Oxford-IIIT TRECVID 2009 – Notebook Paper

Sreekanth Vempati, Mihir Jain, Omkar M. Parkhi, C. V. Jawahar
Center for Visual Information Technology,
International Institute of Information Technology, Gachibowli, Hyderabad, India
Marcin Marszalek, Andrea Vedaldi, Andrew Zisserman
Visual Geometry Group,
Department of Engineering Science, University of Oxford, United Kingdom

Abstract

1. Oxford-IIIT combined: a *spatial pyramid intersection kernel SVM* image classifier, a *sliding-window random-forest* object detector, a *sliding-window intersection kernel SVM* object detector, and a *discriminative constellation model* facial feature extractor. For each of the twenty features, methods were ranked based on their performance on a validation set and associated to successive runs by decreasing performance. For training, TRECVID annotations were manually corrected and augmented with object bounding boxes, and additional training data was used for under-represented features such as *Airplane flying*.

2. The different methods yielded a significantly different performance depending on the feature, as expected by their design.

3. The image classifier worked better for scene-level features such as *Cityscape*, *Classroom*, *Doorway*, while the object detectors worked better for *Boat or ship*, *Bus*, *Person riding a bicycle*, and the face feature extractor worked well for *Female face closeup*.

4. Three conclusions can be drawn: (i) different features are addressed better by specialised methods, (ii) removal of noise from TRECVID annotations (iii) additional data for under-represented features significantly improve performance.

1 Introduction

Our team participated in all 20 categories of high level feature extraction task. We extracted our own keyframes for every shot of both the TRECVID 2009 DEVEL and TEST data sets. The DEVEL set was subdivided in two halves denoted TRAIN and VAL and used for training and validation, respectively. Ground truth labels for the DEVEL keyframes were obtained by transferring TRECVID collaborative annotations. Additional region-of-interest annotations and images were used for training some of the object detectors.

New developments this year include: removing the noise in the annotation data (section 2); using a fast intersection kernel SVM combining two features (section 4.1); extending the region of interest (ROI) in training for object detectors in order to include more context (section 4.2.1); and classifying face tracks into male/female (section 4.4).

2 Cleaning the annotations

We found the collaborative annotations for the TRECVID high level features to be quite noisy: some shots are wrongly annotated, and others are labelled as 'skip' when they are, in fact, unambiguously positive or negative for the feature. To remove this noise in the annotation, we used a weak classifier trained on the noisy data for each high level feature as follows

1. Train a classifier using all the +ves and a subset of -ves in TRAIN and VAL sets according to the Collaborative Annotation.

2. Rerank all the images in the TRAIN+VAL set based on the classifier output.

3. Refine the annotations of the top 5000 ranked images.

In this manner, we could find many of the wrong annotations with minimal manual effort. This refinement was found to be very effective. For example, for the Doorway category the AP performance increased from 0.16 to 0.41 due to annotation cleaning (training on TRAIN and testing on VAL, both of which were corrected).

3 Visual representation

For the key frame classification we used Pyramid Histogram of Visual Words (PHOW) [3]. It is described below. Special features are used for the Female-human-face-closeup category as described in section 4.4.

3.1 Pyramid Histogram of Visual Words

These descriptors consist of visual words which are computed on a dense grid. Here visual words are vector quantized SIFT descriptors [11] which capture the local spatial distribution of gradients.

Local appearance is captured by the visual words distribution. SIFT descriptors are computed at points on a regular grid with spacing M pixels. We have used gray level representations for each image. At each grid point, the descriptors are computed over circular support patches with radii r. Thus, each point is represented by four SIFT descriptors. These dense features are vector quantized into *visual words* using K-means clustering. Here, we have used a vocabulary of 300 words. Each images is now represented by a histogram of these visual word occurrences.

We have used $M = 5$, $K = 300$ and radii $r = 10, 15, 20, 25$. To deal with with empty patches, we zero all SIFT descriptors with L2 norm below a threshold (200).

In order to capture the spatial layout representation, which is inspired by the pyramid representation of Lazebnik $et.al.$ [9] , an image is tiled into regions at multiple resolutions. A histogram of visual words is then computed for each image sub-region at each resolution level.

To summarize, the representation of an appearance descriptor is a concatenation of the histograms of different levels into a single vector which are referred to as Pyramid Histogram of Visual Words (PHOW). Here, we have used three levels for the pyramid representation. The distance between the two PHOW descriptors reflects the extent to which the images contain similar appearance and the extent to which the appearances correspond in their spatial layout.

