INFORMATICA, 2018, Vol. 29, No. 2, 353–369

DOI: http://dx.doi.org/10.15388/Informatica.2018.171

Double Probability Model for Open Set Problem at Image Classification

Dávid PAPP*, Gábor SZŰCS
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2, H-1117, Budapest, Hungary
e-mail: pappd@tmit.bme.hu, szucs@tmit.bme.hu

Received: May 2017; accepted: January 2018

Abstract. In this paper an exploratory classification, so called open set problem, is investigated. Open set recognition assumes there is incomplete knowledge of the world at training time, and unknown classes can be submitted to an algorithm during testing. For this problem we elaborated a theoretical model, Double Probability Model (DPM), based on likelihoods of a classifier. We developed it with double smoothing solution in order to solve technical difficulties avoiding zero values in the predictions. We applied the GMM based Fisher vector for the mathematical representation of the images and the C-SVC with RBF kernel for the classification. The last contributions of the paper are new goodness indicators for classification in open set problem, the new type of accuracies. The experimental results present that our Double Probability Model helps with classification, the accuracy increases by using our proposed model. We compared our method to a state-of-the-art open set recognition solution and the results showed that DPM outperforms existing techniques.

Key words: open world problem, open set, image classification, unknown class, double probability model.

1. Introduction to Open Set Problem

There are many works dealing with multi-class classification that incorporates labelled and unlabelled data. The reason of usage of both of them comes from the costs of the machine learning process. Namely, in some cases labelled instances are often expensive, difficult, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabelled data may be relatively easy to gather, but there has been few ways to use them. This kind of learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice. This is useful in many areas, e.g. person (Szűcs and Marosvári, 2015) and character identification (Zhu and Goldberg, 2009) in multimedia data (the latter one is solved by clustering procedure). This topic belongs to semi-supervised learning theory (Bauml *et al.*, 2013), where there are usually only small amount of labelled data with a large amount of unlabelled data. Semi-supervised learning falls between unsupervised learning and supervised learning,

*Corresponding author.

and it can learn from both labelled and unlabelled instances. This can be combined by active method, like active clustering based classification method, which clusters both the labelled and unlabelled data with the guidance of labelled instances, then queries the label of the most informative instances in an active learning phase and after that classifies the data set (Szűcs and Henk, 2015).

In all researches mentioned above the unlabelled instances belong to known classes (in the test set the new instances should be categorized into one of known classes also), but in an exploratory learning a new type of task occurs.

The task to be addressed is related to what is called open-set or open-world recognition problems (Bendale and Boult, 2015; Scheirer *et al.*, 2014), i.e. classification problems in which the recognition system has to be robust to unseen categories. Formally, given K known classes (categories) in the training set, and the task is not only to classify the new instances into known categories, but also to recognize when an instance does not belong to any of the known classes. This new category is called unknown class, thus the test set contains $K + 1$ classes. The task is an extended version of the single-label classification, because after training K classes the decision should be drawn among $K + 1$ alternatives.

After this formalization we organize the rest of this paper as follows. First we summarize the related literature in this area, then in Section 3 we present our suggestion, so called Double Probability Model (DPM) for open set problem. In the next section our solution for image classification is presented. Section 5 contains the proposed new goodness indicators for classification in this problem, and the next one presents experimental results, finally in the last one we describe our conclusion.

2. Related Work

The aim of our task was to identify data from classes that are not previously seen by a machine learning system during training. There are several works dealing with similar problem, since real-world tasks in computer vision often touch upon open set recognition (i.e. multi-class recognition with incomplete knowledge of the world and many unknown inputs). Some of those works use a new variant of SVM capable to solve the rejection problem, e.g. Support Vector Data Description (SVDD) (Tax and Duin, 2004), and the One-class SVM (Schölkopf *et al.*, 2001; Cevikalp and Triggs, 2012), RO-SVM (Zhang and Metaxas, 2006) determines the instance labels, and the rejection region during the training phase simultaneously. Furthermore, in the literature binary classification models have been proposed specifically for open set visual recognition tasks. Scheirer *et al.* (2014) developed a Compact Abating Probability model (CAP model), where the probability of class membership decreases in value (abates) as points move from known data toward open space. Based on the CAP model, they described a new variant of SVM, the novel Weibull-calibrated SVM (W-SVM) for open set recognition, which combines useful properties of statistical extreme value theory for score calibration with one-class and binary SVMs. Scheirer *et al.* (2014) claim that W-SVM outperforms their previous solutions, namely the 1-vs-Set Machine (Scheirer *et al.*, 2013) and the P_I-SVM (Jain *et al.*, 2014); besides,

they included several other approaches in their experimental evaluation, which were all outperformed by W-SVM. In this paper we compare our solution to the W-SVM and discuss the results (see Section 6.3). The 1-vs-Set Machine algorithm (Scheirer *et al.*, 2013) sculpts the decision space from the marginal distance of one-class or binary SVM with a linear kernel, so that it can reduce open space risk. This approach simply assigns class labels to instances during test. On the other hand, P_I-SVM (Jain *et al.*, 2014) is developed for estimating the unnormalized posterior probability of class inclusion. The idea is based on knowledge of rejection the large set of unknown classes even under an assumption of incomplete class knowledge if an accurate model could be built for positive data for any known class without overfitting. The solution is formulated as modelling positive training data at the decision boundary, where the statistical extreme value theory can help. Bendale and Boult (2015) defined Open World recognition and presented the Nearest Non-Outlier (NNO) algorithm which adds object categories incrementally while detecting outliers and managing open space risk.

