Efficient Classification of Digital Images based on Pattern-features

Angelo Furfaro[1] and Simona E. Rombo[2]
[1]*DIMES, University of Calabria, Italy*
[2]*DMI, University of Palermo, Italy*

Keywords: Image Classification, Bidimesional Pattern Extraction, Irredundant Pattern.

Abstract: Selecting a suitable set of features, which is able to represent the data to be processed while retaining the relevant distinctive information, is one of the most important issues in classification problems. While different features can be extracted from the raw data, only few of them are actually relevant and effective for the classification process. Since relevant features are often unknown a priori, many candidate features are usually introduced. This degrades both the speed and the predictive accuracy of the classifier due to the presence of redundancy in the feature candidate set. We propose a class of features for image classification based on the notion of *irredundant bidimensional pair-patterns*, and we present an algorithm for image classification based on their extraction. The devised technique scales well on parallel multi-core architectures, as witnessed by the experimental results that have been obtained exploiting a benchmark image dataset.

1 INTRODUCTION

Image classification is an active research field and various classification techniques, based on supervised and on unsupervised techniques or on a mix of them, appeared in the literature. Surveys on the topic can be found in (Bosch et al., 2007; Lu and Weng, 2007; Nanni et al., 2012).

Traditional approaches use low-level image features, such as color or texture histograms. Other techniques rely on intermediate representations, made of local information extracted from *interesting* image patches referred to as *keypoints* (Bosch et al., 2007). Image keypoints are automatically detected using various techniques, and then represented by means of suitable descriptors. Keypoints are usually clustered based on their *similarity*, and each cluster is interpreted as a "*visual word*", which summarizes the local information pattern shared among the belonging keypoints (Yang et al., 2007). The set of all the visual words constitutes the *visual vocabulary* or *codebook*. For classification purposes, an image is then represented as a histogram of its local features, which is analogous to the *bag-of-words* model for text documents. Examples of commonly exploited keypoint detectors are: Difference of Gaussian (DoG) (Lowe, 2001), Sample Edge Operator (Berg et al., 2005), Kadir-Brady (Kadir and Brady, 2001). Feature descriptors are often based on SIFT (Scale Invariant Feature Transform) (Lowe, 1999).

Like in many other classification problems, the relevant features to be employed are not known a priori. Therefore, various candidate features can be introduced, many of which are either partially or completely irrelevant/redundant to the target concept (Dash and Liu, 1997). A relevant feature is neither irrelevant nor redundant to the target concept; an irrelevant feature does not affect the target concept in any way, and a redundant feature does not add anything new to the target concept (John et al., 1994). In this work, we analyze a special kind of features for image classification, based on the concept of *bidimensional irredundant pair-patterns*.

Roughly speaking, bidimensional irredundant pair-patterns represent approximate repetitions between pairs of images in an input training set. In more detail, given two images I_1 and I_2, one can superimpose each rectangular sub-portion of I_1 with each rectangular sub-portion of I_2, keeping only those pieces of the two images that match. When all the possible repeated portions between I_1 and I_2 are considered, also taking into account sub-regions that are *similar* but not identical, the number of such features can grow exponentially with the size of I_1 and I_2, and many of the extracted patterns are irrelevant and/or redundant.

Suitable notions of maximality and irredundancy have been introduced for digital images (Apostolico

Furfaro, A. and Rombo, S.
Efficient Classification of Digital Images based on Pattern-features.
DOI: 10.5220/0006955500930099
In *Proceedings of the 5th International Conference on Physiological Computing Systems (PhyCS 2018)*, pages 93-99
ISBN: 978-989-758-329-2

et al., 2008) and successfully exploited for both image compression in (Amelio et al., 2011) and image classification (Furfaro et al., 2013), showing to be useful in encoding the relevant information from input images, by reducing their representation. However, all such approaches work by extracting repetitions from a single input image, whereas for classification purposes the goal is that of singling out useful repetitions able to characterize *a set* of images, representing a class.

A first contribution of the research work presented here is that of extending the notions of maximality and irredundancy introduced in (Apostolico et al., 2008) for pairs of images. In particular, the main idea is that of eliminating all the extracted pair-patterns that are *redundant* with respect to the other ones in the set of candidates. The notion of redundancy here is related to the occurrence of patterns in the images of a class, since if several patterns occur on the same class, then only the most informative ones will be kept into account and they will be used as features for the classification process.

