# Bag-of-Features Model
## Application to Medical Image Classification

Karlijn Zaanen
Supervision: Mitko Veta, MSc., Utrecht University

October 11, 2013

**Abstract**

The aim of this study is to investigate the Bag-of-Features (BoF) model and its application to medical image classification. With this model, images, or parts of them, are represented as an orderless collection of local image descriptors. A visual codebook is learned from a training set of local image descriptors, usually by performing vector quantization by clustering. Each image is then represented by its distribution of visual words from the codebook. To achieve image classification, a classifier such as a Support Vector Machine (SVM) is used with the obtained image representation as the feature vector. This paper presents a review of the literature on BoF methods and compares the most important implementation choices that have been suggested. In addition, the application of the method to medical image classification is discussed.

# 1 Introduction

Automatic classification of (medical) images is a very challenging problem. Over the last decade, the Bag-of-Features (BoF) method has become popular for both texture and object classification. The approach treats an image as an orderless collection of local image descriptors. The first step of the BoF method is feature detection. Points in the image are detected using interest point detectors, or alternatively by sampling from a regular grid or randomly. Feature descriptors are computed over small support regions, so-called 'image patches', around each detected point. In the third step, vector

quantization is used to define a relatively small number of generic local image descriptors which form the visual words of the codebook from an initial pool of descriptors extracted from a training set. Each image is represented by its distribution of visual word counts. The final step is classification using the obtained representation as the feature vector and applying a classifier such as a Support Vector Machine (SVM).
A texture is made up of a repetition of basic primitive elements called textons. When the BoF method is used for texture classification, the visual words in the codebook correspond to the textons. It is not surprising that an orderless approach such as the BoF method is successful for texture classification; stochastic textures are characterized by the identity of their textons, not by their spatial arrangement. Interestingly, the BoF method has also proved effective for object classification and natural scene categorization problems. Especially when large viewpoint changes, clutter and occlusions are present in the images, the BoF method offers advantages over methods that compute global descriptors or include spatial relationships between features [1], [8], [15].
The aim of this study is to investigate the BoF method and its application to medical image classification. Research in this area has been done on a variety of imaging modalities, including computed tomography (CT) lung scans [2], histopathological images [9] and endoscopic colorectal images [10]. In medical images, abnormalities are typically characterized by texture changes. Therefore, the main focus is on texture classification, object classification being less common in medical image classification. This paper presents a review of the literature on the BoF method for texture classification and is structured as follows. In Section 2, the most important implementation choices that have been suggested in literature are compared. Application of the method to medical image classification is discussed in Section 3. Section 4 concludes with a final discussion and directions for future research.

# 2 Components of the Representation

In the following, each component of the BoF method is discussed in detail. For each of the five steps; feature detection (2.1), feature description (2.2), codebook formation (2.3), image representation (2.4) and classification (2.5), the main methods found in literature are discussed and comparisons are made. Figure 1 gives an overview of this section.

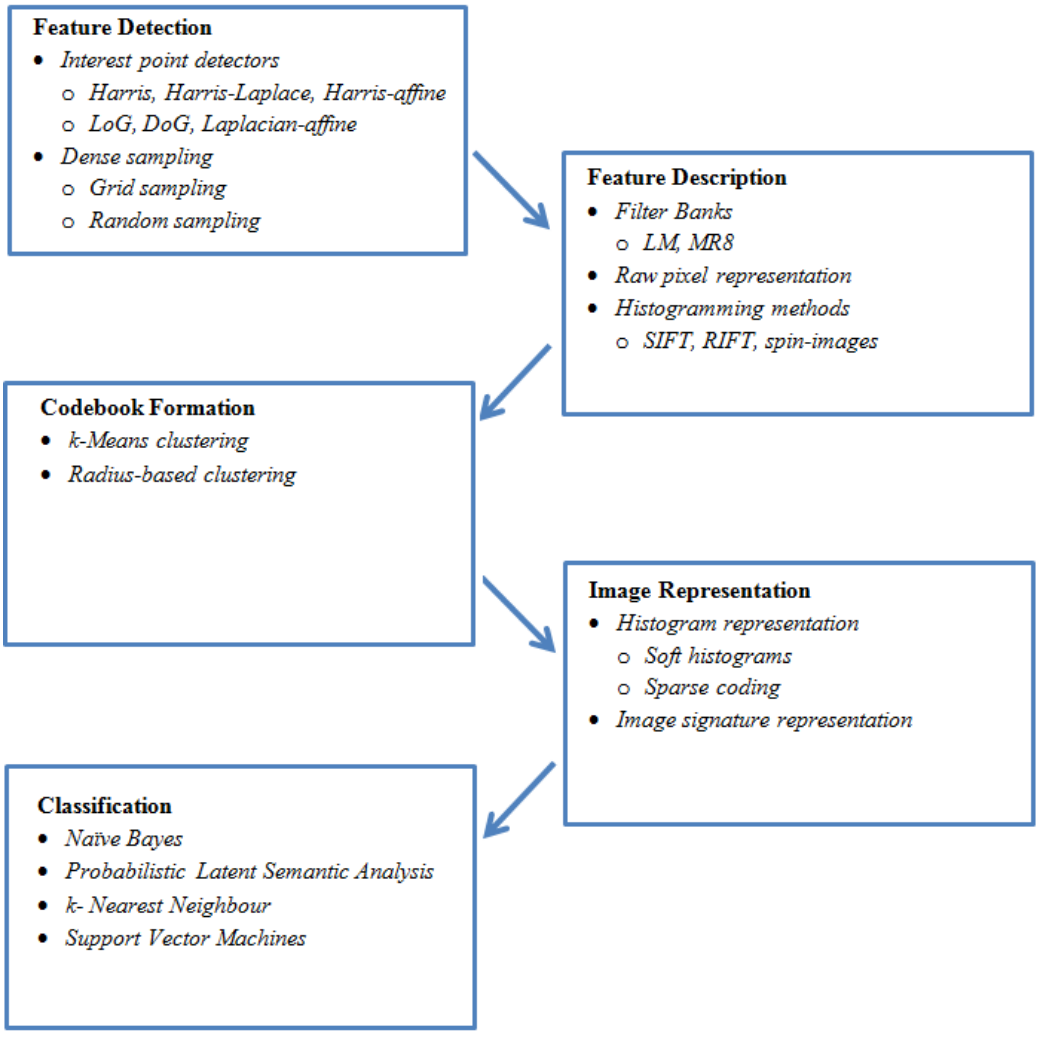



Figure 1: Overview of the BoF method. For each of the five components, the main methods found in literature are listed in the corresponding box.