4 Classification schemes

We have three different methods: (i) for "scene" like categories where we classify the entire image and its spatial layout we use an intersection kernel SVM; (ii) for object like categories, where we search for the localization (position, scale) of the object within the image, we use a sliding window Random Forest classifier or a sliding window intersection kernel SVM classifier; and (iii) for faces we use a Viola-Jones face detector and track the detections throughout the video to associate detections of the same person. Each of these methods is described in the following subsections.

4.1 SVM with Fast Intersection Kernel

An intersection kernel SVM is more powerful than a linear kernel SVM, and using the method of [12] its testing complexity can be reduced from $O(dmn)$ to $O(dn \log m)$, where d is the feature dimensionality, n is the number of keyframes to be classified, and m the number of support vectors. For example, to classify 200K frames requires only 5 minutes (including the time taken for loading the features into memory) on a Intel Xeon CPU @2.00GHz. Both the PHOW and color features are used. Also, when trained on TRAIN set and tested on VAL set, it is found that nearly 20% of the total negative samples available are enough to get an AP which is nearly equal to AP obtained by training all the negative samples.

Category	**AP**	**infAP**
Training Set	**TRAIN**	**TRAIN+VAL**
Testing Set	**VAL**	**TEST**
Cityscape	0.54	0.28
Demonstration or protest	0.45	0.03
Doorway	0.41	0.21
Nighttime	0.40	0.24
Hand	0.39	0.20
Boat or ship	0.23	0.17
Female face closeup	0.20	0.19
Traffic intersection	0.17	0.16
Person playing soccer	0.11	0.31

Figure 1: **Performance of the SVM image classifier.** The figure reports the average precision of the method of Sect. 4.1 when trained on TRAIN and evaluated on VAL and TRECVID inferred AP when trained on TRAIN+VAL. To compute average precision on TRAIN+VAL the complete and cleaned annotations where used. In several cases the difference in AP and infAP is remarkable.

This method worked well for most of the scene like categories, (e.g. *Cityscape*, *Demonstration or protest*, *Hand*). Results obtained on the TEST set for different categories are given in Fig. 3 for the Hand category, and Fig. 4 for the Cityscape category, and quantitative results are reported in Fig. 1.

4.2 Classification by Detection using Random Forests

A random forest multi-way classifier [1, 4, 10, 3, 5, 6, 13, 14, 17, 21, 22, 16] consists of a number of trees, with each tree grown using some form of randomization. The leaf nodes of each tree are labeled by estimates of the posterior distribution over the image classes. Each internal node contains a test that splits the space of data to be classified. An image is classified by sending it down every tree and aggregating the reached leaf distributions.

Here we use Random forest as an object detector, and take the approach of classification by detection for object categories in high-level feature extraction task.

Learning the Object Classifier: Each Random Forest classifier is trained to discriminate between candidate regions that do and do not contain an instance of the object of interest. A one-versus-rest classifier is trained for 4 object categories namely Boat_Ship, Person-riding-a-bicycle, Bus and Hand.

Training Random Forests: The trees we train here are binary and are constructed in a top-down manner. Pyramid Histogram of Visual Words (PHOW) is used with 3 levels for the pyramid representation to represent the appearance. For node test we have a node function (difference of two components of the feature vector) and a threshold. During training of the tree each node has available only a randomly chosen subset of the entire pool of possible node functions (100 such functions were chosen). Training is achieved by finding for each non-terminal node a node function and a threshold which yields maximum information gain within such restricted, randomized search space [3, 21]. We train 100 trees with maximum allowed depth 10 and choose an optimal threshold from 10 randomly selected thresholds for each chosen node function. To further introduce randomness each tree is trained using 67% of samples from training data selected at random. Empirical posterior distribution for the class is stored in the leaf nodes as done in [3].

Classification: The test image is passed down each random tree until it reaches a leaf node. All the posterior probabilities are then averaged and the arg max is taken as the classification of the input image.

