18^{th} *Computer Vision Winter Workshop*
Walter G. Kropatsch, Fuensanta Torres, Geetha Ramachandran (eds.)
Hernstein, Austria, February 4-6, 2013

An Analysis of Diversity in Classification for Object Tracking *

Georg Nebehay Roman Pflugfelder
AIT Austrian Institute of Technology
{georg.nebehay.fl, roman.pflugfelder}@ait.ac.at

Walter Chibamu Peter R. Lewis Arjun Chandra Xin Yao
CERCIA, School of Computer Science
The University of Birmingham
{wcc081, p.r.lewis, a.chandra, x.yao}cs.bham.ac.uk

Abstract. *In this work, we present an analysis of the state of the art in object tracking with respect to diversity found in its main component, an ensemble classifier that is updated in an online manner. By casting object tracking as a classical online learning problem, we are able to employ established measures for diversity and performance from the rich literature on ensemble classification and online learning. We present a detailed evaluation of diversity and performance on standard sequences that have already been used for evaluation purposes in order to gain insight how the tracking performance can be improved.*

1. Introduction

We deal with the problem of single-target model-free **object tracking**, meaning that a single object is to be tracked and no a-priori information about the object is available. Many authors (e.g. [6, 10, 1, 20, 21, 14]) formulate the task of object tracking as a binary classification problem. In [10, 20, 21, 14] ensembles of multiple learners are used as binary classifiers, as they have the favourable properties of being fast to compute and easy to update in an online manner. One of the elements required for accurate prediction in ensembles is error **diversity** [5]. While ensembles of diverse classifiers have been used in object tracking methods, no measurements of diversity in these systems were reported. In this work, we fill this gap by making use of an established measure for diversity of ensemble classifiers and applying it in the context of online learning for object tracking in order to gain insight how the performance of such a system can be improved by manipulating diversity.

The contributions of this paper are as follows: We show how diversity can be measured in an object tracking method by casting it as a classical online learning problem. Furthermore we provide a detailed analysis of the state of the art in object tracking with respect to diversity and performance.

This work is structured the following way: In Sec. 2 we discuss related work in object tracking and machine learning. In Sec. 3 we illustrate and describe object tracking as an online learning problem and describe the state of the art from a conceptual point of view. In Sec. 4, we describe our experimental setup and show how we measure diversity. In Sec. 5 we present our analysis of diversity and performance. Sec. 6 concludes this work.

2. Related Work

In this section, we first review related work in online learning for object tracking, we secondly discuss the concept of diversity in ensembles and thirdly we revisit existing methods in computer vision that make use of diversity measures.

2.1. Online Learning of Object Trackers

Collins et al [6] were the first to employ binary classification in a tracking context, the two

*The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no 257906.

classes being the object and the immediate surrounding. They employ feature selection in order to switch to the most discriminative colour space from a set of candidates and use mean-shift for finding the mode of a likelihood surface, thereby locating the object. In a similar spirit, Grabner et al. [10] perform online boosting and Babenko et al. [1] use multiple instance learning in order to find the location of the object. All of these methods use a form of reinforcement learning, meaning that the prediction of the classifier is directly used to update the classifier. While this approach enables the use of unlabelled data for training, it typically amplifies errors made in the prediction phase, thus leading to a degradation of tracking performance. In [11], this problem is addressed by casting object tracking as a semi-supervised learning problem, where only the first appearance of the object is used for updating. Both Kalal et al. [14] and Santner et al. [21] employ an optic-flow-based mechanism in order to reduce the errors made in the prediction phase and demonstrate superior results.

2.2. Diversity in Ensembles

Ensembles of classifiers often provide better prediction accuracy than any of the individual members of the ensemble [5]. Rather than relying on the output of a single classifier, ensembles combine the outputs of multiple classifiers, which can be done in a variety of ways (e.g. majority voting, averaging). However for the ensemble output to be more accurate than the individual members, it is necessary that the base classifiers be diverse, i.e. in the event that the base classifiers make prediction errors on new data points, then those errors should be different [7]. Combining the ensemble members in a strategic way will ensure that the overall ensemble error can be reduced [19].