## 2.1 Feature Detection

Feature detection is the selection of image locations around which local image descriptors are computed. Early examples of the BoF method are the texton approaches in which features for each pixel are computed, resulting in a dense representation [6]. Another way of obtaining a dense representation is by sampling from a regular grid [10] or randomly [8]. More recent BoF approaches use local interest point detectors to extract characteristic structures in the image, resulting in a sparse set of image patches.
A frequently used interest point detector is the Harris detector, responding to corners in the image. The Harris detector is a computationally efficient and robust operator, but it is only rotation-invariant. The scale-invariant extension is called Harris-Laplace and though it is computationally more complex, it is a much used tool for detecting a large number of patches [11]. The Harris-affine detector is the affine-invariant extension. It is an iterative algorithm that initializes with the Harris-Laplace circular patches. It adapts the original support region in shape and size to compensate for changes in surface orientation and scale by outputting regions in the shape of an ellipse. As many perspective transformations can be locally approximated by affine transformations, affine-invariant detectors could be useful for classification under viewpoint changes.
Another popular interest point detector is the Laplacian-of-Gaussian (LoG), a rotation- and scale-invariant detector, responding to blob-like regions in the image. The LoG can be approximated by the computationally more efficient Difference-of-Gaussians (DoG) [7]. In a similar manner as the Harris detector, an affine extension to the LoG can be made. We will call this the Laplacian-affine detector.
Also, a combination of interest point detectors can be used. Blob detectors, extracting homogeneous areas, can be viewed as complementary to corner detectors, extracting regions of high variability in intensity. As a result, they are often used together to provide more patches and a better coverage of the image.

Advantages of using interest point detectors are that they provide robustness to non-homogeneity of a texture and can reduce the computational cost by selecting fewer but more characteristic points [5]. For scale- or affine-invariant detectors, since the support region is adapted to the changes in scale or surface orientation, an intrinsically invariant representation of the

image patch is obtained. In literature, there is no clear-cut answer to the question whether such intrinsic representation-level invariance is beneficial to classification. Zhang et al. [15] conclude that affine-invariant detectors do not improve classification even for datasets with significant viewpoint changes. They note that detectors with a high degree of invariance are less stable and that the affine normalization process might result in a loss of discriminative information. If the transformations between images are expected to be small, the more robust but less invariant detectors are advantageous. Tuytelaars and Mikolajczyk [11] argue that for classification, as variability within the class dominates variability due to viewpoint changes, affine invariance tends to bring little improvement. However, Lazebnik et al. [5] conclude in their work that especially for datasets where the lack of invariance cannot be compensated by storing multiple prototypes of each texture, representation-level invariance is necessary. Evaluations and comparisons of detectors often use retrieval performance rather than classification performance. Retrieval performance is an indicator of how well each texture class can be modelled by one individual sample. In this case invariance is more important, and evaluations based on retrieval performance thus tend to favour interest point detectors over the dense sampling methods. It can be concluded that the need for invariance depends on the dataset used; only the level of invariance that cannot be compensated by storing multiple prototypes should be added to the detector. In the medical applications discussed in [2], [9], [10], indeed only the dense methods and the DoG sampling, which are respectively non-invariant and scale-invariant representations, are investigated.

The most influencing factor is the number of patches extracted [8], [11]. A large number of patches and a good coverage of the image is crucial for classification purposes. This argument favours the Laplacian detector over the Harris detector, since the Laplacian detector typically selects a denser set of regions than the Harris detector [5]. The dense approaches using sampling on a grid or random sampling are also favoured by this argument as these methods can produce a practically unlimited number of patches. Also, dense approaches guarantee a good coverage of the image independent of image content, while interest point detectors can fail for homogeneous, high-frequency textures [15]. Jurie et al. [4] and Nowak et al. [8] also confirm that, for the task of classification, using sparse interest point based patches often results in a significant loss of discriminant information compared to densely sampled patches.

## 2.2 Feature Description

Feature description is the translation of information in an image patch into a feature vector. The traditional method of feature description for texture classification uses filter banks. Filter banks are usually combined with a dense sampling approach, in which a filter response is obtained at every pixel location. Leung and Malik [6] were the first to propose clustering of the filter responses to obtain a small set of prototype response vectors which form the textons or visual words of a codebook. The filter bank of Leung and Malik (LM) consists of 48 filters and is not rotation-invariant. It is a mix of edge, bar and spot filters at multiple scales and orientations.
Varma and Zisserman [12] achieved rotation invariance by storing only the maximum response over the different orientations for a given filter type and scale. There are different variants of the maximum response (MR) set of Varma and Zisserman, the most used being the MR8 filter bank. They evaluated the performance of the mentioned filter banks on the CUReT database, and found that the rotation-invariant, multi-scale MR8 outperformed all other filter banks.

Filter banks have become less dominant in the field of texture classification. Among the alternatives to filter banks are methods based on raw pixel intensities [2], [8], [9], [13]. Varma and Zisserman [13] proposed a classifier that uses the raw pixel values of fixed size square image patches as feature vectors and called it the Joint classifier. This method considers a $N \times N$ image patch around each pixel. The description of the image patch is simply a vector of length $N^2$ containing the intensity values of each pixel in the image patch. Using raw pixel values with a fixed size patch results in features which are not invariant to scale and rotation. Rotation invariance can be included by aligning all image patches to the dominant orientation of the patch and using a circular patch instead of a square one.

