Electronic Letters on Computer Vision and Image Analysis 16(3):30-45, 2017

MMKK++ algorithm for clustering heterogeneous images into an unknown number of clusters

Dávid Papp* and Gábor Szűcs*

* *Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar Tudósok krt. 2., H-1117, Budapest, Hungary;* {pappd,szucs}@tmit.bme.hu

Received 2nd Feb 2017; accepted 23rd Nov 2017

Abstract

In this paper we present an automatic clustering procedure with the main aim to predict the number of clusters of unknown, heterogeneous images. We used the Fisher-vector for mathematical representation of the images and these vectors were considered as input data points for the clustering algorithm. We implemented a novel variant of K-means, the kernel K-means++, furthermore the min-max kernel K-means plusplus (MMKK++) as clustering method. The proposed approach examines some candidate cluster numbers and determines the strength of the clustering to estimate how well the data fit into *K* clusters, as well as the law of large numbers was used in order to choose the optimal cluster size. We conducted experiments on four image sets to demonstrate the efficiency of our solution. The first two image sets are subsets of different popular collections; the third is their union; the fourth is the complete Caltech101 image set**.** The result showed that our approach was able to give a better estimation for the number of clusters than the competitor methods. Furthermore, we defined two new metrics for evaluation of predicting the appropriate cluster number, which are capable of measuring the goodness in a more sophisticated way, instead of binary evaluation.

Key Words: image clustering, kernel K-means, cluster number, Fisher-vector

1 Introduction

The image grouping is an existing problem [7] in processing large collections of heterogeneous images in order to organize a large set of images into clusters, such that images within the same cluster have similar meaning. Image clustering provides high-level summarization of large image collections, and thus has many useful applications. For example, image repositories (in Media Content Management Systems) are more convenient for users to browse. Grouping is a category of image sorting problem that can be found in many other areas and applications as well. Every day the use of images from mobile devices as evidence in legal lawsuits is more usual and common. Therefore, forensic analysis of mobile device images takes on special importance, which can be based on the identification of the source, specifically on the grouping of images according to their source acquisition [54]. Another area is the World Wide Web, where clustered web image search results can help end users. Furthermore, image grouping can be used to better align the semantics of the Web image and text. Near-duplicate image clustering can be used to group web images into a set of clusters of near-duplicate images according to their visual similarities. The near-duplicate web images in the same cluster could share similar semantics [55]. There is a problem type in evesryday life or in social life where the aim is to summarize image collections that correspond to a single event [39], furthermore in the work [2] the

Correspondence to: pappd@tmit.bme.hu

Recommended for acceptance by Armando Pinho
https://doi.org/10.5565/rev/elcvia.1054
ELCVIA ISSN: 1577 – 5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

task was grouping images into clusters of different events using the image features and related metadata. In agricultural area there is a similar problem, the systematization, grouping of large amount of gathered plants. There is a gap in botanists' knowledge: there are many plant species we know about now and the species yet to be discovered and named, which do not yet "exist" scientifically. This gap is called taxonomic gap [1], that is an open problem, but clustering the images with smart image analytics using computer science and image processing is promising direction in a long way of solution, and this can help with stakeholders in agricultural area, therefore may have a strong impact.

Our work is based on only visual information, without any existence of manual information about the foreground or background, without any user's help and the aim is to cluster the whole image data set, in heterogeneous environment. Our solution is based on image analytics and data mining algorithm. The image processing procedure begins with feature extraction, and gives multidimensional descriptor vectors as mathematical representations of the images. The next stage of our solution is a clustering process, which uses the vector representations of the images as input data points. We constructed a new clustering algorithm, which is the largest contribution of this paper and it contains more cycles for finding a good solution. In the next sections we will describe the implemented solution in more details. We conducted experiments to demonstrate the efficiency of our proposed approach, and we compared our results with the performance of other methods. We used four image sets during the tests (see Figure 1 for example images). We defined novel metrics to measure the goodness of the predicted cluster number for a given data set, which is the second contribution of this paper. We presented the experimental results in the last chapter.

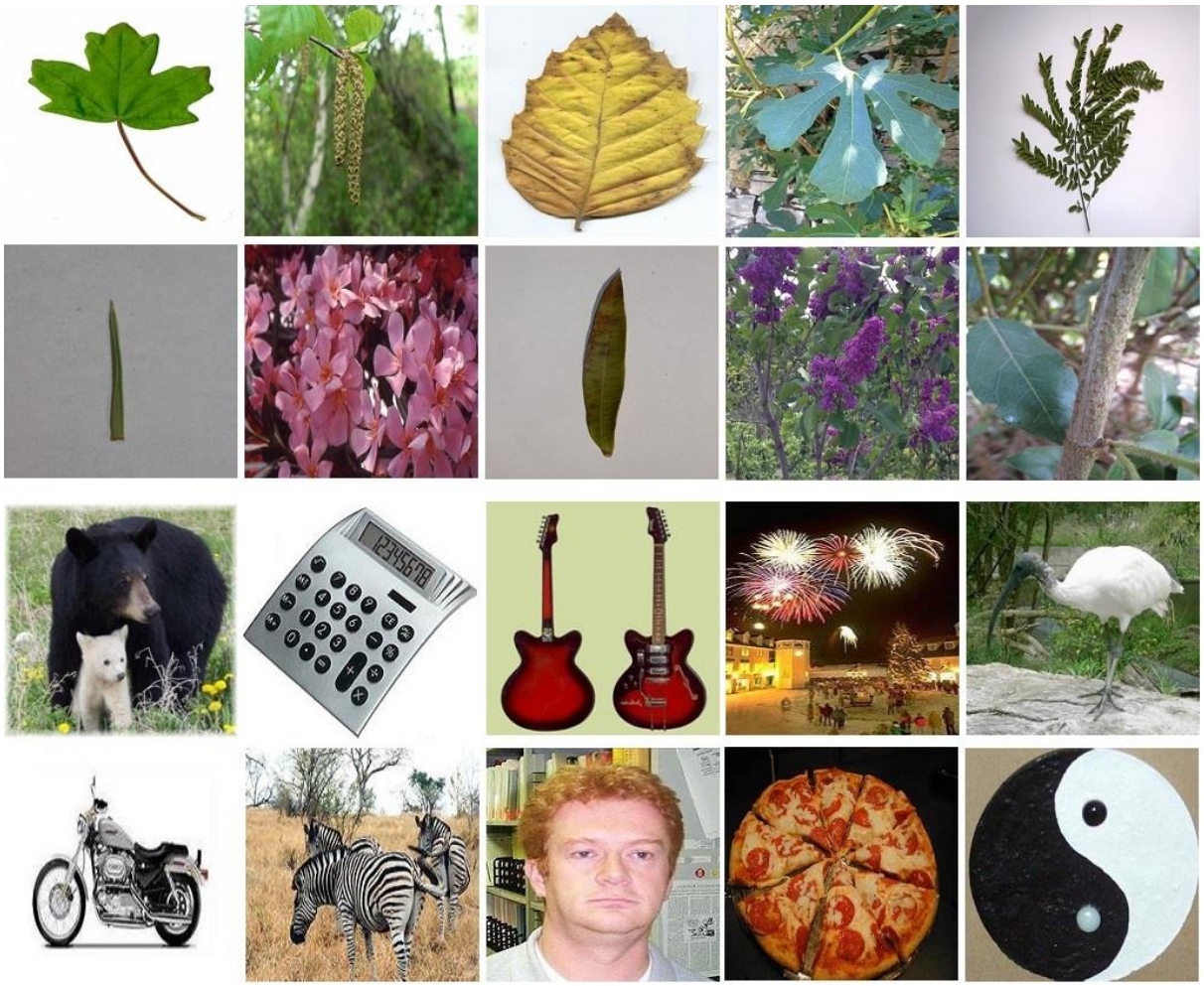

Figure 1 Example images from the test sets. The first two rows show an image from each category of the first test set (Plant10), and the last two rows show examples from the second test set (Cal10)

2 Related Work

There are many works deal with image clustering; in some of them the grouping is in pixel level (low level), where the aim is image segmentation. At the others the goal is to cluster the images itself (higher level), and we focused on this level. The works can be categorized by some viewpoints:

- available information (only visual information or more information),
- granularity (whole images or parts of them),
- existence of an uncluttered background,
- amount of user's help.

