A Flexible Ensemble-SVM for Computer Vision Tasks

Remi Trichet and Noel E. O'Connor
Insight centre for data analytics
Dublin City University, Glasnevin, Ireland
remi.trichet@gmail.com, noel.oconnor@dcu.ie secondauthor@i2.org

Abstract

This paper presents an ensemble-SVM method that features a data selection mechanism with stochastic and deterministic properties, the use of extreme value theory for classifier calibration, and the introduction of random forest for classifier combination. We applied the proposed algorithm to 2 event recognition datasets and the PASCAL2007 object detection dataset and compared it to single SVM and common computer vision ensemble-SVM methods. Our algorithm outperforms its competitors and shows a considerable boost on datasets with a limited amount of outliers.

1. Introduction

Support vector machines (SVM) are a widely used classification and regression technique [8]. By learning a separating hyperplane that maximizes the margin between the data, they show good generalization ability and good results in high dimensional space. Moreover, by automatically weighting the input features, they are robust to noise and almost immune to uninformative data.

However, the intrinsic difficulty of computer vision problems challenges this technique. There are a number of different factors that contribute to this lack of efficacy. First and foremost, is the increasing complexity of computer vision problems. Constantly, more sophisticated models are needed to represent the outliers, high intra-class and low inter-class data variability driven by new challenges. Second, high dimensionality noise from sensors exacerbates this issue, leading to features of limited reliability. Finally, discrepancy between class cardinalities frequently occur. Since SVM was originally developed for binary classification, multi-class problems are treated with several one-vs-all classifiers, creating highly imbalanced sets. By always looking for more complex, finer grain semantic instances, this trend has worsened, even leading to rare category classification sub-problems [39]. Consequently, the issue of imbalanced data has attracted growing attention from the research community [15]. [17] proved that disproportioned datasets degrade SVMs prediction accuracy, especially for non-linearly separable data. Subsequent research on these experiments [39] showed that best performance was obtained for approximately comparable class cardinalities when over-sampling the minority set.

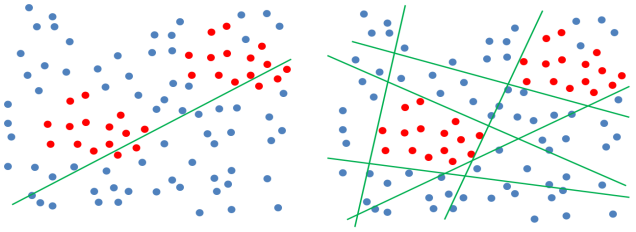

Figure 1. example illustrating how Ensemble of classifiers (*right*) can overcome the limitations of a single classifier (*left*) for 2 properly defined classes in red and blue. The green line represents the hyperplane. Best viewed in colour.

Under these circumstances, SVMs reach their limits as complex, high dimensional data, are rarely linearly separable. Indeed, as illustrated in figure 1, single classifier can fail for even well understood tasks. Consequently, they are often insufficient to tackle complex computer vision problems.

Ensemble-based methods have been shown to overcome the limitations of single classifiers in various domains [19, 14, 16, 13]. These methods combine a set of classifiers (referred to as weak classifiers throughout this paper) into a more accurate strong classifier. Therefore, ensemble-SVMs have received a lot of attention [25, 9, 27, 18, 39, 7, 12] for computer vision problems. Despite the recent rise of very effective machine learning techniques, like deep convolutional networks [2], ensemble of classifiers still attract attention as they can be applied to any classification technique, including deep convolutional networks [36]. For comparison with the state-of-the art purposes, this paper will nevertheless be restricted to ensemble-SVMs.