Varma and Zisserman [13] conducted a thorough comparison between filter bank methods and raw pixel intensity based methods. They conclude that raw pixel representations outperform filter banks with the same support. Originally, filter banks were used because they were believed to increase the signal to noise ratio, extract useful features such as edges or bars at multiple orientations and scales, and achieve dimensionality reduction. A disadvantage of using filter banks is the large support they require which

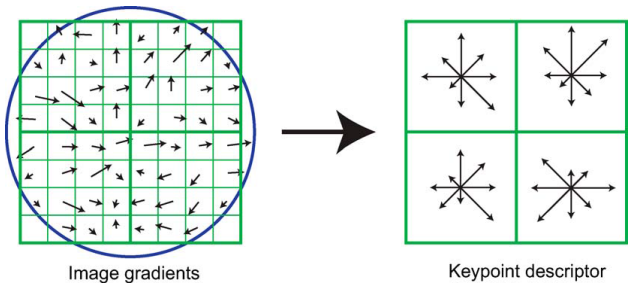



Figure 2: SIFT descriptor computation. First the gradient orientation and magnitude are computed at each sample point in the image patch. A gradient orientation histogram is formed for each of the $2 \times 2$ subregions. The length of each arrow corresponds to the sum of gradient magnitudes in that orientation bin. The SIFT descriptor is formed by concatenating the gradient orientation histograms of each subregion. The figure is reproduced from [7].

results in fewer samples being drawn from each texture, hindering reliable clustering. Also, the blurring in many filters results in a loss of fine detail which can be critical to classification performance. While the MR8 filter bank has achieved better results than other filter banks, the Joint classifier outperforms the MR8 classifier significantly on multiple texture databases [12], [13]. However, raw pixel intensity based methods lack geometric invariance.

To achieve greater geometric invariance, Lowe [7] introduced the Scale Invariant Feature Transform (SIFT). In Figure 2, a schematic view of the SIFT descriptor computation is given. SIFT features are computed from scale-invariant interest points, found with for example the DoG detector. Each detected image patch is first oriented to the dominant gradient direction. The oriented patch is subdivided into a number of regions and a gradient orientation histogram is computed for each subregion. The descriptor vector for the image patch is formed by concatenating the gradient orientation histograms of each subregion. Usually, 8 orientation bins and $4 \times 4$ subregions in a patch are used, resulting in a 128-dimensional feature vector [7], [10].
The Rotation Invariant Feature Transform (RIFT) descriptor is a variant of the SIFT descriptor that is rotation-invariant without having the need to rotate the image patch [5]. Instead of rotating the patch in the dominant gradient direction, a rotation-invariant descriptor is computed directly. The patch is divided into concentric rings of equal width. A gradient orientation

histogram is built within each ring, with the orientation relative to the direction pointing away from the center to achieve rotation invariance. The number of rings and number of bins for gradient orientation are parameters of the descriptor.
The spin-image as proposed by Lazebnik et al. [5] is a similar descriptor, based on raw intensity information. A spin image is a two-dimensional histogram with entries $(d, i)$, $d$ being the distance to the center and $i$ the intensity bin, thus achieving rotation invariance.

In more recent literature, SIFT descriptors are displayed as the standard choice [10]. Lazebnik et al. note that dividing the image patch into subregions prevents the loss of spatial information, as can be the case with filter banks, while the histogramming provides robustness to deformations of the image, in contrast to raw pixel intensity based methods. SIFT descriptors outperformed spin-images and RIFT descriptors, as well as descriptors based on normalized raw pixel intensities in the evaluations in [8], [15].
When computational cost is not a limitation and there is enough data available, it is advantageous to combine complementary features. Best performance can be achieved when combining descriptors based on greylevel values like spin-images, with descriptors based on gradient information such as SIFT or RIFT [13], [15].

We turn to a discussion on the level of invariance a descriptor should possess. Traditionally, research concentrated on obtaining invariance to global 2D transformations such as rotation and scaling [15]. Classification under lighting and viewpoint changes has more recently become a significant subject of research [5], [6], [12], [13].
Note that all of the descriptors mentioned above can be made invariant to global affine transformations in illumination intensity i.e. transformations of the whole image patch of the form $aI(x) + b$, where $I(x)$ is the image intensity and $a, b$ are constants. This can be done by normalizing the intensity of the support region to have zero mean and unit standard deviation before computing the descriptors [12]. For SIFT descriptors invariance to affine illumination transformations is obtained by scaling the norm of each descriptor to unit length [7].
Raza et al. [9] empirically establish that for histopathology image classification, scale- and rotation-invariant features outperform the rotation-invariant and non-invariant features. However, when a large training set is available

with multiple models representing each texture, features with a lower level of invariance tend to perform better than invariant features [5]. Typically, increasing the invariance of a feature results is a loss of discriminative power. Therefore, similar to the conclusions on detector invariance in Section 2.1, only the level of invariance strictly necessary for the particular application, that cannot be compensated by storing multiple prototypes in the training set, should be added to the descriptor [5], [15].
For specific applications, some invariance properties might even be unwanted. For CT image classification, the mean of the intensity values is a physical property, so invariance to affine illumination transformations is not desirable [2]. Similarly, scale-invariant features might not be beneficial to cancer grading based on nucleus size [9].

## 2.3 Codebook Formation

In the BoF approach, vector quantization is used to define a relatively small number of generic local image descriptors which form the visual words of the codebook from an initial pool of descriptors extracted from a training set. This is a data-driven approach: the visual words are learned from the training data. An image is represented by its distribution of the visual words. In literature on the BoF method, the vector quantization method used is almost always a clustering algorithm, therefore we will refer to the vector quantization step as clustering step in what follows. The clustering method, the codebook size and the choice of training set will be discussed in this section.

A variety of clustering methods are used in literature, the most popular one being classical $k$-means clustering. This method aims to minimize the within-cluster sum of squares:

