Push and Pull: Iterative grouping of media

Andrew Gilbert
www.andrewjohngilbert.co.uk
Richard Bowden
r.bowden@surrey.ac.uk

CVSSP
University of Surrey
Guildford, UK

Abstract

We present an approach to iteratively cluster images and video in an efficient and intuitive manor. While many techniques use the traditional approach of time consuming groundtruthing large amounts of data [10, 16, 20, 23], this is increasingly infeasible as dataset size and complexity increase. Furthermore it is not applicable to the home user, who wants to intuitively group his/her own media without labelling the content. Instead we propose a solution that allows the user to select media that semantically belongs to the same class and use machine learning to "pull" this and other related content together. We introduce an "image signature" descriptor and use min-Hash and greedy clustering to efficiently present the user with clusters of the dataset using multi-dimensional scaling. The image signatures of the dataset are then adjusted by APriori data mining identifying the common elements between a small subset of image signatures. This is able to both pull together true positive clusters and push apart false positive examples. The approach is tested on real videos harvested from the web using the state of the art YouTube dataset [18]. The accuracy of correct group label increases from 60.4% to 81.7% using 15 iterations of pulling and pushing the media around. While the process takes only 1 minute to compute the pair wise similarities of the image signatures and visualise the youtube whole dataset.

1 Introduction

In recent years there has been an large increase in the amount of personal media, video and images captured and stored by people. In addition with the advent of the *iPad* and other touch based tablets and smart phones, there has been change in the way people expect to interact with their data. Our aim is rather than training on large labelled datasets to cluster or group media, users would be presented with a basic grouping of their photos and videos and then, through the use of a touch interface, are able to pull together similar media of the same class, and conversely push apart mis-classified false positive groupings.

There are many other approaches that classify or group large sets of images and video. However these generally use a large training dataset that has been groundtruthed [8, 16, 20, 23], this becomes increasingly infeasible as the datasets increase in complexity and size. Other approaches [4] are based around clustering images of the same object, but from varying viewpoints and illumination. The idea of "single shot" learning [21, 25] where a single example is used to learn the class is appealing, but such approaches can be very sensitive to the quality of the training example. Due to the constraints of these approaches, we propose

BMVC 2011 http://dx.doi.org/10.5244/C.25.66

an approach that moves away from training a classifier with a large labelled dataset, and aim to group objects without groundtruthed datasets, instead presenting the user with an initial grouping of the media, and allowing the user to pull together and push apart the media iteratively and interactively.

We use real data harvested from the internet and propose an approach capable of incrementally clustering similar material using the manual identification of a few true positive and false positive examples. In order to provide both scalability and incremental learning, the approach needs to be efficient. We combine two popular data mining tools developed for the text analysis domain to efficiently compute distances between high dimensional representations and dynamically augment the representation with new compound visual words to form an image signature. These tools are applied to the user selected true and false positive examples of the media, to learn rules that are applied to the full corpus of material. The media is then formed into groups of same class using a greedy clustering approach.

Vision researchers have shown the effectiveness of randomized approximate similarity search algorithms. They are designed to preserve query time for even high dimensionality inputs, for various image search application including shape matching, pose inference, bag-of-words indexing and identification of distinctive areas [3, 5, 11, 14, 24]. Quack *et al* [22] applied *Association rule* data mining to supervised object recognition by mining spatially grouped SIFT descriptors. Many solutions to action recognition use a BoW style architecture with SIFT descriptors [11, 15]. While Gilbert *et al* [10] applied a neighbourhood based corner feature classifier learnt using Association Rule data mining. However all these approaches require large amounts of training data, and this work aims to remove this constraint.

With any approach that clusters or correlate images and video, the choice of the input sample representation and similarity measure is important, as they can affect both the size of the database and the query time. We propose, an *Image signature*, as an efficient representation of a feature classifier's response for an input sample. It is designed for domains with large complex datasets, which require efficient computation and good generalisation.

We describe the image signatures and min-Hash distance in section 2. The greedy clustering is presented in section 3, and the APriori association rule mining in section 4. Finally Results and a conclusion is sections 5 and 6 respectively.