2.3. Use of Diversity in Computer Vision

There has been no explicit use of metrics for diversity in online learning applied to object tracking. However, diversity has been explicitly considered more generally in computer vision. Venkataramani and Kumar [23] use a diversity measure to choose between three different decision fusion rules for face recognition. Bertolami and Bunke [2] use diversity measures as indicators for the accuracy of ensemble classification for handwriting recognition. Frinken et al. [9] increase the diversity of a handwriting recognition system by combining different types of classifiers and show that high diversity leads to better results. Levy et al. [16] force classifiers to learn different aspects of the data by minimizing correlation between ensembles and show results on visual recognition problems.

3. State of the Art

In this section, we formally introduce the object tracking problem, and describe the state of the art algorithm in detail.

3.1. Problem

The problem of single-target model-free object tracking is defined the following way. Given a set of images $I_1 \dots I_n$ and an object location L_1, locate the object in every image I_j, with $j > 1$.

The main challenge in object tracking stems from the fact that it is difficult to find features that occur only on the object and not on the background, a phenomenon known as clutter [17].

3.2. Tracking-Learning-Detection

Kalal et al. [14] propose a solution to the problem which they call *Tracking-Learning-Detection* (TLD). TLD consists of two separate components: A **frame-to-frame tracker** that predicts the location of the object in frame I_j by calculating the optical flow between frames I_{j-1} and I_j and transforming L_{j-1} accordingly. Details about this method are in [13]. Clearly, this approach is only feasible as long as the object is visible in the scene and fails otherwise. When the object is presumably tracked correctly (according to certain criteria) the location L_j is used in order to update a **Random Fern** classifier [18] with positive training data from patches close to L_j and negative data from patches that exceed a distance. This classifier is then applied in a sliding-window manner (see Fig. 1) in order to re-initialize the frame-to-frame-tracker after failure. In TLD, two additional stages are used for classification, but we restrict our analysis to the random fern classifier.

3.3. Random Fern Classifier

The Random Fern classifier [18] operates on binary features $f_1 \dots f_n$ calculated on the raw

Figure 1: In TLD, a binary ensemble classifier is used to locate the object of interest by applying it in a sliding-window manner. The ability for multi-scale detection is achieved by scaling the size of the detection window. Image is from the SPEVI[1]dataset.

image data. These features are randomly partitioned into groups of so-called *ferns* $F_1 \dots F_m$ of size s

$$\overbrace{f_1 \dots f_s}^{F_1}, \overbrace{f_{s+1} \dots f_{2s}}^{F_2} \dots \overbrace{f_{(m-1)s+1} \dots f_{ms}}^{F_m}. \quad (1)$$

Ferns essentially are non-hierarchical trees, meaning that the outcome of each fern is independent of the order in which features are evaluated. The main reason for favouring ferns over tree is that they can be implemented extremely efficiently. Unlike the Naive Bayes classifer, where strong independence assumptions are made, features within a fern are modelled as dependent.

3.4. Features

In [18], feature vectors of the following form are used. A feature vector of size s consists of s binary tests performed on gray-scaled image patches. Each test compares the brightness values of two random pixels (See Figure 2). The locations of the tests are generated once at startup and remain constant throughout the rest of the processing. The same set of tests is used with appropriate scaling for all subwindows. Input images are smoothed with a Gaussian kernel to reduce the effect of noise.

[1] http://www.eecs.qmul.ac.uk/~andrea/spevi.html

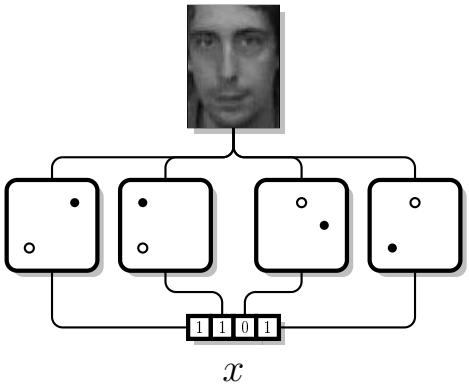

Figure 2: Feature values depends on the brightness values of pairs of two random pixels. In this case, the outcome is the binary string 1101.