3. Double Probability Model

3.1. *Theoretical Model*

In this section we present our Double Probability Model, which is based on likelihoods of a classifier. After training the classifier is able to give predictions with reliability values (scores) for each class. The range of the scores depends on the classifier type (sometimes it is from 0 to 1), but it can be any range; only one condition is required, namely the larger score for class C_i should represent larger likelihood to being member of class C_i. In the training set or in a validation set the instances with corresponding scores are investigated in each class. The ground truth is known in this set, so the positive elements can be selected from each class. Denote the set of scores of the positive and negative instances of class C_i by S_{P_i} and S_{N_i}, respectively. The set of negative instances of a class is the union positive instances of all other classes, as can be seen in Eq. (1).

$$S_{N_i} = \bigcup_{j \neq i} S_{P_j}. \tag{1}$$

In order to get conditional probability that a new instance belongs to class C_i provided by it's score, cumulative distribution function (CDF) of score values in S_{P_i} should be calculated, and we created a "reverse" CDF of values in S_{N_i} (see Eqs. (2) and (3)).

$$F_{P_i}(x) = p(C_i | score < x), \tag{2}$$

$$F_{N_i}(x) = p(\overline{C_i} | score > x). \tag{3}$$

Note that the sum of these probabilities is not always equal to 1 (this is not required). We constructed the so called Double Probability Model based on CDF and "reverse" CDF functions. After these calculations the predicted class should be decided at a new instance.

The focus is on the likelihood of unknown class compared with any of the known classes. Before the comparison, the probabilities of the known classes should be calculated. We get scores ($score_i$ for class i) for a new instance, as outputs of the prediction of the original classifiers, and based on them the probability of the ith class can be expressed as we describe in Eq. (4) and the expression for the probability of class C_{K+1} can be seen in Eq. (5).

$$P_{C_i} = F_{P_i}(score_i) \prod_{j=1, j\neq i}^{K} F_{N_j}(score_j), \tag{4}$$

$$P_{C_{K+1}} = \prod_{j=1}^{K} F_{N_j}(score_j), \tag{5}$$

$$P_{C_{K+1}} > \max_i \{P_{C_i}\}. \tag{6}$$

If the condition described by the inequality in Eq. (6) is true, then the decision in prediction of this new instance will be unknown class. Otherwise the prediction is based on the original classifier, i.e. the decision will be the class with the largest score. The decision in prediction of the jth test instance is formalized in Eq. (7).

$$j_{decision} = \begin{cases} K+1 & | \quad P_{C_{K+1}} > \max_i \{P_{C_i}\}, \\ \operatorname{argmax}_j \{score_j\} & | \quad \text{otherwise.} \end{cases} \tag{7}$$

3.2. *Double Smoothing*

In order to avoid zero probabilities in the product we used smoothing. In this smoothing we add dummy data to the data set, one value to the minimum and another one to the maximum of the range, so we call it double smoothing. This double smoothing method slightly modifies the cumulative distribution function, but it helps with creating non-zero CDF. At the double smoothing the number of data increases by two in each CDF. If the number of scores (i.e. the size of the validation data) is large enough, then the modified CDF (modified by smoothing) tends to the original CDF. Let us suppose that we have N data: $score_1, score_2, \ldots, score_N$. Between $score_i$ and $score_{i+1}$ the value of CDF changes from $\frac{i}{N}$ to $\frac{(i+1)}{(N+2)}$, so the difference of them can cause the smoothing error (se), described in Eq. (8).

$$se = \left| \frac{i+1}{N+2} - \frac{i}{N} \right| = \left| \frac{N-2i}{(N+2)N} \right|. \tag{8}$$

If $i = \frac{N}{2}$, then se is zero, and the maximum of the smoothing error will be at $i = N$, as can be seen in Eq. (9).

$$\max_i(se) = \frac{1}{N+2}. \tag{9}$$

The smoothing error tends to zero as N becomes infinite. Let us denote the number of elements in the original (i.e. before smoothing) CDF of the jth class by N_j. The maximal error caused by double smoothing can be derived by Eq. (10).

$$\textit{Maxerror-smoothing} = \prod_{j=1}^{K} \frac{1}{N_j + 2}. \tag{10}$$

4. Image Classification

We tested the Double Probability Model with image classification. Following the general trend, we applied the BoW (Bag-of-Words) model (Fei-Fei *et al.*, 2007; Chatfield *et al.*, 2011; Lazebnik *et al.*, 2006) for the mathematical representation of the images and we used SVM (Support Vector Machine) (Boser *et al.*, 1992; Cortes and Vapnik, 1995; Chatfield *et al.*, 2011) for classifier. We should note that the DPM can be used with any classification process, as long as it provides probability values for each possible category.

The key idea behind the BoW model is to represent an image (based on its visual content) with so-called visual code words while ignoring their spatial distribution. This technique consists of three steps: (i) feature detection, (ii) feature description, (iii) image description as usual phases in computer vision. For feature detection we used the Harris-Laplace corner detector (Harris and Stephens, 1988; Mikolajczyk and Schmid, 2004), and SIFT (Scale Invariant Feature Transform) (Lowe, 2004) to describe them. Note that we used the default parameterization of SIFT proposed by Lowe; therefore the descriptor vectors had 128 dimensions. To define the visual code words from the descriptor vectors, we used GMM (Gaussian Mixture Model) (Reynolds, 2009; Tomasi, 2004), which is a parametric probability density function represented as a weighted sum of (in this case 256) Gaussian component densities; as can be seen in Eq. (11).

$$p(X \mid \lambda) = \sum_{j=1}^{K} \omega_j g(X \mid \mu_j o_j), \tag{11}$$

where ω_j, μ_j and o_j denote the weight, expected value and the variance of the jth Gaussian component respectively, furthermore $K = 256$. We calculated the λ parameter with ML (Maximum Likelihood) estimation by using the iterative EM (Expectation Maximization) algorithm (Dempster *et al.*, 1977; Tomasi, 2004). We performed K-means clustering (MacQueen, 1967) 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 (see Eq. (12)) was calculated (Perronnin and Dance, 2007; Reynolds, 2009). These vectors were the final representations (image descriptor) of the images.

$$F = \nabla_\lambda \log p(X \mid \lambda) \tag{12}$$

where $\log p(X \mid \lambda)$ is the probability density function introduced in Eq. (11), X denotes the SIFT descriptors of an image and λ represents the parameter of GMM ($\lambda = \{\omega_j \mu_j o_j | j = 1 \dots K\}$).