2 Media Similarity

2.1 Image Signature Description

Local features and descriptors [6, 7, 19] have been proposed for the compact description of an image or video sequence fo classification. They are designed to be invariant to illumination and geometric transformations. The descriptors are often quantized, by K-Means clustering into a smaller set of visual words, otherwise known as a "bag of words" (BoW) [17, 26]. We build on these approaches with the proposal of an "image signature". An image signature is constructed for each input sample; this is similar to a bag-of-words (BoW) histogram representation, and provides a compact, discrete representation of the input sample. However, the signature differs from a traditional static BoW, as the signature is dynamic, and increases in size, to accentuate elements that are found to discriminate between classes. In addition, the architecture is generic and could use any local feature and descriptor set, however for this work we use a descriptor based on mined Harris corners [8], these are spatio-temporal 2D Harris corners temporally and spatially grouped into neighbourhoods.

2.2 Image Signature Similarity

In order to form the similarity between the image signatures, a data mining tool called min-Hash is used and adapted. Originally designed for text analysis [2], it was more recently adapted for the near duplicate detection of images [5]. It is a randomised hashing approach, where the computation is proportional only to the number of input samples rather than the dimension of the vocabulary. This makes it ideal for large image signatures which by definition increase in size and could contain 1000's of elements especially when describing video sequences.

A min-Hash is a function that assigns a number to each image signature, the function has a property that the probability of two sets having the same value of the min-Hash function is equal to their set overlap, i.e. the ratio of the intersection and union of their set representations.

$$sim(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{1}$$

To estimate the overlap of two image signatures, multiple independent min-Hash functions are used. The fraction of the min-Hash functions that assign an identical value to the two sets gives an estimate of the similarity of the two image signatures. To efficiently retrieve images with high similarity, the values of min-Hash functions are formed into short sets called sketches. Similar signatures will have many values of the min-Hash function in common and hence have high probability of having the same sketches. A pair of signatures are a potential match when at least m identical sketch hits are found.

2.3 Weighted min-Hash

The set overlap similarity measure that min-Hash is based on, assumes that all elements of the set, are equally important. However, an image signature is a frequency based histogram therefore, to represent it; a new vocabulary is constructed to allow the min-Hash to represent the weighted histogram [5]. Given a visual vocabulary containing $|X|$ visual words or features, for example $X = \{A, B, C\}$, and where t_i is a vector of the frequency response of the visual words for the input, for example with two input signatures, $t_1 = \{3, 0, 2\}$ $t_2 = \{2, 1, 0\}$. In order to convert the frequency based image signature into a min-Hash based set of uniform symbols, the frequency w of each visual word in t_i^w is used to form the same number of new visual words as the value of w. Therefore, in the example above, the min-Hash feature vocabulary for the two input signatures becomes, $t_1 = \{A1, A2, A3, C1, C2\}$ $t_2 = \{A1, A2, B1\}$. From this representation the min-Hash algorithm in the section above can then be applied directly to the new set representation.

The resulting min-Hash representation of each input sample will then become the signature of the original sample. This compact and descriptive description can then be used to compute the image signature set overlap using standard min-Hash.

3 Greedy Clustering

The min-hash will return pairwise similarities between image signatures, therefore to efficiently cluster the classes, a greedy clustering approach is proposed. The aim is to form

groups of image signatures based on the consistency of the min-Hash result. For each image signature, its set overlap to the rest of the database is ranked according the min-Hash results. Then, instead of only using the highest ranked similarity for a pairwise correlation, the top k results are saved for each signature. Therefore given a set of n signatures $\Phi = \{S_1, ...S_n\}$, where $\forall p \in \Phi$, the k nearest neighbours, $N_k(p)$ where $N_k(p)$ are computed $\subseteq \Phi$ and $|N_k(p)| = k$, this means that

$$\forall o \in N_k(p), \forall \hat{o} \in \Phi : \hat{o} \notin N_k(p) \Rightarrow sim(o,p) \leq sim(\hat{o},p) \tag{2}$$

In a bottom up agglomerative sense the two sets are combined if they have a set overlap greater than $2/3$. where

$$\forall p,q \; N(pq) = N_k(p) \cup N_k(q) \; if \; |N_K(p) \cap N_k(q)| > 2/3k \tag{3}$$

This process then iterates until no further grouping is possible.

3.1 Visualisation

As databases increase in size, it becomes increasingly difficult for an end user to effectively visualise the groupings provided by min-Hash and greedy clustering. In addition it becomes increasingly important to allow iterative selection that allows media incorrectly grouped to be pushed apart and pulled together. Therefore Multidimensional scaling (MDS) is used to visualise the database and the relationships within it. MDS is a data analysis technique that displays the structure of distance-like data as a geometric picture. It has its origins in psychometrics by Torgerson [27], designed to help understand people's judgements of the similarity of members of a set of objects.

