the other class), and the gray regions correspond to the zero symbol (classes that are not considered by the respective dichotomizer h_j). For example, the first classifier is trained to discriminate c_3 against c_1 and c_2 without taking into account class c_4, the second one classifies c_2 against c_1, c_3, and c_4, etc. In this case, the Hamming and Euclidean Decoding classify the test data sample by class c_1. Note that a test codeword can not contain the zero value since the output of each dichotomizer is $h_j \in \{-1, +1\}$.

The analysis of the ECOC error evolution has demonstrated that ECOC corrects errors caused by the bias and the variance of the learning algorithm [43] [1]. The variance reduction is to be expected, since ensemble techniques address this problem successfully and ECOC is a form of voting procedure. On the other hand, the bias reduction must be interpreted as a property of the decoding step. It follows that if a point ρ is misclassified by some of the learnt dichotomies, it can still be classified correctly after being decoded due to the correction ability of the ECOC algorithm. Non-local interaction between training examples leads to different bias errors. Initially, the experiments in [43] show the bias and variance error reduction for algorithms with *global* behavior (when the errors made at the output bits are not correlated). After that, new analysis also shows that ECOC can improve performance of *local* classifiers (e.g., the k-nearest neighbor, which yields correlated predictions across the output bits) by extending the original algorithm or selecting different features for each bit [44].

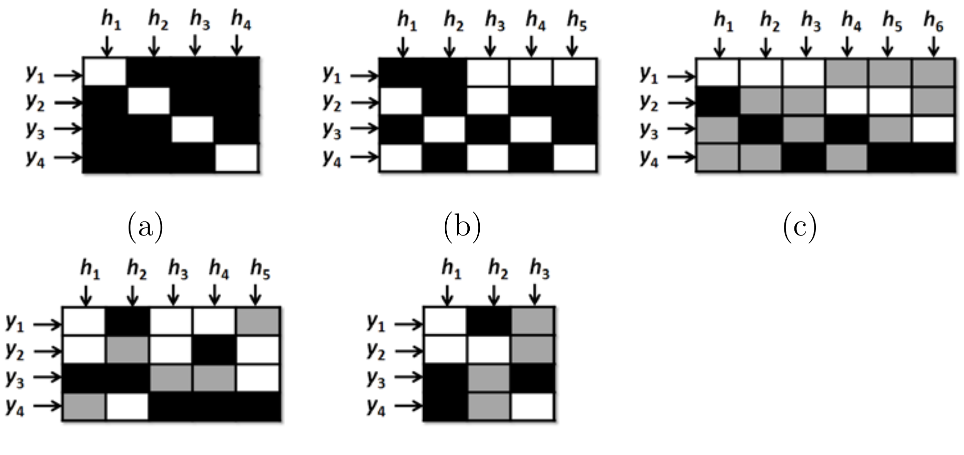
[1] The bias term describes the component of the error that results from systematic errors of the learning algorithm. The variance term describes the component of the error that results from random variation and noise in the training samples and random behavior of the learning algorithm. For more details, see [43].

3.1 Coding designs

In this section, we review the state-of-the-art on coding designs. We divide the designs based on their membership to the binary or the ternary ECOC frameworks.

3.1.1 Binary coding

The standard binary coding designs are the one-versus-all [41] strategy and the dense random strategy [42]. In one-versus-all, each dichotomizer is trained to distinguish one class from the rest of classes. Given N classes, this technique has a codeword length of N bits. An example of an one-versus-all ECOC design for a 4-class problem is shown in fig. 5(a). The dense random strategy generates a high number of random coding matrices M of length n, where the values $\{+1, -1\}$ have a certain probability to appear (usually $P(1) = P(-1) = 0.5$). Studies on the performance of the dense random strategy suggested a length of $n = 10 \log N$ [42]. For the set of generated dense random matrices, the optimal one should maximize the Hamming Decoding measure between rows and columns (also considering the opposites), taking into account that each column of the matrix M must contain the two different symbols $\{-1, +1\}$. An example of a dense random ECOC design for a 4-class problem and five dichotomizers is shown in fig. 5(b). The complete coding approach was also proposed in [42]. Nevertheless, it requires the complete set of classifiers to be measured $(2^{N-1} - 1)$, which usually is computationally unfeasible in practice.



(a) (b) (c)

(d) (e)

Fig. 5. Coding designs for a 4-class problem: (a) one-versus-all, (b) dense random, (c) one-versus-one, (d) sparse random, and (e) DECOC.

3.1.2 Ternary Coding

The standard ternary coding designs are the one-versus-one strategy [45] and the sparse random strategy [42]. The one-versus-one strategy considers all possible pairs of classes, thus, its codeword length is of $\frac{N(N-1)}{2}$. An example of an one-versus-one ECOC design for a 4-class problem is shown in fig. 5(c). The sparse random strategy is similar to the dense random design, but it includes the third symbol zero with another probability to appear, given by $P(0) = 1 - P(-1) - P(1)$. Studies suggested a sparse code length of $15 \log N$ [42]. An example of a sparse ECOC design for a 4-class problem and five dichotomizers is shown in fig. 5(d). In the ternary case, the complete coding approach can also be defined.

Due to the huge number of bits involved in the traditional coding strategies, new problem-dependent designs have been proposed [46][47][48]. The new techniques are based on exploiting the problem domain by selecting the representative binary problems that increase the generalization performance while keeping the code length small. The Discriminant ECOC (DECOC) of [48] is based on the embedding of discriminant tree structures derived from the problem domain. The binary trees are built by looking for the sub-sets of classes that maximizes the mutual information between the data and their respective class labels. As a result, the length of the codeword is only $(n-1)$. The algorithm is summarized in table 1. In fig. 6, a binary tree structure for an 8-class problem is shown. Each node of the tree splits a sub-set of classes. Each internal node is embedded in the ECOC matrix as a column, where the white regions correspond to the classes on the left sub-sets of the tree, the black regions to the classes on the right sub-sets of the tree, and the gray regions correspond to the non-considered classes (set to zero). Another example of a DECOC design for a 4-class problem obtained by embedding a balanced tree is shown in fig. 5(e). [2]

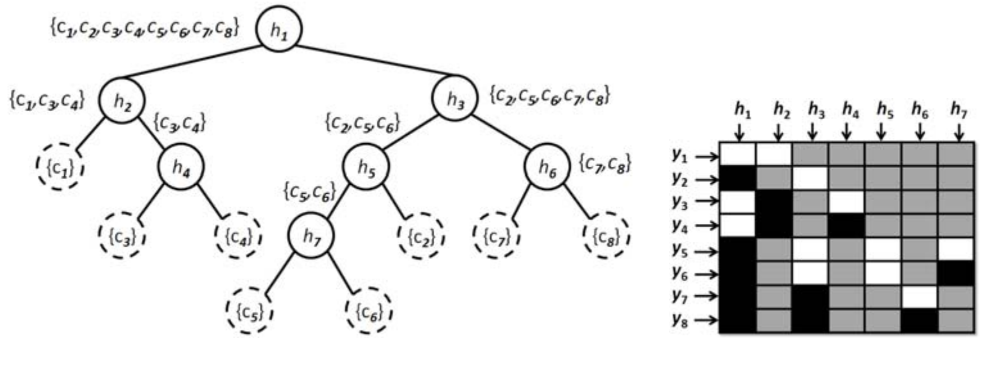


Fig. 6. Example of a binary tree structure and its DECOC codification.

[2] For further information about more recent coding designs the reader is referred to [49] and [50].

Table 1
DECOC algorithm.

```
**DECOC**: Create the Column Code Binary Tree as follows:

Initialize $L$ to $L_0 = \{\{c_1, .., c_N\}\}$

while $|L_k| > 0$
    1) Get $S_k : S_k \in L_k, k \in [0, N-2]$
    2) Find the optimal binary partition $BP(S_k)$ that maximizes the
       fast quadratic mutual information [48].

    3) Assign to the column $t$ of matrix $M$ the code obtained by the
       new partition $BP(S_k) = \{C_1, C_2\}$.

    4) Update the sub-sets of classes $L_k$ to be trained as follows:

        $L_k^{'} = L_k \backslash S_k$
        $L_{k+1} = L_k^{'} \cup C_i$ iff $|C_i| > 1, i \in [1, 2]$
```

3.2 Decoding designs

In this section, we review the state-of-the-art on decoding designs. The decoding strategies (independently of the rules they are based on) are divided depending if they were designed to deal with the binary or the ternary ECOC frameworks.

3.2.1 Binary decoding

The binary decoding designs most frequently applied are: Hamming Decoding [41], Inverse Hamming Decoding [51], and Euclidean Decoding [42].

- *Hamming Decoding*

The initial proposal to decode is the Hamming Decoding measure. This measure is defined as follows:

$$HD(x, y_i) = \sum_{j=1}^{n} (1 - sign(x^j y_i^j))/2 \tag{2}$$

This decoding strategy is based on the error correcting principles under the assumption that the learning task can be modeled as a communication problem, in which class information is transmitted over a channel, and two possible symbols can be found at each position of the sequence [11].

- *Inverse Hamming Decoding*

The Inverse Hamming Decoding [51] is defined as follows: let Δ be the matrix composed by the Hamming Decoding measures between the codewords of M. Each position of Δ is defined by $\Delta(i_1, i_2) = HD(y_{i_1}, y_{i_2})$. Δ can be inverted to find the vector containing the N individual class likelihood functions by means of:

$$IHD(x, y_i) = max(\Delta^{-1} D^T) \tag{3}$$

where the values of $\Delta^{-1} D^T$ can be seen as the proportionality of each class codeword in the test codeword, and D is the vector of Hamming Decoding values of the test codeword x for each of the base codewords y_i. The practical behavior of the IHD showed to be very close to the behavior of the HD strategy [41].

- *Euclidean Decoding*

Another well-known decoding strategy is the Euclidean Decoding. This measure is defined as follows:

$$ED(x, y_i) = \sqrt{\sum_{j=1}^{n} (x^j - y_i^j)^2} \tag{4}$$

3.2.2 Ternary decoding

Concerning the ternary decoding, the state-of-the-art strategies are: Loss-based Decoding [42], and the Probabilistic Decoding [52].

- *Loss-based Decoding*

The Loss-based Decoding strategy [42] chooses the label ℓ_i that is most consistent with the predictions f (where f is a real-valued function $f : \rho \rightarrow R$), in the sense that, if the data sample ρ was labeled ℓ_i, the total loss on example (ρ, ℓ_i) would be minimized over choices of $\ell_i \in \ell$, where ℓ is the complete set of labels. Formally, given a Loss-function model, the decoding measure is the total loss on a proposed data sample (ρ, ℓ_i):

$$LB(\rho, y_i) = \sum_{j=1}^{n} L(y_i^j f^j(\rho)) \tag{5}$$

where $y_i^j f^j(\rho)$ corresponds to the *margin* and L is a Loss-function that depends on the nature of the binary classifier. The two most common Loss-functions are $L(\theta) = -\theta$ (Linear Loss-based Decoding (LLB)) and $L(\theta) = e^{-\theta}$ (Exponential Loss-based Decoding (ELB)). The final decision is achieved by assigning a label to example ρ according to the class c_i which obtains the minimum score.

- *Probabilistic Decoding*

Recently, the authors of [52] proposed a probabilistic decoding strategy based on the continuous output of the classifier to deal with the ternary decoding. The decoding measure is given by:

$$PD(y_i, F) = -log\left(\prod_{j\in[1,\ldots,n]:M(i,j)\neq 0} P(x^j = M(i,j)|f^j) + K\right) \tag{6}$$

where K is a constant factor that collects the probability mass dispersed on the invalid codes, and the probability $P(x^j = M(i,j)|f^j)$ is estimated by means of:

$$P(x^j = y_i^j|f^j) = \frac{1}{1 + e^{y_i^j(\upsilon^j f^j + \omega^j)}} \tag{7}$$

where vectors υ and ω are obtained by solving an optimization problem [52]

- *Loss-Weighted Decoding*

The main objective of the Loss-Weighted decoding is to find a weighting matrix M_W that weights a loss function to adjust the decisions of the classifiers, either in the binary and in the ternary ECOC frameworks. To obtain the weighting matrix M_W, we assign to each position (i, j) of the matrix of hypothesis H a continuous value that corresponds to the accuracy of the dichotomy h_j classifying the samples of class i (8). We make H to have zero probability at those positions corresponding to unconsidered classes (9), since these positions do not have representative information. The next step is to normalize each row of the matrix H so that M_W can be considered as a discrete probability density function (10). This step is very important since we assume that the probability of considering each class for the final classification is the same (independently of number of zero symbols) in the case of not having *a priori* information ($P(c_1) = ... = P(c_{N_c})$). In fig. 7 a weighting matrix M_W for a 3-class problem with four hypothesis is estimated. Figure 7(a) shows the coding matrix M. The matrix H of fig. 7(b) represents the accuracy of the hypothesis classifying the instances of the training set. The normalization of H results in the weighting matrix M_W of fig. 7(c). [3]

[3] Note that the presented Weighting Matrix M_W can also be applied over any decoding strategy.