3.5. Random Ferns in TLD

The posterior probability for each fern is

$$P(y = 1|F_k) = \frac{P(y = 1)P(F_k|y = 1)}{\sum_{i=0}^{1} P(y = i)P(F_k|y = i)}. \quad (2)$$

In TLD, the prior is assumed to be uniform, and the $P(F_k|y = i)$ are modelled as the absolute number of occurrences $\#p_{F_k}$ for positive training data and $\#n_{F_k}$ for negative training data. Therefore, the posterior probability becomes

$$P(y = 1|F_k) = \frac{\#p_{F_k}}{\#p_{F_k} + \#n_{F_k}}. \quad (3)$$

When $\#p_{F_k} = \#n_{F_k} = 0$, then $P(y = 1|F_k)$ is assumed to be 0 as well. Each training instance is used for training only if it was misclassified in the current frame. A decision is obtained by employing a threshold θ on the averaged posterior probabilities

$$\frac{1}{m} \sum_{i=1}^{m} P(y = 1|F_m) \geq \theta. \quad (4)$$

3.6. Non-maximal suppression

When the classifier is applied to all subwindows, ideally there would be exactly one positive at the true location of the object. In practice, several detections will occur [3]. Kalal et al. use methods commonly known as **non-maximal suppression** in order to compress these detections into a single detection. In this work, we ignore this step and focus directly on the output of the classifier.

4. Experimental Setup

We conduct experiments according to the following pattern in order to assess the diversity and the performance of the Random Fern classifier in TLD. We replace the optic-flow based tracker with manually labeled ground truth in order to exclude possible errors caused by mislabeled training examples. since we are interested only in the performance of the classifier. For each frame, we closely follow the predict-update cycle of classical online learning: First we let the classifier predict labels for all subwindows. We then measure performance and diversity using the ground truth values and update the classifier according to the misclassified examples. Each experiment is run 10 times with different seeds for the random number generator. Over these runs, the mean and standard deviation of the selected metrics for performance and diversity are reported. We use the parameters $m = 30, s = 14, \theta = 0.5$ unless noted otherwise.

4.1. Performance Measures

We use the following statistics to measure the performance, based on the occurrences of *True Positives* (TP), *False Negatives* (FN) and *False Positives* (FP) in each frame. TPs, FNs and FPs are found by comparing algorithmic output to manually annotated ground truth. Recall

$$R_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FN}_j}. \tag{5}$$

measures the fraction of positive instances that were correctly classified as positive. Precision

$$P_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j}. \tag{6}$$

measures the fraction of examples classified as positive that are truly positive. The F-measure

$$F_j = \frac{2R_jP_j}{R_j + P_j}. \tag{7}$$

as the harmonic mean combines precision and recall into a single measurement. We calculate R_j, P_j and F_j for each frame and report their average values R, P, F over the whole sequence.

As the set of subwindows is not exhaustive, there will typically be no single subwindow of the same location and the same dimension as the manual annotation. We therefore employ the measure used in the Pascal Visual Object Challenge [8] for overlap between two bounding boxes B_1 and B_2

$$overlap = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{I}{(B_1 + B_2 - I)}. \tag{8}$$

If the overlap between a manual annotation and a subwindow is larger than 0.5, then the label of the subwindow should be predicted as positive in order to be correct.

4.2. Diversity Metrics

Several measures for a quantitative assessment of diversity in ensembles have been proposed in the literature, however there is no universally agreed single definition and measure of diversity [22], thus making it hard to explicitly measure diversity and correlate a measure to higher ensemble accuracy. Brown et al. [5] and Tang et al. [22] have conducted a wide and detailed study of various diversity measures and they both concluded that there is not one unique way of measuring diversity and there is no direct or distinctive relationship between the diversity of an ensemble and its accuracy.

Among the many available diversity measures, the *Q-statistic* [15] is calculated in a pairwise manner for any two classifiers f_i and f_j:

$$Q_{i,j} = \frac{ad - bc}{ad + bc} \tag{9}$$

The symbols a, b, c, d refer to the number of times

- a : f_i and f_j are correct,
- b : f_i is correct, f_j is incorrect,
- c : f_i is incorrect, f_j is correct,
- d : f_i and f_j are incorrect.