As a second contribution, we propose an image classification approach based on the extraction of such bidimensional pair-patterns. In particular, given an input training set of images already classified, the irredundant pair-patterns are extracted and used to build a suitable codebook. After feature selection, image classification is then performed based on the K-Nearest Neighbour approach. The algorithm has been implemented according to the principles of parallel computing, in order to use efficiently the modern multi-core architectures.

We tested the proposed approach on a benchmark image dataset (ZuBud (Shao et al., 2003)) and the preliminary results show that it is able to reach high values of accuracy.

The paper is organized as follows. Section 2 illustrates some preliminary notions, while in Section 3 the proposed classification algorithm is described in detail. In Section 4 we show some preliminary results we obtained on real datasets, also comparing them with those returned by other methods proposed in the literature. Finally, Section 5 draws some conclusive remarks.

2 PRELIMINARY NOTIONS

A digitized image can be represented as a rectangular array I of $N = m \times n$ *pixels*, where each pixel i_{ij} is a *character* (typically, encoding an integer) over an alphabet Σ, corresponding to the set of colours occurring into I (see Figure 1).

(a)

$$I_{[m,n]} = \left\| \begin{array}{cccc} i_{11} & i_{12} & \dots & i_{1n} \\ i_{21} & i_{22} & \dots & i_{2n} \\ . & & & \\ . & & & \\ . & & & \\ i_{m1} & i_{m2} & \dots & i_{mn} \end{array} \right\|$$

(b)

Figure 1: (a) A digitized image (Lena). (b) The corresponding image I over the alphabet of colours $\Sigma = \{c_1, c_2, ..., c_k\}$ (each element i_{ij} represents a pixel of Lena with a specific colour $c_l \in \Sigma$).

We are interested in finding a compact descriptor for a set of images $S_I = \{I_1, I_2, \dots, I_l\}$, able to capture the common features of the images in the set. To this aim, we search for the repetitive content among $I_1, I_2, \dots I_l$ under the assumption that if a small block of such images is sufficently repeated in S_I, then it represents a feature that is characteristic for the set. Such repeated blocks can be not necessarily identical, but somewhat identical unless some pixel, due to different shades or lightness in the original pictures. Thus, in addition to the *solid* characters from Σ, we also deal with a special *don't care* character, denoted by '$*$', that is a wildcard matching any character in $\Sigma \cup \{*\}$. Don't cares are useful in order to take into account *approximate* repetitions. We say that an image P defined on $\Sigma \cup \{*\}$ *occurs* in a larger image I if $P[i, j] = I_i[h+i-1, k+j-1]$, for some position $[h, k]$ in I and for all the positions $[i, j]$ in P.

Given a pair of images I_i and I_j, both of size $m \times n$ and such that $I_a, I_b \in S_I$, a *bidimensional pair-pattern* is an extended image P of size $m' \times n'$ such that:

1. $m' \leq m$ and $n' \leq n$;
2. there is at least one solid character adjacent to each edge of P;
3. there exist positions $[h_a, k_a]$ in I_a and $[h_b, k_b]$ in I_b such that P occurs in both I_a and I_b.

The notion of *bidimensional pair-pattern* extends that of 2D motif already introduced in (Apostolico et al., 2008) and applied to digital images in (Amelio et al., 2011; Furfaro et al., 2017). The main difference here is that the bidimensional pair-pattern (referred to simply as *pattern* in the following) is extracted from pairs of images, whereas the 2D motif represents repetitions occurring in the same image.

When approximate occurrences are taken into account, and patterns with don't cares are thus considered, the number of all the possible patterns that one can extract from an input image can grow drastically, often becoming exponential in the size of the input image. In order to limit such a growth, suitable notions of *maximality* and *irredundancy* have been proposed for bidimensional patterns (Apostolico et al., 2008; Rombo, 2009; Rombo, 2012). Since such notions concern patterns extracted from only one image, we extend them here for pairs of images. To this aim, given a pattern P occuring in some of the images in S_I (we say that P is covered on S_I), its *occurrence list* $\mathcal{L}_P = \{1, 2, \ldots, k\}$ is made of the indices of those images $\{I_1, I_2, \ldots, I_k\}$ in S_I where P occurs.

Maximal Pattern. Let $\mathcal{P} = P_1, P_2, \ldots, P_f$ be a set of patterns covered on S_I, and let $\mathcal{L}_{P_1}, \mathcal{L}_{P_2}, \ldots, \mathcal{L}_{P_f}$ be their occurrence lists, respectively. A pattern P_i is *maximal* in $\mathcal{P}$ if and only if there exists no pattern $P_j \in \mathcal{P}$, $j \neq i$, such that P_i occurs in P_j and $|\mathcal{L}_{P_i}| = |\mathcal{L}_{P_j}|$. In other words, P_i cannot be substituted by P_j without loosing some of the P_i occurrences in S_I.