Some metadata can be used for the grouping [53][50][40][2], which can help for better clustering; but in our task only visual information was available for grouping. Not only the metadata, but also other high level semantic informational structure can provide additional information. For example Object Relation Network (ORN) [8] is an informational structure to capture image semantics. ORN is a graphical model that links objects in an image through meaningful relations. Therefore, an image can be described by the ontology class assignments in the ORN [7][6].

Based on granularity different purposes can be defined at the grouping of images: the aim can be clustering the images itself (i.e. not the parts of them, like in [39][2]) or objects [27][3] (can be seen) in the parts of the images. The first task is easier, because there is not required to distinguish important and unimportant parts in the images.

Considering the subset of object clustering the next viewpoint in the task categorization is the existence of background. Sometimes there is nothing disturbing, interfering background, only an object can be seen in each image [27][34], but sometimes it is difficult to separate the foreground and the background. However, in our task we used heterogeneous image sets, so some of the objects are in unknown background and some of them have no background at all. Some works [33] have used such image databases – like Columbia Object Image Library [38] with 1440 gray scale image database representing 20 objects – where there is no any background in the image (i.e. the background is black), so the foreground is the object itself. In some works the user helps the system (e.g. the user gives the number of the clusters [39]), but in the beginning of our work we have defined a fully automatic clustering without any user help.

At comparison of our work with others, in spite of many image clustering papers, there are only a few works where the aim and the details of the task are similar to our purposes. A recent work deals with the problem of summarizing image collections that correspond to a single event [39]. For this purpose several clustering algorithms were used, K-means, Hierarchical clustering using complete linkage [13], Hierarchical clustering using single linkage [48], Partitioning Around Medoids (PAM) [30], Affinity Propagation [19] and the Farthest First Traversal Algorithm [21]. In the experiences the K-means algorithm gave the best results, but the numbers of clusters were fix (K=10 in the collection) or in the other alternative the user should give it. Another paper suggests two clustering algorithms (K-Means and Fuzzy K-Means) for image grouping [44], but the system was not tested, so there is no information about the results, thus we cannot compare them.

There are some pioneer image clustering researches [43][33] and a promising work [3]. In an early paper [43] Qiu presented a stochastic algorithm to jointly cluster images and their description features, but the work was only theoretical without any goodness indicator for measurement of the results.

A similar work [33] dealt with only such images, where the background did not take the problem to be more complicated; but in our environment the objects can be found in a various, heterogeneous background.

Another investigated paper [3] works with only known clustering algorithm (k-means, partitioning around medoids: PAM, fuzzy c-means, and hierarchical). The largest difference between the earlier publication and our suggestion is the usefulness of the solution, because our system can be used in more general cases. The earlier published solution contains only color-based feature extraction methods: 3x3x3, 5x5x5 and 6x6x6 quantized RGB histogram (27, 125 and 216 bins) and a 32-, 128-, and 256-cell quantized HMMD (MPEG-7-compatible) histogram [25] (32, 128 and 256 bins). These feature extraction methods are not able to grasp variety of an object type (with different shape and illumination). The tested image set consists of traffic signs, which always look like similar, thus the method is not capable to use in heterogeneous environment. However, in our solution we have used more sophisticated feature extraction methods, which are able to represent and

handle larger variety of objects, thus due to wider application area our system can be considered as more general (and more useful).

Hancer E. and Karaboga D. [26] created a comprehensive survey of methods related to automatic cluster number evaluation, however the most similar works to our paper are the Silhouette method, which was first described by Peter J. Rousseeuw [47] and the cluster validity proposed in [52]. In this paper we compare the performance of our solution to these techniques. The former provides a succinct graphical representation of how well each object lies within its cluster. The silhouette coefficient contrasts the average distance of elements in the same cluster with the average distance of elements in other clusters. Objects with a high silhouette value are considered well clustered; objects with a low value may be outliers. The entire clustering is displayed by a single plot, allowing an overview of the relative quality of the clustering and the configuration of the data. This index works well with K-means clustering, and is also used to determine the optimal number of clusters. Siddheswar Ray and Rose H. Turi used cluster validity [52] to determine the number of clusters in K-means clustering and they applied this technique in color image segmentation. Cluster validity is the ratio of the average of all intra-cluster distances and the minimum of the inter-cluster distances, and by minimizing this metric, the number of clusters can be determined automatically. We chose these techniques for comparison, because our solution also focuses on finding the optimal cluster number.

3 Image Representation

It is important to represent the visual content of the images using sophisticated and state-of-the-art techniques, since the descriptor vectors of images will be the input of the clustering algorithm. We used BoW (Bag-of-Words) [17][29][32] model to represent an image (based on its visual content) with so-called visual code words while ignoring their spatial distribution. Firstly, to construct such visual code words, we selected keypoints in the images with the Harris-Laplace corner detector [5][37], then we used SIFT (Scale Invariant Feature Transform) [35] to extract and describe the local attributes of those keypoints. Note that we used the default parameterization of SIFT proposed by Lowe; therefore, we got descriptor vectors with 128 dimensions. There are several low level features, like RGB histogram or HOG (Histogram of Oriented Gradients) [12], but we chose SIFT because it is a robust and frequently used feature. It could be used in every grid point in the images, this is the "dense" version of SIFT, but Harris-Laplace corner detector is more feasible, because this takes the most important points, the keypoints in the images.

We used GMM (Gaussian Mixture Model) [46][51][4] to define the visual code words from the descriptor vectors, which is a parametric probability density function represented as a weighted sum of (in our case 256) Gaussian component densities, as can be seen in Equation 1.

$$p(X|\lambda) = \sum_{j=1}^{K} \omega_j g\left(X|\mu_j, \sigma_j\right) \tag{1}$$

where X is the concatenation of all SIFT descriptors, ω_j, μ_j and σ_j denote the weight, expected value and variance of the j^{th} Gaussian component, respectively, and $K = 256$. We calculated the λ parameter, which includes the parameters of all Gaussian functions, with ML (Maximum Likelihood) estimation by using the iterative EM (Expectation Maximization) algorithm [51][16][24]. We performed K-means clustering [36] over all the descriptors with 256 clusters to get the initial parameter model for the EM. The next step was to create a descriptor that specifies the distribution of the visual code words in an image, called high-level descriptor. To represent an image with high-level descriptor, the GMM based Fisher-vector [18][42] was calculated. This is an appropriate and complex descriptor vector, because this is able to take the semantic essence of the picture, and this is already validated in classification problems [18][42][41][22]. The Fisher-vector is computed from the SIFT descriptors of an image based on the visual code words by taking the derivative of the logarithmic of the Gaussian functions (see Equation 2), thus it describes the distribution of the visual elements for an image. These vectors were the final representations (image descriptor) of the images, and we used them as the input data for the clustering algorithm.

$$F = \nabla_\lambda \log p(X|\lambda) \tag{2}$$

where $p(X|\lambda)$ is the probability density function introduced in Equation 1, X denotes the SIFT descriptors of an image and λ represents the parameter of GMM ($\lambda = \{\omega_j, \mu_j, \sigma_j | j = 1 \dots K\}$).