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathbf{c}_i||^2, \tag{1}$$

where $\mathbf{x}_j$ is a descriptor vector of an image patch in the training set and $\mathbf{c}_i$ is the center of cluster $C_i$. The $k$-means algorithm is usually started with $k$ random cluster centers, and iteratively adds descriptor vectors to the cluster for which the cluster center is closest by. Disadvantages of the method are that it only converges to local optima of Equation (1) and that the number of clusters formed is a parameter of the method which must be determined

in advance. A solution to the second problem could be to use agglomerative clustering, in which clusters are merged until the average intra-cluster distance becomes too big. However, this is computationally expensive, and becomes infeasible for larger datasets [5]. Pelleg and More [22] propose a more efficient $k$-means based clustering algorithm that determines the number of clusters within a given range, termed $x$-means. Csurka et al. [1] state that for accurate classification, a global minimum of the objective function (1) or the most accurate clustering in feature space is not necessary. They run $k$-means several times with different sizes and different sets of initial cluster centers and then choose the codebook with the lowest classification error on the test set.
Jurie and Triggs [4] argue that radius-based clustering works better than the traditional $k$-means clustering for densely sampled image patches from heterogeneous textures. Patches with intermediate frequency are the most informative for classification: the high frequency patches are excessive in all classes and thus tend to contain little information. Traditional $k$-means clustering assigns too many small-sized clusters to dense areas in feature space. This results in a non-uniform and suboptimal coding in which many clusters code the less-informative high frequency image patches. The radius-based clustering method of Jurie and Triggs enforces a minimum cell size in feature space to prevent this. In their codebooks, more visual words are dedicated to coding the intermediate-frequency patches. In tests on the ETH80, Agarwal-Roth and Xerox7 datasets, the proposed method outperformed the $k$-means based codebook. However, in the medical applications [2], [9], [10], $k$-means is still used for its simplicity and established performance on homogeneous textures.

Although the optimal size of the codebook depends on the particular application, in general it is a matter of expressiveness versus generalization and computational efficiency [6]. At first, increasing the codebook size leads to significant improvements in performance, since the distribution of local features can be approximated more accurately by the visual words. However, increasing the codebook size beyond a certain point does not improve performance and only increases the computational cost. A decrease in performance beyond a certain codebook size is even reported in [8], [12], which can be attributed to overfitting of the data. For a specific data set, the point at which no improvement is further achieved can be established empirically by forming differently sized codebooks on the training set and choosing the

codebook with the best performance on the validation set [12].
Jurie and Triggs experiment with building a large codebook and subsequently using feature selection to prune it down. They select a subset of the visual words by choosing the words that maximize an informativeness score such as mutual information or odds ratio, or train a SVM on the complete codebook and then use only the visual words with the highest weight. They empirically establish that the latter method performs best. However, the performance of the complete codebook is still superior to the reduced one, so feature selection should only be used if the computational cost is a limitation for the particular application.

We now turn to a discussion on the training set to be used for the codebook formation. Let $m$ be the number of classes. The most straightforward choice is to form a training set containing images of each of the $m$ classes. One can either cluster the descriptors from the entire training set, i.e. cluster the descriptors from all images of all classes at once. Alternatively, the clustering can be done per class. In that case, the codebook is built by forming $n$ visual words for each of the $m$ classes, resulting in a codebook of size $mn$. The codebook can be pruned down by merging cluster centers that lie too close together and discarding the centers that have too few data assigned to them. As the class-wise method prevents clustering a large amount of data at once, it is computationally more efficient than clustering on the entire training set.
Leung and Malik [6] form a codebook by learning visual words per class, but only use a subset of all of the texture classes. The idea behind this is that generic, local features will be described by the visual words learned from the subset of textures, so that the codebook will be able to describe the rest of the textures as well. Their codebook is built by forming $n$ visual words for each of the $s < m$ texture classes in the reduced training set, resulting in a codebook of size $sn$. The performance of such a codebook is adequate, but inferior to the performance of a codebook trained on all texture classes. Therefore, in later work, Varma and Zisserman [13] use all texture classes for learning the codebook.
Nowak et al. [8] test whether codebooks should be formed with a specific application in mind. They conclude that indeed it is best to design a codebook for a particular task, but note that codebooks trained on random images also work adequately. Even codebooks containing random SIFT vectors have considerable discriminative power. This indicates that the BoF method is

relatively insensitive to the choice of training images and the visual words in the codebook.

## 2.4 Image Representation

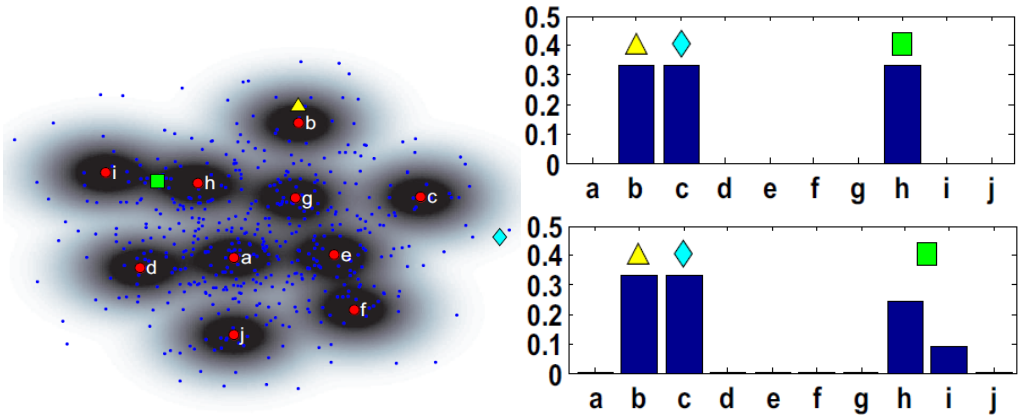
Usually in the BoF method, an image representation is obtained by mapping the descriptor vector of each image patch to the nearest visual word in the codebook with a vector quantizer. An image can then be represented as a histogram of frequency counts of the visual words. This representation discards all information on spatial arrangement of the descriptors. For scene categorization, spatial information preserving methods such as spatial pyramid matching (SPM) can be useful [14]. The SPM method is an extension to the BoF image representation, in which the image is partitioned into segments at multiple scales. A BoF histogram is computed for each segment, and the histograms are concatenated to form the feature vector of the entire image. We will not discuss spatial information preserving methods further, since generally in medical images relevant morphology can appear anywhere and spatial arrangement is not important.
Nowak et al. [8] compare the standard BoF approach, in which an image is represented by a histogram of visual word counts, to two alternative image representations. They note that using the frequency of visual words directly as feature vector is not optimal in combination with their linear SVM classifiers. Using a linear kernel: $K_{linear}(\mathbf{h}_1, \mathbf{h}_2) = \mathbf{h}_1^T \mathbf{h}_2$, the effects of a non-uniform distribution of occurrence counts degrade classification performance. Two options to convert the frequency counts to an alternative image representation are presented. The first option is to form a binary indicator vector, where the index of a visual word is 1 if the count is non-zero, and 0 otherwise. The second option is to form a binary indicator vector using a threshold of the count chosen to maximize the mutual information between the visual word and the class label on the training set. In [4], [8] tests were done on the Xerox7, Pascal-01 and Agarwal-Roth datasets. It was concluded that the original histogram representation and the binary indicator vector with mutual information based thresholding worked best; both methods gave similar performances with linear kernel SVMs. The use of a non-linear kernel such as the $\chi^2$ kernel, Equation (11), reduces the potential negative effects of non-uniform distribution of occurrence counts. Therefore, usually the original BoF histogram image representation is used in combination with a non-linear distance metric between histograms.