Testing and Retraining: For training number of positive and negative data samples are required. The ground truth object instances (Region of Interest or ROIs) for a class, plus a number of jittered instances (both from original and flipped training images), are used as positive samples. Regions that do not overlap the target object instances by more than 20% are used as negative samples. The aspect ratio of the the detector window is found from the aspect ratios of the ROIs in the training set. While testing we use a sliding window approach, where a detector window is applied at all positions and scales of an image. The aspect ratio of the the detector window is found from the aspect ratios of the ROIs in the training set. Because of this the number of possible negative samples is exorbitantly large and it is important to find a proper representative sub-set. This is done by *bootstrapping* or *retraining* each classifier as follows:

- Train a classifier using positive and negative samples from DEVEL data.
- Run the classifier over the training images.
- Compare the detections with the ground truth ROIs, and label them as false positives if the overlap is less than 20%.
- The top 30K to 35K false positives are used as hard negative samples and are added to the negative set for retraining.

The retrained classifier is then run on the TEST data. The maximum detection score in a frame is taken as the confidence of that frame.

4.2.1 Extended ROIs

There are a lot of variations in shape and appearance of objects, such as caused by extreme viewpoint changes, that are not well captured by a single template (or aspect ratio). It is common to use multiple templates to encode view or pose variations, for example separate templates for frontal and side views of faces and cars [15]. To interpret the variations in TRECVID data it would require many templates. Applying them over such a large dataset for testing/bootstrapping is computationally very expensive. To deal with this we use what we call as *Extended ROIs*.

Extended ROIs are obtained by extending the original ROIs such that its aspect ratio becomes the selected one and it lies at the center of the extended ROI. This allows us to use only one aspect ratio and still cover for large range of aspect ratios while testing. Fig. 2 shows the original ROIs from DEVEL set and their Extended versions. While training Extended ROIs were used as the positive samples. Then all the training ROIs as well as the detector window (while testing) had the same aspect ratio. The detected windows were also extended ones with the object at its center.

The Extended ROI can give a significant performance boost. For example, for Boat_Ship the AP increased from 0.198 to 0.411 when Extended ROIs were used compared to not using them (training on the TRAIN set, testing on the VAL set).

Fig. 5 shows ranked shots for the Boat-Ship category using the Random Forest classifier.

4.3 Sliding-window intersection kernel SVM

In addition to the random forest object detector, we tested a classifier based on an intersection kernel sliding-window SVM. The method is a simplified version of [18], using only one feature (PHOW), and one stage of the cascade (additive kernel) on top of coarsely sampled region-of-interest. This is sufficient as no accurate localisation is required. The detector is used to reject false positives retrieved by a whole-image classifier, using the same features. It performed better than the random-forest detector for the *Bus* and *Person riding a bicycle* classes.

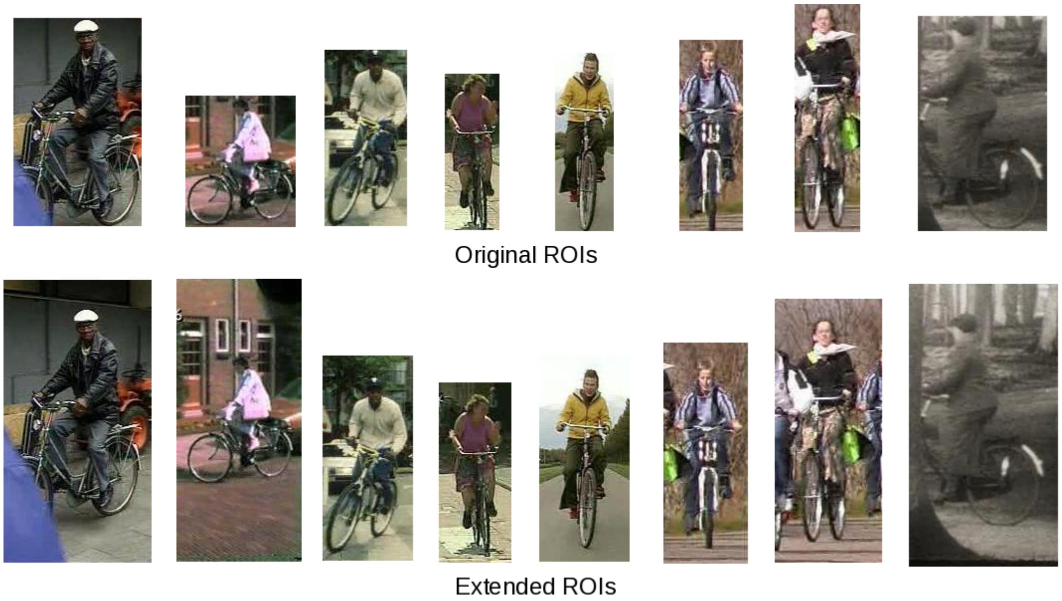

Figure 2: Top row shows the examples of original ROIs for class Person-riding-a-bicycle, and their extended ROIs are shown in the bottom row. Note that all the extended ROIs have same aspect ratio.