It works by visualising the structure of the set of image signatures from the confusion matrix distances formed by the min-Hash. Each image signature is represented by a point in a multidimensional space, and the points are arranged in this space so that the distances between pairs of points have the strongest possible relationship to the similarities among all the pairs of objects. That is, to project each point $x_i \in \Re^2$ in our visualisation space we find the set of vectors $X = \{x_i\}_{i=1}^n$ which minimise the stress function

$$stress = \sum_{i=0}^{n} \sum_{j=i}^{n} \|x_i - xj\| - sim(s_i - s_j) \tag{4}$$

A solution is then be found by a simple numerical optimization technique. Effectively the visualisation means that two similar objects are represented by two points that are close together, and two dissimilar objects are represented by two points that are far apart. It allows us to project the pairwise metric space of min-Hash into a 2D euclidean space for visualisation.

4 Pushing and Pulling Image signatures

With increasingly large datasets, we propose to move away from using large training sets. This will mean that initially the visualisation of the min-Hash and greedy clustering will place false positive results within groups, and also there will be false negatives that will not be grouped. Therefore we propose to "push" false positive classifications apart and to "pull" false negatives closer together. To achieve this, APriori association rule mining is used. An

association rule of the form $A \Rightarrow B$ is evaluated by looking at the relative frequency of its antecedent and consequent parts i.e. the set elements A and B. The support of the rule $A \Rightarrow B$ is

$$sup(A \Rightarrow B) = \frac{|\{T \mid T \in D, (A \cup B) \subseteq T\}|}{|D|} \tag{5}$$

and measures the statistical significance of the rule. The confidence of a rule is then calculated as

$$\begin{aligned} conf(A \Rightarrow B) &= \frac{sup(A \cup B)}{sup(A)} \\ &= \frac{|\{T \mid T \in D, (A \cup B) \subseteq T\}|}{|\{T \mid T \in D, A \subseteq T\}|} \end{aligned} \tag{6}$$

In summary, support for the rule is the probability of the joint occurrence of A and B i.e. $P(A,B)$ while confidence is the conditional probability $P(B|A)$ for greater details see [9].

In addition to the set elements being frequent, they must also be discriminative with respect to the negative set. To achieve this, the algorithm is run on image signatures from both the positive and negative sets. The image signatures of all examples are appended with a label, α, that identifies if the set is a positive or negative example. The results of data mining then include rules of the form $(A,B) \Rightarrow \alpha$ and an estimate of $P(\alpha|A,B)$ is given by the confidence of the rule. As the Transaction database contains both positive and negative training examples $P(\alpha|A,B)$ will be large only if (A,B) occurs frequently in the positive examples but infrequently in the negative examples. If (A,B) occurs frequently in both positive and negative examples, then $P(\alpha|A,B)$ will remain small. There is a more detailed explanation of association rule mining in [10].

The confidence and support thresholds are adjusted depending on whether the desired action is to pull the image signatures closer, or to push them apart. When pulling signatures closer, the confidence is set at 100% to ensure an association rule is only found if the elements are within all the positive sets and none of the negative sets. These elements are then accentuated to pull the positive signatures closer together in the min-Hash space.

When pushing the signatures apart, the support is set at 100%, to ensure that it identifies elements of both the positive and negative images signatures that are common to all signatures - these elements are then removed to decrease the similarity between the positive and negative image signatures in the min-Hash space.

4.1 Applying Mining Rules

After the APriori association rule mining has been performed on the positive and negative image signatures, the resultant rules are applied to all the signatures. Each image signature is taken in turn. In a pull operation, for each rule returned from the mining, if the image signature contains the elements of the rule, an additional min-Hash element is added. This accentuates the elements common within the positive image signatures to pull them spatially closer.

For example, using the image signatures from section 2.3, $t_1 = \{A1, A2, A3, C1, C2\}$ $t_2 = \{A1, A2, B1\}$. If the association rule returned from the mining was a subset of t_i, e.g. Ax where x is any number, the element $(A4)$ would be added to set t_1 and the element $(A3)$ would be added to set t_2. This process would be repeated over all the input sets, however,

if the set does not contain the subset (Ax), it would not be incremented. This increased weighting on the subset (A) would "pull" together sets that contain subset (A) together over time improving accuracy. In addition, the mining can return association rules that contain multiple subsets that together are descriptive and distinctive. Using the same example, if the mining returns the rule $(A2, B1)$, it would not be appended to t_1 as the set does not contain any (B) elements. However, it would be appended to t_2, making $t_2 = \{A1, A2, B1, AB1\}$. This means that for the min-Hash permutations to match, both sets would have to contain the symbol $AB1$, not just a subset. This has the ability to reduce the confusion between classes further.