The method heavily relies on two factors: The selection

978-1-5090-3811-4/16/$31.00

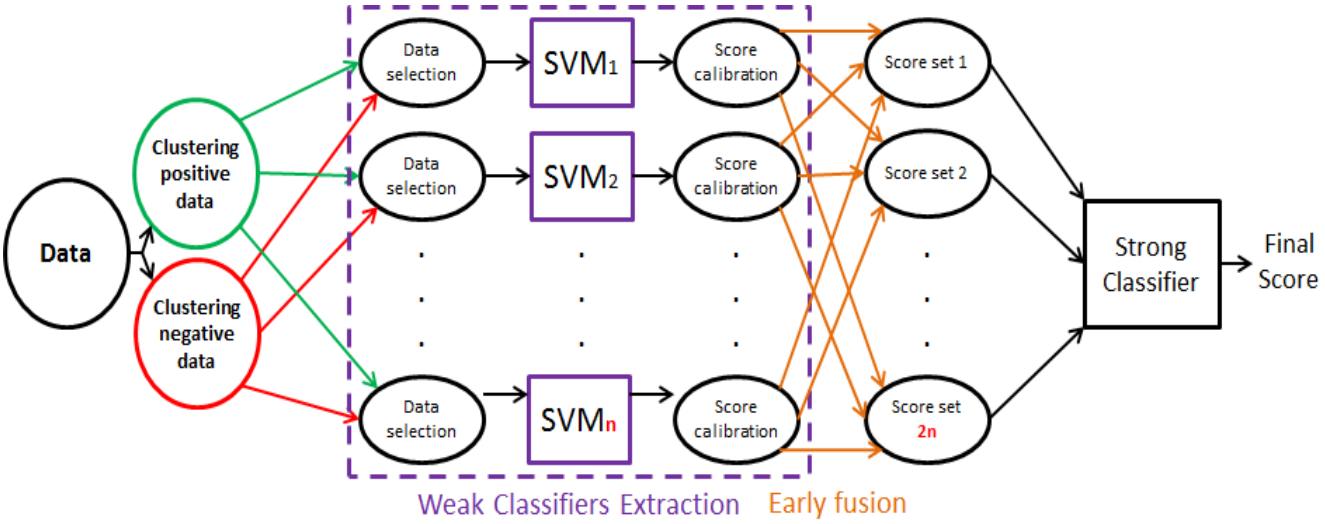

Figure 2. Method overview. Best viewed in color.

of appropriate subsets of data on which to build the weak classifiers and the type of strong classifier utilized.

So far, data selection techniques have been two-fold: stochastic [18, 12] and deterministic [9, 27, 39, 7]. Stochastic approaches are grounded on probabilistic draw to model data uncertainty, whereas deterministic ones aim to guide the selection toward areas of the feature space that require deeper attention. If the former boasts flexibility, the latter offers a more meaningful exploration of the feature space. In this paper we propose a new ensemble-SVM method that takes advantage of both these complementary approaches.

A lot of existing work aims at finding the optimal strong classifier that makes the best of its weak pendants outputs [31, 30, 10, 12, 16], with the predominance of deep-SVM [39, 1]. However, limited attention has been given to the calibration of the weak classifiers before their combination [27]. In this paper, we utilize extreme value theory (EVT) for this purpose. We also study the use of random-forests as the strong classifier.

Therefore, our work augments ensemble-SVM techniques in 3 ways:

- Proposing a new data selection mechanism, that harnesses independent Gaussian distribution centred on cluster centres. It efficiently balances deterministic and stochastic approaches while offering a solution to the imbalanced data issue.
- Applying extreme value theory (EVT) for the calibration of the weak classifier.
- Exploring further weak classifier combination methods, by introducing a random forest based strong classifier.

The rest of this paper is organized as follows. Section 2 analyses the related work and section 3 describes our ensemble-SVM method. Experiments are presented in section 4. Section 5 concludes this paper.

2. Related Work

Ensemble SVM techniques can broadly be decomposed into two steps: the extraction of weak classifiers and the combination of their outputs into a strong classifier.

Pertinent weak classifier extraction relies on insightful data subsampling of the original training set. Two types of methods have been investigated. Early attempts select data randomly using techniques like bagging [18] or genetic algorithms [12]. Extensions allowing the use of different weak classifier kernels [38] or infinite ensembles [23] have also been proposed. Despite its flexibility, this type of approach lacks guidance toward important areas of the feature space. Indeed, frequently misclassified sections of the feature space or borders between category groups should have more attention when determining the subsets. Bearing these shortcomings in mind, more recent work explored data partitions. This deterministic type of approach clusters the feature space into non-overlapping subsets. The weak classifiers are then built on these subsets, or a combination of them. Methods include hyperplane partitioning [25], binary trees partitioning [26], or typical clustering [9, 33]. Following a different direction, [7] builds up, at each iteration, a new weak classifier based on the top misclassified data. Finally, exemplar-SVMs [27] is a thought-provoking approach that brings this principle to its extreme. In this method, a swarm of weak classifiers are built using one instance of the rare class opposed to all instances