$Q_{i,j}$ is closer to 1 if the output of the labels is less diverse and it is closer to -1 if their output is more diverse. An overall measure for the diversity of an ensemble of size n is then obtained by averaging all of the pairwise measurements

$$Q = \frac{1}{n} \sum_i \sum_j Q_{i,j}. \tag{10}$$

We employ the Q-statistic as a measure for diversity in each frame and report averaged values over

Seq.		Combination rule mean	majority
Davd	F	0.66±0	0.68±0
Jump	F	0.69±0	0.69±0
Ped1	F	0.38±1	0.41±1
Ped2	F	0.65±1	0.63±1
Ped3	F	0.81±1	0.80±1
Car	F	0.85±0	0.85±0

Table 1: Mean vs majority

the whole sequence. As the Q-statistic requires individual labels from the ensemble members, we use majority voting instead of the mean rule. As Table 1 suggests, performance is not affected by this measure.

4.3. Sequences

We employ the following six sequences that are shown in Fig. 3 for conducting our evaluation. These sequences were used in [24, 12] for evaluating object tracking methods. The sequence **David** consists of 761 frames and shows a person walking from an initially dark setting into a bright room and undergoing various changes in appearance. **Jumping** consists of 313 frames and shows a person jumping rope causing motion blur. **Pedestrian 1** (140 frames), **Pedestrian 2** (338 frames) and **Pedestrian 3** (184 frames) show pedestrians being filmed by an unstable camera. The sequence **Car** consists of 945 frames showing a moving car, exposed to low contrast recording and undergoing multiple occlusions. The appearance of the car itself stays constant over the run of the sequence.

Figure 3: The data set used for evaluation. First row: David, Jumping, Pedestrian 1. Second row: Pedestrian 2, Pedestrian 3, Car.

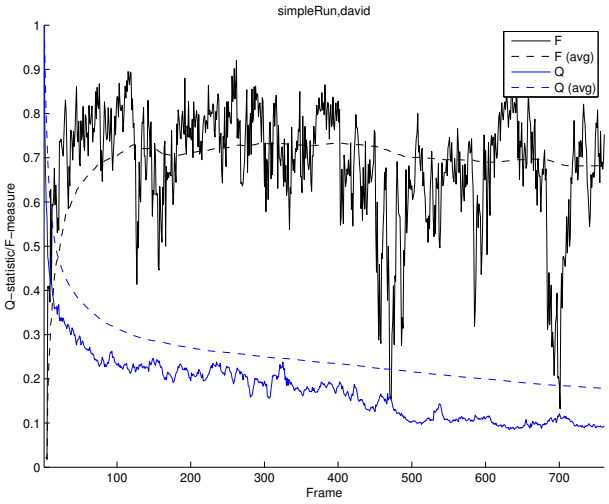

Figure 4: Single run on the sequence *David*. Q-statistic (blue), F-measure (black). Dashed lines denote the running average. Low values of Q imply high diversity. High values of F indicate good performance.

5. Diversity Analysis of TLD

In this section we report the results of the experiments we conducted. Firstly, we inspect a run on single sequence visually. Secondly, we explore the effect of varying the parameters of the system on the selected metrics. Thirdly, we artificially decrease diversity in the system and analyse the results.

5.1. Single Run

The evolution of Q and F when measured on the sequence *David* is shown in Figure 4. F increases up to frame 110, while Q continues to decrease until the end of the sequence. The decrease of Q is caused by more training data becoming availabe, which leads to more diverse classification results, since initially all classifiers are biased towards the background class. While one might be tempted to attribute the increase of F to the decrease of Q, it is actually caused by improved lighting conditions in the scene itself. In a similar fashion, performance drops around frames 480 and 700 due to perturbations in the sequence, while Q remains unaffected. These results generalize to runs on all other sequences in the dataset and support the observations already known from the literature that there is no explicit link between diversity measures and performance measures.