In contrast, if a push operation is performed, for each rule returned from the mining, if the image signature contains the elements of the rule, the min-Hash element would be *removed*. This would remove similarity between the positive and negative image signatures and in the MDS visualisation be spatially pushed apart. Using the example from section 2.3. If the association rule returned from the mining was a subset of t_i, e.g. Ax where x is any number, the element $(A3)$ would be *removed* from set t_1 and the element $(A1)$ would be removed from set t_2, and further repeated with the other image signatures and returned rules. This would reduce the set overlap between the positive and negative image signatures to ungroup them in the MDS visualisation.

The min-hash and greedy clustering process can then be repeated and the MDS visualisation redrawn to illustrate the improved grouping of the media

5 Results

To illustrate the process, and evaluate the quality of the clustering and categorisation approach the approach is demonstrated on the *YouTube* dataset [18]. This dataset is formed of user generated videos from the internet and consists of eleven categories: *basketball shooting, cycling, diving, golf, horse riding, juggling, play swings, tennis swinging, trampolining, volleyball, and dog walking*. There are 1168 videos and they exhibit large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions. Some examples of the dataset are shown in Figure 1.

(a) Cycling (b) Juggling (c) Basketball

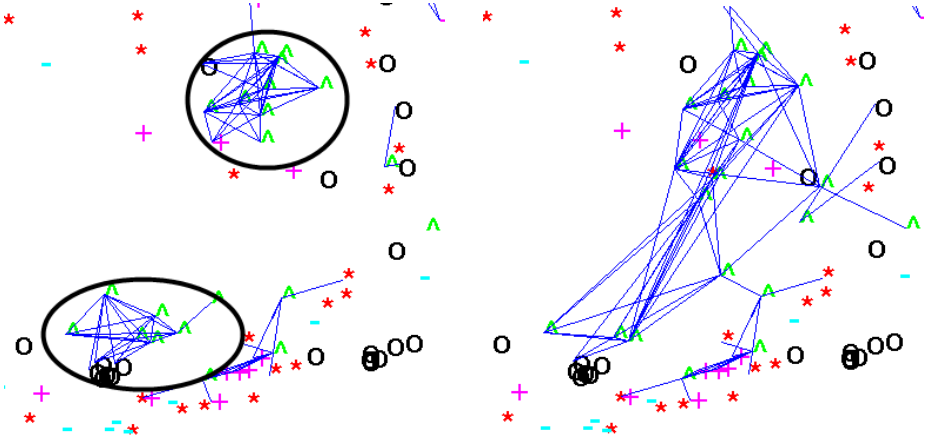
Figure 1: *YouTube* dataset examples

5.0.1 Feature Representation

Most feature representations of the image and video content are based on the popular bag-of-words representation [5, 15], however in this framework the feature representation used by Gilbert *et al* [8] is employed. These are compound corner classifiers, that are based

on a set of spatio-temporal Harris [12] corner interest points. The classifiers were trained on a different dataset, the 6 class action recognition dataset, *KTH* [23]. This means that it is unlikely the the feature classifiers are optimal for the *YouTube* dataset. However, the classifiers were trained to discriminate between different types of motion, and the 6 *KTH* classifiers are concatenated into a single classifier to provide an effective description on the motion in another temporal dataset, the *YouTube* dataset.

Each image signature for each video contains around 2000 elements, and the overall initial vocabulary of elements is 4500. Figure 2(a) shows a subset of the initial groupings for the class *Diving* from the *YouTube* dataset, each symbol represents a different class. It



(a) Initial greedy clustering result of the class Diving

(b) Grouping of diving class after pulling together two separate groups

Figure 2: The lines indicate the grouping of the class diving from the *YouTube* dataset before and after pull groups together

can be seen that there are a number of groups of true positive examples but also many false positives. Overall for the *YouTube* dataset, there are initially 60.4% true positive groupings and 21.4% false positive groupings. Figure 3 shows examples of the true and false positive classification of videos within the two circles of Figure 2(a). The false positive examples

(a) (b) (c)

Figure 3: Positive examples of image signatures from the *Diving*

generally contain the same vertical motion of diving as is the case of the golf swing 4(a), or the ball bouncing in Figure 4(b).