Tamaki et al. [10] experiment with class-wise concatenation of visual words to reduce the computational load of clustering a large amount of visual words for their endoscopic images. $k$-Means clustering is performed on each class separately, and for each class, a codebook is formed. An image representation is obtained by concatenating all of the histogram representations of the class-specific codebooks. In experiments on their endoscopic image set, this method has similar performance as a global codebook, formed by clustering the data from all classes together. Although the global codebook tends to perform a little better for most codebook sizes, the class-wise concatenation is chosen in their application for its reduced computational cost.

An image representation method based on soft histograms has been proposed by van Gemert et al. [3]. The method is termed codeword uncertainty (CU). Here, descriptors are not assigned one hard label, but spread in multiple bins with a technique based on kernel density estimation. The CU method distributes probability mass to all relevant codewords:

$$CU(\mathbf{w}, \mathbf{x}_i) = \frac{1}{M} \frac{K_\lambda(d(\mathbf{w}, \mathbf{x}_i))}{\sum_{j=1}^{k} K_\lambda(d(\mathbf{v}_j, \mathbf{x}_i))}, \tag{2}$$

where $M$ is the number of descriptors in the image, $k$ is the number of visual words in the codebook, $\mathbf{x}_i$ is a descriptor, $\mathbf{w}, \mathbf{v}_i$ are visual words of the codebook and $d(\cdot, \cdot)$ is a distance function between visual words. In [3] the Euclidean distance function is used, together with the Gaussian-shaped kernel of Equation (10). The scale parameter $\lambda$ is determined by cross-validation. The entry of the codeword $\mathbf{w}$ in the image representation is then obtained by summing over all descriptors $\mathbf{x}_i$. Figure 3 gives a graphical illustration of the difference between the traditional histogram image representation and the CU representation. The biggest difference can be appreciated at the descriptor depicted with a green square. In the traditional representation, this descriptor only contributes weight to its nearest visual word in the codebook, **h**. In the CU representation it contributes most of its weight to codeword **h**, a considerable amount to its second closest visual word **i** and an insignificant amount to the remaining codewords. Especially for small sized codebooks and a high dimensional feature space, the method provides increased robustness to the curse of dimensionality. CU outperforms the traditional histogram representation method in the tests on the Scene-15, Caltech-101 and Caltech-256 datasets.




(a) (b)

Figure 3: (a) Codebook with visual words $a, b, ..., j$ formed by radius-based clustering of the blue descriptors in the training set. The green square, yellow triangle and blue diamond are the descriptors in novel images to be encoded. (b) Traditional image representation (upper) compared to the codeword uncertainty image representation (lower). The figure is reproduced from [3].

More recently, research has been done on replacing vector quantization by sparse coding. Using vector quantization, the index of only one of the visual words in the codebook may be nonzero, leading to coarse reconstruction of the image. Sparse coding represents every descriptor in the query image as a linear combination of a few visual words from the codebook [20]. The codebook can be learned to be optimal for use with sparse coding, as is done by Yang et al. [14]. We will not consider this, and suppose the usual clustering based codebook fixed. Given a descriptor $\mathbf{x}_i$ and fixed codebook $\mathbf{V}$, the sparse coding optimization problem becomes:

$$\min_{\mathbf{h}_i} ||\mathbf{x}_i - \mathbf{h}_i\mathbf{V}||_2^2 + \gamma|\mathbf{h}_i|_1, \tag{3}$$

where $\mathbf{h}_i$ is the sparse coding representation of descriptor $\mathbf{x}_i$. The last term of the expression relates to the sparsity constraint. The minimization is done for all descriptors, and the whole image representation is obtained by averaging over all descriptors:

$$\mathbf{h} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{h}_i, \tag{4}$$

where $M$ is the number of descriptors in the image. Using sparse coding, the reconstruction error is typically much lower than using vector quantization. A drawback of sparse coding is the increase in computational cost.

It is interesting to note that an alternative to the histogram image representation has been introduced by Rubner et al. [23]. The proposed image signature $S = \{(\mathbf{c}_1, u_1), ..., (\mathbf{c}_m, u_m)\}$ has become popular in BoF methods. Here $m$ is the number of clusters, $\mathbf{c}_i$ is the center of the $i$-th cluster, and $u_i$ is the weight of the cluster (i.e. the size of the cluster divided by the total number of descriptors extracted from the image). This allows more flexibility, since the centers of the clusters can either be codewords from a global codebook, clusters of descriptors in an individual image, or individual features. The second option, clustering within images, is mostly used. This way, a different codebook is obtained for each image. Especially in high dimensions, the effects of vector quantization and binning are reduced compared to using histograms with a global codebook [5]. Also, the computational cost of training decreases significantly as the building of a global codebook is avoided. The signature image representation is a variable-size representation rather than fixed-size such as the histogram representation. An agglomerative clustering algorithm can be used to determine the amount of clusters for each image. This is especially useful when selecting image patches using an interest point detector. Then the number of patches per image can vary significantly, and choosing a fixed number of clusters per image can be hard. Compared to the histogram representation, the image signature can achieve a better balance between expressiveness and efficiency by varying the number of clusters per image [23].

## 2.5 Classification

In this section, we mention briefly two classification models that originate from text document analysis: Naïve Bayes and Probabilistic Latent Semantic Analysis (PLSA). In more detail we discuss the $k$-Nearest Neighbour ($k$-NN) classifier and SVMs which are more widely used for texture classification.