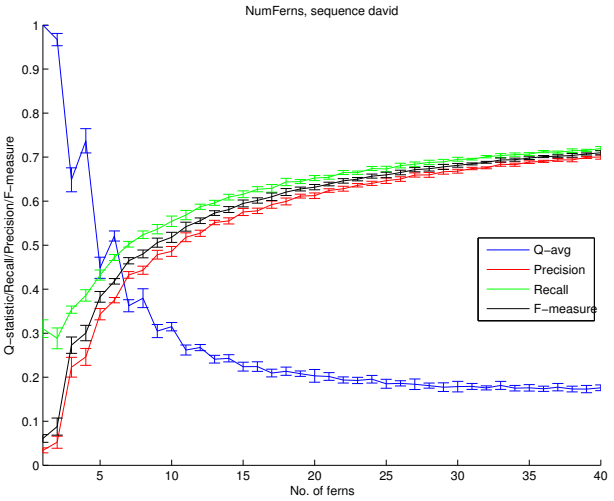

Figure 5: Both diversity and performance exhibit a convergent behaviour when the number of ferns m is increased.

		Number of ferns m			
Seq.		10	20	30	40
Davd	Q	0.31±1	0.20±1	0.18±1	0.18±1
	F	0.52±1	0.63±1	0.68±0	0.71±0
Jump	Q	0.30±1	0.22±1	0.21±1	0.22±1
	F	0.51±1	0.64±1	0.69±0	0.71±0
Ped1	Q	0.34±1	0.27±1	0.26±1	0.26±1
	F	0.30±2	0.38±1	0.41±1	0.43±1
Ped2	Q	0.28±2	0.27±2	0.28±1	0.29±1
	F	0.46±2	0.58±1	0.63±1	0.66±1
Ped3	Q	0.45±3	0.44±2	0.44±1	0.45±1
	F	0.66±2	0.76±1	0.80±1	0.81±1
Car	Q	0.38±2	0.27±1	0.28±1	0.28±1
	F	0.77±1	0.83±0	0.85±0	0.86±0

Table 2: Depending on the sequence, the limit of Q is approached at different values of m, an effect not observable for F.

		Number of features s				
Seq.		8	11	14	17	20
Davd	Q	0.51±27	0.25±9	0.19±2	0.27±1	0.37±1
	F	0.26±19	0.49±12	0.64±3	0.66±0	0.62±1
Jump	Q	0.37±18	0.18±4	0.23±1	0.35±1	0.52±1
	F	0.29±18	0.48±12	0.64±4	0.67±1	0.66±1
Ped1	Q	0.19±6	0.16±1	0.27±1	0.45±1	0.69±2
	F	0.33±7	0.44±1	0.42±1	0.29±1	0.11±1
Ped2	Q	0.12±3	0.15±1	0.28±1	0.44±1	0.63±2
	F	0.56±4	0.64±1	0.65±1	0.60±1	0.43±1
Ped3	Q	0.33±9	0.32±2	0.45±1	0.60±2	0.74±2
	F	0.51±15	0.74±3	0.79±1	0.80±1	0.79±1
Car	Q	0.53±27	0.26±6	0.29±1	0.43±2	0.58±2
	F	0.34±28	0.70±10	0.84±1	0.85±0	0.84±0

Table 3: Diversity and performance exhibit a large variation when the number of features s is small. When s is large, classifiers tend to overfit.

5.2. Effect of Parameters

The parameter m steers the number of classifiers in the ensemble. Breiman [4] proved that an ensemble of randomized decision trees does not overfit as more trees are added, meaning that performance does not decrease. It is not that clear how the parameter m affects diversity. In Figure 5 we plot Q and F against m for the sequence *David*. Increasing s leads to a convergent behaviour of Q, similar to the performance metric. Q appears to converge more quickly than the performance metrics, a hypothesis supported by the data presented in Table 2. For all sequences, Q converges more quickly than F, but it depends on the sequence when the limit of Q is approached.