Given a coding matrix M,

1) Calculate the matrix of hypothesis H:

$$H(i,j) = \frac{1}{m_i} \sum_{k=1}^{m_i} \gamma(h_j(\wp_k^i), i, j) \tag{8}$$

$$\text{based on} \quad \gamma(x_j, i, j) = \begin{cases} 1, & \text{if} \quad x_j = M(i,j) \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

2) Normalize H so that $\sum_{j=1}^{n} M_W(i,j) = 1, \forall i = 1, ..., N_c$:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)},$$
$$\forall i \in [1, ..., N_c], \quad \forall j \in [1, ..., n]$$

Given a test input $\wp$, decode based on:

$$d(\wp, i) = \sum_{j=1}^{n} M_W(i,j) L(M(i,j) \cdot f(\wp, j)) \tag{10}$$

Table 2

Loss-Weighted algorithm.

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix} \quad H = \begin{bmatrix} 0.955 & 0.955 & 1.000 & 0.000 \\ 0.900 & 0.800 & 0.000 & 0.000 \\ 1.000 & 0.905 & 0.805 & 0.805 \end{bmatrix} \quad M_W = \begin{bmatrix} 0.328 & 0.328 & 0.344 & 0.000 \\ 0.529 & 0.471 & 0.000 & 0.000 \\ 0.285 & 0.257 & 0.229 & 0.229 \end{bmatrix}$$

(a) (b) (c)

Fig. 7. (a) Coding matrix M of four hypotheses for a 3-class problem. (b) Matrix H of hypothesis accuracy. (c) Weighting matrix M_W.

The Loss-weighted algorithm is shown in table 2. As commented before, the loss functions applied in equation (10) can be for example the linear or the exponential ones. The linear function is defined by $L(\theta) = \theta$, and the exponential loss function by $L(\theta) = e^{-\theta}$, where in our case θ corresponds to $M(i,j) \cdot f^j(\wp)$. Function $f^j(\wp)$ may return either the binary label or the confidence value of applying the j^{th} ECOC classifier to the sample $\wp$.[4]

[4] For further information about more recent decoding designs the reader is referred to [53].

4 ECOC-Rank

Retrieval systems retrieve huge amount of data for each query. Thus, sorting the results from most to less relevant cases is required. Based on the framework and application there exist different ways for ranking the results based on the associated criteria.

In the decoding process of the ECOC framework, a "distance" associated to each class is computed. This "distance" could be then interpreted as a ranking measure. But this ranking is the most trivial way for sorting the results. Moreover, the output of the ECOC system does not take into account any semantic relationship among classes, which may be beneficial in retrieval applications. As an example of an image retrieval system, suppose the query of "Dog". In the feature space, it is possible that there exists high similarity between "Dog" and "Bike", so based on features, the ranking will be higher for "Bike" than for some other class which can be semantically more similar to "Dog", such as "Cat". On the other hand, it is easy to see that similarity based on features also is important, and thus, a tradeoff based on appearance and semantics is required. Thus, our goal is to embed class semantics and contextual information in the ranking process. For this purpose, we define two matrices that will be used to vote the ranking process: one based on adjacency and another one based on ontology. These matrices are $n \times n$ matrices for n number of classes, where each entry represents the similarity between two classes. By multiplying the ranking vector of the ECOC output by these matrices, we can improve the retrieval results. In the rest of this section we describe the design of the adjacency matrix, ontology matrix, and their use to modify the output ECOC ranking.

4.1 Adjacency Matrix M_A

As we discussed, our goal is to enhance the primarily ranking based on ECOC output. First, we use class similarities in feature space and define an adjacency matrix.

There are different approaches in literature for measuring the similarity between two classes, Support Vector Machines margins and the distance between cluster centroid are two common approaches. Here, we follow a method similar to the second approach. However, just considering the cluster centroid would not be an accurate criteria for non-Gaussian data distributions. Instead, we re-cluster each class data into a few number of clusters and measure the mean distance of centroid of the new set of representant.

Since the objective is to alter the ranking, the defined adjacency matrix should be converted to a measure of likelihood, which means that the more two classes are similar, the more the new measure among them should be higher. Thus, we compute the inverse of the distance for each element and normalize each column of the matrix to one to give the same relevance to each of the classes similarities. The details of this procedure are described in algorithm 3.

Table 3
Adjacency Matrix M_A computation.

```
Given the class set $c = \{c_1, c_2, .., c_n\}$ and their associated data $W = \{W_{c_1}, .., W_{c_n}\}$ for $n$ classes

**For each** $c_i$
    1) Run $k$-means on $W_{c_i}$ set and compute the cluster centroids for class $c_i$ as $m_i = \{m_{i1}, .., m_{ik}\}$

Construct distance matrix $M_D$ as follows:

**For each pair of classes** $c_p$ **and** $c_q$
    1) $M_D(p,q) = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} \delta(m_{pi}, m_{qj})}{k^2}$, being $\delta$ a similarity function

Convert distance matrix $M_D$ to adjacency matrix $M_A$ as follows:

**For each pair of classes** $c_p$ **and** $c_q$
    1) $M_A(p,q) = \frac{1}{M_D(p,q)}$

Normalize each column $p$ of $M_A$ as follows:
    1) $M_A(p,q) = \frac{M_A(p,q)}{\sum_{i=1}^{n} M_A(i,p)}$
```

Look at the toy problem of Figure 8. In the example, for each class three representant are computed using k-means. Then the distance among all pairs of representant are computed for a pair of classes, obtaining an adjacency distance for that two classes as $M_D(1,3) = \frac{8+10+9+7+9+8+7.5+9.5+8.5}{9} = 8.5$. After that, the remaining positions of M_D are obtained in the same way, obtaining the following distance matrix M_D:

$$M_D = \begin{pmatrix} 1 & 4 & 8.5 \\ 4 & 1 & 10 \\ 8.5 & 10 & 1 \end{pmatrix}$$

Finally, the adjacency matrix is computed changing the distance to a value of likelihood and normalizing each column of the matrix. Then we obtain the final adjacency matrix M_A for the toy problem as:

$$M_A^{\text{Likelihood}} = \begin{pmatrix} 1 & 0.25 & 0.12 \\ 0.25 & 1 & 0.1 \\ 0.12 & 0.1 & 1 \end{pmatrix}, M_A = \begin{pmatrix} 0.73 & 0.18 & 0.08 \\ 0.19 & 0.74 & 0.07 \\ 0.09 & 0.08 & 0.81 \end{pmatrix}$$

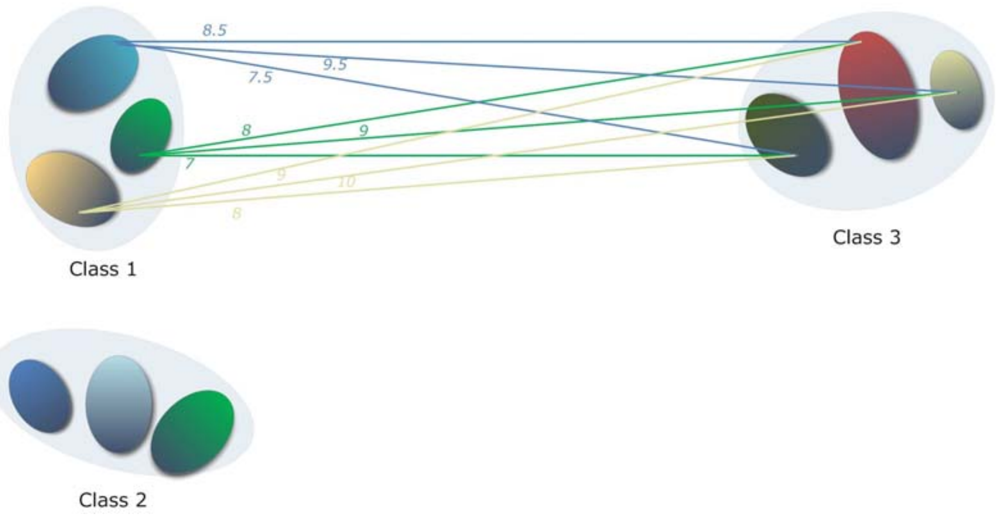


Fig. 8. Toy problem for a 3-class classification problem. For each class, three representant are computed using k-means. Then the distance among all pairs of representant are computed for a pair of classes. This distance is used in next steps to fill the adjacency matrix M_A values.

4.2 Ontology Matrix M_O

The process up to here considered the relationship between classes by means of computational methods. However, some times no matter how good the system is, it needs the human knowledge. Here, we try to ”inject” human knowledge of semantic similarity between classes into the system.

Taxonomy based on ontology is a tree or hierarchical classification which is organized by subtype-supertype relationships or in another word parent-child relationship. For example, Dog is a subtype of Animal. The authors of Caltech256 data set compiled a taxonomy for all the categories included in their data set. Based on this taxonomy, we also defined a similar one for the MSR-CORID data set, which will be used to validate our methodology in the results section. Both taxonomies are shown in Figures 10 and 11, respectively.

Here we try to construct a similarity matrix like we did for the adjacency matrix, but now the similarity of classes is computed by means of the taxonomy tree.

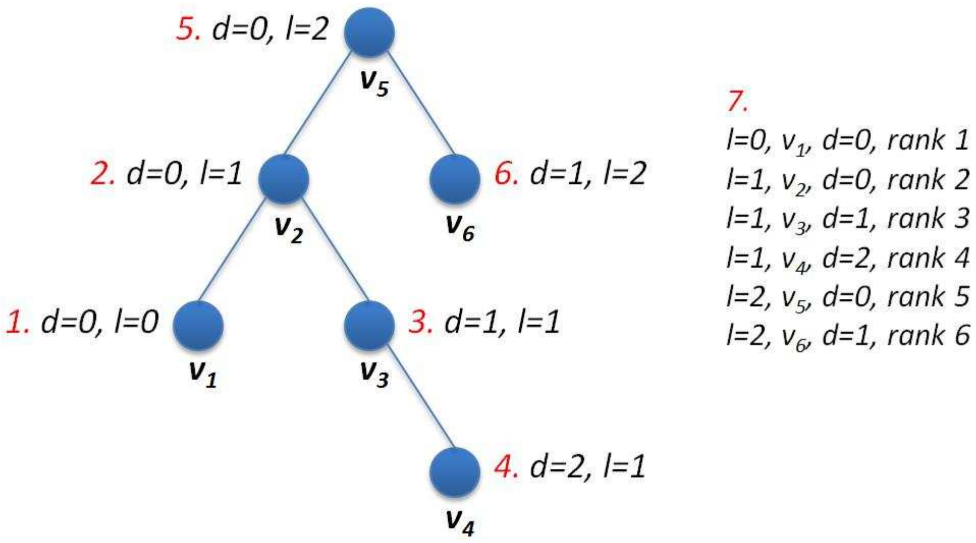


Fig. 9. Example of the ontology distance computation of vertex v_1 to the rest of vertices. The steps of the distance computation are sorted and showed in red. The final ranking is shown in the last step of the distance computation. This final ranking is then normalized and used as a ontology likelihood.

In order to compute the distance among classes based on the taxonomy, we look for common ancestor of nodes within the tree. Each category is represented as a leaf, and the non-leaf vertices correspond to abstract objects or super-categories. The less distance of the two leafs to their common ancestor, the less is their distance. We construct the similarity matrix by crawling the tree from a leaf and rank all other leaves based on their distance. When we start from each leaf and crawl up the tree, at each step the current node is being explored based on depth-first search algorithm. In this search, the less depth leaves get higher rank.