4 Proposed automatic image clustering solution

4.1 Determination of cluster number

The basis of our approach is the well-known K-means algorithm [36]. It has two important inputs, the initial cluster centers, and the number of clusters (K). In our case the value of K was unknown, since this would require prior knowledge of the test set, and our algorithm aims to deal with unknown image collections. The K-means minimizes the sum of squared distances from all points to their cluster centers, so the results will be compact and well-separated clusters (ideally). Because of that the compactness of clusters can be measured by using an internal evaluation technique, which estimates how well the data fit into K clusters. There are several existing internal evaluation measures that can be used in K-means clustering, for example the Davies-Bouldin index [11], the Dunn index [15], the cluster validity [52], and the Silhouette coefficients [47]. The latter two measures were introduced in the papers that suggest procedures to find the number of clusters, but in different environment. Cluster validity was proposed to segment color images, however our goal was to cluster heterogeneous images based on their representatives (Fisher-vectors). The main difference between these problems is that the input space in case of image segmentation is complete (i.e. every pixel represents an input data point); while in our case the space allotted by the Fisher-vectors is rather sparse. Moreover, in case of complete space the cluster centers are some particular points from the input data, but in case of sparse space the cluster centers can be new data points.

In this paper we define the *strength* of the clustering by $vRDI$, which is based on the intra-cluster and inter-cluster distances; by this measure we were able to compare clustering results with different cluster numbers and then select the most suitable one. We calculated the intra-cluster distance of a cluster by averaging the Euclidean distances of all data vectors from their cluster center, as can be seen in Equation 3. Equation 4 describes the inter-cluster distance of two clusters, which is the Euclidean distance between their furthermost element pair. Smaller intra-cluster distance implies tighter cluster and larger inter-cluster distance refers for better separated clusters, therefore we aim to minimize the former and maximize the latter one.

$$intraCD = D'(C_l) = \sum_{x_i \in C_l} \|x_i - z_l\| \tag{3}$$

$$interCD = D(C_l, C_j) = \max_{\{x_i \in C_l, y_k \in C_j\}} \|x_i - y_k\| \tag{4}$$

where z_l is the center of C_l cluster, x_i and y_k are data vectors in C_l and C_j clusters respectively, and $\|x - y\|$ denotes the Euclidean distance between vector x and y. The goal is to assess the whole grouping, so the averages of the above metrics were taken over all clusters (over all pairs of clusters in case of inter-cluster distance), nevertheless this does not change the need to look for extreme values. We define $vRDI$ (Ratio of Distances between Intra and inter) as the ratio of these measures, as can be seen in Equations 5-7; thereby lower $vRDI$ value suggests more desirable clustering and more appropriate cluster number.

$$\operatorname*{avg}_{1 \le C_l \le K} D'(C_l) = \frac{1}{n} \sum_{l=1}^{K} \sum_{x_i \in C_l} \|x_i - z_l\| \tag{5}$$

$$\operatorname*{avg}_{1 \le I < J \le K} D(C_l, C_j) = \frac{1}{K * (K-1)/2} \sum_{l=1}^{K-1} \sum_{j=l+1}^{K} \max_{\{x_i \in C_l, y_k \in C_j\}} \|x_i - y_k\| \tag{6}$$

$$vRDI = \frac{\underset{1\le C_l\le K}{\text{avg}} D^{'}(C_l)}{\underset{1\le C_l<C_j\le K}{\text{avg}} D(C_l, C_j)} \tag{7}$$

where K denotes the number of clusters and n represents the number of data vectors.

In order to choose the optimum cluster number, some candidate K values should be examined, so an upper ($maxK$) and lower ($minK$) limit should be set. The algorithm cycles through these candidates, and in each iteration it calculates the $vRDI$ measure. Then K with minimum $vRDI$ is selected as the number of clusters. This procedure works well with low dimensional data and with similar shapes of clusters, however, a Fisher-vector consists of 65791 dimensions. We applied an upgraded version of K-means in the algorithm, and we did not use the traditional K-means.

4.2 Kernel K-means++

The basic K-means performs less efficiently when the clusters are not linearly separable, or the data contains arbitrarily shaped clusters of different densities. Because of that we used kernel K-means [9][14] (an extension of K-means), which is mapping the data points from input space to a higher dimensional feature space through a nonlinear transformation ϑ. Then K-means is applied in the feature space to solve the clustering problem, since in this new space the data points are linearly separable and the separators correspond to nonlinear separators in input space. This procedure is called kernel trick and it allows operating in a high-dimensional, implicit (often called imaginary) feature space by simply computing the inner products between the images of all pairs of data in the feature space. These inner products can be expressed by so-called kernel functions of the data pairs; examples of certain functions are shown below in Table 1. Usually these kernels are used to directly provide the inner product without explicitly defining transformation ϑ.

Table 1 Examples of frequently used kernel functions

Linear kernel	$Ker(x_i, x_j) = x_i^T \times x_j$
Gaussian kernel	$Ker(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$
Polynomial kernel	$Ker(x_i, x_j) = (x_i \times x_j + 1)^d$
Sigmoid kernel	$Ker(x_i, x_j) = tanh(k(x_i \times x_j) + \theta)$

We mentioned before that K-means aims to minimize the sum of squared distances from all points to their cluster centers (see Equation 8). The objective of kernel K-means is to solve the same minimization problem, but in the feature space $x_i \to \vartheta(x_i)$, as it can be seen in Equation 9. We use the same notations that were introduced in the previous sub-section, thus z_l is the center of cluster C_l and in feature space it can be written as in Equation 10, where n_l denotes the number of data vectors in cluster C_l. The squared distance between the transformed data $\vartheta(x_i)$ and the cluster center z_l can be expressed by Equation 11.

$$\mathrm{E} = \min\left(\sum_{l=1}^{K}\sum_{x_i\in C_l}\|x_i - z_l\|^2\right) \tag{8}$$

$$\mathrm{E} = \min\left(\sum_{l=1}^{K}\sum_{x_i\in C_l}\|\vartheta(x_i) - z_l\|^2\right) \tag{9}$$

$$z_l = \frac{\sum_{x_j\in C_l}\vartheta(x_j)}{n_l} \tag{10}$$

$$\left\| \vartheta(x_i) - \frac{\sum_{x_j \in C_l} \vartheta(x_j)}{n_l} \right\|^2 = \vartheta(x_i) \times \vartheta(x_i) - \frac{2 \times \sum_{x_j \in C_l} \vartheta(x_j) \times \vartheta(x_i)}{n_l} + \frac{\sum_{x_j, x_k \in C_l} \vartheta(x_j) \times \vartheta(x_k)}{{n_l}^2} \tag{11}$$

As we can see, transformed data points are only present as part of an inner product, therefore we can substitute them with their kernel representatives. We used Gaussian kernel in the algorithm and we created a kernel matrix Ker where $Ker_{ij} = Ker(x_i, x_j)$. After that the clusters can be obtained by solving an optimization problem (see Equation 12) as described in [14][9].

$$\mathrm{E} = \min \left(\sum_{l=1}^{K} \sum_{x_i \in C_l} \|\vartheta(x_i) - z_l\|^2 \right) \approx \max(trace(UKerU')) \tag{12}$$

where U is the optimal normalized cluster membership matrix.