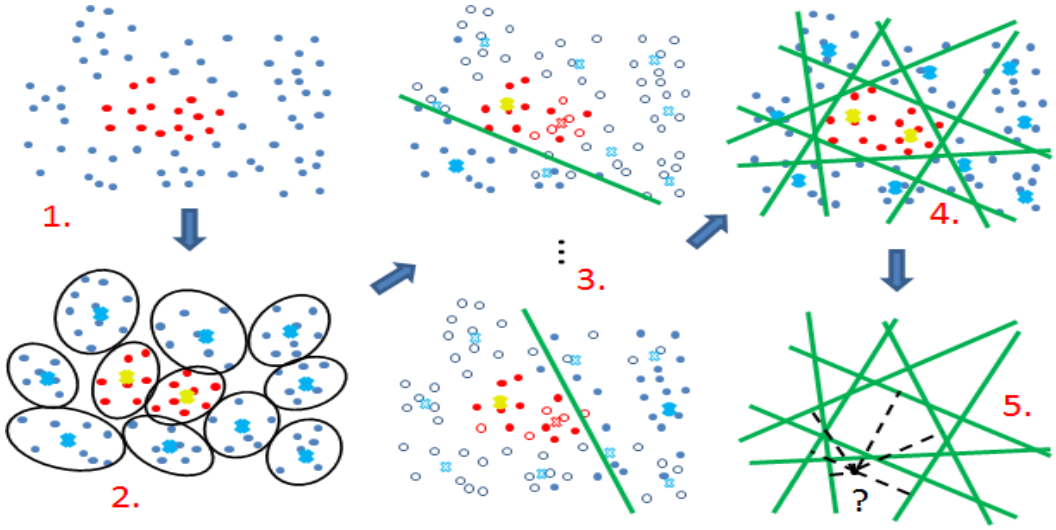

Figure 3. Overview of our weak classifier extraction method. 1- Representation of 2 categories of data (blue and red) in the feature space. 2- Clustering of positive and negative instances in the feature space. Cluster centres are depicted with crosses. 3- Classifier learning based on data selected with independent Gaussian distribution centred on cluster centres. Classifiers are depicted with green lines and unselected data with hollow points. 4- Final set of weak classifiers. 5- During training each new instance is evaluated thanks to the combination of all classifier responses. Best viewed in colour.

of other (negative) classes. The method has the advantage of efficiently dealing with rare classes (i.e., classes with low cardinalities) but treats highly imbalanced, therefore biased, data. Despite their recent success partitioning approaches lack the probabilistic foundation of the stochastic ones that allows dealing with data variability and uncertainty. Clustering alone cannot address the complex nature of computer vision problems. Consequently, the learned weak classifiers, based on rigid subsets, are suboptimal.

Our claim in this paper is that a trade-off between deterministic and stochastic approaches is needed.

Various classifier combination strategies exist [31, 30]. Besides straightforward majority voting, sum [10], weighted sum [12], boosting [16, 30], neural network [3], and deep SVM [39, 1] have been proposed. The latter was found to be the best performing among these techniques [39]. As weak classifiers are not equally useful, properly weighting them is paramount, as demonstrated in [16]. SVMs automatically learn and combine weights for each weak classifier, which explains their good performance compared to majority voting or summing. Moreover, their good generalization for a wide range of tasks and their reduced effort for parameter tuning make them a more stable choice. More recently, [25] outperformed deep SVM with a min-max modular framework, but these experiments were only conducted on text document and protein classification.

3. Ensemble-SVM

This section is divided according to the step sequence that data undergoes in our Ensemble-SVM. The first step is the determination of the weak classifiers. Positive and negative samples are clustered separately and data subsets drawn using the possible combination between clusters of positive and negative samples. Data selection is grounded on two independent Gaussian probability distributions respectively centred on the selected positive and negative clusters. We run one SVM for each subset. Second, we independently calibrate the classifier scores using extreme value theory [35] to fit the classifier scores to a probability value. Third, we augment the weak classifier set through early fusion. Finally, we combine them through a strong classifier. Figure 1 illustrates this process.

3.1. Weak Classifier

Our weak classifier generation technique, depicted in figure 3, can be perceived as a combination of kmeans-SVM [33] and bagging [4].

To obtain meaningful weak classifiers that more comprehensively cover the data configuration, we divide the feature space into a set K of groups using k-means clustering. Positive and negative examples are clustered separately with respect to their data ratio within the database, therefore yielding two non-overlapping sets P and N of positive and negative clusters, with $\{P\} + \{N\} = \{K\}$. We can then extract $|P| \times |N|$ weak classifiers based on all possible combina-

tion between positive and negative clusters.