Finally, like in the case of the adjacency matrix, we need to convert distances into a measure of likelihood by inverting the values, and normalizing each column of the ontology matrix M_O to give the same importance for the taxonomy of all the classes. The whole process of computing the taxonomy distance and the ontology matrix is explained in algorithm 4 by means of recursive functions. Figure 9 shows an example of an ontology distance computation for the previous toy problem shown in Figure 8.

The final ontology matrix M_O obtaining after computing all ranking from ontology distance and likelihood computation are the followings:

Table 4
Ontology Matrix M_O computation.

```
Given the class set $c = \{c_1, c_2, .., c_n\}$ and the taxonomy graph $G$

For each leaf vertex $v_i$ in $G$, $i \in [1, .., n]$, where $n$ is the number of classes
  1) Visiting vertex $v_j = v_i$, Up Level $l = 0$, Depth $d = 0$
     Position list for each vertex $v_p$: $M_P(v_p) = [L_{v_p}, D_{v_p}]$
     where $L_{v_p}$ is the level of $v_p$ and $D_{v_p}$ is the depth of $v_p$
  2) Do while there are unvisited vertices
     1) $VisitVertice(v_j)$
     Function VisitVertice($v_p$):
     If $v_p$ is not visited
         visitChild($v_p$)
         if $\exists$ $parent(v_p)$
             $l = l + 1$
             $M(v_p) = [l, d]$
             $VisitVertice(parent(v_p))$
     Function VisitChild($v_p$):
     for each child $v_p^c$ of $v_p$:
         if $v_p^c$ has not been visited:
             if child($v_p^c$) $! = \varnothing$
                 VisitChild($v_p$)
             else
                 $d = d + 1$
         $M(v_p) = [l, d]$
  3) Filling the ranks
     $r = 0$
     for $\nu = [1, .., \max(l)]$
         for $\omega = [1, .., \max(d)]$
             if $v_q | M_P(v_q) = [\nu, \omega]$ is a leaf vertex of $G$
                 $M_O(i, q) = r$
                 $r = r + 1$

Convert distance matrix $M_D$ to ontology matrix $M_O$ as follows:

For each pair of classes $c_p$ and $c_q$
  1) $M_O(p, q) = \frac{1}{M_O(p,q)}$

Normalize each column $p$ of $M_O$ as follows:
  1) $M_O(p, q) = \frac{M_O(p,q)}{\sum_{i=1}^{n} M_O(i,p)}$
```

$$M_O^{\text{Ranking}} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 4 & 3 & 1 & 2 & 5 & 6 \\ 4 & 3 & 2 & 1 & 5 & 6 \\ 5 & 2 & 3 & 4 & 1 & 6 \\ 6 & 3 & 4 & 5 & 2 & 1 \end{pmatrix}, M_O^{\text{Likelihood}} = \begin{pmatrix} 1.0000 & 0.5000 & 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.5000 & 1.0000 & 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.2500 & 0.3333 & 1.0000 & 0.5000 & 0.2000 & 0.1667 \\ 0.2500 & 0.3333 & 0.5000 & 1.0000 & 0.2000 & 0.1667 \\ 0.2000 & 0.5000 & 0.3333 & 0.2500 & 1.0000 & 0.1667 \\ 0.1667 & 0.3333 & 0.2500 & 0.2000 & 0.5000 & 1.0000 \end{pmatrix}$$

$$M_O = \begin{pmatrix} 0.4082 & 0.2041 & 0.1361 & 0.1020 & 0.0816 & 0.0680 \\ 0.2041 & 0.4082 & 0.1361 & 0.1020 & 0.0816 & 0.0680 \\ 0.1020 & 0.1361 & 0.4082 & 0.2041 & 0.0816 & 0.0680 \\ 0.1020 & 0.1361 & 0.2041 & 0.4082 & 0.0816 & 0.0680 \\ 0.0816 & 0.2041 & 0.1361 & 0.1020 & 0.4082 & 0.0680 \\ 0.0680 & 0.1361 & 0.1020 & 0.0816 & 0.2041 & 0.4082 \end{pmatrix}$$

4.3 Altering ECOC output rank using M_A and M_O

Given the output vector $D = \{d_1, .., d_n\}$ of the ECOC design, where d_i represents the distance of a test sample to codeword i of the coding matrix, first, we convert the vector D to a measure of likelihood by inverting each position of D and normalizing the vector as follows:

$$D^L = \frac{1}{\sum_{i=1}^{n} \frac{1}{d_i}} \left\{ \frac{1}{d_1}, .., \frac{1}{d_n} \right\} \tag{11}$$

Then, using the previous M_A and M_O matrices, the new altered rank R is obtained by means of a simple multiplication, as shown in eq.(12).

$$R = D^L \cdot M_A \cdot M_O \tag{12}$$

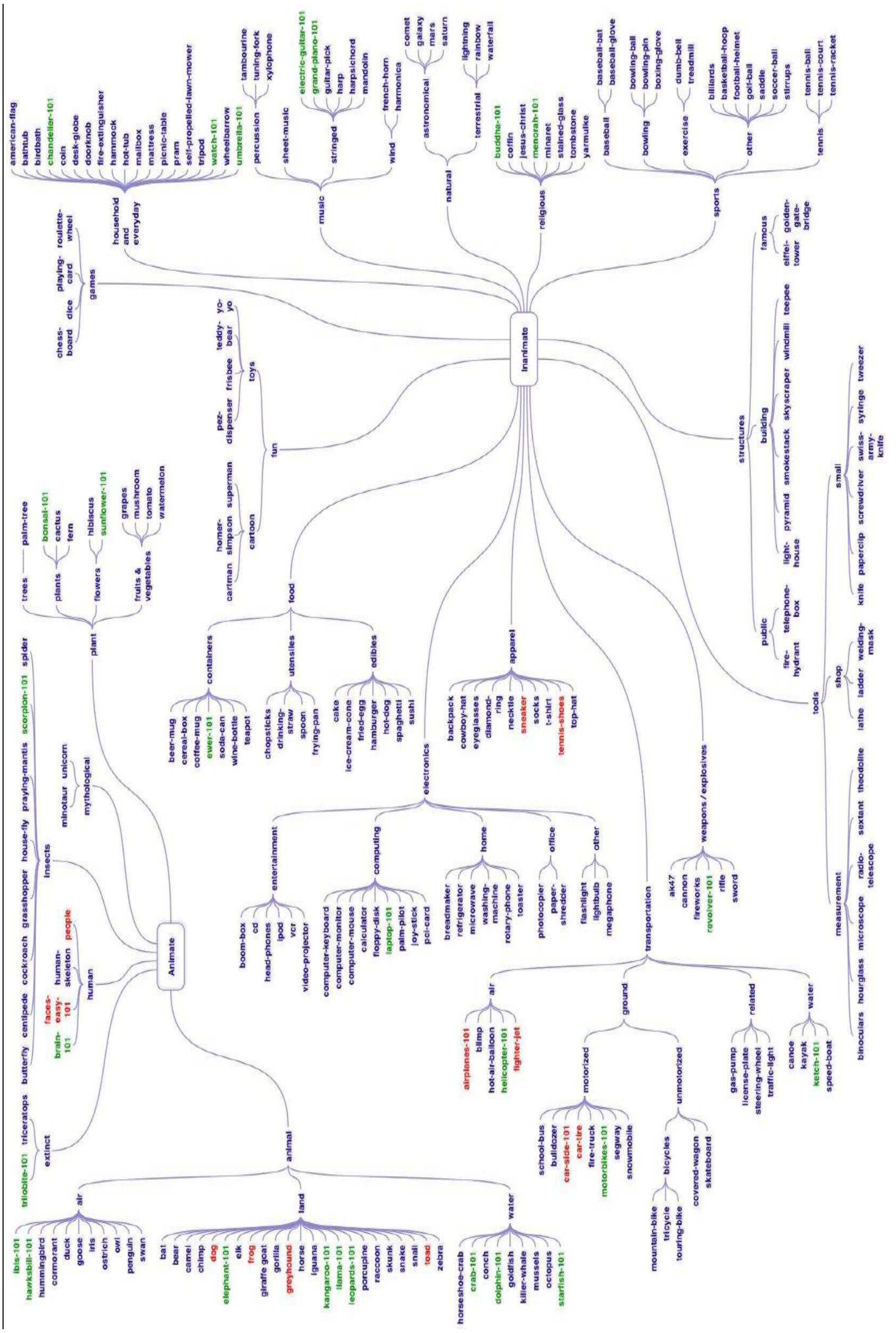


Fig. 10. A taxonomy of Caltech 256 categories created by hand. At the top level these are divided into animate and inanimate objects. Green categories contain images that were borrowed from Caltech 101. A category is colores red if it overlaps with some other category (such as 'dog' and 'greyhound').

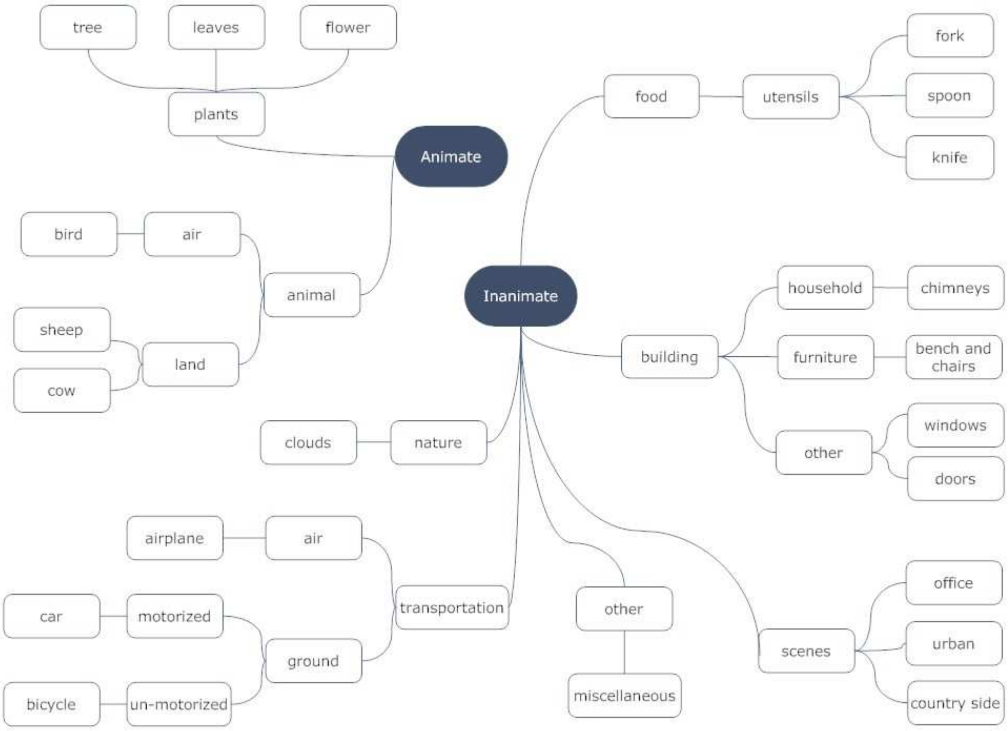


Fig. 11. Taxonomy of object categories of the MSRCORID data set.

5 Results

We divide the results into two types, validating the C-BOVW methodology and validating the ECOC-Rank methodology.

5.1 Contextual Bag-Of-Visual Words

Before the presentation of the results of the C-BOVW methodology, first, we discuss the data, methods, software details, and validation protocol of the experiments.

Data: The data used in the experiments consist of 15 categories from public Caltech 101 [54] and Caltech 256 [55] repository data sets. One sample for each category is shown in Figure 12. For each category, 50 samples were used, 10 samples to define the BOW and another 40 images to define new test sentences.

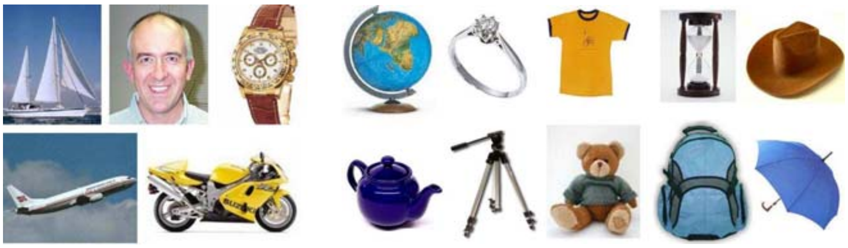


Fig. 12. Considered categories from the Caltech 101 and Caltech 256 repositories.