Algorithm 1 gives an overview of the procedure of kernel K-means; furthermore we used plusplus cluster center initialization (see Algorithm 2) before the iterative steps, which was proposed by D Arthur and S Vassilvitskii [10]. This approach aims to spread the initial cluster centers to reduce randomness and speed-up the convergence. The first cluster center is randomly selected from the data points, after that each subsequent cluster center is chosen from the data points with probability proportional to its squared distance from the closest existing cluster center. This method yields considerable improvement in the final error of K-means, furthermore this seeding lowers the computation time. This is why we used this initialization technique, since kernel K-means tries to minimize the same error function and thus the effects are expected to be similar.

Algorithm 1 Kernel K-means

input: data vectors: $X = \{x_1, x_2, \dots, x_n\}$; number of clusters: K

Initialize cluster centers $z_l;\ l = 1 \dots K$

1. Compute distances of each data point to all cluster centers by using Equation 11

2. Assign each data point to the closest cluster center

3. Update cluster centers based on Equation 10

4. If not converged, go to step 1, otherwise return the clusters and calculate E by using Equation 12

Algorithm 2 Plusplus cluster center initialization

input: data vectors: $X = \{x_1, x_2, \dots, x_n\}$; number of clusters: K

1. Randomly choose a data point from X as first cluster center z_1

2. Calculate $Dist(x_i)$ for all $x_i \in X$, which denotes the distance to the closest cluster center

3. Take a new center z_l, choosing $x_i \in X$ with the following probability: $\frac{Dist(x_i)^2}{\sum_{x_i \in X} Dist(x_i)^2}$

4. Repeat step 2-3 until K centers are selected

4.3 MMKK++

Our proposed algorithm, the *min-max kernel K-means plusplus (MMKK++)* focuses on estimating the cluster size that is optimal for the respective input data. We combined the approaches described in the previous sub-sections to determine the "goodness" of a candidate cluster number: the strength of the clustering was evaluated on the results that were given by kernel K-means++ clustering. We defined a minimal cluster number ($minK$) and a maximal cluster number ($maxK$), then each candidate K' value was examined between them. Finally, we chose the clustering result with the lowest corresponding $vRDI$ measure.

Due to the randomness of K-means, and thus kernel K-means, cluster size prediction is not deterministic. However according to the LLN (law of large numbers) the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed. It

is an important law for random events, since it "guarantees" stable long-term results. In order to estimate K, our algorithm runs multiple times, and the expected value comes from a large number of observations. Min-max kernel K-means++ aims to further minimize the error function E (given in Equation 12) over the input data to select the most appropriate clustering, which will be the final clustering of the input data. This is necessary even using plusplus initialization, because it is not eliminating the randomness of the initial cluster center distribution. After estimating the cluster number K, the algorithm applies kernel K-means++ l times on the data and chose the clustering that gave the lowest error. Algorithm 3 shows a pseudo code of our proposed approach (MMKK++).

Algorithm 3 MMKK++: min-max kernel K-means plusplus

input: data vectors: $X = \{x_1, x_2, \dots, x_n\}$; integers: $minK, maxK, m, l$
Let LLN_Matrix be a matrix with m rows and $maxK - minK + 1$ columns
for $\forall\ i\ =\ 1 \dots m$
    for $\forall\ K' = minK \dots maxK$
        Plusplus initialization of cluster centers
        Perform kernel K-means on $X = \{x_1, x_2, \dots, x_n\}$ with K' as cluster number
        Calculate $vRDI$ metric
        Save its value into $LLN\_Matrix(i, K')$
    end for
end for
Estimate the cluster size (K) based on the LLN_Matrix
Introduce $minE = \infty$
for $\forall\ i\ =\ 1 \dots l$
    Plusplus initialization of cluster centers
    Perform kernel K-means on $X = \{x_1, x_2, \dots, x_n\}$ with K as cluster number
    Let C be the cluster centers
    Let A be the assignment of the data vectors
    if $E < minE$
        $minE = E$
        Save the $\{C, A\}$ pair
    end if
end for
The saved $\{C, A\}$ defines the final clustering

We define a matrix called LLN_Matrix (see Equation 13), which is actually a memory and every row of it stores the corresponding $vRDI$ measure for each candidate cluster number K'. This matrix has as many rows as many runs (m) are performed to approximate the expected value of the optimal cluster number.

$$LLN\_Matrix(i, K') = vRDI_{K'} \tag{13}$$

where $i = 1 \dots m,\ j = minK \dots maxK$ and $vRDI_{K'}$ is the strength of clustering with cluster number K'. MMKK++ predicts the cluster size based on this memory matrix. As we mentioned in Section 4.1, lower $vRDI$ metric implies better clustering result, thus an optimal cluster number K can be estimated from each row of LLN_Matrix; i.e. K_i is the cluster number with the lowest corresponding $vRDI$ measure in the i^{th} run, as can be seen in Equation 14.

$$K_i = \left\{ K^* \,\middle|\, vRDI_{K^*} = \min_{K' \in \theta} \{vRDI_{K'}\} \right\} \tag{14}$$

where θ denotes the set of all possible cluster numbers and $i = 1 \dots m$. We define 3 different techniques to calculate the final cluster number, which are the following:

- **freq K:** the most frequently occurring K_i in m observations
- **avg K:** the rounded average of all K_i in m observations
- **avg Metric:** K_i that corresponds to the minimum of column-wise rounded averages of LLN_Matrix

5 Experimental Results

5.1 Experimental environment

In this section we present the experimental results of our proposed min-max kernel K-means++ approach. We compared our method with two additional techniques, the Silhouette coefficients Rousseeuw [47] and the cluster validity [52]. Both of these competitor methods are internal evaluation techniques, and in order to test them the calculation of $vRDI$ metric in MMKK++ were replaced with one and the other. This means that we actually tested these internal evaluation approaches in our proposed algorithmic environment.

$$validity = \frac{\underset{1 \le C_l \le K}{\text{avg}} \; D'(C_l)}{\underset{1 \le C_l < C_j \le K}{\min} D(C_l, C_j)} \tag{15}$$

$$Dunn = \frac{\underset{1 \le C_l < C_j \le K}{\min} D(C_l, C_j)}{\underset{1 \le C_l \le K}{\max} D'(C_l)} \tag{16}$$

As can be seen in Equation 15 above the cluster validity is the ratio of the average of all intra-cluster distances and the minimum of the inter-cluster distances, thus it throws away additional information about the remaining cluster distribution. Moreover, they defined the distance between two clusters as the distance of their centroids. This cluster validity is a variant of Dunn index [15], because the composition of these measures are similar (see Equation 15-16), however validity uses the average intra-cluster distances instead of the maximum one; and it explicitly defines the calculation procedure of intra- and inter-cluster distances. Therefore our $vRDI$ measure can also be considered as a variant of Dunn index.