The parameter s steers the complexity of each individual classifier. When s is small, the object and the background are difficult to distinguish, leading to underfitting. Due to the large number of misclassified instances in every frame, the output of the classifier essentially oscillates. As it can be seen in Table 3, this leads both to an increase of Q and a reduction of F. It can also be seen that in this case results exhibit a large variance, as the locations of the binary tests dramatically influence discriminability between classes. When s is large, the classifier overfits the training instances and small variations in the object appearance lead to the image being classified as background, unless sufficient training data is available. which is not the case for sequences *Pedestrian 1* and *Pedestrian 2*. Interestingly, the location of the global minimum of Q is dependent on the sequence.

The parameter θ directly influences recall and precision. High values of θ lead to an improvement of precision, as false positives are filtered out, and to a degradation of recall. Low values of θ lead to the inverse effect. Intuitively, high values of θ should lead to a reduction of diversity, as the output of the individual classifiers become more similar. In Table 4, the parameter θ is varied for all sequences. Surprisingly, Q decreases monotonically when θ is increased. Again, the explanation for this effect is that high values of

Seq.		Threshold parameter θ 0.1	0.3	0.5	0.7	0.9
Davd	Q	0.21±1	0.19±1	0.18±1	0.17±1	0.16±1
	P	0.28±2	0.56±1	0.67±0	0.74±0	0.74±1
	R	0.82±0	0.76±0	0.70±0	0.58±1	0.34±1
Jump	Q	0.24±1	0.22±1	0.21±1	0.21±1	0.20±1
	P	0.36±1	0.59±0	0.68±0	0.76±0	0.78±1
	R	0.85±0	0.77±0	0.70±0	0.58±0	0.35±1
Ped1	Q	0.30±1	0.27±1	0.26±1	0.26±1	0.25±1
	P	0.23±1	0.38±1	0.45±1	0.53±1	0.52±1
	R	0.53±1	0.44±1	0.38±0	0.26±1	0.14±1
Ped2	Q	0.31±1	0.29±1	0.28±1	0.27±1	0.26±1
	P	0.35±1	0.53±1	0.62±1	0.74±1	0.77±2
	R	0.71±1	0.68±1	0.65±1	0.58±1	0.45±1
Ped3	Q	0.47±1	0.45±1	0.44±1	0.44±1	0.42±1
	P	0.53±1	0.68±1	0.76±1	0.84±1	0.87±1
	R	0.92±1	0.87±1	0.83±1	0.75±1	0.57±1
Car	Q	0.31±1	0.29±1	0.28±1	0.27±1	0.26±1
	P	0.62±1	0.76±0	0.81±0	0.84±0	0.86±0
	R	0.95±0	0.92±0	0.90±0	0.85±0	0.73±0

Table 4: Increasing θ leads to an increase of diversity due to many positive instances being misclassified and used for training.

theta lead to many positive instances being misclassified, and therefore the set of positive training data becoming larger. As already observed in Sec. 5.1 this leads to a reduction of Q.

5.3. Overlapping Feature Sets

In this section we present a novel algorithm for reducing the total number of features that have to be computed by artificially decreasing diversity in the system. We modify Eq. 1 in order to create overlapping feature sets between the classifiers. More precisely, we reuse ω features of fern $(i-1) \mod m$ in fern i.

$$\overbrace{f_1 \dots \underbrace{f_{s-\omega} \dots f_s}_{\text{classifier 2}} \dots f_{2s-\omega}}^{\text{classifier 1}} \dots \quad (11)$$

In order to keep the number of different features in the system constant, we increase s by ω. In Table 5 results are shown. Clearly Q increases when ω is increased. The results of the experiments $s = 20$ and $s = 20, \omega = 6$ are identical even though the total number of features in the system is decreased from 600 to 420, thus effectively increasing the efficiency of the system by reusing computations of features.