For the classification subtask we used a variation of SVM, the C-SVC (C-support vector classification) (Boser *et al.*, 1992; Cortes and Vapnik, 1995) with RBF (Radial Basis Function) kernel. The one-against-all technique was applied to extend the SVM for multi-class classification. We used Platt's (Platt, 2000) approach as probability estimator, which is included in LIBSVM (A Library for Support Vector Machines) (Chang and Lin, 2011; Huang *et al.*, 2006). At this point we can decide whether to use the Double Probability Model for filtering out the test samples that possibly came from a previously unseen category, or keep the original predictions of the classifier (SVM). The CDF and reverse CDF (Eqs. (2) and (3)) can be calculated based on the class membership probabilities (in a validation set).

5. New Goodness Indicators for Classification in Open Set Problem

We call the instances with known class, and the instances with unknown class in test set by known test samples and unknown test samples, respectively. Note that the unknown classes are different from the known classes and the learning system has no information about their existence or size. The aim of the proposed model is to detect the unknown test samples with the greater accuracy. The detection part is covered by the DPM, but there are some different ways of calculating the accuracy to take these detections into consideration. The traditional accuracy (Eq. (13)) is not an appropriate indicator for measuring the goodness of the results, because it does not consider the unknown (unseen) categories; i.e. even so a test sample belongs to an unknown class it will automatically be classified into one of the known categories, what reduces the accuracy and this reduction depends on the ratio of the unknown test samples.

We introduce so called extended accuracy denoted by $Accuracy_E$: it discards the result of the test samples that are predicted as unknown, and then it calculates the accuracy on this reduced result set (see Eq. (14)). This way we are able to measure the efficiency of our proposed model by comparing it to the general case when the test samples are not filtered out.

$$Accuracy = \frac{\sum_{i \in K \cup U} I(Y_i' = Y_i)}{|K \cup U|}, \quad Y_i \in C_K \cup C_U,\ Y_i' \in C_K, \tag{13}$$

$$Accuracy_E = \frac{\sum_{i \in K \cup U} I((Y_i' = Y_i) \& (Y_i' \in C_K))}{\sum_{i \in K \cup U} I(Y_i' \in C_K)}, \quad Y_i, Y_i' \in C_K \cup C_U, \tag{14}$$

where I is an indicator function and its value is 1 if the condition in Equation 14 is true, otherwise 0. The K and U are the sets of known and unknown instances (in the test set), C_K, C_U are the sets of known and unknown classes, respectively (the unknown label is only one class, but C_K typically contains more known classes). Furthermore, Y_i and Y_i' denote the real and predicted class label of the ith image.

With the above modification we eliminate the test samples that are predicted as unknown by the DPM. While this method of calculation is good for comparison, it does not accurately reflect the classification power on the unknown; therefore a new type of accuracy (see Equation 15) is needed to evaluate such open set problem, denoted by $Accuracy_O$ (where the subscript O refers for open set problem). The decision for a test sample is drawn among $K + 1$ alternatives; thus with the $Accuracy_O$ we evaluate those decisions among $K + 1$ categories.

$$Accuracy_O = \frac{\sum_{i \in K \cup U} I(Y_i' = Y_i)}{|K \cup U|}, \quad Y_i, Y_i' \in C_K \cup C_U. \tag{15}$$

We can use the traditional recall (R), precision (P) metrics on the decisions of DPM, i.e. the percentage of the correctly filtered out images. We calculate these metrics in the following ways:

$$R_{filter} = \frac{\sum_{i \in U} I(Y_i' = Y_i)}{|U|}, \quad Y_i \in C_U, Y_i' \in C_K \cup C_U, \tag{16}$$

$$P_{filter} = \frac{\sum_{i \in U'} I(Y_i' = Y_i)}{|U'|}, \quad Y_i \in C_K \cup C_U, Y_i' \in C_U, \tag{17}$$

where $U' = \{instance_i | Y_i' \in C_U\}$.

6. Experimental Results

6.1. *Experimental Environment*

For conducting our experiments, we used the Caltech101 (Fei-Fei *et al.*, 2004) collection which consists of 8677 images from 101 categories; and we created numerous data sets by randomly sampling the classes from the total data set. These subsets fit into six different types. The training set was formed from 70% of the images in the randomly selected known classes, and the test set contains the other 30% images from the known classes complemented by all of the unknown images. The reason behind isolating the unknown images is that the learning system is not allowed to use them, so all unknown images are basically unknown test samples. We randomly selected some of the known classes, and repeated this operation 20 times, so that we can take the average of the 20 results. We chose two different unknown sets, and we defined three different numbers of known classes to sample, therefore we had total of six types, as can be seen in Table 1. As we mentioned previously, we had 20 data sets of every type, so total of 120 data sets. In the rest of the paper we will consider only the types, instead of the individual data sets one by one; later on when we present the results of the data types, we mean the averaged results of the 20 individuals.

In case of Airplanes5, Airplanes10 and Airplanes20 data sets the known categories were sampled from 100 classes, while in case of Faces5, Faces10 and Faces20 data sets

Table 1
Types of the data sets which were created by randomly selecting the categories of Caltech101 collection.

Name	Number of known classes	Unknown set
Airplanes5	5	airplanes
Airplanes10	10	airplanes
Airplanes20	20	airplanes
Faces5	5	faces + faces easy
Faces10	10	faces + faces easy
Faces20	20	faces + faces easy

they were sampled from 99 classes. This means that the number of unknown classes were 1 and 2; the number of unknown test samples were 800 and 870, respectively. We created such basic test bed where the numbers of known test samples and unknown test samples were equal, and we achieved this by randomly selecting the appropriate amount from the larger set in each test (i.e. downscaled the unknown set if the known set was smaller and vice versa). We measured the results at 11 sampling points, as the percentage of the number of unknown test samples were increasing from 0 to 50 (by 5 percent at each step, so the basic test bed was modified according to this downsampling). Note that during the training our machine learning solution can use only known images (and none of the images with unknown class), and it has not got any information about "unknown ratio in the test set", so DPM does not know how many images should be filtered out as unknown instance.