Methods: We compare the C-BOVW with the classical BOVW model. For both methods, the same initial set of regions is considered in order to compare both strategies at the same conditions. About 200±20 object regions are found by image using the Harris-Affine detector [56] and described using the SIFT descriptor [3]. The visual words are obtained using the public open source k-means software from [57]. After computing the final words and representant, multi-class classification is performed using an one-versus-one ECOC methodology with different base classifiers: Mean Nearest Neighbor (NMC), Fisher Discriminant Analysis with a previous 99% of PCA (FLDA), Gentle Adaboost with 50 iterations of decision stumps (G-ADA), Linear Support Vector Machines with the regularization parameter $C = 1$ (Linear SVM), and Support Vector Machines with RBF Kernel with C and γ parameters set to 1 (RBF SVM) [5] . Finally, we use the Linear Loss-weighted decoding to obtain the class label [58].

[5] We decided to keep the parameter fixed for the sake of simplicity and easiness of replication of the experiments, though we are aware that this parameter might not be optimal for all data sets.

Software details: The project was implemented in Python and the data were stored in MySQL data set. For the multi-class classification we used the Error-Correcting Output Codes library implemented in Matlab/Octave [6].

Validation protocol: We used the sentences obtained by the 50 samples of each category and performed stratified ten-fold cross-validation evaluation.

5.1.1 Caltech 101 and 256 classification

In this experiment, we started classifying from three Caltech categories increasing by 2 up to 15. For each step, different number of visual words are computed: 30, 40, and 50. These numbers are obtained by performing ten iterations of the merging procedure (experimentally tested). In order to compare the BOVW and C-BOVW methods at the same conditions, the same detected regions and descriptions are used for all the experiments. The order in which the categories are considered is the following: (1-3) airplane, motor-bike, watch, (4-5) tripod, face, (6-7) ketch, diamond-ring, (8-9) teddy-bear, t-shirt, (10-11) desk-globe, backpack, (12-13) hourglass, teapot, (14-15) cowboy-hat, and umbrella. The obtained results applying ten-fold cross-validation are graphically shown in Figure 13 for the different ECOC base classifiers. Note that the classification error significantly varies depending on the ECOC classifier. In particular, Gentle Adaboost obtains the best results, with a classification error inferior to 0.2 in all the tests when using 30 C-BOVW words. Independently of the ECOC classifier, in most of the experiments the C-BOVW model obtains errors inferiors to those obtained by the classical BOVW. BOVW only obtains slightly better results in the case of Gentle Adaboost for eleven classes and 50 visual words.

An important remark of the C-BOVW model is about the selection of the number of merging iterations. This parameter has a decisive impact over the generalization capability of the new visual dictionary. First iterations of the merging procedure use to fuse very close feature-words which belong to different visual words whereas final merging iterations fuse more far regions of the feature-space. Thus, a large number of iterations could be detrimental since the new merged words could be too general for discriminating among sentences of different object categories. Thus, this parameter should be estimated for each particular problem domain (i.e. applying cross-validation over a training and a validation subset). In the previous experiment we checked that ten merging iterations obtains significant performance improvements, though we are aware that this parameter could be not optimal for all the data sets.

[6] http://ecoclib.sourceforge.net/

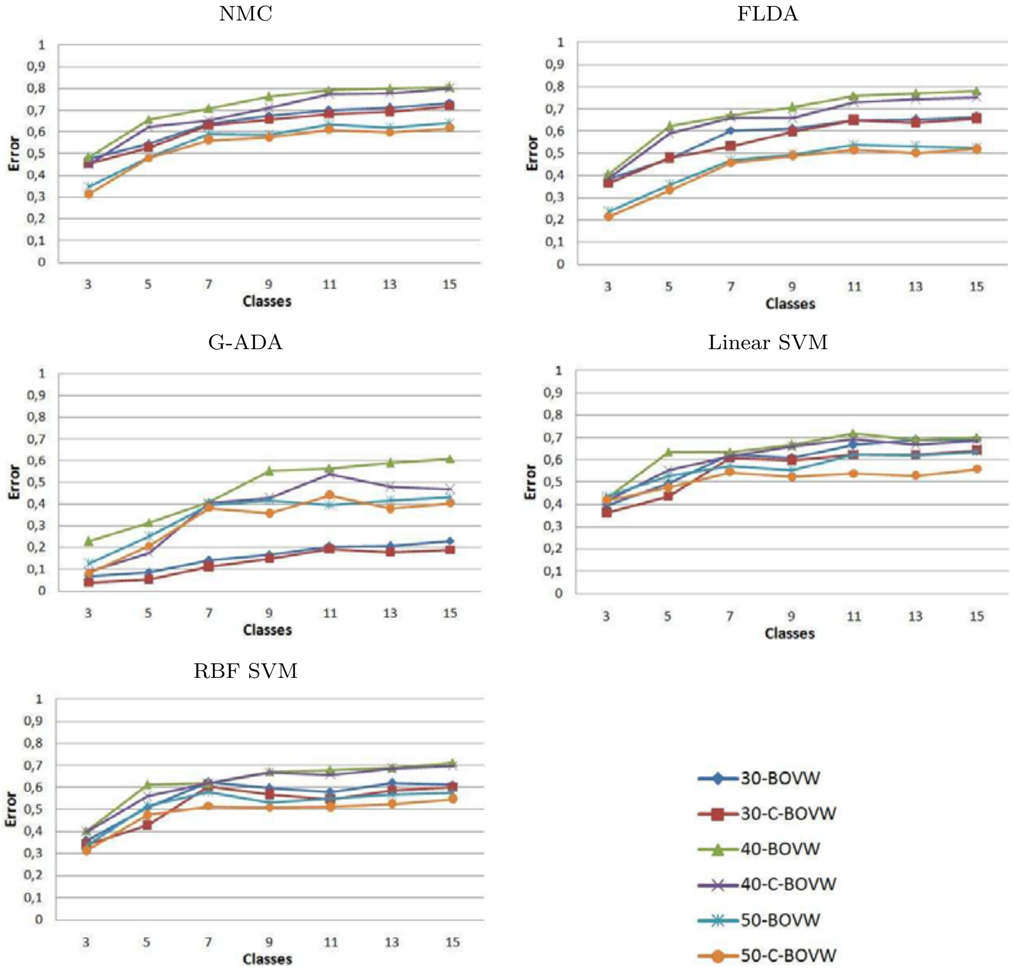


Fig. 13. Classification results for the Caltech categories using BOVW and C-BOVW dictionaries for different number of visual words and ECOC base classifiers.

Before the presentation of the results of the ECOC-Rank methodology, first, we discuss the data, methods and parameters, software details, and validation protocol of the experiments.

Data: The data used in our experiments consist on two public data sets: Caltech 256 [55] and 'Microsoft Research Cambridge Object Recognition Image data set' (MRCORID) [59].

Methods and parameters: We use the same methods and parameters for computing BOVW and CBOVW than at the previous experiments for the Caltech 256 [55] data set and 'Microsoft Research Cambridge Object Recognition Image data set data set' [59]. For the ECOC classification, One-versus-one method with Gentle Adaboost with 50 decision stumps and RBF SVM classifiers has been used. We use the Linear Loss-weighted decoding to obtain the class label [58]. For the adjacency matrix construction, the k parameter of k-means has been experimentally set to 3. For ranking the hist count we looked for one to seven matches at the first 15 positions using vector ontology and semantic distances of 0.001 and 0.0001.

Software details: The project was implemented in Python and the data was stored in MySQL data set. Error-Correcting Output Codes and Ranking code were implemented in Matlab.

Validation measurements: In order to analyze the retrieval efficiency, we defined an ontology distance based on taxonomy trees to look for the retrieved classes at the first positions of the ranking process based on the confusion matrix.

As explained in the previous section, the ranking result R is a sorted set of classes, where the first items have the highest rank.

In retrieval problems, we are looking for an interval of positions to look for the target objects. In our case, retrieving classes, we need to define a validation measure among classes. For this purpose, we define an ontology distance m based on the taxonomy tree and adjacency matrices. Each class c_i in R is accepted if its ontology distance d_i compared to the true label class is less than m. The accepted results at the end of the list R are not desired, so another parameter k (positions) is used to analyze the results of the first positions of the ranking. **If there are more than N (accepted count) accepted classes based on the value of m at the first positions defined by k, then we achieve a test hit**. In order to perform a realistic analysis, we included this validation procedure in a stratified 10-fold evaluation procedure. The algorithm that summarizes the retrieval validation is shown in table 5.

Table 5
ECOC-Rank evaluation.

```
Given the sorted list of classes based on their rank $R = \{r_1, .., r_n\}$

**For each item $r_i$ in the top $k$ positions of $R$**

$acceptedCount = 0$
  1) $d = OntologyDistance(r_i, TrueLabel)$
  2) if $d < m$ then $acceptedCount += 1$

1) If $acceptedCount > N$ then $Hit$
```

We also apply statistical Friedman and Nemenyi test to look for statistical significance among the obtained performances [60].

5.2.1 Caltech 256 retrieval evaluation

Some samples of the Caltech 256 data set are shown in Figure 14. The ontology matrix M_O computed for this data set and BOVW features are shown in Figure 15. In this case, we defined an ontology distance of 0.001 and 0.0001 for Adaboost ECOC base classifier based on the ontology tree of Figure 10 and the ontology distance defined in previous chapters. For both distances we computed the CBOVW and BOVW features for this data set for different values of the k first positions and number of hits. The results are shown in the performance surfaces of Figure 16 and 17, respectively. **The performances are also shown in Table 6 estimated as the mean performance surface for each experiment.** We have used this performance evaluation since it is more general than the classical ROC curve.

Fig. 14. Caltech 256 samples.

Table 6
Performances of Caltech 256 data set for different methods and parameters using Gentle Adaboost ECOC base classifier and ontology distance evaluation.

Problem	Adjacency	Ontology	Adjacency & Ontology	ECOC-raw
m=0.001 CBOVW	0.4635	0.6902	0.4974	0.5604
m=0.001 BOVW	0.4394	0.6901	0.4389	0.5530
m=0.0001 CBOVW	0.0843	0.1485	0.0829	0.0809
m=0.0001 BOVW	0.0718	0.1479	0.0719	0.0785

Comparing CBOVW and BOVW methods for all the previous experiments and both classifiers, we compute the mean rank for both strategies. The rankings are obtained estimating each particular ranking r_i^j for each problem i and each method j, and computing the mean ranking R for each method as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of problems. We obtained a ranking for CBOVW of 1.00 and a ranking for BOVW of 2.00. This means that though the improvement of CBOVW is of small difference in most of the experiments, it achieves always the best performance, and CBOVW is preferred as the first choice for this particular data set using Adaboost base classifier.

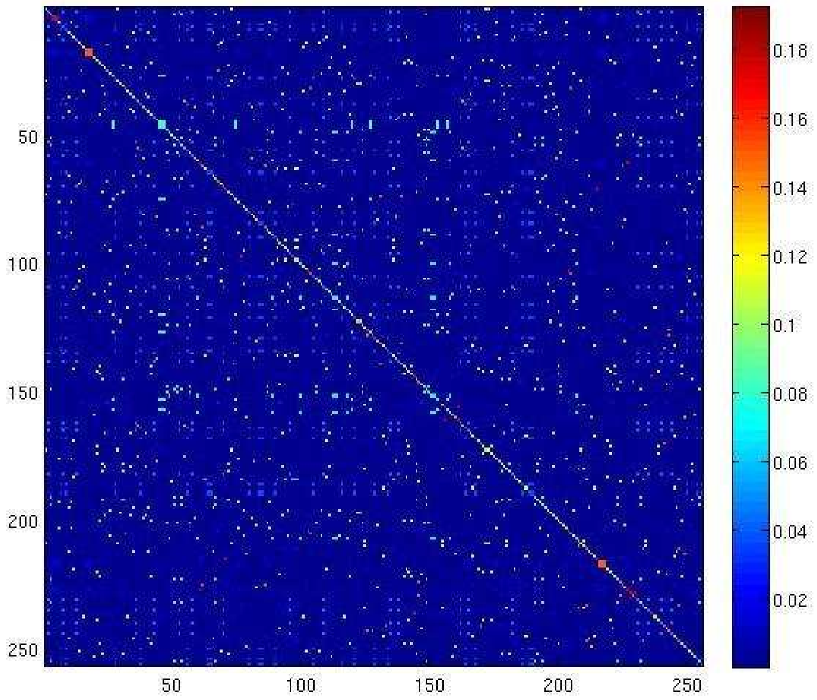


Fig. 15. Ontology matrix M_O computed for the Caltech data set and CBOVW.