Both Naïve Bayes and PLSA model the conditional probability of a bag of features given a class. An image is assigned to the class that has maximal posterior probability given the visual word counts in that image. The Naïve Bayes model assumes conditional independence of the visual words given the

class. The classifier has the advantage of being simple and fast. Csurka et al. [1] and Jurie and Triggs [4] experimented with using a Naïve Bayes classifier for object and texture classification, but in their BoF frameworks the classification performance of SVMs proved superior.
PLSA is a technique based on the idea that each image consists of a mixture of intermediate or so-called 'hidden' topics. In scene classification for example, if the two-stage process would find intermediate topics such as a building and a car, the model would increase the probability of the whole image to be an urban scene [17]. Since images of textures generally do not contain multiple intermediate topics, PLSA has less relevance to texture analysis and medical image classification. In the medical applications considered here [2], [9], [10], neither the Naïve Bayes classifier nor PLSA is used.

The $k$-NN classifier simply stores all training feature vectors. A new image is assigned to the class that has the most votes among the $k$ closest training images in feature space. The Nearest Neighbour (NN) classifier is a special case of the $k$-NN classifier with $k$ equal to one. In earlier texture classification applications of Leung and Malik [6] and Varma and Zisserman [13], the NN classifier is used together with the $\chi^2$ metric for comparing histograms:

$$D_{\chi^2}(\mathbf{h}_1, \mathbf{h}_2) = \frac{1}{2} \sum_{i=1}^{m} \frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}. \tag{5}$$

Alternatively, the $k$-NN classifier can be applied with the signature image representation using the Earth Mover's Distance (EMD):

$$EMD(S_1, S_2) = \frac{\sum_{i,j} f_{ij} d(c_{1i}, c_{2j})}{\sum_{i,j} f_{ij}}, \tag{6}$$

where $d$ is a distance measure between cluster centers and $f_{ij}$ the flow value between cluster centers $c_{1i}$ and $c_{2j}$ that can be determined by solving a linear programming problem [23]. $k$-NN classifiers tend to work well if the distance function is good and there is a lot of data. A disadvantage of the method is that although the training phase is computationally inexpensive, cost of classification can be high. Another drawback is that $k$-NN classifiers use all attributes instead of learning which are most important. In [15] tests were done on the standard texture databases UIUCTex, KTH-TIPS, Brodatz and

CUReT and in [2] tests were done on a CT lung images database. In both studies, SVMs outperformed the $k$-NN classifiers.

SVMs have gained popularity especially for high-dimensional problems. In the BoF method they have become the standard choice classifier [1]. The decision function for a test sample $\mathbf{h}$ for a two-class case has the form:

$$P(\mathbf{h}) = \text{sign}\left(\sum_i \alpha_i y_i K(\mathbf{m}_i, \mathbf{h}) - b\right) \tag{7}$$

where $\mathbf{m}_i$ is a training sample and $y_i \in \{-1, 1\}$ is the class label of $\mathbf{m}_i$. The weight $\alpha_i$ is learned in the training phase and is non-zero for only a fraction of the training models, namely the support vectors. During training, the threshold parameter $b$ is learned as well. For the histogram image representation, different kernel functions $K(\cdot, \cdot)$ can be used, including the linear, histogram intersection (HI), Radial Basis Function (RBF), $\chi^2$ kernel. The RBF kernel is also called the Gaussian kernel. For the signature image representation the EMD kernel is mostly used:

$$K_{linear}(\mathbf{h}_1, \mathbf{h}_2) = \mathbf{h}_1^T \mathbf{h}_2, \tag{8}$$

$$K_{HI}(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{k} \min(h_{1i}, h_{2i}), \tag{9}$$

$$K_{RBF}(\mathbf{h}_1, \mathbf{h}_2) = \exp\left(-\gamma |\mathbf{h}_1, \mathbf{h}_2|^2\right), \tag{10}$$

$$K_{\chi^2}(\mathbf{h}_1, \mathbf{h}_2) = \exp\left(-\frac{\gamma}{2} \sum_{i=1}^{m} \frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}\right), \tag{11}$$

$$K_{EMD}(S_1, S_2) = \exp\left(-\gamma \frac{\sum_{i,j} f_{ij} d(c_{1i}, c_{2j})}{\sum_{i,j} f_{ij}}\right). \tag{12}$$

The SVM framework defines a decision boundary for a two-class problem. There is no straightforward multi-class extension to SVMs. To solve multi-class problems, the one-against-one method or the one-against-all method can be used. For a problem with $c$ classes, one has to train a SVM for each of the $\frac{c(c-1)}{2}$ pairs of classes in the one-against-one method. The class of a novel image is determined by letting all the SVMs vote, assigning the image to the class with the maximum number of votes. For the one-against-all

approach, one SVM is trained per class. Each SVM is trained on the class versus the rest. The image is assigned to the class for which the decision function of its SVM has the highest value. A disadvantage of the latter method is the unbalancedness in the training data for all of the SVMs. According to Tamaki et al. [10] and Zhang et al. [15] both methods give similar results. In most articles, the one-against-one method is adopted without discussion.

SVMs tend to work very well in practice, especially for high dimensional problems such as the typical BoF classification problem. Disadvantages of SVMs are the significant computation and memory requirements. During training time, a matrix of kernel values for each pair of images in the training set must be computed. Especially for non-linear kernels, the training phase is computationally expensive and scaling up to large datasets encountered in real-word applications is a non-trivial problem. In addition, the soft margin parameter $C$ has to be established via cross-validation, increasing computational cost further. For some kernels, such as the RBF kernel, the scaling parameter $\gamma$ has to be established in a similar manner. In [15] this was solved by setting $\gamma$ equal to the mean value of the distances between all training images which was found empirically to give good results.
Different kernel types were tested in [15], best classification results were achieved with the $\chi^2$ kernel and EMD kernel, which had similar performance. The EMD kernel was chosen to avoid building a global codebook, thereby saving computational time. The classification performance of several histogram based kernels was also compared in [10]. On this endoscopic image dataset, the $\chi^2$ kernel achieved the best performance. However, a linear kernel was chosen for its reduced computational time.
Yang et al. [14] find that sparse coding in combination with a simple linear SVM classifier performs better than the computationally expensive non-linear kernels that are needed in combination with vector quantization to obtain good performance. The classes can be made more linearly separable in feature space when using sparse coding. Although the sparse coding step is more expensive than vector quantization, the combination with a linear kernel speeds up training and enables scaling to much larger datasets.