6.2. *Evaluation of Double Probability Model*

In the following we present the experimental results of our proposed Double Probability Model in six diagrams (Fig. 1) and in four tables (Tables 2–5). The diagrams in Fig. 1 show that our proposed model has a positive influence on the $Accuracy_E$; in case of each test the usage of DPM is beneficial, because it is able to filter out many unknown images.

Regarding P_{filter} the Faces20 and Airplanes20 tests were the best, in these cases the predictions of our model were approximately 75% correct (see Table 5 for details). We got the lowest P_{filter} on the Faces5 test and it was 0.563 which means that the true positive detections were higher then the false ones even in this "worst" case.

We also calculated the R_{filter} metric for every test and we experienced that increasing the percentage of the unknown test samples in the whole test set R_{filter} did not significantly change (± 0.02), thus we only present this metric at the last sampling point (i.e. when numbers of known and unknown test samples were equal): *Airplanes* : 0.741, *Faces*5 : 0.730, *Airplanes*10 : 0.551, *Faces*10 : 0.611, *Airplanes*20 : 0.234, *Faces*20 : 0.550. We can see that in the majority of tests our proposed model detected more than half of the unknown test samples, moreover, in case of Airplanes5 and Faces5 only a quarter of the unknown set remained undetected.

The tables below summarize the total result of our experiments with Double Probability Model. The meaning of the first column at the left is similar to the x-axis of the diagrams above, it represents the percentage of the unknown test samples. In addition to

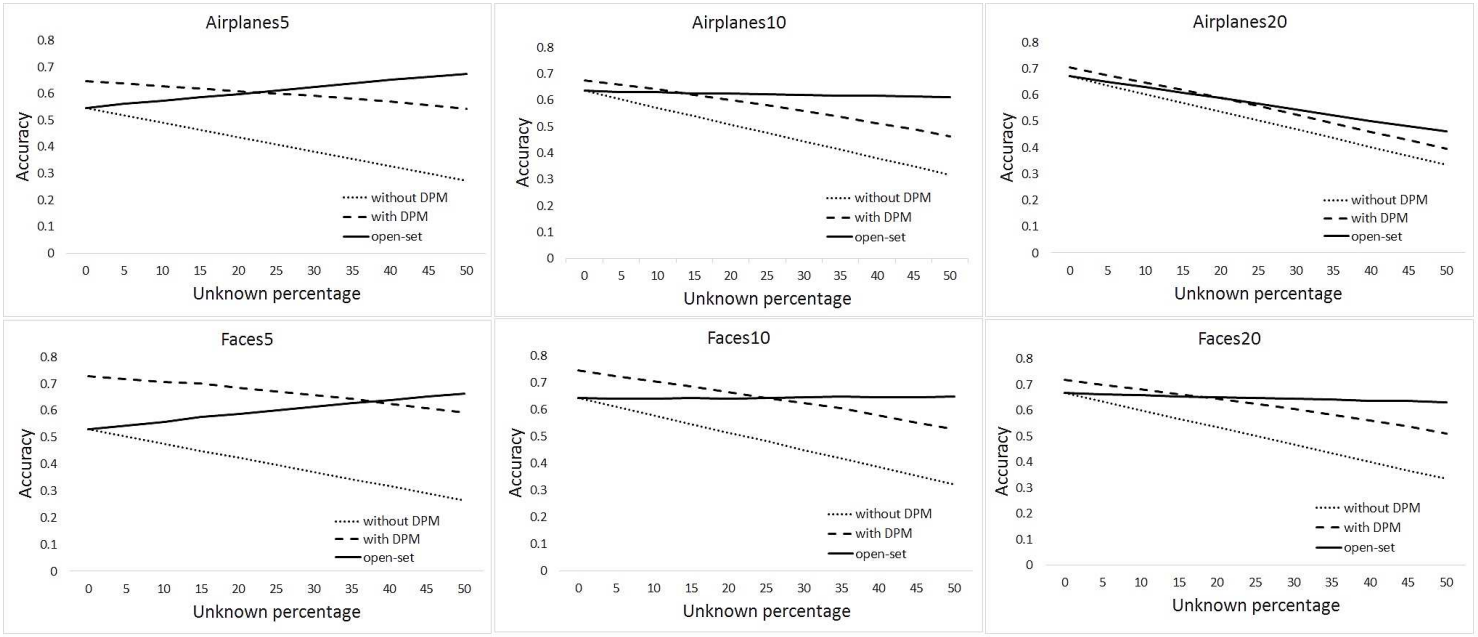

Fig. 1. Results got on the six different data set types. Each diagram shows the $Accuracy_E$ (average $Accuracy_E$ of 20 tests) with or without using our proposed Double Probability Model and the $Accuracy_O$ (average $Accuracy_O$ of 20 tests); represented as dashed, dotted and solid lines, respectively. The accuracy is on the y-axis and the percentage of the unknown test samples is on the x-axis.

Table 2
$Accuracy_E$, Q1 and Q3 metrics with or without using DPM evaluated on the results of Airplanes5 and Faces5 test data set types.

%	Airplanes5						Faces5					
	Without DPM			With DPM			Without DPM			With DPM		
	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3
0	0.547	0.419	0.634	0.647	0.557	0.720	0.531	0.398	0.630	0.730	0.638	0.807
5	0.517	0.396	0.601	0.639	0.547	0.720	0.502	0.376	0.596	0.718	0.620	0.792
10	0.490	0.377	0.569	0.629	0.534	0.716	0.475	0.357	0.567	0.707	0.587	0.792
15	0.463	0.355	0.536	0.619	0.523	0.716	0.449	0.336	0.534	0.701	0.564	0.792
20	0.436	0.334	0.505	0.610	0.502	0.716	0.423	0.318	0.502	0.686	0.530	0.792
25	0.409	0.313	0.474	0.601	0.485	0.716	0.397	0.297	0.472	0.672	0.518	0.792
30	0.382	0.292	0.444	0.592	0.464	0.716	0.370	0.277	0.440	0.657	0.494	0.792
35	0.354	0.272	0.412	0.582	0.437	0.716	0.344	0.258	0.408	0.643	0.463	0.792
40	0.327	0.251	0.380	0.570	0.412	0.716	0.318	0.238	0.378	0.625	0.427	0.792
45	0.300	0.230	0.348	0.558	0.382	0.715	0.291	0.218	0.346	0.610	0.389	0.792
50	0.273	0.210	0.317	0.543	0.361	0.702	0.265	0.199	0.315	0.593	0.344	0.792

Table 3
$Accuracy_E$, Q1 and Q3 metrics with or without using DPM evaluated on the results of Airplanes10 and Faces10 test data set types.