Irredundant Pattern. A pattern P_i that is maximal in $\mathcal{P}$, having occurrence list $\mathcal{L}_P$ in S_I, is also *irredundant* in $\mathcal{P}$ if there not exist any maximal pattern $P_j \in \mathcal{P}$, $j = 1, \ldots, h$, such that P_i occurs in P_j and $\mathcal{L}_P = \mathcal{L}_{P_1} \cup \mathcal{L}_{P_2} \cup \ldots \cup \mathcal{L}_{P_h}$, up to some offsets.

3 THE CLASSIFICATION ALGORITHM

This section describes the proposed image classification procedure whose pseudo-code is reported in Algorithm 1.

The algorithm takes in input an image dataset $S_I = \{I_1, I_2, \ldots, I_l\}$, which is a priori partitioned into h classes $C_1, C_2, \ldots, C_h$, and a test image I for which the algorithm will predict the membership class. For each image in the input dataset, the set of irredundant bidimensional pair-pattern motifs is extracted by overlapping it with all the other images in its class, as described by the extraction procedure (PATTERNEXTRACTION) reported in Algorithm 2. The overall set made of the extracted motifs constitutes the *codebook* $\mathcal{D}$. Then, for each image I_j, we build an array w_j, having as many entries as the codebook size. w_j is the histogram of the occurrences of the codebook patterns into I_j, i.e. $w_j[i]$ is the number of occurrences of the motif m_i in the image I_j.

The REMOVEBORDER procedure, invoked at line 12 of Algorithm 2, removes from the input pattern those border zones made up of only don't cares.

In order to classify a test image I, its histogram w with respect to the codebook pattern is computed, as for the training set images.

The final step, which actually outputs the classification label for I, uses the well-known classification algorithm k-Nearest Neighbour ($k-$NN). This step is applied by calculating all the distances between each array w_j and the array w obtained from the image that has to be classified. The output class is that having the larger consensus among the k images that score the lowest values for the distances from I.

Algorithm 1: Classification algorithm.

Require: set S_I of l images of size $m \times n$ images partitioned in h classes $C_1, C_2, \ldots, C_h$; a test image I; an integer k;
Ensure: the label x of the class predicted for I
 /* *training phase* */
 $\mathcal{D}' = \emptyset$
 for each class C_j in $C_1, C_2, \ldots, C_h$
  B_j =PATTERNEXTRACTION(C_j)
  $\mathcal{D} = \mathcal{D} \cup B_j$
 end for

 for each image I_j in S_I
  let w_j be an empty array of $|\mathcal{D}|$ integer values
  for each motif m_i in $\mathcal{D}$
   $w_j[i]$ =COMPUTEOCCURRENCES(m_i,I_j)
  end for
 end for
 /* *testing phase* */
 let w be an empty array of $|\mathcal{D}|$ integer values
 for each motif m_k in $\mathcal{D}$
  $w[k]$ = COMPUTEOCCURRENCES(m_k,I)
 end for

 let E be an empty array of l real values
 for each image I_j in S_I
  $E[j] = dist(w_j, w)$
 end for
 sort E in increasing order

 let S be the set of the first k elements in E
 return the label x of the most popular class in S

Algorithm 2: Extraction of irredundant patterns.

Require: set S_I of l images of size $m \times n$
Ensure: set I_{S_I} of irredundant patterns of S_I

$I_{S_I}' = \emptyset$
for each image I_a in S_I
  for each image $I_b \neq I_a$ in S_I
    for each position $[i_a, j_a]$ in I_a
      for each position $[i_b, j_b]$ in I_b
        extract a pattern P_{ab} such that
        if $I_a[i_a, j_a] \neq I_b[i_b, j_b]$ **then**
          $P_{ab}[i, j] = *$
        else
          $P_{ab}[i, j] = I_a[i_a, j_a]$
        end if
        $P_{ab} =$ RemoveBorder(P_{ab})
        $I_{S_I}' = I_{S_I}' \cup P_{ab}$
      end for
    end for
  end for
end for

for each pattern P in I_{S_I}'
  compute the occurrence list of P in S_I
end for
$I_{S_I} = I'_{S_I} \setminus \{$all the redundant patterns in $I'_{S_I}\}$

3.1 Distance Notions

An important aspect in classification algorithms is the choice of the distance notion employed to measure the similarity among objects. This choice may have in some cases a great impact on the algorithm performances. As discussed in the next section, the proposed classification technique has been tested with different definitions of the distance functions. Some of them are *normalized* versions of classical distance functions which take into account also the standard deviation of the training samples for each dimensions. In particular, we used: the Euclidean distance and its normalized version, the Hamming distance, and a normalized Manhattan distance.