# 3 Medical Image Classification

Computer Aided Diagnosis (CAD) tools have been investigated since around 1980 [19]. The common attitude towards CAD has moved from being the ultimate tool for obtaining a fully automated diagnosis to being a second opinion to a medical expert. The present clinical practice in which digital images can be stored in central systems and large numbers of images are available, would be well suited for such applications. CAD systems have become reality in aiding the detection of abnormalities in routine scans, for example for breast lesion detection in mammographs. Our focus is on medical image classification by which we mean another type of CAD system: one that classifies whole images, or predefined regions. To be able to use a classification system as a second opinion, it is important that the CAD system provides interpretable results. The output is preferably a soft classification, i.e. the posterior probabilities of the image belonging to each of the classes, rather than a hard class label. PLSA and Naïve Bayes classifiers directly provide such posterior probabilities as output. Both $k$-NN classifiers and SVMs can be adapted relatively easily to give a soft classification [18].

In medical studies, groups tend to use their own datasets rather than benchmark datasets for both training and testing, making it hard to compare the absolute classification performances between different CAD systems. It would be beneficial if large medical datasets become publicly available, as is the case for standard texture databases. This would create the opportunity to directly compare different CAD systems. Also, it could provide individual studies with more data, which is beneficial to both training and testing [21].

We review three recent works that use the BoF approach for classification of CT lung scans [2], histopathological images [9] and endoscopic colorectal images [10]. Here, ground truth labels for the training and test images are obtained by combined assessment of two or more medical experts.
Gangeh et al. [2] experiment with a BoF image representation for the classification of CT lung images. They focus on improving the assessment of emphysema in CT images, classifying between normal tissue (NT), centrilobular emphysema (CLE), paraseptal emphysema (PSE). In their experiments, 168 regions of interest (ROIs) are used, evenly distributed over the classes. Random sampling is used with non-normalized, raw pixel intensities used as descriptor. The choice to use non-normalized raw pixel values is motivated

by the fact that in CT images, the mean intensity directly indicates a physical property of the tissue. The codebook is formed by $k$-means clustering and their system uses a SVM with RBF kernel. They report a classification performance of 96% on their test set.
Raza et al. [9] apply the BoF approach to histopathological images and aim to classify renal cell carcinoma (RCC) into four subtypes: clear cell (CC), chromophobe (CH), oncocytoma (ON) and papillary (PA). The tissue samples are taken from renal tumors. The study contained 106 histopathological images evenly spread over the classes. A DoG interest point detector with SIFT descriptors gave best performance on the particular dataset. $k$-Means clustering was used to form the codebook, a linear SVM yielded a classification performance of 88%.
Tamaki et al. [10] aim to develop a CAD system for colorectal tumor classification. They use images obtained with a Narrow Band Imaging (NBI) endoscope. A training set of 908 NBI images was collected before April 2010, and a separate test dataset containing another 504 NBI images was collected after April 2010. This set up is much like it would occur in clinical practice: training the system on 'old' images and applying it to new images. Their prototype system extracts SIFT features at regular grid points and forms the codebook by class-wise $k$-means clustering of visual words. They use a linear kernel SVM classifier. As the computation time for classification is only 60 ms, approximately 15 frames per second, the possibility exists to build a real-time application in which frames of NBI videoendoscopy are classified by feeding them to the SVM. Their system is adapted to produce a soft classification of class probability estimates. A classification performance of 93% was achieved on the test dataset.
Comparing the three applications built, we conclude that the work by Tamaki et al. on NBI endoscopic images is most likely to become clinical practice in the near future. First of all, the soft classification output makes the tool more useful as a second opinion to a medical expert. Secondly, their method of producing a training and test dataset is the way one would encounter in clinical practice, in contrast to the manually balanced datasets used by Gangeh et al. and Raza et al. Lastly, the parameter tuning of Tamaki et al. is much more extensive, aiming for the best implementation choice of each component of the BoF method instead of focusing on just a single or a few components.

# 4 Discussion

For each of the components of the BoF method, being feature detection, feature description, codebook formation, image representation and classification, various methods have been suggested in literature. Currently the general approach in texture analysis is to use dense sampling together with SIFT feature descriptors. For codebook formation, typically $k$-means clustering or radius-based clustering is used. The image is either represented by a histogram of visual word counts or by the signature representation. Newer image representations based on soft histograms or sparse coding are promising, but they have not become standard choice yet. The use of a SVM as classifier has become common practice. The $\chi^2$ or EMD kernel tends to give the best classification performances, but linear kernel SVMs are used frequently for their reduced computational cost. It must be stressed that the optimal choices, or parameter tunings, can vary significantly for different datasets. This means that when designing a classification system for a particular application, dedicated effort should be made to find the best choices and tune parameters in all of the components of the BoF method.
Research should continue to focus on finding more effective detectors and descriptors, as well as investigating more advanced methods for combining multiple detectors and descriptors. The medical applications discussed [2], [9], [10], do not incorporate the newest developments in the BoF approach such as sparse coding or soft histograms in their applications. Future research should focus on transferring novel developments in texture analysis to medical applications. Another interesting direction is to investigate the possibility to extend the BoF approach to 3D data obtained for example with Magnetic Resonance Imaging (MRI) or CT.

# References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray. *Visual Categorization with Bags of Keypoints*. Workshop on statistical learning in computer vision, European Conference on Computer Vision - ECCV 2004, vol. 1, p. 22-38, 2004.

[2] M.J. Gangeh, L. Sørensen, S.B. Shaker, M.S. Kamel, M. de Bruine, M. Loog. *A Texton-Based Approach for the Classification of Lung*

*Parenchyma in CT Images*. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2010, vol. 6363, p. 595-602, 2010.

[3] J.C. van Gemert, J. Geusebroek, C.J. Veenman, and A.W.M. Smeulders. *Kernel Codebooks for Scene Categorization*. European Conference on Computer Vision - ECCV 2008, vol. 5304, p. 696-709, 2008.

[4] F. Jurie, B.Triggs. *Creating Efficient Codebooks for Visual Recognition*. Tenth IEEE International Conference on Computer Vision - ICCV 2005, vol. 1, p. 604-610, 2005.