%	Airplanes10						Faces10					
	Without DPM			With DPM			Without DPM			With DPM		
	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3
0	0.635	0.561	0.717	0.676	0.609	0.727	0.643	0.601	0.734	0.745	0.654	0.851
5	0.602	0.532	0.679	0.658	0.607	0.709	0.609	0.570	0.695	0.723	0.620	0.830
10	0.571	0.504	0.645	0.642	0.590	0.686	0.578	0.539	0.659	0.704	0.604	0.797
15	0.539	0.477	0.608	0.620	0.556	0.668	0.545	0.509	0.623	0.686	0.588	0.767
20	0.508	0.448	0.573	0.602	0.524	0.643	0.514	0.480	0.587	0.663	0.573	0.758
25	0.476	0.420	0.537	0.580	0.503	0.606	0.482	0.450	0.550	0.642	0.567	0.734
30	0.444	0.393	0.501	0.560	0.491	0.575	0.449	0.420	0.513	0.624	0.528	0.734
35	0.412	0.365	0.465	0.536	0.469	0.559	0.417	0.390	0.476	0.604	0.494	0.711
40	0.381	0.337	0.430	0.514	0.441	0.544	0.385	0.361	0.440	0.577	0.459	0.676
45	0.349	0.309	0.394	0.490	0.414	0.521	0.353	0.330	0.404	0.551	0.414	0.655
50	0.318	0.281	0.359	0.463	0.382	0.485	0.321	0.301	0.367	0.527	0.381	0.634

the averaged metric we also included the Q1 and the Q3 (first and third quartiles) statistical indicators to give a comprehensive view about the performance of our model. Figure 1 already showed that the averaged results are better when we use DPM, but by looking at and comparing the Q1, Q3 values of $Accuracy_E$ in Tables 2–4 we can notice that even every Q1 and Q3 is higher in case of using Double Probability Model; moreover, in some cases the Q1 with DPM outperforms the Q3 without DPM. Based on these results we conclude that our proposed model efficiently filters out the unknown test samples.

As we discussed before, $Accuracy_O$ is the appropriate evaluation of the results and its value is barely changing (see left part of Table 5) as we increase the percentage of the unknown test samples. The reason for this is that our model is able to classify the known and unknown test samples as efficient as SVM classifies the known classes. For example,

Table 4
$Accuracy_E$, Q1 and Q3 metrics with or without using DPM evaluated on the results of Airplanes20 and Faces20 test data set types.

%	Airplanes20						Faces20					
	Without DPM			With DPM			Without DPM			With DPM		
	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3	AVG	Q1	Q3
0	0.671	0.607	0.701	0.705	0.644	0.757	0.668	0.604	0.710	0.718	0.652	0.763
5	0.637	0.576	0.666	0.675	0.614	0.753	0.634	0.573	0.674	0.700	0.640	0.752
10	0.604	0.545	0.630	0.648	0.589	0.724	0.601	0.543	0.639	0.682	0.635	0.744
15	0.570	0.516	0.596	0.618	0.561	0.688	0.567	0.513	0.603	0.663	0.622	0.733
20	0.537	0.485	0.560	0.589	0.533	0.654	0.534	0.483	0.568	0.645	0.594	0.723
25	0.503	0.455	0.526	0.558	0.503	0.616	0.501	0.453	0.532	0.625	0.570	0.714
30	0.470	0.425	0.491	0.526	0.471	0.590	0.467	0.422	0.497	0.605	0.542	0.704
35	0.436	0.394	0.456	0.493	0.439	0.550	0.434	0.392	0.462	0.583	0.508	0.691
40	0.403	0.364	0.421	0.460	0.407	0.513	0.401	0.362	0.426	0.560	0.472	0.678
45	0.369	0.334	0.386	0.428	0.378	0.478	0.367	0.332	0.390	0.537	0.435	0.664
50	0.336	0.303	0.351	0.395	0.347	0.434	0.334	0.302	0.355	0.509	0.400	0.640

Table 5
$Accuracy_O$ and P_{filter} metrics evaluated on the results of all types of test data sets; A and F denote Airplanes and Faces, respectively.

	$Accuracy_O$						P_{filter}					
%	A5	F5	A10	F10	A20	F20	A5	F5	A10	F10	A20	F20
0	0.547	0.531	0.635	0.643	0.671	0.668	0.000	0.000	0.000	0.000	0.000	0.000
5	0.562	0.545	0.632	0.640	0.649	0.663	0.108	0.069	0.126	0.076	0.129	0.143
10	0.573	0.559	0.632	0.641	0.630	0.660	0.186	0.132	0.235	0.151	0.268	0.263
15	0.586	0.576	0.627	0.642	0.609	0.655	0.259	0.198	0.314	0.222	0.365	0.357
20	0.598	0.587	0.625	0.640	0.589	0.652	0.325	0.251	0.392	0.284	0.445	0.445
25	0.612	0.602	0.622	0.641	0.568	0.648	0.389	0.308	0.460	0.346	0.511	0.516
30	0.626	0.615	0.620	0.644	0.545	0.645	0.447	0.364	0.523	0.409	0.562	0.578
35	0.640	0.629	0.618	0.648	0.524	0.641	0.500	0.417	0.579	0.467	0.612	0.629
40	0.652	0.640	0.616	0.646	0.501	0.638	0.547	0.466	0.627	0.517	0.654	0.677
45	0.664	0.654	0.614	0.646	0.482	0.635	0.590	0.517	0.674	0.564	0.701	0.719
50	0.675	0.663	0.612	0.647	0.461	0.630	0.632	0.563	0.715	0.610	0.739	0.757

in case of Airplanes10 the $Accuracy_O = 0.635$ at the first sampling point (where every test sample is known) and $Accuracy_O = 0.612$ at the last sampling point, while it only slightly fluctuates between them.