The formal definitions of these functions are reported in Table 1, where $s_k^2 = \frac{1}{l-1}\sum_{i=1}^{l}(w_i[k] - \overline{w}[k])^2$, i.e. the variance of the k^{th} dimension, $\overline{w}[k] = \frac{1}{l}\sum_{i=1}^{l} w_i[k]$, i.e. the mean value for the k^{th} dimension, and $\delta(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases}$

3.2 Algorithm Parallelization

The algorithm has been implemented according to the principles of parallel computing. Some details about the implementation are provided below.

Table 1: Used distances.

Euclidean	$d_2(w_i, w_j) = \sqrt{\sum_{k=1}^{n}(w_i[k] - w_i[k])^2}$
Normalized Euclidean	$d_{2_s}(w_i, w_j) = \sqrt{\sum_{k=1}^{n}\frac{(w_i[k]-w_i[k])^2}{s_k^2}}$
Hamming	$d_H(w_i, w_j) = \sum_{k=1}^{n}\delta(1 - \delta(w_i[k], 0), \delta(w_j[k], 0))$
Normalized Manhattan	$d_{M_s}(w_i, w_j) = \sum_{k=1}^{n}\frac{\lvert w_i[k]-w_j[k]\rvert}{s_k}$

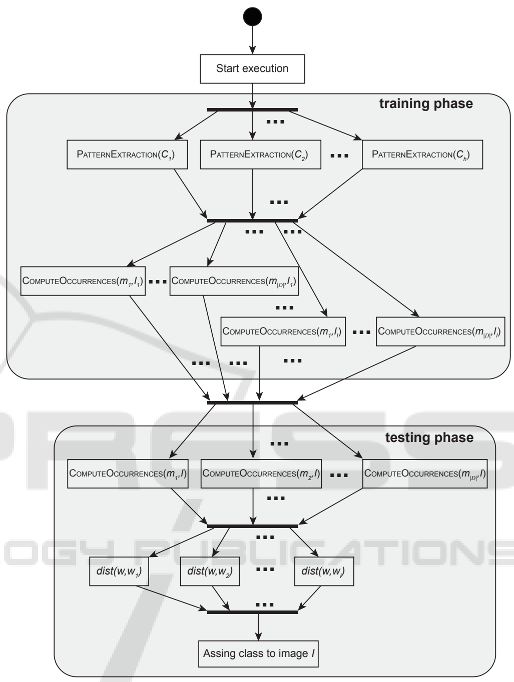

Figure 2: Parallel execution of Algorithm 1.

Both training and testing phases have been parallelized. In particular, a pool of $n_{core} + 1$ working threads has been created, where n is the number of available cores on the underlying architecture. Each iteration of the *for* loops can be transformed into a separate task, which can be submitted for execution and accomplished by a worker thread. In particular, regarding the *for* loop at line 3, for each class we generate a single task consisting in the execution of the PatternExtraction procedure on the relevant test images; similar considerations hold regarding the *for* loop of line 6, where a single task is the execution of the ComputeOccurrences procedure on each pair made of a pattern and training image.

As for the testing phase, the *for* loop of line 13 is parallelized like it has been done for that of line 6. Finally, the computation of the distances among

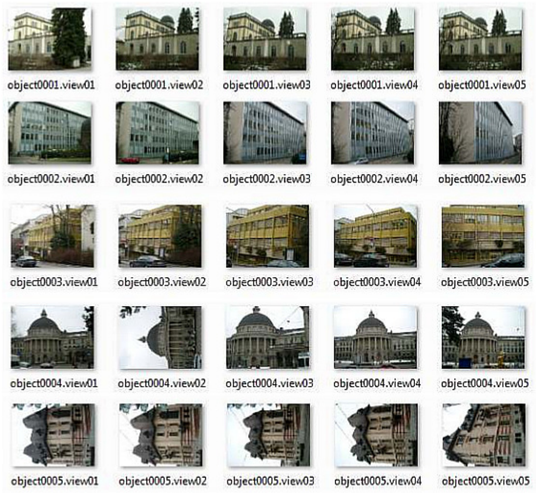

Figure 3: Images from the ZuBud dataset.

the histogram of a test image and those of the training images (line 17) is performed in parallel as well.