Table 2 Cardinality of the clusters in the Plant10 and Cal10 test sets

	Cluster #1	Cluster #2	Cluster #3	Cluster #4	Cluster #5
Plant10	51	23	41	19	82
Cal10	27	19	22	46	26
	Cluster #6	Cluster #7	Cluster #8	Cluster #9	Cluster #10
Plant10	54	45	48	73	49
Cal10	222	45	90	53	60

We used four image sets in our experiments: the first two are subsets of different popular image collections (see example images in Figure 1) and both of them consist of 10 clusters. The first test set (called Plant10) contains 485 images, which were selected from the training data of PlantCLEF [28][20][49] competition published by the LifeCLEF campaign under ImageCLEF. The second test set (called Cal10) is a subset of the Caltech101 [31] and Caltech256 [23] image collections and contains 610 images. In Table 2 we summarized the distribution of the images between the clusters. We used the union of the Plant10 and Cal10 images as third test set (called Merged20) which is a real heterogeneous collection, since it includes pictures of leafs, flowers, food, vehicles, animals and people. Furthermore, some of the images are drawings and the rest are photos as can be seen in Figure 2, the first row corresponds to the "ying-yang" cluster, which is almost entirely consists of drawings; on the other hand, the second row shows some example images from the "ibis" cluster where the images are photos; finally, the "pizza" cluster in the third row is a mixed cluster from this point of view, because it is half drawing and half photo. The fourth test set was the compete Caltech101 image set without the "noise" category, so it contained a total of 8677 images from 101 categories (clusters).

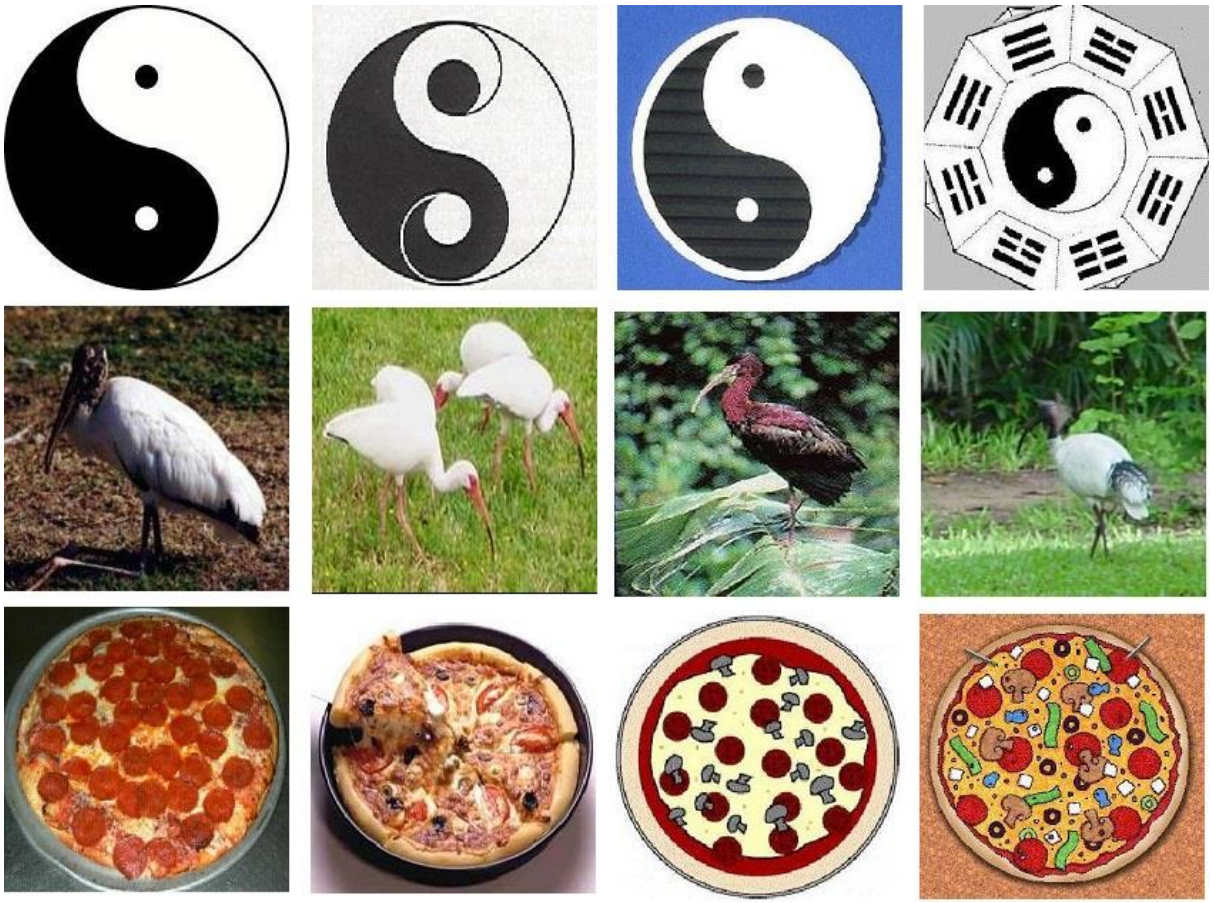

Figure 2 Heterogeneous images from the test sets

5.2 New metrics for evaluation of the cluster numbers

We used Rand Index (RI) [45] to evaluate how similar the predicted clusters (returned by the algorithm) were to the ground truth clustering. This metric is well-known and commonly used to measure the percentage of the correct decisions made by the algorithm. However, it cannot tell us any information about the number of clusters or the adequacy of this number. The easiest way for this is the direct comparison of the "real" cluster number and the predicted cluster number, but it is a rather raw technique since it only gives us a binary decision and the magnitude of the difference in case of mismatch. Therefore, we introduce a new metric the Cluster Number Indicator based on Rand Index (CNI-RI), which is capable of measuring this in a more sophisticated way, as can be seen in the following Equation.

$$CNI\text{-}RI = \frac{RI_K}{\max\limits_{K' \in \theta}\{RI_{K'}\}} \tag{17}$$

where K denotes the estimated cluster number, θ denotes the set of all possible cluster numbers and RI_K denotes the Rand Index with K clusters. The value of CNI-RI is 1 if the predicted cluster number gives the highest RI among the candidates. Note that the cluster number with the highest RI and the "real" cluster number is not necessarily the same, because this metric takes the clustering error into consideration. In many cases a clustering algorithm is not capable to perfectly cluster the data, even if the value of K is known; therefore it is possible that the algorithm achieves a higher percentage of correct decisions with a different K. To calculate this metric we evaluated the Rand Index for each candidate K' values for each test sets.

Furthermore, we define an adjusted scale of RI values for each test set, based on the evaluation of the metric; i.e. the set of values were transformed from the 0-1 scale to the min-max RI scale. Let us denote the adjusted RI (aRI) of K with aRI_K, it can be calculated as described in Equation 18.

$$aRI_K = \frac{RI_K - \min_{K' \in \theta}\{RI_{K'}\}}{\max_{K' \in \theta}\{RI_{K'}\} - \min_{K' \in \theta}\{RI_{K'}\}} \tag{18}$$

There are many other external evaluation measures, e.g. entropy, purity, mutual information and F-measure, but they are useful when the goal is to evaluate a clustering result with a fix cluster number and measure the similarity between the prediction and the ground truth. Of course it is possible to indirectly infer to the goodness of the estimated cluster number K, but it also can be misleading; for example purity is maximal if every data point can be found in a separate cluster. CNI-RI and aRI directly measure the adequacy of the cluster number K, therefore we only evaluated these metrics.