The overall results showed that the DPM is a useful technique to find the unknown test samples. One possible downside of the model is that it is less successful in case of small number of positive samples per category, because DPM cannot set up accurate CDF and reverse CDF if this issue is present; although this "negative" attribute is due to the way of its composition.

6.3. *Comparison with the Weibull-Calibrated SVM (W-SVM)*

In this subsection we present the results of the comparison of our proposed Double Probabilty Model and the state-of-the-art W-SVM introduced by Scheirer *et al.* (2014). We tested the W-SVM on each data sets (total of 120) and evaluated the $Accuracy_O$ and P_{filter} metrics, then compared them to the ones given by DPM. Figures 2 and 3 show the $Accuracy_O$ and the P_{filter} metrics, respectively. As can be seen in the diagrams below, DPM has better performance in case of each data set type than W-SVM, and this implies that it would (most likely) outperform all the other techniques that were tested in Scheirer *et al.* (2014). Table 6 gives a summary of the comparison by presenting the values of $Accuracy_O$ and P_{filter} for each data set types given by DPM and W-SVM.

The W-SVM is basically built up from θ one-class SVMs trained on positive examples and θ one-against-all binary SVMs, where θ denotes the number of classes. It has two parameters: one of them is δ_τ (fixed to 0.001 for all experiments in Scheirer *et al.*, 2014), which is used to adjust the minimum threshold to consider data points in CAP model, and δ_R is the level of confidence needed in the estimation of W-SVM. It is important to note that W-SVM was introduced and validated on LETTER and MNIST data sets, where the recognition rate is higher than in image collections that contain photos of outdoor, natural scenes. Therefore, we suspected that a parameter optimization is necessary before going on and testing the W-SVM on each data sets. We used a separate 10-class data set for the optimization and found that $\delta_\tau = 0.1$ and $\delta_R = 0.1$ is an appropriate setting for such type of images (the default setting of W-SVM is $\delta_\tau = 0.001$ and $\delta_R = 0.1$). We decided not to modify the value of δ_R, because by systematically increasing or decreasing this parameter, the $Accuracy_O$ and P_{filter} were not converging to a global maximum. On the other hand, increasing δ_τ resulted better detection rate on the unknown test samples up to a point ($\delta_\tau = 0.1$), where the number of false positive detections became high and it started to decrease both the $Accuracy_O$ and P_{filter} metrics. In Fig. 2, we present the results of W-SVM, which were produced by the default and the optimized settings; thereby the difference between these options were demonstrated and therefore Fig. 3 and Table 6 show only the results given by the optimized W-SVM.

We highlighted the higher values in each pair of rows in the table above and as can be seen, Double Probability Model has better performance than the Weibull-calibrated SVM at almost each case. There are only a few examples when W-SVM gave higher metrics and most of them got on the Faces5 data set. From these results we may conclude that our solution is more efficient than the W-SVM and the other methods that were previously overcome by it.

7. Conclusion

We presented our theoretical model called Double Probability Model, which is based on likelihoods of any classifier. The proposed model creates cumulative distribution functions on the positive samples and reverse cumulative distribution functions on the negative ones

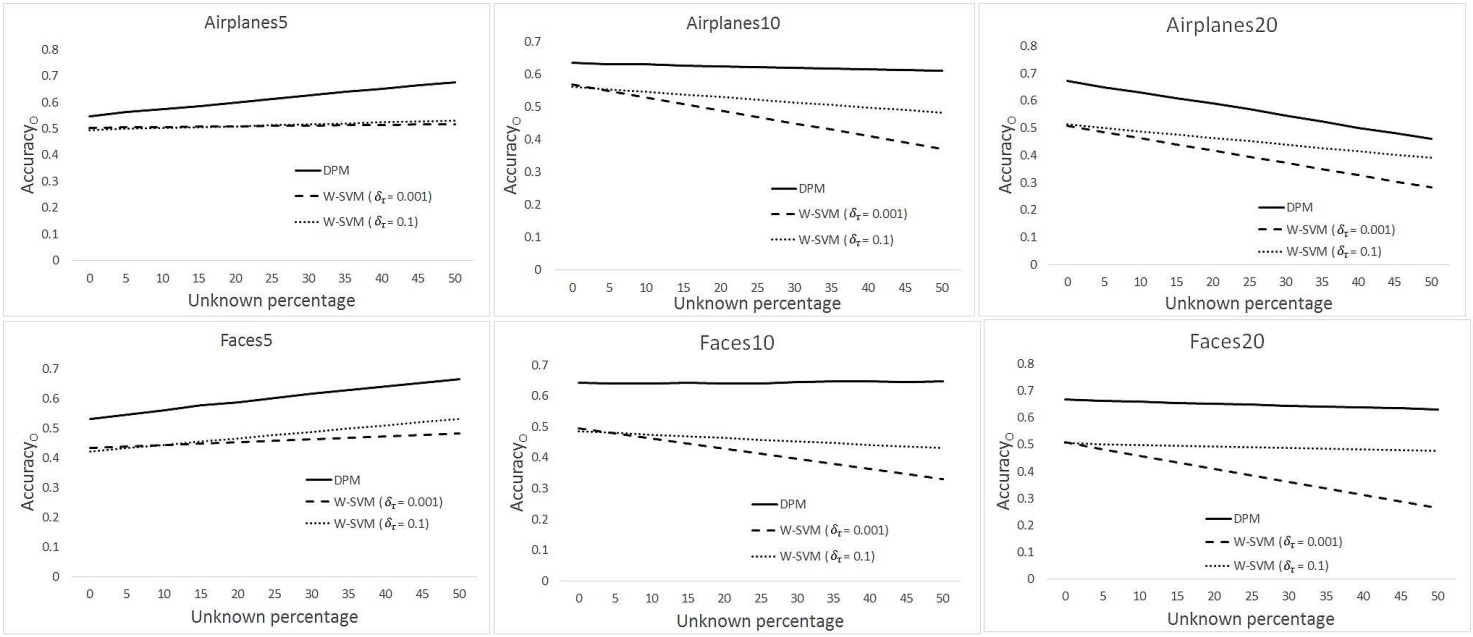

Fig. 2. Results got on the six different data set types by evaluating DPM and CAP W-SVM with $\delta_\tau = 0.001$ and $\delta_\tau = 0.1$ parameter settings; represented as solid, dashed and dotted lines, respectively. The $Accuracy_O$ (average $Accuracy_O$ of 20 tests) is on the y-axis and the percentage of the unknown test samples is on the x-axis.