We introduce randomness into the process as follows: We use bagging (also sometimes called bootstrapping) [4] to create the aforementioned *k* training subsets from the main training set *T*. This technique, designed to improve the accuracy and stability of machine learning algorithms for classification and regression problems, was first introduced to be used with decision trees [5], but can also be applied to other methods. The algorithm is as follows: The training set of size $|T|$ is sampled uniformly with replacement to create k new training sets. Each training set has the size $|R| < |T|$. Bagging has shown in [5] that it can give substantial gains in accuracy. The original bagging algorithm considers data points equiprobable. In our case, for a better fit to the extracted clusters, the data used to determine each weak classifier are selected based on two independent Gaussian probability distributions respectively centred on the selected positive and negative clusters P_i and N_j. This process also allows us to deal with imbalanced data. One important question at this stage is about determining an optimal ratio *R* of data to select. In order to free the system from this parameter, we extract classifiers for *k* values R_k of *R*. For all our experiments, we use 0.25, 0.45, 0.65, and 1, leading to a pool of $n = |P| \times |N| \times 3 + 1$ weak classifiers. In practice, these values can be modified without impacting the results as long as they broadly cover the scale variations.

Let us denote $C\left(P_i, N_j, R_k, V\right)$ a weak classifier determined from a distribution sampled from the parameter set (P_i, N_j, R_k, V), with R_k the percentage of data selected from the training set and *V* the validation set. In order to increase performance and stability of the system, for each weak classifier $C\left(P_i, N_j, R_k, V\right)$, we sample d distributions (P_i, N_j, R_k, V) with the same parameters and keep the best classifier $C_d\left(P_i, N_j, R_k, V\right)$. The corresponding scoring function is validated utilizing out-of-bag estimation [5] (i.e the non-selected training data).

A straightforward way to score each run would be according to a classification metric (like average precision for instance). However, in order to give more importance to the yet uncertainly classified data point, we use a variation on this metric. A weight $w\left(x\right)$ is associated to each data point x. It reflects its proper classification according to the $m\left(0 < m < n\right)$ weak classifiers that have already been determined, and is calculated as:

$$w(x) = \frac{\left(1 + \sum_{1\ldots m} conf\left(C(P_i, N_j, R_k, V), x\right)\right)}{m+1} \quad (1)$$

with $conf\left(C\left(.\right), x\right)$ the confidence value for point *x* to belong to its ground truth class according to classifier $C(.)$. We extend this definition to the output of a classifier; Finally, the weighted average precision score $M(C(.))$ is calculated as:

$$M(C(.)) = AP(w(C(.))) \quad (2)$$

Please note that this updated metric no longer represents the classifier average precision.

$$M(C(.)) \leq AP(C(.)) \quad (3)$$

We use $d = 3$ for all our experiments. In practice, increasing *d* beyond this value doesn't lead to any further performance improvement.

Dynamic programming:

An obvious consequence of imbalanced datasets is $|P| << |N|$. As *K* increases, the *K* clusters get smaller and closer, therefore leading to similar weak classifiers, and ultimately, redundant information. So, for a given positive subset P_i, we restrict *N* to its *nn*-nearest neighbours in order to reduce the computation costs. We then have a pool of $n = 3nn|P| + 1$ weak classifiers. For all our experiments, we used *nn=5*, which significantly reduces the processing time without harming the method performance.

3.2. Calibration

When undertaken, weak classifier calibration is typically done by rescaling their output values to fit the $[0\ldots 1]$ range:

$$R(C(x)) = \frac{C(x) - min(C(x))}{max(C(x)) - min(C(x))} \quad (4)$$

with $C(x)$ the output of a weak classifier $C(.)$ for the data *x*. However, this simple normalization doesn't take the SVM hyperplane position into account. Moreover, their distributions can be radically different. For better calibration, we employ the multi-attribute strategy from [34]. This method stems from extreme value theory (EVT) [35] and converts the confidence score into a positive class probability. Assuming the availability of a training set on which the best scoring values are positive, EVT shows that the positive label probability can be reliably modelled from the highest negative values, or tail of the negative values. A Weibull distribution $F(x, k, \lambda)$ is first fit on these values:

$$F(x, k, \lambda) = 1 - e^{-(x/\lambda)^k} \quad (5)$$

with *k* and λ the distribution parameter to be determined. Then, the CDF of this distribution, representing the probability of a data sample label to be positive, is used as normalization score. See [34] for details. This normalization step is similar in spirit to the one performed in examplar-SVMs [27] but differs by the use of the fitting function of eq. (5)