[5] S. Lazebnik, C. Schmid, J. Ponce. *A Sparse Texture Representation Using Local Affine Regions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27(8), p. 1265-1278, 2005.

[6] T. Leung, J. Malik. *Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons*. International Journal of Computer Vision, vol. 43(1), p. 29-44, 2001.

[7] D.G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60(2), p. 91-110, 2004.

[8] E. Nowak, F. Jurie, B. Triggs. *Sampling Strategies for Bag-of-Features Image Classication*. European Conference of Computer Vision - ECCV 2006, vol. 3954, p. 490-503, 2006.

[9] S.H. Raza, R.M. Parry, R.A. Moffit, A.N. Young, M.D. Wang. *An Analysis of Scale and Rotation Invariance in the Bag-of-Features Method for Histopathological Image Classification*. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2011, vol. 6893, p. 66-74, 2011.

[10] T. Tamaki, J. Yoshimuta, M. Kawakami, B. Raytchev, K. Kaneda, S. Yoshida, Y. Takemura, K. Onji, R. Miyaki, S. Tanaka. *Computer-aided colorectal tumor classification in NBI endoscopy using local features*. Medical Image analysis, vol. 17(1), p. 78-100, 2013.

[11] T. Tuytelaars and K. Mikolajczyk. *Local Invariant Feature Detectors: A Survey*. Foundations and Trends in Computer Graphics and Vision, vol. 3(3), p. 177-280, 2008.

[12] M. Varma, A. Zisserman. *A Statistical Approach to Texture Classification from Single Images*. International Journal of Computer Vision, vol. 62(1-2), p. 61-81, 2005.

[13] M. Varma, A. Zisserman. *A Statistical Approach to Material Classification Using Image Patch Exemplars*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31(11), p. 2032-2047, 2009.

[14] J. Yang, K. Yu, Y. Gong, T. Huang. *Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification*. IEEE Conference on Computer Vision and Pattern Recognition - CVPR 2009, p. 1794-1801, 2009.

[15] J. Zhang, M. Marszaek, S. Lazebnik, C. Schmid. *Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study*. International Journal of Computer Vision, vol. 73(2), p. 213-238, 2007.

**Extra Articles**


[16] Y. Amit, D. Geman. *A Computational Model for Visual Selection*. Neural Computation, vol. 11(7), p. 1691-1715, 1999.

[17] A. Bosch, A. Zisserman, X. Munoz. *Scene Classification via pLSA*. European Conference of Computer Vision - ECCV 2006, vol. 3954, p. 517-530, 2006.

[18] C.C. Chang, C.J. Lin. *LIBSVM: a Library for Support Vector Machines*. ACM Transactions on Intelligent Systems and Technology, vol. 2(3), 2011.

[19] K. Doi. *Computer-Aided Diagnosis in Medical Imaging: Historical Review, Current Status and Future Potential*. Computerized Medical Imaging and Graphics, vol. 31(4-5), p. 198-211, 2007.

[20] M. Elad, M. Aharon. *Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries*. IEEE Transactions on Image Processing, vol. 15(12), p. 3736-3745, 2006.

[21] B. van Ginneken, B.M. ter Haar Romeny, M.A. Viergever. *Computer-Aided Diagnosis in Chest Radiography: A Survey*. IEEE Transactions on Medical Imaging, vol. 20(12), p. 1228-1241, 2001.

[22] D. Pelleg, A.W. Moore. *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. International Conference on Machine Learning - ICML 2000, p. 727-734, 2000.

[23] Y. Rubner, C. Tomasi, L.J. Guibas. *The Earth Mover's Distance as a Metric for Image Retrieval*. International Journal of Computer Vision, vol. 40(2), p. 99-121, 2000.

# Samenvatting

Het doel van deze literatuurstudie is het onderzoeken van het 'Bag-of-Features' (BoF) model en de toepassingen van dit model op classificatie van medische beelden. Met classificatie van beelden bedoelen we het plaatsen van een beeld in een categorie aan de hand van wat er in het beeld te zien is. Voorbeelden van catergorieën zijn 'auto' of 'huis'. We geven een computer-algoritme eerst een aantal trainingsbeelden, zodat geleerd kan worden wat elk van de categorieën typeert. Als het algoritme op die manier is getraind, wordt het gebruikt om nieuwe beelden te classificeren in één van de categorieën. In medische beeldclassificatie kunnen de categorieën bijvoorbeeld zijn: 'normaal weefsel' en 'tumorweefsel'.
Het BoF model is een specifieke manier van beeld representatie die we in deze literatuurstudie onderzoeken. Het BoF model representeerd een beeld als een collectie van locale indicatoren. Een locale indicator beschrijft bijvoorbeeld de grijswaarden, of er wel of niet een rand is, of dat er een sterke overgang van licht naar donker is op een bepaald gebied in het beeld. Tijdens de trainingsfase wordt aan de hand van deze verzameling indicatoren een codeboek geleerd, door de indicatoren die op elkaar lijken allemaal als hetzelfde codewoord te beschrijven. Zo kan een nieuw beeld worden beschreven als een verzameling van codewoorden. Een belangrijk aspect van de BoF methode is dat de beelden worden vergeleken aan de hand van het aantal van elke codewoord in het beeld. Hierbij wordt de locatie waar het codewoord voorkomt niet meegenomen in de beeldrepresentatie. Een beeld wordt dus gerepresenteerd als een distributie (vaak een histogram) van codewoorden uit het codeboek. Om tot beeldclassificatie te komen, wordt een classificatie algoritme toegepast met de verkregen representatie als invoer.
Oorspronkelijk werd de BoF methode vooral toegepast voor het classificeren van beelden van textuur en materialen. Het is gebleken dat ook voor classificatie van beelden van objecten de methode succesvol is. Toepassingen op het gebied van medische beeldclassificatie worden sinds kort onderzocht. We zien dat de nieuwste technieken in de BoF methode nog maar weinig worden toegepast in medische beeldclassificatie. In dit werk geven we een overzicht van de recente literatuur over het BoF model en vergelijken we voor de belangrijkste componenten verschillende opties die in de literatuur zijn gesugereerd. Verder kijken we specifiek naar de toepassingen van de BoF methode voor medische beeldclassificatie.