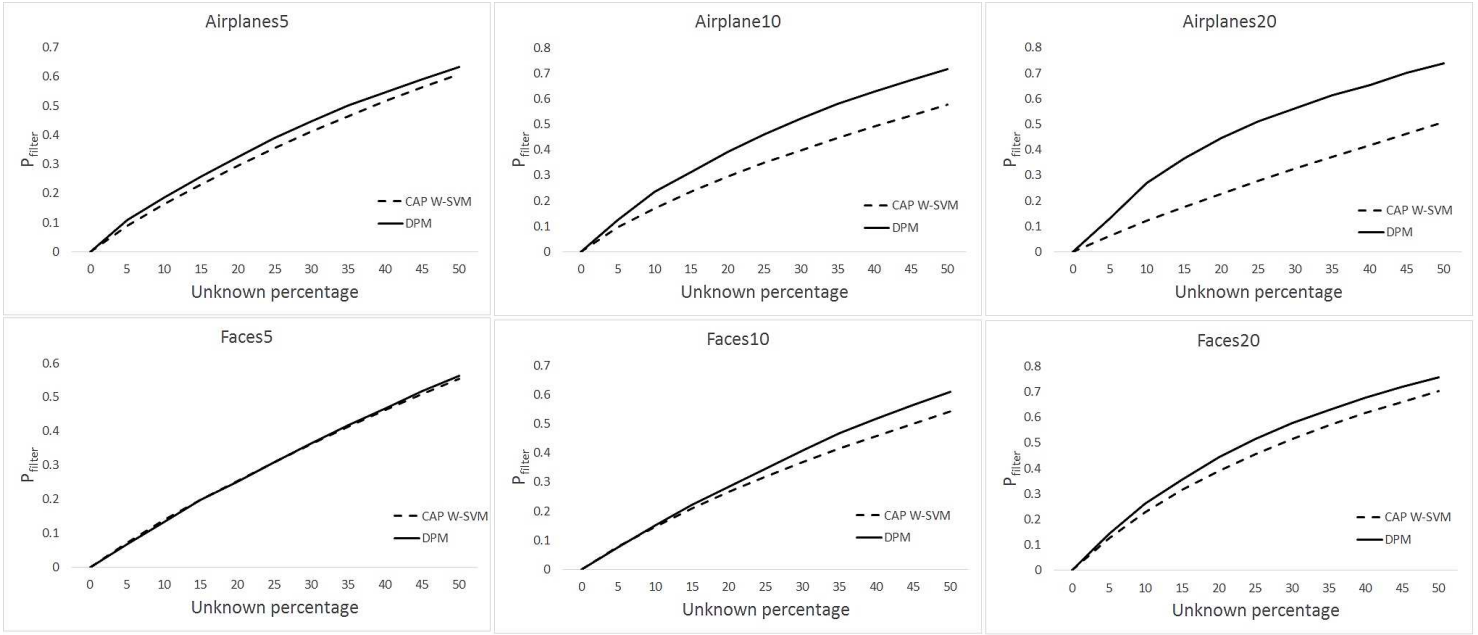

Fig. 3. Percentages of the correctly detected unknown test samples got on the six different data set types by evaluating DPM and CAP W-SVM; represented as solid and dotted lines, respectively. The P_{filter} (average P_{filter} of 20 tests) is on the y-axis and the percentage of the unknown test samples is on the x-axis.

Table 6
Comparison of $Accuracy_O$ and P_{filter} metrics got on the results of all types of test data sets between DPM and W-SVM methods; A and F denote Airplanes and Faces, respectively.