4 EXPERIMENTAL RESULTS

This section describes the experimental campaign that has been performed in order to test and analyze the results of the proposed image classification algorithm. The algorithm has been implemented in Java. A standard benchmark dataset, the ZuBuD dataset, which is described by the next subsection, has been used. The classifier performances have been measured from two points of view: in terms of its accuracy and in terms of scalability of its parallel execution. In particular, the experiments have been executed on a machine with an i7 processor with 2.00 Ghz per core, and 8GB of RAM.

4.1 The ZuBuD Dataset

The ZuBuD (Zurich Building Image Database) (Shao et al., 2003) is a famous collection of images, often used to test classification algorithms. The dataset is freely available, and it is considered a standard in the literature to test this kind of classifiers. It consists of a collection of photos that have been done to 201 buildings of Zurich. For each building, the training set contains five pictures, each of which represents the building from a different point of view. Each image in the training has a size of 640×480 pixels: in total, there are 1005 images in the training set. Then, we have the test set, that consists of 115 photos of buildings. These images have a size of 320×240 pixels, and represent some of the buildings contained into the training set in various angulations and various points

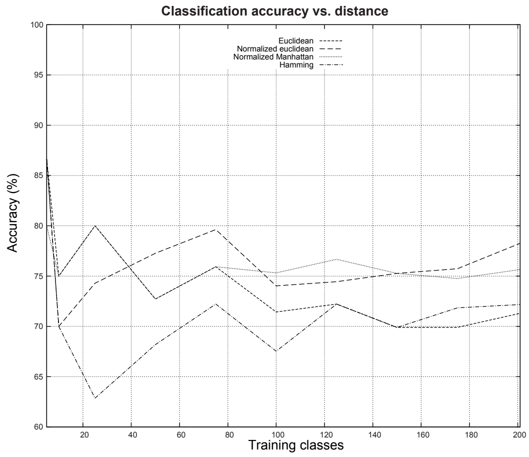

Figure 4: Accuracy.

Table 2: Accuracy vs. distance (201 classes).

Distance	Accuracy
Euclidean	71.30%
Normalized euclidean	78.26%
Normalized Manhattan	75.65%
Hamming	72.17%

of view. In general, the photos have numerous characteristics of heterogeneity: in fact, they have been taken with two cameras and in different periods time of the year. The only condition consistent between all photos relating to a same building are the conditions of illumination, which remain almost identical.

4.2 Accuracy Analysis

The algorithm has been tested with an increasing number of training classes and with various kinds of distances. This allows us to study how the accuracy evolves with the increasing of the number images and, consequently, of the extracted patterns. Figure 4 shows how the accuracy tends to decrease with the increase of the training set size till 100 classes, then it increases and settles down to about 70%.

As it can be seen from Table 2, the best score has been achieved with the Normalized Euclidean distance, which with 201 classes scored 78.26% of accuracy, i.e 90 test images out of 115 were correctly classified.

4.3 Scalability

In this kind of algorithms performance is a crucial aspect, especially with big datasets. As explained in Section 3.2, the classifier was parallelized both in the training phase and in the testing phase: this allows us to use efficiently the modern multicore architectures.

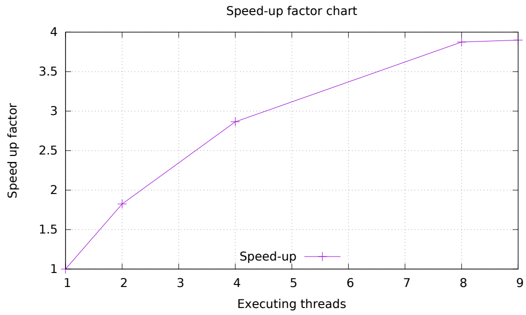

Figure 5: Speed up.

We also did a work of optimization on all of the data structures used during the execution. However, there is no guarantee that all of the real world machines on which the algorithm will run have at their disposal the same number of physical cores. Accordingly, it should be noted as the performance evolve in relation to varying the number of threads that are running simultaneously on the machine in question: this aspect is summarized in the speed-up plot of Figure 5. The results are related to an experiment conducted with 10 training classes.

In this speed-up factor chart we can see how the increase of number of threads, running simultaneously on the machine, have a good effect on the execution time. Clearly, beyond a certain number, i.e., that of the physical processors, there are no further considerable improvements. The machine where the experiments were performed is equipped with 8 cores and by using a pool of 8 threads we reached a factor speed-up of nearly 4.