Now we compare if any of the four variants of ranking strategies is preferred against the rest. For this purpose, considering all previous experiment, we compute the mean rank of each strategy as explained before. The obtained ranks are shown in Table 7.

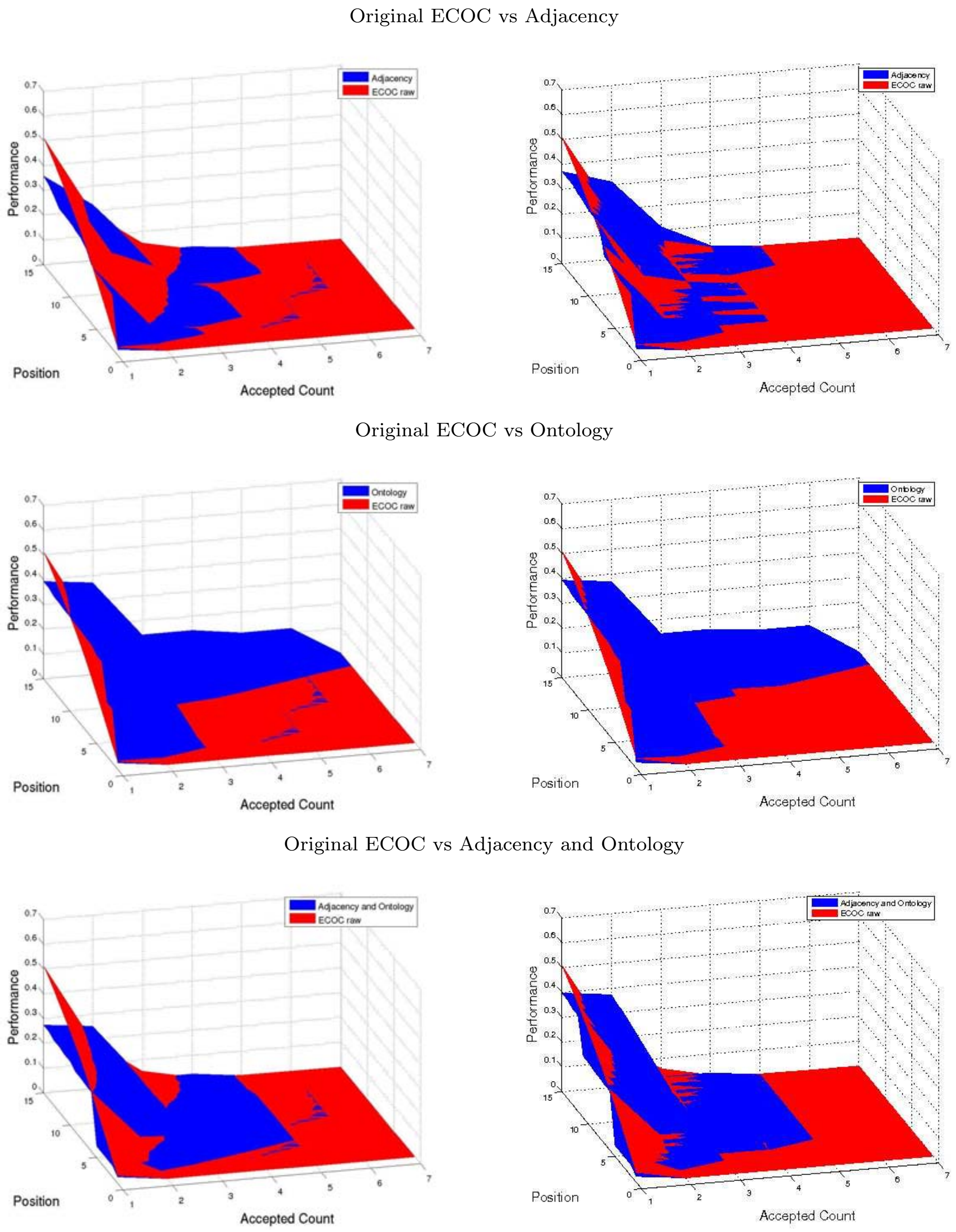
Table 7
Ranking of Caltech 256 data set for the different ECOC-Rank configurations considering all experiments with ontology distance evaluation.

Adjacency	Ontology	Adjacency & Ontology	ECOC-raw
3.2500	1.0000	3.3750	2.5000

In order to analyze if the difference between methods ranks are statistically significant, we apply the Friedman and Nemenyi tests. In order to reject the null hypothesis that the measured ranks differ from the mean rank, and that the ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \qquad (13)$$

In our case, with $k = 4$ ranking designs to compare and $N = 4$ experiments, $X_F^2 = 10.08$. Since this value has shown to be undesirable conservative [60],


Original ECOC vs Adjacency

Original ECOC vs Ontology

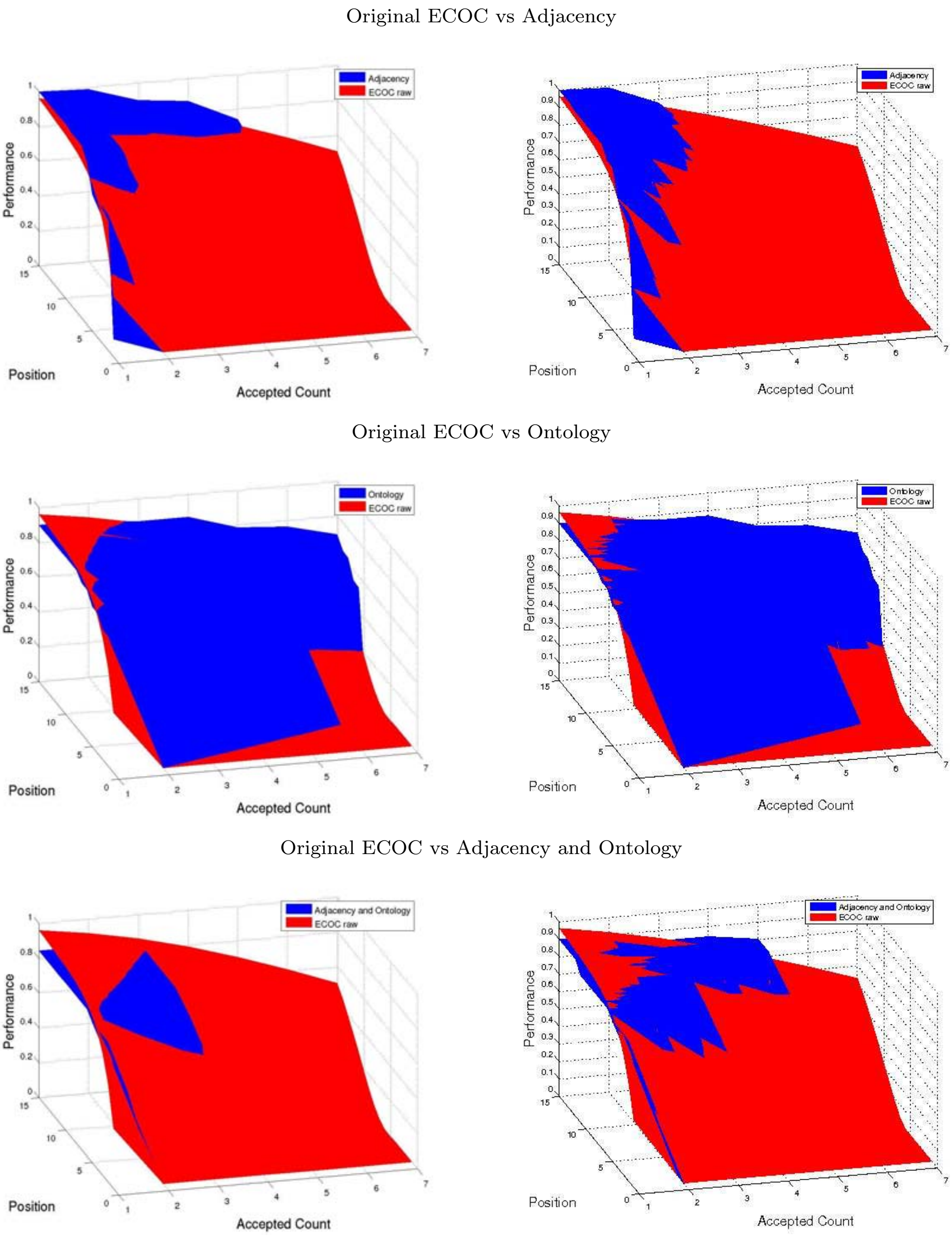
Original ECOC vs Adjacency and Ontology

Fig. 16. Results on Caltech 256 data set for ontology distance m=0.0001 and Gentle Adaboost ECOC base classifier. Left column using BOVW and right column using CBOVW.

Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{14}$$

Applying this correction we obtain $F_F = 15.82$. With four methods and four


Original ECOC vs Adjacency

Original ECOC vs Ontology

Original ECOC vs Adjacency and Ontology

Fig. 17. Results on Caltech 256 data set for ontology distance m=0.001 and Gentle Adaboost ECOC base classifier. Left column using BOVW and right column using CBOVW.

experiments, F_F is distributed according to the F distribution with $((k-1),(k-1)(N-1))$ degrees of freedom (3 and 9 in our case). The critical value of $F(3,9)$ for 0.05 is 3.86. As the value of F_F is higher than 3.86 we can reject the null hypothesis. One we have checked for the non-randomness of the results, we can perform a post hoc test to check if one of the techniques can be singled out. For this purpose we use the Nemenyi test - two techniques are significantly different if the corresponding average ranks differ by at least the critical difference value (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{15}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$. In our case, when comparing four methods with a confidence value $\alpha = 0.10$, $q_{0.10} = 1.53$. Substituting in eq.15, we obtain a critical difference value of 1.39. Since the difference of any technique rank with the Ontology alteration of the ECOC-Rank is higher than the CD, we can infer that the Ontology alteration approach is significantly better than the rest with a confidence of 90% in the present experiments.

5.2.2 *Microsoft Research Cambridge Object Recognition Image data set retrieval evaluation*

One sample for each category of this data set is shown in Figure 18. The ontology matrix M_O computed for this data set and BOVW features are shown in Figure 19. In this case, we have defined an ontology distance of 0.001 and 0.0001 for Adaboost ECOC base classifier based on the ontology tree of Figure 11 and the ontology distance defined in previous chapters. For both distances we computed the CBOVW and BOVW features for this data set for different values of the k first positions and number of hits. The results are shown in the performance surfaces of Figure 20 and 21, respectively. The performances are also shown in Table 8 estimated as the mean performance surface for each experiment.

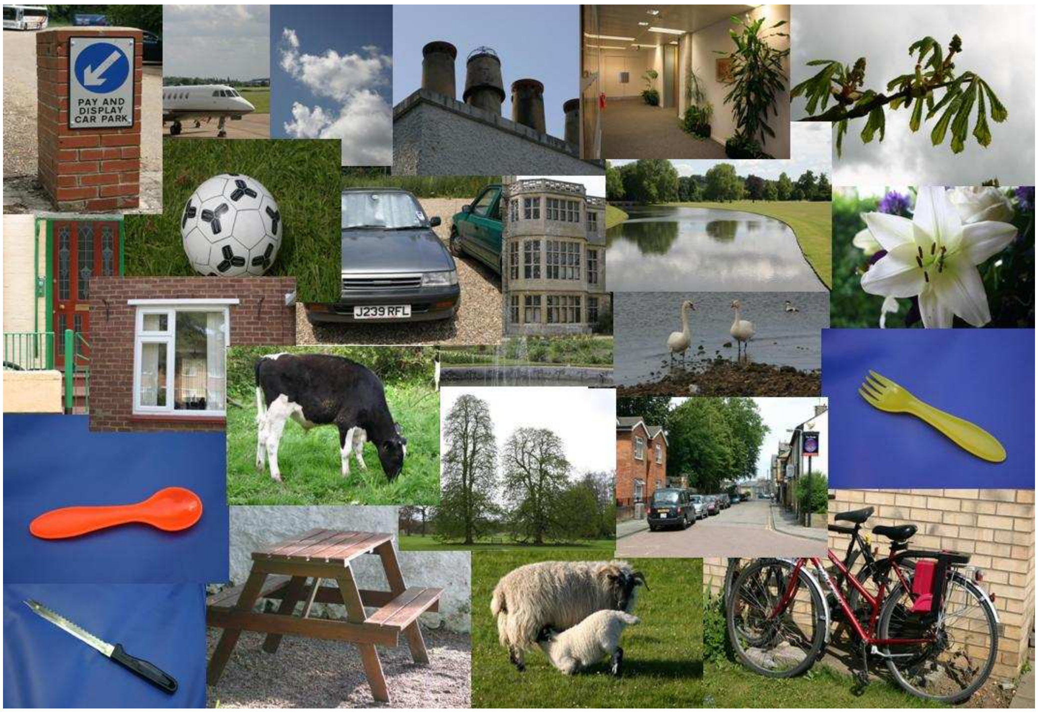


Fig. 18. One representant for each category of the Microsoft Research Cambridge Object Recognition Image data set.