(a) (b)

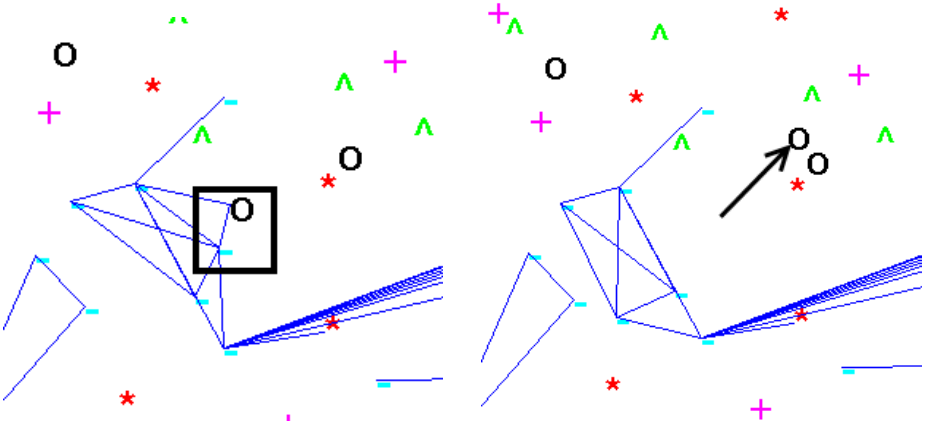
Figure 4: False positive examples of image signatures from the *Diving*

5.1 Pulling the groups together

To improve this result the user can iteratively pull together groups of positive classifications. The aim is to pull together the two areas highlight by circles in Figure 2(a), performing the mining to identify common elements of the true positive image signatures and accentuating these to pull the true image signatures closer. Figure 2(b) shows the groupings after selecting all the videos within the two marked circled groups. In Figure 2(b), the two groups are more interlinked, this reflected by the increased accuracy of correctly grouping *diving* examples by 10%. In addition, a number of the false positive links were removed as the true positive links have increased in strength.

5.2 Pushing apart Groups

To remove the false positive assignments from the greedy clustering, the incorrect image signatures can also be pushed apart. The process is illustrated by taking the image signatures of the sets that are to push apart; these are then mined to identify the common elements. These elements can then be removed from all image signatures with the aim to push part the signatures. This process can illustrated by using the *jumping* class of the *YouTube* dataset, Figure 5(a) shows a subset of the initial grouping of the class *jumping*. Then in Figure 5(b) is the



(a) Initial greedy clustering result of the class Jumping

(b) Grouping of jumping class after pulling together two separate groups

Figure 5: The lines indicate the grouping of the class; Jumping from the *YouTube* dataset before and after pull groups together

result of pushing part the two image signatures within the black square in Figure 5(a). In this example it reduces the false positive rate for the jumping class by 5%, reducing cross class confusion. Repeating this process of pushing apart and pulling together image signatures can result in the overall accuracy of the correctly grouped media from the *YouTube* dataset increasing from 60.4% to 81.7%, in only 15 iterations. Each iteration takes around 1 minutes to complete on a standard desktop computer. This includes the complete re-computation of the pair wise similarities of the image signatures and visualisation of the dataset. Obviously which examples are chosen by the user effects the accuracy at each iteration, but this example gives an indicative results

5.3 Comparison to other approaches

For a comparison with more standard published approaches, we adopt the commonly used Leave-One-Out Cross-Validation. More specifically, for the YouTube dataset, adopting the settings given in [18], the dataset was divided into 25 subsets, out of which 24 subsets were used for training and the remaining subset was used for testing. Where for each unseen test subset, the other 24 subsets were used to adjust the signatures using the pulling together of positive image signatures . This process was repeated four times/iterations, and each time, five true positive image signatures and a single false positive signature were selected to correct misclassification. However as only 4 iterations were used, only 24 videos needed to be groundtruthed unlike the 1121 videos used by other approaches. The semantic clusters were iteratively built on the 24 training subsets, and then classified by performing a nearest neighbour assignment to the closest class. Table 1 shows the results for our signature min-Hash approach compared to other recently published results on the same dataset.

Table 1: Accuracy of *YouTube* dataset

Approach	Accuracy
Cinbis [13]	75.2%
Liu [18]	71.2%
Bregonzio [1]	63.1%
Baseline min-Hash	56.4%
Sig min-Hash	79.7%

It is important to note in this experiment, while the grouping requires minimal data, the features themselves have been learnt over the entire *KTH* training data. This serves to highlight that features that are learnt for classification, may not be the best features for grouping or clustering using naive distance metrics, but through the use of signatures, these features can be reweighed appropriately to increase performance.