5.3 Results

In this sub-section we present the detailed results of the tests. We executed the MMKK++ with the above mentioned three different internal evaluation techniques: $vRDI$ metric, Silhouette coefficient, cluster validity. As can be seen in Algorithm 3, we used the LLN_Matrix to store these indexes for the multiple runs and for the candidate cluster numbers. During our experiments, we set both m and l to 100; in case of the first two test sets the $minK$ was set to 2 and the $maxK$ to 20, in case of the third test set we changed the $maxK$ to 40 (since the ground truth cluster size was doubled as well); finally in case of Caltech101 test set we set $minK$ to 5 and $maxK$ to 150. Based on the LLN_Matrix we used three different techniques to predict the cluster number: *freq K*, *avg K*, *avg Metric*. Thereby we had total of 9 different techniques and we tested them on each data set, then evaluated the RI, CNI-RI and aRI measures (based on the estimated K). We summarized the total results in Table 3 and Figure 3, both of them consist of four sub-parts and present the Plant10, Cal10, Merged20 and Caltech101 in the upper left, upper right, bottom left and bottom right parts, respectively.

As can be seen from the results of Plant 10 in Table 3, the "real" (10) cluster number was only predicted by using the $vRDI$ measure, however the highest CNI-RI was achieved when the images were clustered into 11 clusters, as can be seen in Figure 3. The other two methods estimated 8 and 12 as closest cluster numbers to 10, and as we can see in the last three columns of Table 3 the evaluated metrics are higher in case of $K = 12$ than in case of $K = 8$, even though both of them differ from the ground truth number by 2. This is the reason we use complex metrics to evaluate the results, because $K = 8$ and $K = 12$ could improperly be considered as similar results in terms of difference.

In case of the Cal10 test set, all of the techniques gave similar results, but in spite of the other two methods, with our proposed $vRDI$ metric the predicted cluster number is 9 with each estimation technique (*freq K*, *avg K*, *avg metric*). The results in the bottom left sub-table of Table 5 show that the Merged20 is a more difficult image set to cluster. The ground truth cluster number is 20, and the closest prediction to that number was 17, estimated by using $vRDI$ and *avg K* technique. Also, the CNI-RI was highest with 26 clusters (see bottom left sub-figure in Figure 3), what means that the clustering algorithm was not confident clustering this image collection. The last test set was even more complex due to the large number of clusters and images. The best prediction was 57 by using $vRDI$ measure and *avg K* methods. As can be seen in the following table, $vRDI$ metric with *avg K* technique gave the best (or it was one among multiple best) results in every cases, thus we highlighted the corresponding rows.

During the experiments we integrated the Silhouette and cluster validity methods into the MMKK++ algorithm, so we only tested the improved version of these techniques and compared the results to the $vRDI$. Summarizing the results in the tables we can conclude that our method outperforms the others in the literature, in spite of the fact that we built the competitor methods into our framework. Note that the difference in CNI-RI values for the different evaluation methods were not too high, because of the many TN (True Negative) decisions. Therefore, the slightest difference in values could in fact imply significant difference in the actual structure of the clusters.

Table 3 Summary of the results got on each test set. The abbreviations SC, CV, avg M and est. K represent the Silhouette coefficient, cluster validity, average Metric and estimated K, respectively.

Plant10						**Cal10**					
		est. K	RI	CNI-RI	aRI			est. K	RI	CNI-RI	aRI
SC	freq K	7	0.741	0.853	0.613	SC	**freq K**	***9***	***0.851***	***0.998***	***0.993***
	avg K	7	0.741	0.853	0.613		**avg K**	***9***	***0.851***	***0.998***	***0.993***
	avg M	12	0.828	0.953	0.877		avg M	6	0.777	0.912	0.637
CV	freq K	8	0.764	0.879	0.682	CV	**freq K**	***9***	***0.851***	***0.998***	***0.993***
	avg K	8	0.764	0.879	0.682		avg K	11	0.846	0.992	0.969
	avg M	7	0.741	0.853	0.613		**avg M**	***9***	***0.851***	***0.998***	***0.993***
vRDI	**freq K**	***10***	***0.839***	***0.965***	***0.909***	*vRDI*	**freq K**	***9***	***0.851***	***0.998***	***0.993***
	avg K	***10***	***0.839***	***0.965***	***0.909***		**avg K**	***9***	***0.851***	***0.998***	***0.993***
	avg M	8	0.764	0.879	0.682		**avg M**	***9***	***0.851***	***0.998***	***0.993***
Merged20						**Caltech101**					
		est. K	RI	CNI-RI	aRI			est. K	RI	CNI-RI	aRI
SC	freq K	11	0.793	0.84’	0.648	SC	freq K	25	0.940	0.841	0.862
	avg K	10	0.788	0.835	0.635		avg K	28	0.944	0.880	0.972
	avg M	16	0.898	0.953	0.895		avg M	22	0.930	0.816	0.958
CV	freq K	7	0.756	0.801	0.560	CV	freq K	30	0.949	0.977	0.901
	avg K	7	0.756	0.801	0.560		avg K	35	0.954	0.983	0.925
	avg M	12	0.867	0.920	0.822		avg M	39	0.955	0.984	0.929
vRDI	freq K	14	0.877	0.930	0.845	*vRDI*	freq K	55	0.962	0.990	0.958
	avg K	***17***	***0.914***	***0.969***	***0.931***		**avg K**	***57***	***0.962***	***0.991***	***0.960***
	avg M	16	0.898	0,953	0.895		avg M	49	0.959	0.988	0.947

Caltech101 is a collection of images that was created for testing classification approaches. Numerous different types of difficulties are present in this image set that makes it hard to perform unsupervised learning on it. For example, there are two separate classes named “Faces” and “Faces easy”, while both of them contains faces, therefore it is easily possible that a clustering algorithm merges these categories, because of the high similarity between the images. The predictions in case of this data set were far from the real cluster number as we can see in the bottom right sub-table of Table 3: the “best” estimations from Silhouette, cluster validity and $vRDI$ were 28, 39 and 57 respectively. Despite the poor results, $vRDI$ outperformed the competitor methods at testing of Caltech101 images.

In Figure 3 we present the results of the evaluation of the CNI-RI and aRI metrics on the test sets. As can be seen in the diagrams, the values of these metrics are very close to each other in those cases where the cluster number is close to the ground truth. Considering that the algorithm aims to cluster images based on their visual content, it is acceptable; e.g. the classification of leafs based on plant species is a way to categorize them, but two leafs are possible (even likely) to be visually similar independently of their species. From this aspect the algorithm makes no large mistake if it merges two clusters with photos of leafs from different species. When the estimated cluster number was lower than the one given as ground truth, this kind of “mis-clustering” was a typical reason for the “wrong” prediction. Another possibility is to split up clusters; e.g. it is possible that some images have different background then others in the same cluster, but the backgrounds could be grouped into two or more separate “background types”. In this case the algorithm may choose to split the cluster based on the “hidden clusters” that were discovered in the backgrounds, and this causes a higher estimated cluster number.