For our experiments, we independently normalize each weak classifier score set. Following [34] strategy, we take as tail the highest values not exceeding 10% of the total scores. In practice this calibration step improves the system performance from 1 to 2% compared to the one in eq. (4)

3.3. Early Fusion

We augment our set of weak classifiers *W* by combining their scores. As weak classifiers may display various confidence values or even opposite decisions, this combination aims to emphasize reliable outputs. Our combination strategy is grounded on consensus agreement. We assume that a reliable classifier will feature similar classifiers over the dataset for at least $p-1$ other classifiers. We define the similarity between 2 classifiers *i* and *j* as difference between their scores:

$$sim(W_i, W_j) = \phi(W_i, W_j) \tag{6}$$

with $\phi(W_i, W_j)$ a comparison metric. Any comparison metric can be used for this purpose. In our experiments, we employed the χ^2 distance.

Therefore, for each classifier, we look for its *p-1* closest classifiers and define this classifier subset as $S(W_i)$.

$$S(W_i) = \{W_k | k = 1 \dots p \quad \forall W_j \in W sim(W_i, W_k) \leq sim(W_i, W_j)\} \tag{7}$$

Then, we associate each data sample to its *q* closest subsets. The distance between a data point *d* and a classifier subset $S(W_i)$ is computed as the variance $\sigma(d, S(W_i))$ of the data point *d* scores over $S(W_i)$ subset.

$$\sigma(d, S(W_i)) = \frac{1}{p} \sum_{k=1}^{p} (W_{k,d} - \mu)^2 \tag{8}$$

with μ the mean score of data point *d* over $S(W_i)$ subset. Note that this similarity will be based on only *p* values. be *S(d)* this set of classifier subsets for point *d*. The subset scores are then calculated as follows:

$$S(W_i)_d = \sum_{k=1}^{p} W_{k,d}/p \quad if \quad W_k \in S(d) \qquad \sum_{k=1}^{n} W_{k,d}/n \quad if \quad W_k \notin S(d) \tag{9}$$

For all our experiments, we used *p=5* and *q=3*. the pseudo code in algorithm 3.3 recapitulates this early fusion process.

We then have a final weak classifier pool of cardinality $n = 6nn|P| + 1$.

3.4. Strong Classifier

We experimented with two types of strong classifiers. First, we combine weak classifiers with a deep SVM that have been shown to outperform other combination methods in [39]. It consists in using another SVM to aggregate the output of several SVMs. More formally, let *n* be the number of weak classifiers, and let *c* be the strong classifier of upper-layer SVM. The upper layer SVM is trained on a held-out set, which is sampled from the training set. The strong classifier *cSVM(x)* for a test vector *x* is determined by $cSVM(x) = c(C_1(x), \dots, C_n(x))$.

Second, we employed random forests [5]. Random forest is a widely used ensemble learning technique harnessing the output of multiple decision trees. It was first proposed to solve the classification problem [5], and was later extended to handle regression problems. In this paper, the tree construction is grounded on typical entropy optimization *S*:

$$S = \sum_{k=0}^{n} -p_i log(p_i) \tag{10}$$

And the final confidence score for instance *x* is obtained by voting:

$$P(C_i(x)) = \frac{1}{T} \sum_{t=0}^{T} c_t(C_i) \tag{11}$$

where *T* is the forest size, $c_t(C_i)$ is the count for category C_i at the leaf node of the t^{th} decision tree. The parameter set for Random Forest classifiers includes the number of decision trees *T*, the number of sampled feature dimensions N_f and the max tree depth *D*. They were selected by measuring out-of-bag errors (OOB) [5]. It was computed as the average of prediction errors for each decision tree, using the non-selected training data.

Data: a set of *n* weak classifiers W_i with weak classifier score for the data point d, $W_{i,d}$
Result: The addition of *n* new weak classifiers
initialization;
for *each* W_i **do**
  for *each* $W_j \quad j = 1 \dots n; \quad j \neq i$ **do**
    compute the similarity $sim(W_i, W_j)$ between classifiers i and j with eq. (6);
  end
  Find its *p* classifiers with the closest similarity ;
end
for *each data sample d* **do**
  Associate d to its *q* closest subsets with eq. (8) ;
end
for *each new classifier* $S(W_i)$ **do**
  Compute the new classifier scores with eq. (9) ;
end

Algorithm 1: Weak classifier early fusion.