6 Conclusion

This paper has presented a novel approach to the problem of classifying and clustering video sequences. Rather than using a large dataset, the user selects small subsets of data to learn common rules. The efficient nature of the the two data mining tools, min-Hash and APriori association rule mining ensure the whole process is fast and responsive to the user. This user

led training allows for high accuracy with reduced computation time compared to traditional train / test approaches, Furthermore the iterative approaches effectively means the user needs only supervise a small subset of the data. In the future, we aim to build on this success and incorporate additional features into the generic image signature to further improve accuracy.

References

[1] M. Bregonzio, J. Li, S. Gong, and T. Xiang. Discriminative Topics Modelling for Action Feature Selection and Recognition. *In Proc of British Machine Vision Conference (BMVC'10)*, 2010.

[2] A. Brooder. On the Resemblance and Containment of Documents. *In Proc SEQS: Seqeunces'91*, 1998.

[3] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'10)*, 2010.

[4] O. Chum, J. Philbin, J. Sivic, M. Isard, and A Zisserman. "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval ". *Proc. IEEE International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.

[5] O. Chum, J. Philbin, and A. Zisserman. Near Duplicate Image Detection: min-Hash and tf-idf Weighting. *In Proc. of BMVA British Machine Vision Conference (BMVC'08)*, 2008.

[6] N. Dalah and B. Triggs. Histograms of Oriented Gradient for Human Detection. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. "Behavior Recognition via Sparse Spatio-temporal Features". *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 65–72, 2005.

[8] Andrew Gilbert, John Illingworth, and Richard Bowden. "Scale Invariant Action Recognition Using Compound Features Mined from Dense Spatio-Temporal Corners". *In Proc. of International Conference on Computer Vision (ICCV'09)*, I:222–233, 2009.

[9] Andrew Gilbert, John Illingworth, and Richard Bowden. iGroup: Weakly supervised image and video grouping. *In Proc of IEEE International Conference on Computer Vision (ICCV1)*, 2011.

[10] Andrew Gilbert, John Illingworth, and Richard Bowden. Action recognition using-mined hierarchical compound features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 883 – 897, 2011.

[11] K. Grauman and T. Darrell. Pyramid match hashing: Sub-linear time indexing over partial correspondences. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007.

[12] C. Harris and M. Stphens. "A Combined Corner and Edge Detector". *In Proc. of Alvey Vision Conference*, pages 189–192, 1988.

[13] N. Ikizler-Cinbis and S. Sclaroff. Object, Scene and Actions: Combining Multiple Features for Human Action Recognition. *In Proc. of European Conference on Computer Vision (ECCV'10)*, 2010.

[14] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2008.

[15] I. Laptev and T. Lindeberg. "Space-time Interest Points". *In Proc. of IEEE International Conference on Computer Vision (ICCV'03)*, pages 432–439, 2003.

[16] I. Laptev and Pérez. "Retrieving Actions in Movies". *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, 2007.

[17] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.16. URL http://dx.doi.org/10.1109/CVPR.2005.16.

[18] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos "in the Wild". *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[19] D. G. Lowe. "Object Recogntion from Local Scale-Invaraint Features". *In Proc. of IEEE International Conference on Computer Vision (ICCV'98)*, pages 1150–1157, 1998.

[20] M. Marszalek, I. Laptev, and C. Schmid. Actions in Context. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[21] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang. Hierarchical Space-time Models, Enabling Efficient Search for Human Actions. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 2568 – 2574, 2009.

[22] T. Quack, V. Ferrari, B. Leibe, and L.C. Gool. "Efficient Mining of Frequent and Distinctive Feature Configurations". *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, 2007.

[23] C. Schuldt, I. Laptev, and B. Caputo. Recognizing Human Actions: a Local SVM Approach. *In Proc. of International Conference on Pattern Recognition (ICPR'04)*, III:32–36, 2004.

[24] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. *In Proc of IEEE International Conference on Computer Vision (ICCV03)*, 2003.

[25] E. Shechtman and M. Irani. Space-time Behavior-based Correlation -or- How to Tell if two Underlying Motion Fields are Similar without Computing them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2045 – 2056, 2007.

[26] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. *In Proc of IEEE International Conference on Computer Vision (ICCV05)*, 2005.

[27] W. S. Torgerson. *In Psychometrika*, 17:401 – 4192574, 1952.