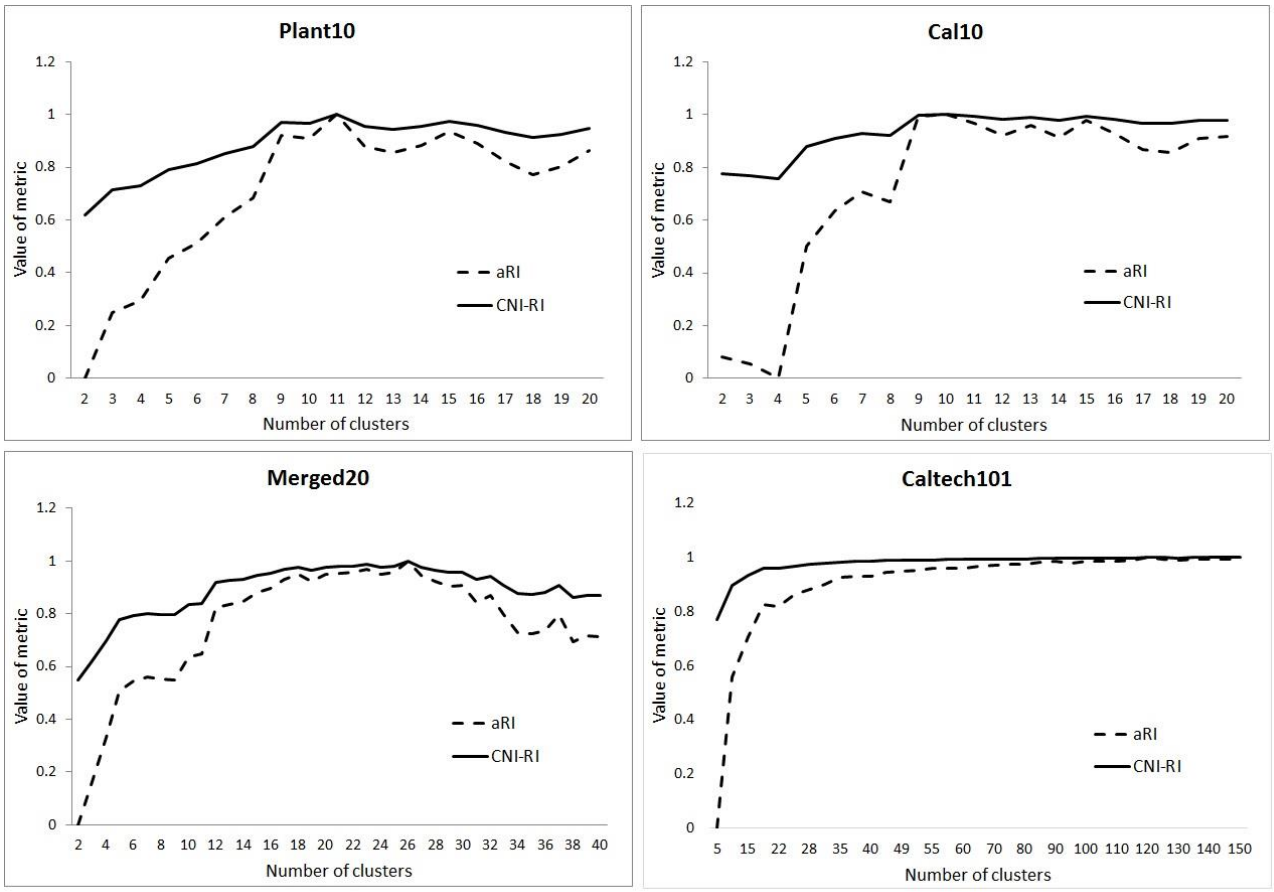

Figure 3 Visualization of the values of CNI-RI and aRI metrics got on each test set

6 Conclusion

In this paper we presented the proposed MMKK++ (min-max kernel K-means plusplus) algorithm for clustering whole images with prediction the number of clusters of unknown, heterogeneous images. We used only the visual content of the images to represent them with descriptor vectors using the state-of-the-art Fisher-vector. The mathematical representations were the input data for our clustering algorithm. This approach uses our proposed new internal evaluation procedure ($vRDI$) to evaluate the adequacy of the given cluster number. We examined some candidate cluster numbers to predict the most appropriate one. The algorithm uses the law of large numbers to estimate the value of K, so it runs multiple times and the prediction comes from a large number of observations. We conducted experiments on four test sets: two of them were subsets of larger collections, the third one was the union of the first two, and the fourth was the Caltech101 collection. Furthermore we defined two new metrics for evaluation of predicting the appropriate cluster number, which are capable of measuring the goodness in a more sophisticated way, instead of binary evaluation. We evaluated the results and compared the proposed $vRDI$ measure to two other techniques, to the Silhouette coefficient and to the cluster validity. The results showed that the $vRDI$ slightly outperforms the other methods, and it also showed that the MMKK++ can be considered as a useful tool for automatic clustering of heterogeneous images.

References

[1] Abrego N, Salcedo I, "Taxonomic gap in wood-inhabiting fungi: identifying understudied groups by a systematic survey", *Fungal Ecology*, 15:82-85, 2015. doi: 10.1016/j.funeco.2013.12.007

[2] Ahsan U, Essa I, "Clustering Social Event Images Using Kernel Canonical Correlation Analysis", *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, 814-819, 2014. doi: 10.1109/CVPRW.2014.124

[3] Borba G B, Gamba H R, Marques O, Mayron L M, "An unsupervised method for clustering images based on their salient regions of interest", *Proceedings of the 14th annual ACM international conference on Multimedia*, 145-148, 2006. doi: 10.1145/1180639.1180681

[4] Browne R P, McNicholas P D, Sparling M D, "Model-based learning using a mixture of mixtures of Gaussian and uniform distributions", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):814-817, 2012. doi: 10.1109/tpami.2011.199

[5] C Harris, M Stephens, "A combined corner and edge detector", *Proceedings of the Alvey Vision Conference*, 23.1-23.6, 1988. doi:10.5244/C.2.23

[6] Chen N, Prasanna V K, "A bag-of-semantics model for image clustering", *The Visual Computer*, 29(11):1221-1229, 2013. doi: 10.1007/s00371-013-0785-5

[7] Chen N, Prasanna V K, "Semantic image clustering using object relation network", *In Computational Visual Media (CVM 2012)*, 59-66, 2012. doi: 10.1007/978-3-642-34263-9_8

[8] Chen N, Zhou Q –Y, Prasanna V K, "Understanding web images by object relation network" *In* Proceedings of the 21st International Conference on World Wide Web, 291-300, 2012. doi: 10.1145/2187836.2187876

[9] Chitta R, Jin R, Havens T C, Jain A K, "Approximate kernel k-means: Solution to large scale kernel clustering", In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge discovery and data mining, 895-903, 2011. doi: 10.1145/2020408.2020558

[10] D Arthur, S Vassilvitskii, "k-means++: The advantages of careful seeding", *In SODA*, 1027–1035, 2007.

[11] D L Davies, D W Bouldin, "A cluster separation measure", IEEE Transactions on Pattern Analysis Machine Intelligence, 1:224-227, 1979. doi: 10.1109/tpami.1979.4766909

[12] Dalal, Navneet, and Bill Triggs, "Histograms of oriented gradients for human detection", Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 1:886-893, 2005. doi: 10.1109/cvpr.2005.177

[13] Defays D, "An efficient algorithm for a complete link method", The Computer Journal, 20(4):364-366, 1977. doi: 10.1093/comjnl/20.4.364

[14] Dhillon I, Guan Y, Kulis B, "Kernel k-means: Spectral Clustering, and Normalized Cuts", In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 551-556, 2004. doi: 10.1145/1014052.1014118

[15] Dunn J C, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3(3):32-57, 1973. doi: 10.1080/01969727308546046

[16] Everingham M, Van Gool L, Williams C K I, Winn J, Zisserman A, "The PASCAL Visual Object Classes (VOC) Challenge", International Journal of Computer Vision, 88(2):303 338, 2010. doi: 10.1007/s11263-009-0275-4

[17] Fei-Fei L, Fergus R, Torralba A, "Recognizing and Learning Object Categories", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[18] Florent Perronnin, Chris Dance, "Fisher kernel on visual vocabularies for image categorization", *Computer Vision and Pattern Recognition (CVPR)*, 2007. doi: 10.1109/cvpr.2007.383266

[19] Frey B J, Dueck D, "Clustering by passing messages between data points", *Science*, 315(5814):972-976, 2007. doi: 10.1126/science.1136800

[20] Goeau H, Joly A, Bonnet P, Selmi S, Molino J F, Barthélémy D, Boujemaa N, "LifeCLEF plant identification task 2014", *In CLEF working notes*, 2014.