Method type	UCFsports		Youtube		PASCAL VOC 2007	
Single SVM	DT+BoW[37]	88.2%	DT+BoW[37]	84.2%	PHOW+BoW[6]	53.42%
	DT+FV	90.3%	DT+FV[22]	90.69%	PHOW+FV[6]	61.69%
					HoG+detection[27]	0.39%
Ensemble-SVM	DT+BoW+E-SVM	90.48%	DT+BoW+E-SVM	86.46%	PHOW+BoW+E-SVM	51.34%
(state-of-the-art)	DT+FV+E-SVM	91.47%	DT+FV+E-SVM	90.05%	PHOW+FV+E-SVM	58.95%
	ST-neighbourhoods[20]	87.27%			exemplar-SVMs[27]	22.7%
Ours (deep SVM)	DT+BoW	96.3%	DT+BoW	92.7%	PHOW+BoW	56.1%
	DT+FV	94.18%	DT+FV	91.59%	PHOW+FV	63.8%
					HoG+detection	25.2%
Ours (RF)	DT+BoW	95.34%	DT+BoW	93.46%	PHOW+BoW	55.7%
	DT+FV	95.37%	DT+FV	91.85%	PHOW+FV	63.7%
					HoG+detection	25.9%

Table 1. Comparison of our method with single classifier and existing ensemble-SVM methods. *DT* - Dense trajectories. **BoW** - Bag of Words. **FV** - Fisher vector encoding. **E-SVM** - ensemble-SVMs with bagging. **RF** - Random Forest. **detection** - object detection setting.

4. Experiments

This section first details the datasets and our experimental setup. Results and their analysis follow in the last subsection.

4.1. Datasets

We use 3 different datasets of various complexities to assess our ensemble-SVM algorithm. The UCFsports dataset [32] contains 150 videos sequences at a resolution of 720×480, depicting 9 sport actions under various viewpoints and settings: swinging, diving, kicking a ball, weight-lifting, horse-riding, running, skateboarding, high-bar swinging, golfing and walking. This is an easy dataset, featuring still videos, limited intra-class variations, significant inter-class variations and background related to the action. We follow the original setup [32] using leave-one-out cross validation for a pre-defined set of folds. Average accuracy over all classes is reported as performance measure.

The YouTube dataset [24] contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. Despite similar backgrounds and the absence of unrelated footage, the difficulties include variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination conditions. The dataset contains a total of 1168 sequences. We follow the original setup [24] using leave-one-out cross validation for a pre-defined set of 25 folds. Average accuracy over all classes is reported as performance measure.

The PASCAL VOC 2007 object recognition dataset [11] contains about 10000 images split into train, validation, and test sets, and labelled with 20 object classes. Significant noise, small objects, intra category variation, and inter-category similarities (ex: motorcycle and bicycle) make this dataset a challenge. A one-vs-all SVM classifier is learned and evaluated independently for each category. The performance is measured as mean Average Precision (mAP) across all classes.

4.2. Experimental setup

Videos are first rescaled to a 640×480 resolution. We then employed DT features [37]. Images are represented with either PHOW [28] or HoG [21] features. In the case of PHOW features, we replicated the parametrization from [6]. We employed HoG features within the object detection setup of exemplar-SVMs [27].

Features are encoded utilizing Bag-of-Words (BoW) [22] or Fisher vector (FV) coding [29]. BoW is based on k-means clustering with hard-assignment. The codebook size is 4000, determined over 500K randomly sampled feature vectors. The final histogram is then L2-normalized. SVMs with χ^2 kernels are further employed for BoW. We encode FV based on a mixture of 256 Gaussians. Each component is first independently power-normalized and the descriptors are then power- and intra-normalized. Linear SVMs are further utilized for all these runs. Due to the dataset small size, *N=30* (approximately 80 weak classifiers) is used for experiments on UCFsports. The same parameter is used with the Youtube datasets for comparison (see next subsection). N=150 (approximately 180 weak classifiers) is employed on PASCAL2007 dataset.