4.4 Face Detection and Classification

This section describes our face detection approach (for details see [2] which presents a real-time version of the method described below). The aim here is to find video footage of people where their face is visible with a low false positive rate. The same processing pipeline is applied to all frames of the training data and test data. In the training data, a very high precision was achieved at low recalls.

4.4.1 Face detection and tracking

The first stage of processing is frontal face detection which is done using the Viola- Jones cascaded face detector [19]. When a new individual has been detected, a kernel-based regressor is trained to track that individual such that the tracking performance is both fast and more robust to non-frontal faces in comparison to cascaded face detection [20]. Face detection is used to collect several exemplars of an individuals face which may vary in pose and expression. A training set consisting of image patches that are offset from the face center and at a slightly different scale, and the respective transformations back to the original face location and scale, are artificially generated from the face detections. This dataset is used to train a kernel-based regressor to estimate the position (x, y) and scale (w) of a face.

4.4.2 Feature localization

The output of the face tracker gives an approximate location and scale of the face, but does not provide a confidence in this measure. To achieve a low false positive rate, features at the corners of the eyes, nose and mouth are located to verify the existence of a face. Where multiple successive frames achieve a poor localization confidence, the track is terminated. To locate the features, a model combining a generative model of the feature positions with a discriminative model of the feature appearance is applied. The probability distribution over the joint position of the features is modelled using a mixture of Gaussian trees, a Gaussian mixture model in which the covariance of each component is restricted to form a tree structure with each variable dependent on a single parent variable. This model is an extension of the single tree proposed in [8], and further details can be found in [7].

4.4.3 Feature Descriptor

Output of Feature Localization is used to locate 9 facial features at the corners of the eyes, nose and mouth. Four additional features are added at the centre of both eyes, mouth and nose. The face region defined by the facial features is normalized with respect to a canonical face to reduce the effects of scale and out-of-plane rotations of the head. An affine transformation is computed between the canonical feature set and the facial features. Using the affine transformation, regions that were circular in the canonical reference frame are extracted from the corresponding elliptical regions in the tracked face. These 13 image patches are then normalized to have zero mean and unit variance, and are concatenated to form a single vector that represents the persons face.

4.4.4 Female Face Classification

For the Female-human-face-closeup category of the high level feature extraction task, tracks obtained from the tracker were first filtered to remove any having an average face size less than 50% of the frame area. Then in the DEVEL data the remaining tracks were manually labeled as male and female faces. Middle frame of each of the tracks was used for classification purpose. Using this we received classification scores at track level. These results were then transfer to shot level results using shot boundaries. During this process, In case of more than one track assigning its score to a shot, best score was retained. Finally shots were ranked based on the classification scores they received.

Acknowledgement We are grateful to the UK-India Education and Research Initiative (UKIERI) for financial support, and to ERC grant VisRec.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation, 9(7)*, 1997.

[2] N. E. Appostoloff and Andrew Zisserman. Who are you? real time person identification. In *British Machine Vision Conference*, 2007.

[3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Internation Conference on Computer Vision*, 2007.

[4] L. Breiman. Random forests. *ML Journal, 45(1)*, 2001.

[5] H. A. Chipman, E. L. George, and R. E. McCulloch. Bayesian ensemble learning. In *Neural Information Processing Systems*, 2006.

[6] T. Deselaers, A. Criminisi, J. Winn, and A. Agarwal. Incorporating on-demand stereo for real-time recognition. In *Computer Vision and Pattern Recognition*, 2007.

[7] Mark Everingham and Andrew Zisserman. Regression and classification approaches to eye localization in face images. In *International Conference on Automatic Face and Gesture Recognition*, 2006.

[8] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. In *International Journal of Computer Vision*, pages 55–79, 2005.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, June, 2006.

[10] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. In *Pattern Analysis and Machine Intelligence, 28(9)*, 2006.

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

[12] S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition*, 2008.

[13] R. Mar'ee, P. Geurts, J. Piater, and L Wehenkel. Random subwindows for robust image classification. In *Computer Vision and Pattern Recognition*, 2005.