Table 8
Performances of Microsoft Research Cambridge Object Recognition Image data set for different methods and parameters using Gentle Adaboost ECOC base classifier and ontology distance evaluation.

Problem	Adjacency	Ontology	Adjacency & Ontology	ECOC-raw
m=0.001 CBOVW	0.3154	0.1764	0.2997	0.1572
m=0.001 BOVW	0.3154	0.1744	0.2996	0.1568
m=0.0001 CBOVW	0.1777	0.0671	0.1576	0.0665
m=0.0001 BOVW	0.1777	0.0659	0.1576	0.0667

The same analysis is shown in Figure 22, Figure 23 and Table 9 for RBF SVM

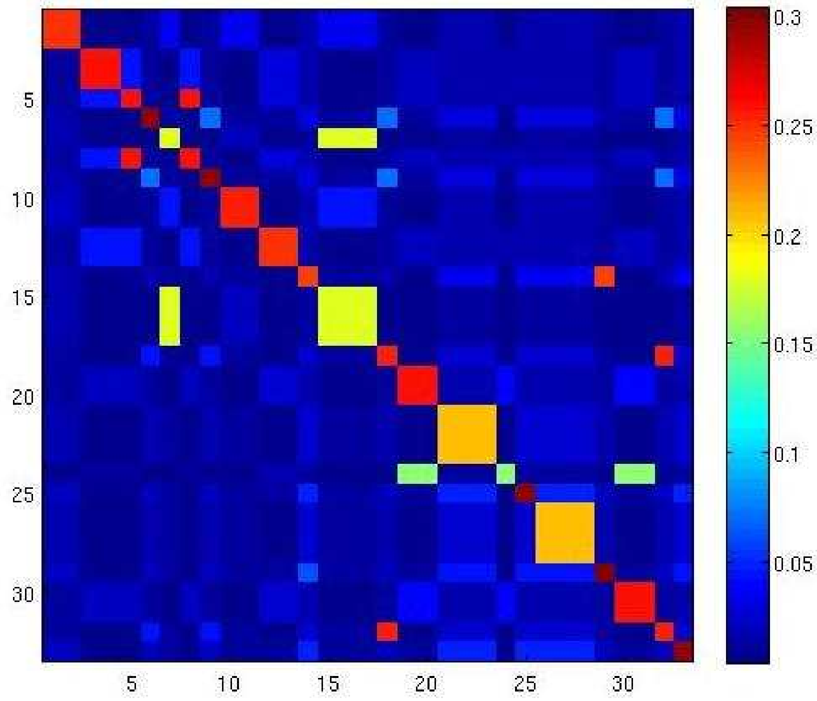


Fig. 19. Ontology matrix M_O computed for the Microsoft Research Cambridge Object Recognition Image data set and CBOVW.

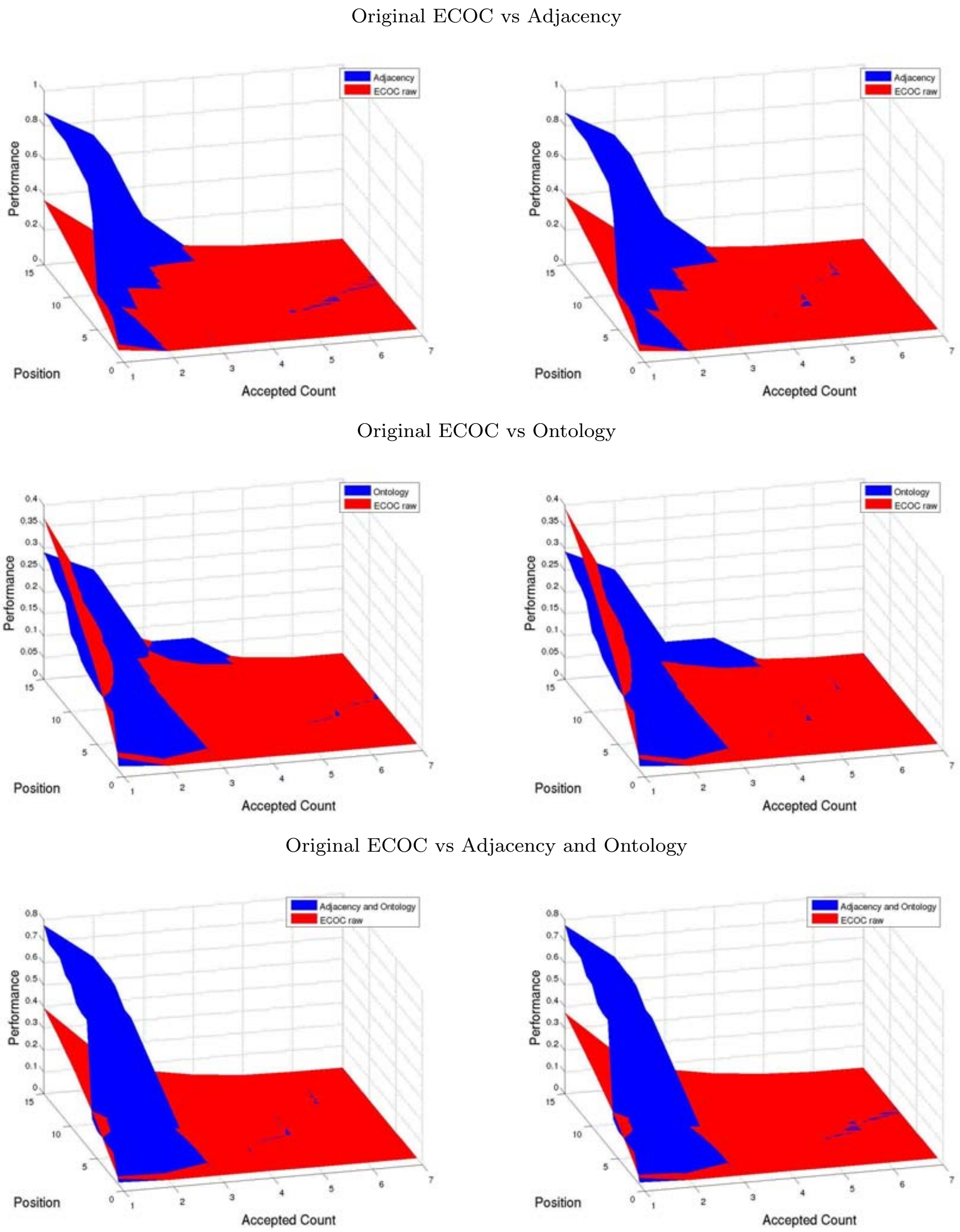
ECOC base classifier, respectively.

Table 9
Performances of Microsoft Research Cambridge Object Recognition Image data set for different methods and parameters using RBF SVM ECOC base classifier and ontology distance evaluation.

Problem	Adjacency	Ontology	Adjacency & Ontology	ECOC-raw
m=0.001 CBOVW	0.3156	0.1777	0.2995	0.2019
m=0.001 BOVW	0.3714	0.1798	0.3001	0.2038
m=0.0001 CBOVW	0.1777	0.0688	0.1583	0.1004
m=0.0001 BOVW	0.2511	0.0676	0.1577	0.0950

Comparing CBOVW and BOVW methods for all the previous experiments and both classifiers, we compute the mean rank for both strategies as explained in the previous section. We obtained a ranking for CBOVW of 1.3750 and a ranking for BOVW of 1.4375. Although the difference is not high, COBW if preferred as the first choice. Note that also the improvements in the case of Adaboost base classifiers are more significant for CBOVW than when using SVM for this particular data set.

Now we compare if any of the four variants of ranking strategies is preferred against the rest. For this purpose, considering all previous experiment, we compute the mean rank of each strategy as explained before. The obtained ranks are shown in Table 10.


Original ECOC vs Adjacency

Original ECOC vs Ontology

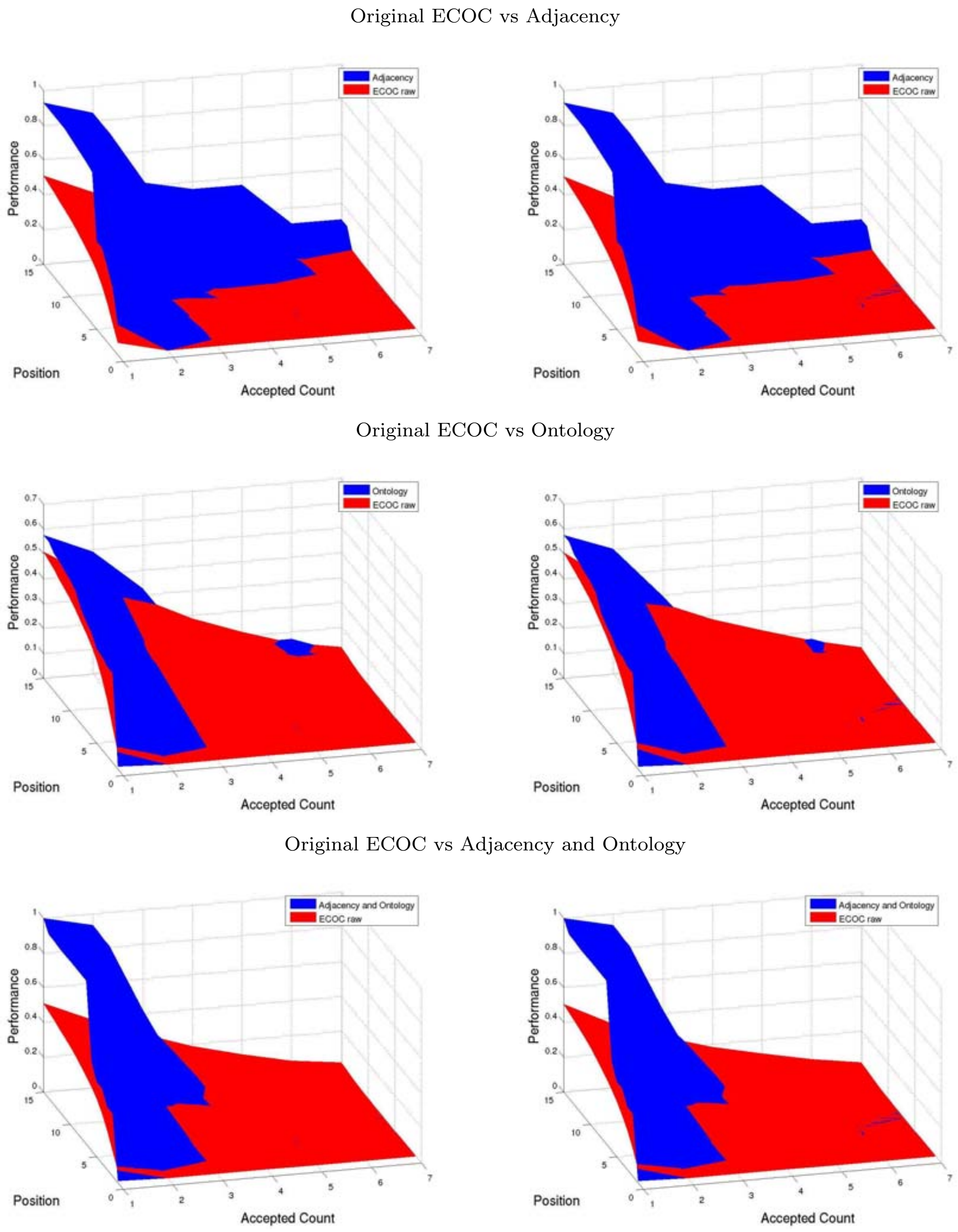
Original ECOC vs Adjacency and Ontology

Fig. 20. Results on Microsoft Research Cambridge Object Recognition Image data set for ontology distance m=0.0001 and Gentle Adaboost ECOC base classifier. Left column using BOVW and right column using CBOVW.