For reasons of brevity, we focus our experiments on comparison with the state-of the art. More extensive analysis is planned in the future. All our runs are compared with single SVM results. We also implemented a classical ensemble-SVM with bagging, which is among the most commonly employed ensemble-SVM methods. The same number of weak classifiers as our method and calibration

Dataset	UCFsports	Youtube	PASCAL
encoding	BOW	BOW	FV
strong classifier	Deep-SVM	RF	RF
N	30	30	150
category 1	100.00%	83.46%	82.72%
category 2	94.44%	93.99%	69.29%
category 3	96.00%	99.31%	54.08%
category 4	100.00%	96.22%	72.65%
category 5	91.66%	94.02%	31.11%
category 6	84.61%	87.07%	72.05%
category 7	100.00%	94.21%	84.64%
category 8	100.00%	91.89%	62.94%
category 9	100.00%	95.74%	54.97%
category 10		97.88%	49.68%
category 11		94.29%	62.07%
category 12			48.91%
category 13			82.94%
category 14			73.54%
category 15			88.63%
category 16			32.76%
category 17			55.41%
category 18			53.91%
category 19			84.95%
category 20			57.69%
mean	**96.30%**	**93.46%**	**63.75%**

Table 2. Detailed results for our best run for the UCFsports, Youtube, and PASCAL VOC 2007 datasets. The metric is Accuracy for UCFsports and Youtube datasets, AP for PASCAL VOC 2007. **BoW** - Bag of Words. **FV** - Fisher vector encoding. **RF** - Random Forest.

from eq. (4) are used for this baseline. Deep SVMs are also employed as strong classifier.

4.3. Results and Analysis

Results are presented in table 1. Detailed results for our best run is provided in table 2. Results on the UCFsports and Youtube dataset show an impressive performance with respectively 96.3% and 93.46% average accuracy. These constitute a 9.16% and 9.26% increment compared to single classifier scores, 5.82% and 7% compared to our ensemble-SVM with bagging baseline. To the best of our knowledge, they are the best results to date on these datasets. Fisher vector encoding results are slightly lower than their Bag-of-Words pendants on these benchmarks. We explain it by the datasets being close to exhaustion.

Improvement on the PASCAL VOC2007 dataset are more modest. Our ensemble-SVM method outperforms the single-SVM baseline by 2.7% and 2.1% for respectively, Bag-of-Words and Fisher vector encoding. We assume that this is due to the high amount of outliers, impacting ensemble-SVMs more strongly than on single-SVMs. Three evidences tend to confirm our intuition. First, the size of the dataset doesn't seem to be accountable for it. Indeed, the method still performs well on the Youtube dataset that features difficulties similar to UCFsports and is tested with the same value of *N*, but is 7.78 times bigger. Second, the ensemble-SVM with bagging baseline also performs poorly on this dataset. Finally, our method still performs well under the object detection setting used for exemplar-SVMs [27], even exceeding its competitor with a 3.2% margin. This object detection framework, using a sliding window approach, leads to positive samples with a reduced amount of outliers. A thorough study on the influence of outliers on ensemble-SVMs has to be undertaken for formal proof of this assumption.

Random forest strong classifier perform better than deep SVMs, on average. When tested on UCFsports dataset with 5 different values of *N* (see figure 4), its performance is 1.1% higher. However, due to slight mean improvement and occasional predominance of the latter, we would advise testing both, when faced with unknown data.

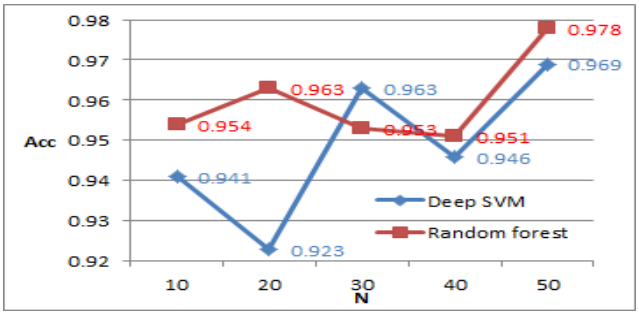

Figure 4. Influence of parameter *N* on results for the UCFsports dataset.

Figure 4 reports the study of parameter *N* impact on results for the UCFsports dataset. The conclusion matches straightforward intuition: the higher, the better. If a theoretical plateau is reached as the cluster moved toward limited cardinality (for *N=50*, we have on average 3 data samples per clusters for the UCFsports dataset), the computational cost is the limiting factor for big datasets.

5. Conclusion

This paper explored variations on ensemble-SVMs, namely a data selection mechanism with stochastic and deterministic properties, the use of extreme value theory for classifier calibration, and the introduction of random forest as classifier combiner. The method showed competitive results compared to the state-of-the art and major performance boost when applied to data with limited outliers. Future work includes a thorough study of early fusion possible combinations and their actual impact on performance. Also,

applying ensemble methods to deep convolutional networks is envisioned.

6. Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289.

References

[1] A. Abdullah, R. C. Veltkamp, and M. A. Wiering. An ensemble of deep support vector machines for image categorization. *SoCPaR*, pages 301–306, 2009.

[2] A.Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[3] S. Bengio, R. Collobert, and Y. Bengio. A parallel mixture of svms for very large scale problems. *Neural Computation*, 14(5), 2002.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] L. Breiman. Random forests. *Machine Learning*, 2001.

[6] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *BMVC*, 2011.

[7] G. Chen, M. Giuliani, D. S. Clarke, A. K. Gaschler, and A. Knol. Action recognition using ensemble weighted multi-instance learning. *ICRA*, 2014.

[8] C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20:273–297, 1995.

[9] D. Dai, M. Prasad, and L. V. Gool. Ensemble partitioning for unsupervised image categorization. *ECCV*, 2012.

[10] R. Duin, J. Kittler, M. Hater, and J. Mates. On combining classifiers. *PAMI*, 20(3), 1998.

[11] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The pascal visual obiect classes challenge 2007 (voc2007) results. *Technical report, Pascal Challenge*, 2007.

[12] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee. Classifier ensemble optimization for human activity recognition in smart homes. *ICUIMC*, 2013.

[13] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *JMLR*, 15:3133–3181, 2014.

[14] R. Hamid. Ensemble learning methods for human activity recognition. *Ensemble Learning: Methods and Applications, Springer*, pages 251–272, 2012.

[15] H. B. He and E. A.Garcia. Learning from imbalanced data. *IEEE Transaction on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[16] Y. Ivanov and R. Hamid. Weighted ensemble boosting for robust activity recognition in video. *International Journal of Machine Graphics and Vision*, 4(2), 2007.

[17] N. Japkowicz. Learning from imbalanced data sets: a comparison of various strategies. *AAAI Workshop on Learning from Imbalanced Data Sets, Tech Rep. WS-00-05*, 2000.

[18] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S.-Y. Bang. Support vector machine ensemble with bagging. *Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science*, 2388:397–408, 2002.

[19] J. Kittler, Y. Li, J. Matas, and R. Sanchez. Combining evidence in multimodal personal identity recognition systems. *Intl . Conference on Audio- and Video-Based Biometric Authentication*, 1997.

[20] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. *CVPR*, 2010.

[21] I. Laptev, M. Marszaek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008.

[22] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. *ECML*, 4(1):415, 1998.

[23] H.-T. Lin and L. Li. Infinite ensemble learning with support vector machines. *ECML*, 2005.

[24] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. *CVPR*, 2009.

[25] B.-L. Lu, X. Wang, Y. Yang, and H. Zhao. Learning from imbalanced data sets with a min-max modular support vector machine. *Frontiers of Electrical and Electronic Engineering*, 6(1):56–71, 2011.

[26] G. Madjarov, D. Gjorgjevikj, and T. Delev. Ensembles of binary svm decision trees. *ICT Innovations 2010 Web Proceedings ISSN*, pages 1857–7288, 2010.

[27] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. *ICCV*, 2011.

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[29] F. Perronnin and C. Dance. Fisher kenrels on visual vocabularies for image categorizaton. *CVPR*, 2006.

[30] R. Polikar. Ensemble learning. *Scholarpedia*, 4(1):2776, 2009.

[31] M. Ponti. Combining classifiers: from the creation of ensembles to the decision fusion. *SIBGRAPI-T*, 2011.

[32] M. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. *CVPR*, 2008.

[33] L. Rokach. Pattern classification using ensemble methods. *Singapore: World Scientific*, 2010.

[34] W. Scheirer, N. Kumar, P. N. B. Terrance, and E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. *CVPR*, 2012.

[35] W. J. Scheirer, A. Rocha, R. Micheals, and T. E. Boult. Robust fusion: Extreme value theory for recognition score normalization. *ECCV*, 2010.

[36] Y. Tang. Deep learning with linear support vector machines. *ICML*, 2013.

[37] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. *CVPR*, 2011.

[38] X. Wang, X. Liu, N. Japkowicz, and S. Matwin. Ensemble of multiple kernel svm classifiers. *Advances in AI Springers*, 2014.

[39] R. Yan, Y. Liu, R. Jin, and A. G. Hauptmann. On predicting rare classes with svm ensemble in scene classification. *ICASSP*, 2003.