Seq.		Feature overlap ω (s) 0 (14)	2 (16)	4 (18)	6 (20)
Davd	Q	0.18±1	0.24±1	0.30±1	0.37±1
	F	0.68±0	0.67±0	0.65±1	0.61±1
Jump	Q	0.21±1	0.30±1	0.41±1	0.51±2
	F	0.69±0	0.69±0	0.68±0	0.66±1
Ped1	Q	0.26±1	0.39±1	0.55±2	0.69±3
	F	0.41±1	0.33±1	0.22±1	0.12±2
Ped2	Q	0.28±1	0.39±1	0.51±1	0.63±2
	F	0.63±1	0.61±1	0.54±3	0.42±2
Ped3	Q	0.44±1	0.56±1	0.66±1	0.74±2
	F	0.80±1	0.80±1	0.78±1	0.77±1
Car	Q	0.28±1	0.38±2	0.48±2	0.58±2
	F	0.85±0	0.85±0	0.84±0	0.84±0

Table 5: Increasing the number of overlapping features ω and the number of features per fern s leads to a reduced number of total features that have to be evaluated while performance and diversity remain constant when compared to Table 3.

6. Conclusion and Future Work

In this work, we presented an analysis of TLD with respect to diversity and showed how it is influenced by the intrinsic parameters of the random fern classifier. Increasing m leads to more diversity. There is a value of s where diversity is maximal, but the specific value depends on the sequence. Increasing θ also increases diversity.

We showed a novel way of reducing the total number of features that have to be evaluated, while keeping performance constant. This effectly reduces the computing time of the classifier, as a less features have to be evaluated.

As only misclassified examples are used for training in TLD, the classifier highly overfits the training data. This does not to lead to a reduction in performance as long as sequences contain enough training examples. When short sequences with severe changes in appearance occur, performance is affected in a negative way.

In future work, we would like to address this issue by creating a classifier with an ability to better generalise to unseen training examples, a property that is important when applying a trained classifier in a different scenerario, e.g. in a different camera. We also plan to explicitly increase diversity in the system by making use of algorithms similar to minimal correlation learning [16].

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, Aug. 2011.

[2] R. Bertolami and H. Bunke. Diversity analysis for ensembles of word sequence recognisers. In D.-Y. Yeung, J. Kwok, A. Fred, F. Roli, and D. Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 4109 of *Lecture Notes in Computer Science*, pages 677–686. Springer Berlin Heidelberg, 2006.

[3] M. B. Blaschko. *Branch and Bound Strategies for Non-maximal Suppression in Object Detection*, volume 6819 of *Lecture Notes in Computer Science*, chapter 28, pages 385–398. 2011.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[5] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion*, 6:5–20, 2005.

[6] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[7] T. G. Dietterich. Ensemble methods in machine learning. In *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857*, pages 1–15. Springer, 2000.

[8] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[9] V. Frinken, T. Peter, A. Fischer, H. Bunke, T.-M.-T. Do, and T. Artieres. Improved handwriting recognition by combining two forms of hidden markov models and a recurrent neural network. In *Computer Analysis of Images and Patterns*, volume 5702 of *Lecture Notes in Computer Science*, pages 189–196. 2009.

[10] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260–267, Washington, DC, USA, 2006. IEEE Computer Society.

[11] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-Line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision*, volume 5302, pages 234–247, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[12] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–56. IEEE, June 2010.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-Backward Error: Automatic Detection of Tracking Failures. In *International Conference on Pattern Recognition*, pages 23–26, 2010.

[14] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, July 2012.

[15] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, May 2003.

[16] N. Levy and L. Wolf. Minimal correlation classification. In *Computer Vision – ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 29–42. Springer Berlin Heidelberg, 2012.

[17] E. Maggio and A. Cavallaro. *Video Tracking: Theory and Practice*. Wiley, 2011.

[18] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):448–461, Mar. 2010.

[19] R. Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21 –45, third quarter 2006.

[20] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. pages 1393–1400, 2009.

[21] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 723–730. IEEE, June 2010.

[22] E. Tang, P. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, Oct. 2006.

[23] K. Venkataramani and B. V. K. V. Kumar. Role of statistical dependence between classifier scores in determining the best decision fusion rule for improved biometric verification. In *2006 international conference on Multimedia Content Representation, Classification and Security*, MRCS'06, pages 489–496, 2006.

[24] Q. Yu, T. B. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *European Conference on Computer Vision*, volume 5303 of *Lecture Notes in Computer Science*, pages 678–691, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.