Table 10

Ranking of Microsoft Research Cambridge Object Recognition Image data set for the different ECOC-Rank configurations considering all experiments with ontology distance evaluation.

Adjacency	Ontology	Adjacency & Ontology	ECOC-raw
1.0000	3.6250	2.0000	3.3750



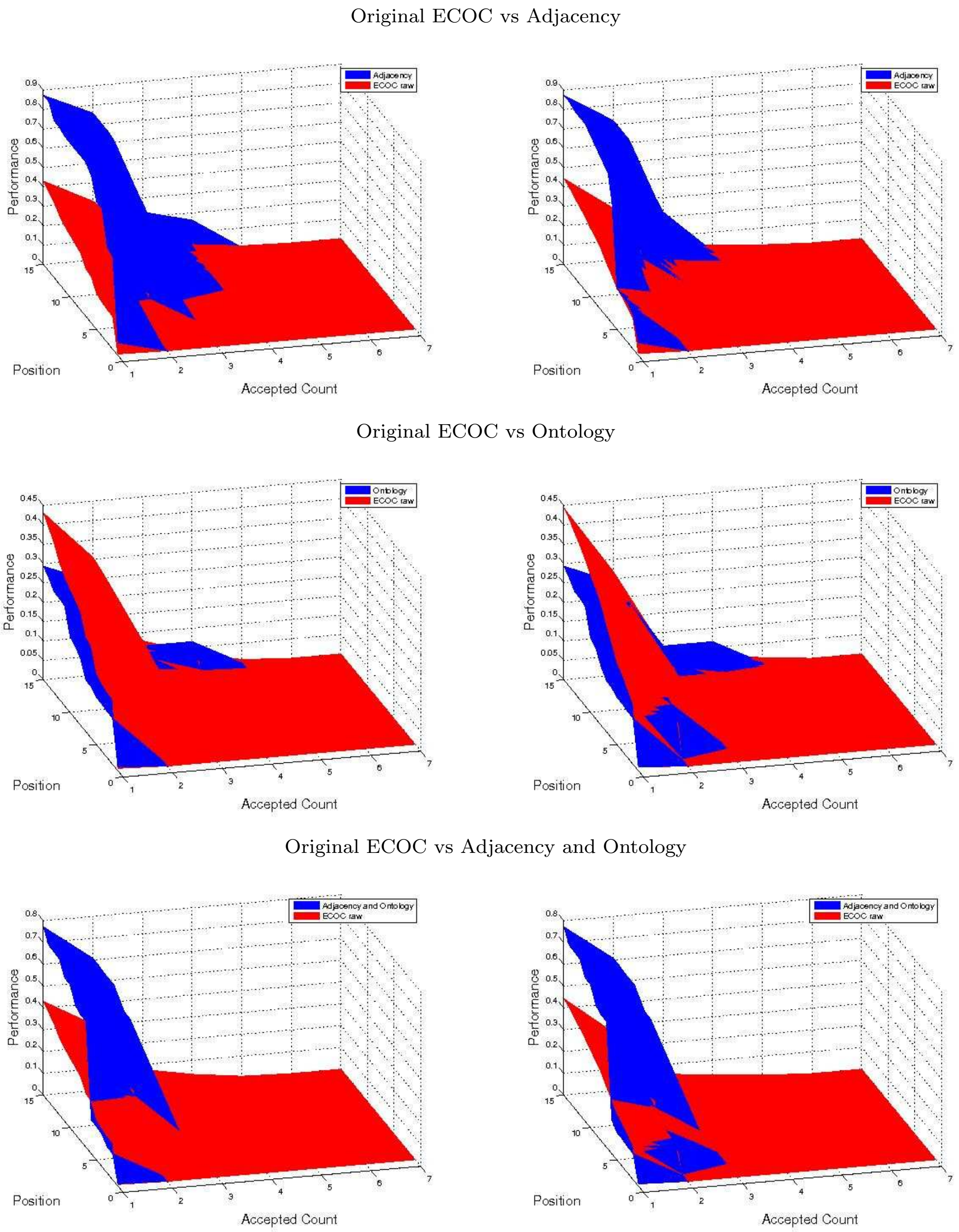
Original ECOC vs Adjacency

Original ECOC vs Ontology

Original ECOC vs Adjacency and Ontology

Fig. 21. Results on Microsoft Research Cambridge Object Recognition Image data set for ontology distance m=0.001 and Gentle Adaboost ECOC base classifier. Left column using BOVW and right column using CBOVW.

In order to analyze if the difference between methods ranks are statistically significant, we apply the previous Friedman and Nemenyi tests. In our case, with $k = 4$ ranking designs to compare and $N = 8$ experiments, $X_F^2 = 21.75$. Applying the Iman and Davenport correction, we obtain $F_F = 67.66$. With four methods and eight experiments, F_F is distributed according to the F distribution with 3 and 21 degrees of freedom. The critical value of $F(3, 21)$

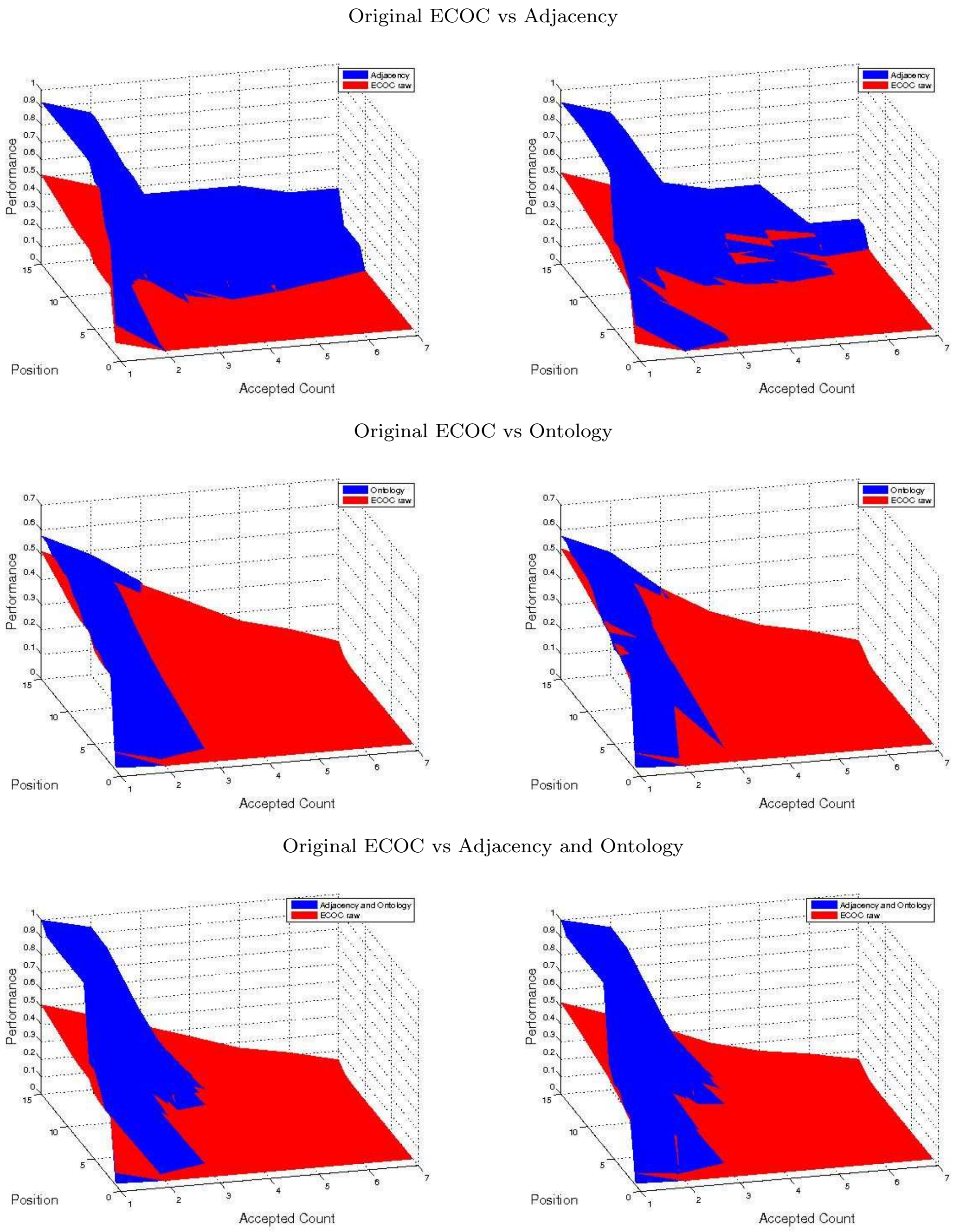

Original ECOC vs Adjacency

Original ECOC vs Ontology

Original ECOC vs Adjacency and Ontology

Fig. 22. Results on Microsoft Research Cambridge Object Recognition Image data set for ontology distance m=0.0001 and RBF SVM ECOC base classifier. Left column using BOVW and right column using CBOVW.

for 0.05 is 3.07. As the value of F_F is higher than 3.07 we can reject the null hypothesis. One we have checked for the for the non-randomness of the results, we apply the Nemenyi test. In our case, when comparing four methods with a confidence value $\alpha = 0.10$, $q_{0.10} = 1.53$. Substituting in eq.15, we obtain a critical difference value of 0.98. Since the difference of any technique rank with the Adjacency alteration of the ECOC-Rank is higher than the CD, we


Original ECOC vs Adjacency

Original ECOC vs Ontology

Original ECOC vs Adjacency and Ontology

Fig. 23. Results on Microsoft Research Cambridge Object Recognition Image data set for ontology distance m=0.001 and RBF SVM ECOC base classifier. Left column using BOVW and right column using CBOVW.

can infer that the Adjacency alteration approach is significantly better than the rest with a confidence of 90% in the present experiments.

6 Conclusion

In this work we re-formulated the bag-of-visual-words model so that geometrical information of significant object region descriptions are taken into account in the dictionary construction step. In this sense, regions which have slightly different descriptors because of small displacements in the region detection process can be merged together in a same visual word. The method is based on the definition of a contextual-space and a feature-space. The first space codifies the geometrical properties of regions meanwhile the second space contains the region descriptions. A merging process is then used to fuse feature words based on their proximity in the contextual-space. The new dictionary is learnt in an Error-Correcting Output Codes design to perform multi-class object categorization. The results when spatial information is taken into account showed significant performance improvements compared to the classical approach for different number of object categories and visual words.

Moreover, we also applied the proposed contextual bag-of-visual-words to deal with class retrieval problems. The original output ECOC vector is used as a measure of ranking for retrieval purposes. In order to include contextual and semantic information in a retrieval system we defined two matrices that alter the output ECOC vector to improve ranking and retrieval. The contextual information in included by the definition of an adjacency matrix which positions store a value of likelihood based on the inverse distance of classes analyzing class representant in feature space. The semantic information is included in the retrieval process by defining an ontology matrix by means of a taxonomy tree and a new ontology distance procedure. Finally, both adjacency and ontology matrices multiply the initial ECOC output to obtain a more realistic class ranking and perform class retrieval. The new ECOC-Rank procedure showed to outperform classical ECOC output values when retrieving classes based on semantic information. Furthermore, using the contextual bag-of-visual-words in the ECOC-Rank procedure also showed to obtain significant performance improvements compared to classical approaches.

As future lines, we plan to test alternatives for the definition of adjacency and ontology matrices of the ECOC-Rank methodology. We also want to test for on-line methods which can analyze adjacency/taxonomy definitions over different types of data to look for the best combination of matrices that will improve class retrieval in a problem dependent way.

Publications

Mehdi Mirza-Mohammadi, Sergio Escalera, and Petia Radeva, "Contextual-Guided Bag-Of-Visual-Words Model for Multi-class Object Categorization", *Conference on Computer Analysis of Images and Patterns*, pp. 748-756, Germany, 2009.