5 CONCLUDING REMARKS AND DISCUSSION

This paper presented an image classification technique based on bidimensional motifs. The analysis of the experimental results, suggests that this technique is effective and obtains good performances both in terms of accuracy and of scalability. We must not however lose sight of the enormous complexity that the classification problem presents inherently, especially as it regards the search space that we can explore in order to improve the accuracy of classification. In fact, at present, it is possible to customize the classifier in a considerable number of parameters, each of which can assume a range of values potentially very high. In some cases, the wrong choice of these parameters can also degenerate performance at an extent as to make the execution times unacceptable. In this sense, research is responsible for identifying situations where the instrument maximizes percentage of correctly classified images with reasonable performances. As regards the implementation, the actual classifier version runs in a maximally efficient on a single machine with multi-core architecture: one of the steps for the future might be to offer an implementation that can deliver computing in cluster machines or distributed environments, for example in cloud scenarios. In this way, we may improve timing performance and be able to experience the behavior of the classifier in further complex scenarios. To be taken into account is also the positive influence that has had the introduction of distance normalization: this aspect has led a substantial improvement in the accuracy of the algorithm, and is indeed a solid starting point for future research. It might also be interesting to try to replace the final classification technique, currently *k*-NN, with other types of classifiers coming for example from the pool of statistical ones. Another open question is to see how non-exact chromatic matching is influential on the technique. Moreover, suitable invariants, such as rotations and/or translations, can be considered in the 2D basis generation, by extending the proposed approach to this aim. Finally, a comparison with more recent techniques for image classification (Chan et al., 2015; Kumar et al., 2017; Maggiori et al., 2017) will be object of our investigation.

ACKNOWLEDGEMENTS

The authors are grateful to Michele Bombardieri for his help in the implementation of a preliminary version of the software presented here. Moreover, their research has been partially supported by a project financed by INDAM titled “Elaborazione ed analisi di Big Data modellati come grafi in vari contesti applicativi”, under the program GNCS 2018.

REFERENCES

Amelio, A., Apostolico, A., and Rombo, S. E. (2011). Image compression by 2D motif basis. In *Data Compression Conference (DCC 2011)*, pages 153–162.

Apostolico, A., Parida, L., and Rombo, S. E. (2008). Motif patterns in 2D. *Theoretical Computer Science*, 390(1):40–55.

Berg, A., Berg, T., and Malik, J. (2005). Shape matching and object recognition using low distortion correspondences. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 26–33.

Bosch, A., Muñoz, X., and Martí, R. (2007). Review: Which is the best way to organize/classify images by content? *Image Vision Comput.*, 25(6):778–791.

Chan, T., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2015). Pcanet: A simple deep learning baseline for image classification? *IEEE Trans. Image Processing*, 24(12):5017–5032.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1:131–156.

Furfaro, A., Groccia, M. C., and Rombo, S. E. (2013). Image classification based on 2d feature motifs. In *Flexible Query Answering Systems*, pages 340–351. Springer.

Furfaro, A., Groccia, M. C., and Rombo, S. E. (2017). 2D motif basis applied to the classification of digital images. *Computer Journal*, 60(7):1096–1109.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International*, pages 121–129.

Kadir, T. and Brady, M. (2001). Saliency, scale and image description. *Int. J. Comput. Vision*, 45(2):83–105.

Kumar, A., Kim, J., Lyndon, D., Fulham, M. J., and Feng, D. D. (2017). An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE J. Biomedical and Health Informatics*, 21(1):31–40.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proc. of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lowe, D. (2001). Local feature view clustering for 3D object recognition. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 682–688.

Lu, D. and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870.

Maggiori, E., Tarabalka, Y., Charpiat, G., and Alliez, P. (2017). Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geoscience and Remote Sensing*, 55(2):645–657.

Nanni, L., Lumini, A., and Brahnam, S. (2012). Survey on LBP based texture descriptors for image classification. *Expert Syst. Appl.*, 39(3):3634–3641.

Rombo, S. E. (2009). Optimal extraction of motif patterns in 2D. *Information Processing Letters*, 109(17):1015–1020.

Rombo, S. E. (2012). Extracting string motif bases for quorum higher than two. *Theoretical Computer Science*, 460:94–103.

Shao, H., Svoboda, T., and Gool, L. V. (2003). Zubud - Zurich building database for image based recognition. Technical Report TR-260, Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland.

Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, MIR '07, pages 197–206.