[14] F. Moosman, B. Triggs, and F. Jurie. Fast discriminative visual codebook using randomized clustering forests. In *Neural Information Processing Systems*, 2006.

[15] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Conference on Image and Video Retrieval*, 2000.

[16] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In *British Machine Vision Conference*, 2008.

[17] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition*, 2008.

[18] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. *Proc. of the International Conference on Computer Vision*, 2009.

[19] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, 2001.

[20] Oliver M. C. Williams, Andrew Blake, and Roberto Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Internation Conference on Computer Vision*, 2003.

[21] J. Winn and A. Criminisi. Object class recognition at a glance. In *Computer Vision and Pattern Recognition*, 2006.

[22] P. Yin, A. Criminisi, J.Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *Computer Vision and Pattern Recognition*, 2007.

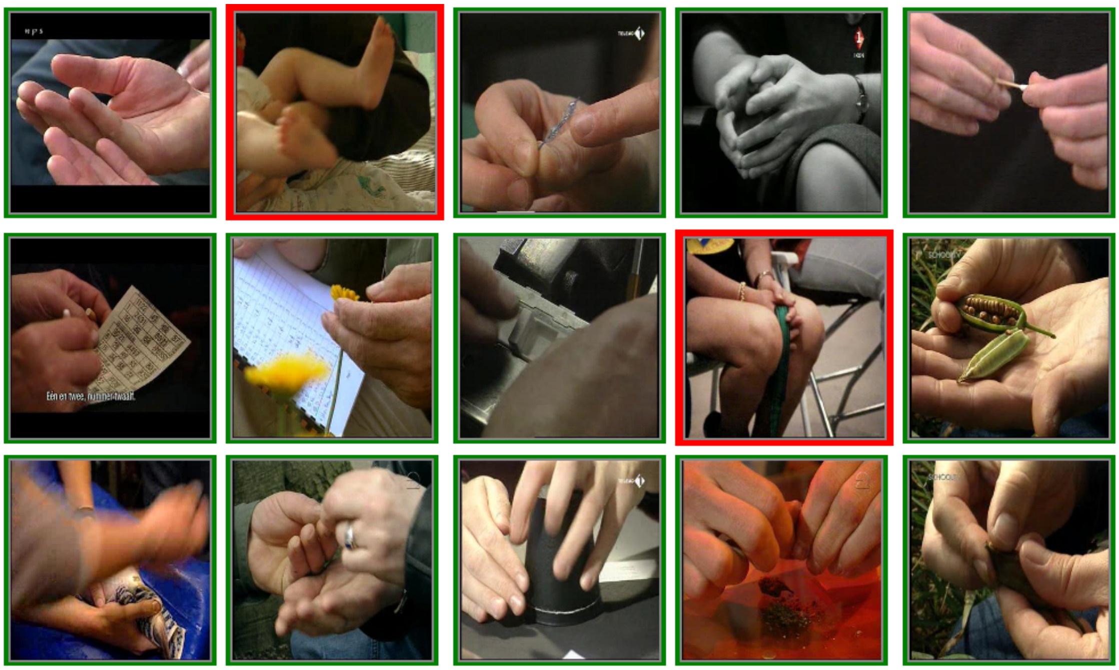

Figure 3: Top 15 shots from the TEST set (shown by keyframe) using a SVM with intersection kernel for the Hand category