[21] Gonzalez T F, "Clustering to minimize the maximum intercluster distance", *Theoretical Computer Science*, 38:293-306, 1985. doi: 10.1016/0304-3975(85)90224-5

[22] Gosselin, P. H., Murray, N., Jégou, H., & Perronnin, F., "Revisiting the fisher vector for fine-grained classification", *Pattern Recognition Letters*, *49*:92-98, 2014. doi: 10.1016/j.patrec.2014.06.011

[23] Griffin G, Holub AD, Perona P, "Caltech-256 object category data set", Technical Report, 2007.

[24] Gupta M R, Chen Y, "Theory and Use of the EM Algorithm", *Signal Processing*, 4(3):223-296, 2010. doi: 10.1561/2000000034

[25] Hahnel M, Klunder D, Kraiss K F, "Color and texture features for person recognition", *In Neural Networks. Proceedings IEEE* International Joint Conference, 1:647-652, 2004. doi: 10.1109/ijcnn.2004.1379993

[26] Hancer, E., Karaboga, D, "A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number", Swarm and Evolutionary Computation, 32, 49-67, 2017. doi: 10.1016/j.swevo.2016.06.004

[27] Ho J, Yang M H, Lim J, Lee K C, Kriegman D, "Clustering appearances of objects under varying illumination conditions", *In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1-11, 2003. doi: 10.1109/cvpr.2003.1211332

[28] Joly A, Müller H, Goëau H, Glotin H, Spampinato C, Rauber A, Bonnet P, Vellinga W P, Fisher B, "Lifeclef 2014: multimedia life species identification challenges" *In Proceedings of CLEF 2014*, 2014. doi: 10.1007/978-3-319-11382-1_20

[29] K Chatfield, V Lempitsky, A Vedaldi, A Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods", *British Machine Vision Conference*, 76.1-76.12, 2011. doi: 10.5244/c.25.76

[30] Kaufman L, Rousseeuw P J, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[31] L Fei-Fei, R Fergus, P Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 106(1):59-70, 2004. doi: 10.1109/cvpr.2004.383

[32] Lazebnik S, Schmid C, Ponce J, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, 2:2169-2178, 2006. doi: 10.1109/cvpr.2006.68

[33] Le Saux B, Boujemaa N, "Unsupervised robust clustering for image database categorization", IEEE Proceedings In *Pattern Recognition, 16th International Conference on*, 259-262, 2002. doi: 10.1109/icpr.2002.1044678

[34] Lim J, Ho J, Yang M H, Lee K C, Kriegman D, "Image clustering with metric, local linear structure, and affine symmetry", *In Computer Vision-ECCV*, 456-468, 2004. doi: 10.1007/978-3-540-24670-1_35

[35] Lowe D G, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 60(2):91-110, 2004. doi: 10.1023/b:visi.0000029664.99615.94

[36] MacQueen J, "Some methods for classification and analysis of multivariate observations", *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281-297, 1967.

[37] Mikolajczyk K, Schmid C, "Scale & affine invariant interest point detectors", *International Journal on Computer Vision* 60(1):63-86, 2004. doi: 10.1023/b:visi.0000027790.02288.f2

[38] Nene S A, Nayar S K, Murase H, "Columbia object image library (coil-20)", Technical report, Columbia University, 1996. http://www.cs.columbia.edu/CAVE/. Accessed 24 April 2015

[39] Papagiannopoulou C, Mezaris V, "Concept-based Image Clustering and Summarization of Event-related Image Collections". *In Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia*, 23-28, 2014. doi: 10.1145/2660505.2660507

[40] Paróczi Zs, Fodor B, Szűcs G, "Re-Ranking the Image Search Results for Relevance and Diversity in MediaEval 2014 Challenge", *In Working Notes Proceedings of the MediaEval 2014 Workshop*, Barcelona, Spain, October 16-17, Paper 26, 2014.

[41] Perronnin, F., Liu, Y., Sánchez, J., & Poirier, H, "Large-scale image retrieval with compressed fisher vectors", *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3384-3391, 2010. doi: 10.1109/cvpr.2010.5540009

[42] Perronnin, Florent, Jorge Sánchez, and Thomas Mensink, "Improving the fisher kernel for large-scale image classification", *Computer Vision–ECCV 2010*, 143-156, 2010. doi: 10.1007/978-3-642-15561-1_11

[43] Qiu G, "Image and feature co-clustering", In Pattern Recognition (ICPR), Proceedings of the 17th International Conference on, 4:991-994, 2004. doi: 10.1109/icpr.2004.1333940

[44] Rahmani M K I, Pal N, Arora K, "Clustering of Image Data Using K-Means and Fuzzy K-Means", *International Journal of Advanced Computer Science and Applications (IJACSA)*, 5(7):160-163, 2014. doi: 10.14569/ijacsa.2014.050724

[45] Rand W M, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association. American Statistical Association*. 66(336):846-850, 1971. doi:10.2307/2284239

[46] Reynolds D A, "Gaussian Mixture Models", *Encyclopedia of Biometric Recognition*, Springer, February, 659-663, 2009. doi: 10.1007/978-0-387-73003-5_196

[47] Rousseeuw, P. J., "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, 20:53-65, 1987. doi: 10.1016/0377-0427(87)90125-7

[48] Sibson R, "SLINK: an optimally efficient algorithm for the single-link cluster method", *The Computer Journal*, 16(1):30-34, 1973. doi: 10.1093/comjnl/16.1.30

[49] Szűcs G, Papp D, Lovas D, "Viewpoints Combined Classification, Method in Image-based Plant Identification Task", *In Working Notes for CLEF 2014 Conference*, 1180:763-770, 2014.

[50] Szűcs G, Paróczi Zs, Vincz D, "BMEMTM at mediaeval 2013 retrieving diverse social images task: Analysis of text and visual information", *In Working Notes Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19, CEUR-WS. org, ISSN 1613-0073, 2013.

[51] Tomasi C, "Estimating Gaussian mixture densities with EM: A tutorial (Tech. rep., Duke University)", *Chinese Journal of Electron Devices*, 15-18, 2004.

[52] Turi R H, Ray S, "Determination of the Number of Clusters in Colour Image Segmentation", *SCSSE Monash University*, Clayton Vic Australia, 2000.

[53] Vandersmissen B, Tomar A, Godin F, De Neve W, Van de Walle R, "Ghent University-iMinds at MediaEval 2013 Diverse Images: Relevance-Based Hierarchical Clustering", *In MediaEval*, 2013.

[54] Villalba L J G, Orozco A L S, Corripio J R, "Smartphone image clustering", *Expert Systems with Applications*, 42(4):1927-1940, 2015. doi: 10.1016/j.eswa.2014.10.018

[55] Zhou N, Fan J, "Automatic image–text alignment for large-scale web image indexing and retrieval", *Pattern Recognition*, 48(1):205-219, 2015. doi: 10.1016/j.patcog.2014.07.001