%	$Accuracy_O$						P_{filter}						Method
	A5	F5	A10	F10	A20	F20	A5	F5	A10	F10	A20	F20	
0	**0.547**	**0.531**	**0.635**	**0.643**	**0.671**	**0.668**	0.000	0.000	0.000	0.000	0.000	0.000	DPM
	0.495	0.420	0.563	0.486	0.513	0.505	0.000	0.000	0.000	0.000	0.000	0.000	W-SVM
5	**0.562**	**0.545**	**0.632**	**0.640**	**0.649**	**0.663**	**0.108**	0.069	**0.126**	0.076	**0.129**	**0.143**	DPM
	0.499	0.432	0.554	0.480	0.500	0.502	0.088	**0.073**	0.095	**0.080**	0.063	0.126	W-SVM
10	**0.573**	**0.559**	**0.632**	**0.641**	**0.630**	**0.660**	**0.186**	0.132	**0.235**	**0.151**	**0.268**	**0.263**	DPM
	0.502	0.444	0.546	0.475	0.488	0.499	0.162	**0.139**	0.170	0.147	0.121	0.228	W-SVM
15	**0.586**	**0.576**	**0.627**	**0.642**	**0.609**	**0.655**	**0.259**	0.198	**0.314**	**0.222**	**0.365**	**0.357**	DPM
	0.505	0.455	0.539	0.469	0.476	0.497	0.230	**0.199**	0.236	0.210	0.175	0.315	W-SVM
20	**0.598**	**0.587**	**0.625**	**0.640**	**0.589**	**0.652**	**0.325**	0.251	**0.392**	**0.284**	**0.445**	**0.445**	DPM
	0.509	0.465	0.531	0.464	0.464	0.494	0.295	**0.254**	0.295	0.266	0.227	0.390	W-SVM
25	**0.612**	**0.602**	**0.622**	**0.641**	**0.568**	**0.648**	**0.389**	0.308	**0.460**	**0.346**	**0.511**	**0.516**	DPM
	0.513	0.476	0.523	0.458	0.451	0.491	0.355	**0.310**	0.349	0.318	0.277	0.457	W-SVM
30	**0.626**	**0.615**	**0.620**	**0.644**	**0.545**	**0.645**	**0.447**	**0.364**	**0.523**	**0.409**	**0.562**	**0.578**	DPM
	0.516	0.487	0.515	0.452	0.439	0.488	0.412	0.362	0.399	0.368	0.325	0.516	W-SVM
35	**0.640**	**0.629**	**0.618**	**0.648**	**0.524**	**0.641**	**0.500**	**0.417**	**0.579**	**0.467**	**0.612**	**0.629**	DPM
	0.520	0.498	0.507	0.447	0.427	0.485	0.465	0.413	0.447	0.415	0.372	0.569	W-SVM
40	**0.652**	**0.640**	**0.616**	**0.646**	**0.501**	**0.638**	**0.547**	**0.466**	**0.627**	**0.517**	**0.654**	**0.677**	DPM
	0.523	0.509	0.499	0.442	0.415	0.483	0.515	0.462	0.491	0.458	0.418	0.617	W-SVM
45	**0.664**	**0.654**	**0.614**	**0.646**	**0.482**	**0.635**	**0.590**	**0.517**	**0.674**	**0.564**	**0.701**	**0.719**	DPM
	0.527	0.520	0.491	0.436	0.402	0.480	0.564	0.509	0.535	0.501	0.464	0.661	W-SVM
50	**0.675**	**0.663**	**0.612**	**0.647**	**0.461**	**0.630**	**0.632**	**0.563**	**0.715**	**0.610**	**0.739**	**0.757**	DPM
	0.530	0.531	0.483	0.430	0.390	0.477	0.608	0.554	0.576	0.541	0.509	0.702	W-SVM

(i.e. on the union of the positive samples of all other classes) for each category. Using these functions DPM estimates whether a test sample is coming from an unseen category. In order to avoid zero probabilites our model applies double smoothing. We tested the DPM at image classification, where the representation of the images were based on visual content and we used SVM for classifier. To evaluate and compare our model we defined new goodness indicators, which are extended and modified (open-set problem) variants of the general accuracy and are able to measure the influence of DPM. Our experiments showed that the proposed Double Probability Model is able to filter out a large portion of the unknown test samples, thus it increases the classification accuracy, and it outperformed the prior state-of-the-art W-SVM.

References

Bauml, M., Tapaswi, M., Stiefelhagen, R. (2013). Semi-supervised learning with constraints for person identification in multimedia data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3602–3609.

Bendale, A., Boult, T. (2015). Towards open world recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1893–1902.

Boser, B., Guyon, I., Vapnik, V. (1992). A training algorithm for optimal margin classifier. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152.

Cevikalp, H., Triggs, B. (2012). Efficient object detection using cascades of nearest convex model classifiers. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893.

Chang, C.-C., Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27.1–27.27.

Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In: *Proceedings of the 22nd British Machine Vision Conference*, pp. 76.1–76.12.

Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

Dempster, A., Laird, N., Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1), 1–38.

Fei-Fei, L., Fergus, R., Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision*.

Fei-Fei, L., Fergus, R., Torralba, A. (2007). Recognizing and learning object categories. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Harris, C., Stephens, M. (1988). A combined corner and edge detector. In: *Proceedings of the Alvey Vision Conference*, pp. 23.1–23.6.

Huang, T.-K., Weng, R.C., Lin, C.-J. (2006). Generalized Bradley–Terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7, 85–115.

Jain, L.P., Scheirer, W.J., Boult, T.E. (2014). Multi-class open set recognition using probability of inclusion. In: *European Conference on Computer Vision*, pp. 393–409.

Lazebnik, S., Schmid, C., Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 2169–2178.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281–297.

Mikolajczyk, K., Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1), 63–86.

Perronnin, F., Dance, C. (2007). Fisher kernel on visual vocabularies for image categorization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In: *Advances in Large Margin Classifiers*, pp. 61–74.

Reynolds, D.A. (2009). *Gaussian mixture models.* In: *Encyclopedia of Biometric Recognition*, pp. 659–663.

Scheirer, W., Rocha, A., Sapkota, A., Boult, T.E. (2013). Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1757–1772.

Scheirer, W.J., Jain, L.P., Boult, T.E. (2014). Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11), 2317–2324.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 1443–1471.

Szűcs, G., Henk, Zs. (2015). Active clustering based classification for cost effective prediction in few labeled data problem. *Economy Informatics*, 15(1), 5–13.

Szűcs, G., Marosvári, B. (2015). Half and fully automatic character identification in movies based on face detection. *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, 7, 80–92.

Tax, D.M.J., Duin, R.P.W. (2004). Support vector data description. *Machine Learning*, 54, 45–66.

Tomasi, C. (2004). *Estimating Gaussian Mixture Densities with EM – A Tutorial*. Technical report, Duke University.

Zhang, R., Metaxas, D. (2006). RO-SVM: Support vector machine with reject option for image categorization. In: *Proceedings of the 17th British Machine Vision Conference*, pp. 1209–1218.

Zhu, X., Goldberg, A.B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1), 1–130.

D. Papp was born in 1990 in Hungary and he has received BSc and MSc in computer science (at specialization of media informatics) from Budapest University of Technology and Economics (BME) and now he is a PhD student in computer science at the same university.

G. Szűcs was born in 1970 in Hungary. He has received MSc in electrical engineering and PhD in computer science from Budapest University of Technology and Economics (BME) in 1994 and in 2002, respectively. His research areas are data, multimedia mining, content based image retrieval, semantic search. He is an associate professor at Department of Telecommunications and Media Informatics of BME. The number of his publications is more than 80. He is the president of the Hungarian Simulation Society (EUROSIM), he is the leader of the research group DCLAB (Data Science and Content Technologies). He has earned János Bolyai Research Scholarship of the Hungarian Academy of Science some years ago.