Mehdi Mirza-Mohammadi, Francesco Ciompi, Sergio Escalera, Oriol Pujol, and Petia Radeva, "Ranking Error-Correcting Output Codes for Class Retrieval", 4rth Computer Vision Center Workshop, New Trends and Challenges, 2009.

CBOVW nomenclature

Table 11
CBOVW nomenclature

$C = \{(c_1, w_1^C), .., (c_q, w_q^C)\}$ - Contextual space, where w_i^C is the ith word of the contextual-space

$D = \{(\mathbf{x}_1, l_1), .., (\mathbf{x}_m, l_m)\}$ - $\mathbf{x}_i$ is an object sample of label $l_i \in [1, .., n]$ for a n-class problem

$F = \{(f_1, w_1^F), .., (f_q, w_q^F)\}$ - Feature space, where w_i^F is the ith word of the feature-space

I - merging steps

K - number of clusters

M - Contextual-feature relational matrix

ρ - Region parameter

$R = \{(r_1, w_1), .., (r_v, w_b)\}$ - Representant set, where r_v is a representant for word w_i, $i \in [1, ..b]$ for b words

$S = \{(s_1, l_1), .., (s_m, l_m)\}$ - Word sentences, where s_i is the sentence of sample $\mathbf{x}_i$

W - list of feature words to be merged

$X_i = \{(x_1, y_1, \rho_1^1, \rho_1^2, \rho_1^3), .., (x_j, y_j, \rho_j^1, \rho_j^2, \rho_j^3)\}$ - Set of detected regions

$X_i^r = \{r_1, .., r_j\}$ - Region descriptors, where r_j is the description of the jth detected region of sample $\mathbf{x}_i$.

x - x coordinate

y - y coordinate

z_i - Number of representant for word w_i

Error-Correcting Output Codes nomenclature

Table 12
Error-Correcting Output Codes nomenclature

Δ - Matrix composed by the Hamming distances between the codewords of M

ρ^j - j^{th} feature of the object (data sample) ρ

C - Set of classes

c_i - Class i

$d(y_{i_1}, y_{i_1})$ - Decoding measure between codewords of classes c_{i_1} and c_{i_2}

d - Distance

$\{f_1, ..., f_n\}$ - Set of continuous hypotheses, $f_j \in R$

H - Matrix of accuracy of hypotheses

$\{h_1, ..., h_n\}$ - Discrete hypotheses set, $h_j \in \{+1, -1\}$

$\ell = \{\ell_1, .., \ell_N\}$ - Set of labels

$L(\theta)$ - Loss-based function of parameter θ

L - Sets of classes labels

$l(\rho) = \ell_i$ - Label function of data sample ρ is ℓ_i

M_W - Matrix of weights

$M \in \{-1, +1\}^{N \times n}$ - Binary coding matrix

$M \in \{-1, 0, +1\}^{N \times n}$ - Ternary coding matrix

m - Number of objects

N - Number of classes

n - Number of binary problems

$P(X)$ - Probability of item X

P - Prior

υ, ω - Optimization parameters

W - Set of weighting values

w - Weight

X - Set of objects

x - Test codeword

y_i - Codeword of class c_i

ECOC-Rank nomenclature

Table 13
ECOC-Rank nomenclature

$c = \{c_1, c_2, .., c_n\}$ - Class set

Δ - Matrix composed by the Hamming

D_{v_p} - Depth of v_p

$D = \{d_1, .., d_n\}$ - Output ECOC vector

D^L - Likelihood output ECOC vector

d_i - Distance of a test sample to codeword i of the coding matrix

d - Distance

G - Taxonomy graph

L_{v_p} - Level of v_p

M_A - Adjacency matrix

M_D - Distance matrix

M_O - Ontology matrix

$M_P(v_p)$ - Position list of vertex v_p

$m_i = \{m_{i1}, .., m_{ik}\}$ - Centroid of class c_i

n - Number of classes

R - Altered ECOC ranking

v_i - Vertex of G

$W = \{W_{c_1}, .., W_{c_n}\}$ - Data of classes

References

[1] O. A., T. A., Modeling the shape of the scene: a holistic representation of the spatial envelope, in: International Journal in Computer Vision, Vol. 42, 2001, pp. 145–175.

[2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. V. Gool, A comparison of affine region detectors, Int. J. Comput. Vision 65 (1-2) (2005) 43–72.

[3] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[4] S. Belongie, J. Malik, J. Puzicha, Shape context: A new descriptor for shape matching and object recognition, in: NIPS, citeseer.ist.psu.edu/belongie00shape.html, 2000, pp. 831–837.

[5] C. Bishop, Graphical models, Pattern Recognition and Machine Learning, Springer (2006) 359–422.

[6] S. Kumar, T. Kanade, H. Schneiderman, J. Lafferty, A. Blake, Models for learning spatial interactions in natural images for context-based classification.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: ECCV, 2004, pp. 1–22.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: CVPR, 2007, pp. 1–8.

[9] O. Chum, J. Philbin, J. Sivic, A. Zisserman, Automatic query expansion with a generative feature model for object retrieval, in: ICCV, 2007, pp. 1–8.

[10] G. Carneiro, A. Jepson, Flexible spatial models for grouping local image features, in: CVPR, Vol. 2, 2004, pp. 747–754.

[11] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes 2 (1995) 263–282.

[12] R. Datta, D. Joshi, J. Li, J. Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, ACM Comput. Surv. 40 (2) (2008) 1–60.

[13] M. G. J. Eakins, Content-based image retrieval, Technical Report.

[14] P. L. Stanchev, D. Green, B. Dimitrov, High level color similarity retrieval, International Journal Information Theories and Applications 10 (2003) 283–287.

[15] V. Mezaris, I. Kompatsiaris, M. G. Strintzis, An ontology approach to object-based image retrieval, in: In Proc. IEEE Int. Conf. on Image Processing (ICIP03, 2003, pp. 511–514.

[16] L. G. J. Ren, Y. Shen, A novel image retrieval based on representative colors, in: Proceedings of the Image and Vision Computing, N.Z., 2003, pp. 102–107.

[17] C.-Y. Chiu, H.-C. Lin, S.-N. Yang, Texture retrieval with linguistic descriptions, in: PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Springer-Verlag, London, UK, 2001, pp. 308–315.

[18] I. C. I.K. Sethi, Mining association rules between low-level image features and high-level concepts, in: Proceedings of the SPIE Data Mining and Knowledge Discovery, Vol. 3, 2001, pp. 279–290.

[19] H. Feng, T.-S. Chua, A bootstrapping approach to annotating large image collection, in: MIR 2003: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, ACM, New York, NY, USA, 2003, pp. 55–62.

[20] T.-S. C. C.-H. L. R. Shi, H. Feng, An adaptive image content representation and segmentation approach to automatic image annotation, in: International Conference on Image and Video Retrieval (CIVR), 2004, pp. 545–554.

[21] D. S. C.P. Town, Content-based image retrieval using semantic visual categories, in: Society for Manufacturing Engineers, Technical Report MV01-211, 2001.

[22] F. M. J.-A. Vailaya, A., H. Zhang, Image classification for content-based indexing, IEEE Trans. Image Process 10 (1) 117–130.

[23] L. Z. H.-J. Z. B. Z. F. Jing, M. Li, Learning in region-based image retrieval, in: Proceedings of the International Conference on Image and Video Retrieval (CIVR2003), 2003, pp. 206–215.

[24] M. S. V. Mezaris, I. Kompatsiaris, An ontology approach to object-based image retrieval, in: Proceedings of the ICIP, Vol. 2, 2003, pp. 511–514.

[25] L. Zhang, F. Lin, B. Zhang, Support vector machine learning for image retrieval, in: ICIP (2), 2001, pp. 721–724.

[26] J. R. Smith, C. S. Li, Decoding image semantics using composite region templates, in: CBAIVL 98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries, IEEE Computer Society, Washington, DC, USA, 1998, p. 9.

[27] Y. P. Y. Zhuang, X. Liu, Apply semantic template to support content-based image retrieval, in: Proceedings of the SPIE, Storage and Retrieval for Media Databases, Vol. 3972, 1999, pp. 442–449.

[28] J. W. W. Chang, Metadata for multi-level content-based retrieval, in: Third IEEE Meta-Data Conference, 1999.

[29] D. Cai, X. He, Z. Li, W.-Y. Ma, J.-R. Wen, Hierarchical clustering of www image search results using visual, textual and link information, in: MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, ACM, New York, NY, USA, 2004, pp. 952–959.

[30] H. Feng, R. Shi, T.-S. Chua, A bootstrapping framework for annotating and retrieving www images, in: MULTIMEDIA 2004: Proceedings of the 12th annual ACM international conference on Multimedia, ACM, New York, NY, USA, 2004, pp. 960–967.

[31] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, Pattern Recogn. 40 (1) (2007) 262–282.

[32] L. G. Valiant, A theory of the learnable, Communications of the ACM (1984) 1134–1142.

[33] B. K. Natarajan, Machine Learning: A Theoretical Approach, Morgan Kaufmann, 1991.

[34] T. Hofmann, Probabilistic latent semantic analysis, In Proceedings of UAI '99 (1999) 289–296.

[35] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[36] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, CVPR (2005) 524–531.

[37] R. B. E. A. Z.-A. Sivic, J., W. Freeman, Discovering object categories in image collections, Technical Report A.I. Memo 005.

[38] M. B. Blaschko, Kernel methods in computer vision: Object localization, clustering, and taxonomy discovery, PhD dissertation.

[39] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2006, pp. 2169–2178.

[40] J. Niebles, L. Fei-Fei, A hierarchical model of shape and appearance for human action classification., in: Proceedings of IEEE Intern. Conf. in Computer Vision and Pattern Recognition(CVPR)., 2007.

[41] M. Pelikan, D. E. Goldberg, E. Cantú-Paz, Learning machines, in: McGraw-Hill, 1965.

[42] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, Vol. 1, 2002, pp. 113–141.

[43] T. Dietterich, E. Kong, Error-correcting output codes corrects bias and variance, in: P. of the 21th International Conference on Machine Learning (Ed.), S. Prieditis and S. Russell, 1995, pp. 313–321.

[44] F. Ricci, D. Aha, Error-correcting output codes for local learners, European conference on machine learning 1398 (1998) 280–291.

[45] T.Hastie, R.Tibshirani, Classification by pairwise grouping, NIPS 26 (1998) 451–471.

[46] W. Utschick, W. Weichselberger, Stochastic organization of output codes in multiclass learning problems, in: Neural Computation, Vol. 13, 2004, pp. 1065–1102.

[47] K. Crammer, Y. Singer, On the learnability and design of output codes for multi-class problems, in: Machine Learning, Vol. 47, 2002, pp. 201–233.

[48] O. Pujol, P. Radeva, J. Vitrià, Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes, in: Trans. on PAMI, Vol. 28, 2006, pp. 1001–1007.

[49] O. Pujol, S. Escalera, P. Radeva, Optimal node embedding in error correcting output codes, Vol. 14, 2008, pp. 713–725.

[50] S. Escalera, D. Tax, O. Pujol, P. Radeva, R. Duin, Sub-class problem-dependent design of error-correcting output codes, Vol. 30, 2008, pp. 1041–1054.

[51] T. Windeatt, R. Ghaderi, Coding and decoding for multi-class learning problems, in: Information Fusion, Vol. 4, 2003, pp. 11–21.

[52] A. Passerini, M. Pontil, P. Frasconi, New results on error correcting output codes of kernel machines, in: IEEE Transactions on Neural Networks, Vol. 15, 2004, pp. 45–54.

[53] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 99 (1).

[54] Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories.

[55] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Tech. Rep. 7694, California Institute of Technology (2007).
URL http://authors.library.caltech.edu/7694

[56] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, T. Kadir, L. V. Gool, A comparison of affine region detectors, IJCV 65 (1-2) (2005) 43–72.

[57] M. De Hoon, S. Imoto, J. Nolan, S. Miyano, Open source clustering software, Bioinformatics 20 (9) (2004) 1453–1454.

[58] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, in: Transactions in Pattern Analysis and Machine I Intelligence, IEEE computer Society Digital Library. IEEE Computer Society, 2008.
URL <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.266>

[59] Microsoft research cambridge object recognition image database.
URL http://research.microsoft.com/en-us/downloads/b94de342-60dc-45d0-830b-9f6eff91b301/default.aspx

[60] J. Demsar, Statistical comparisons of classifiers over multiple data sets, JMLR 7 (2006) 1–30.