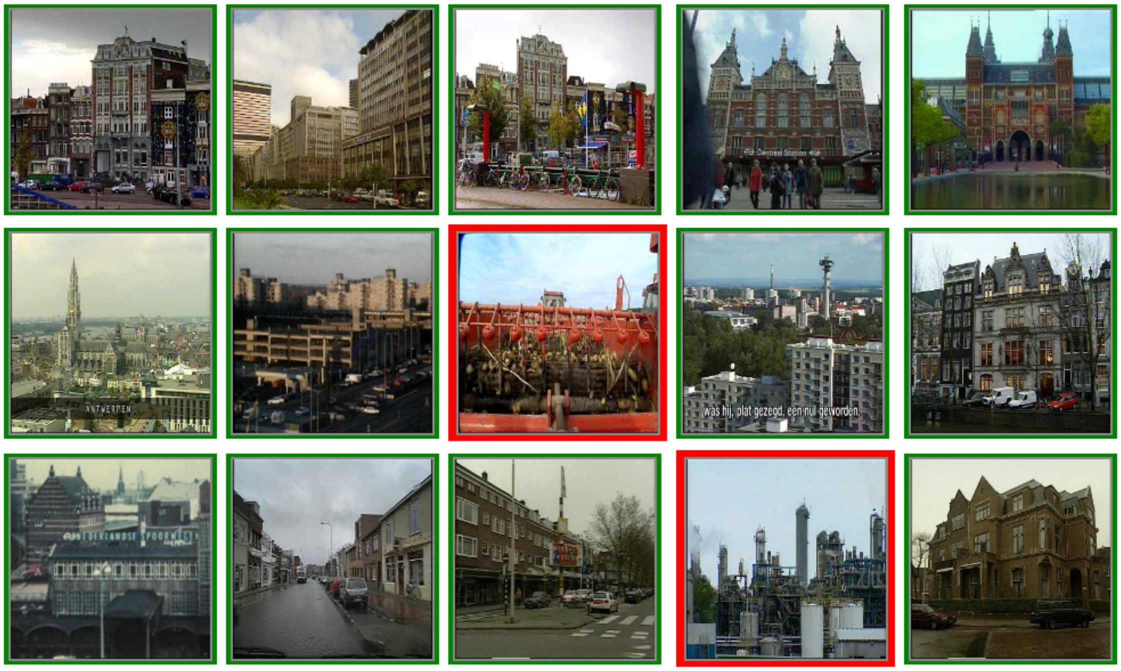

Figure 4: Top 15 shots from the TEST set (shown by keyframe) using a SVM with intersection kernel for the Cityscape category

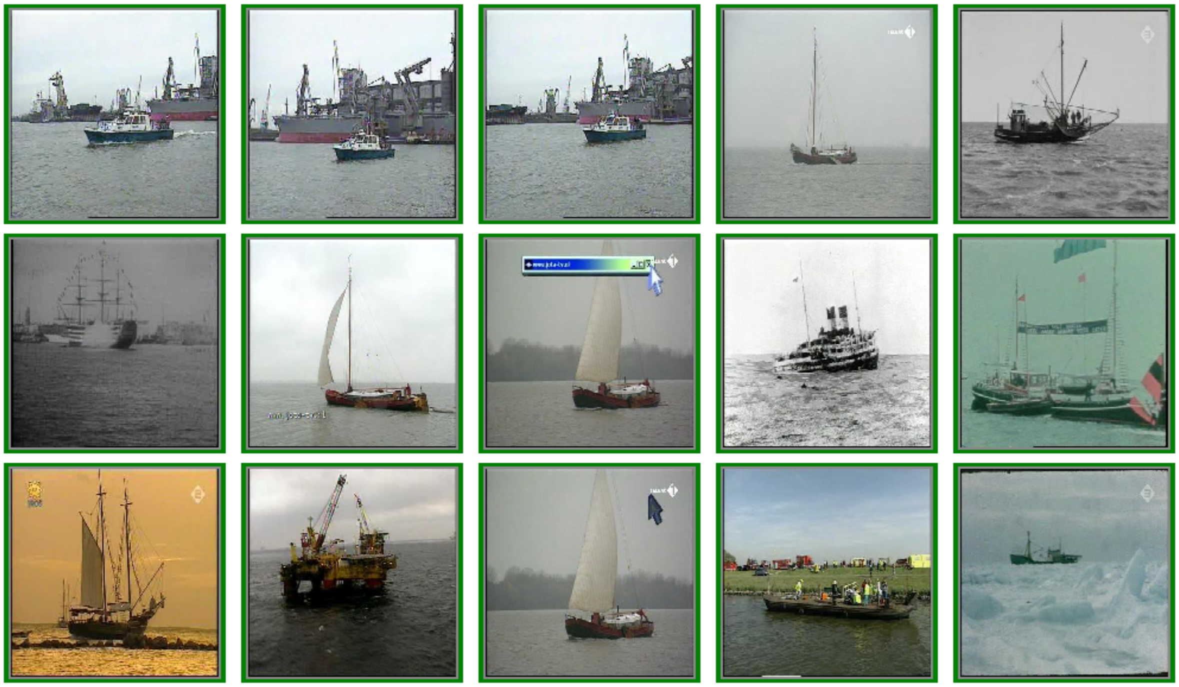

Figure 5: Top 15 shots from the TEST set (shown by keyframe) using a Random Forest Classifier for the Boat Ship category

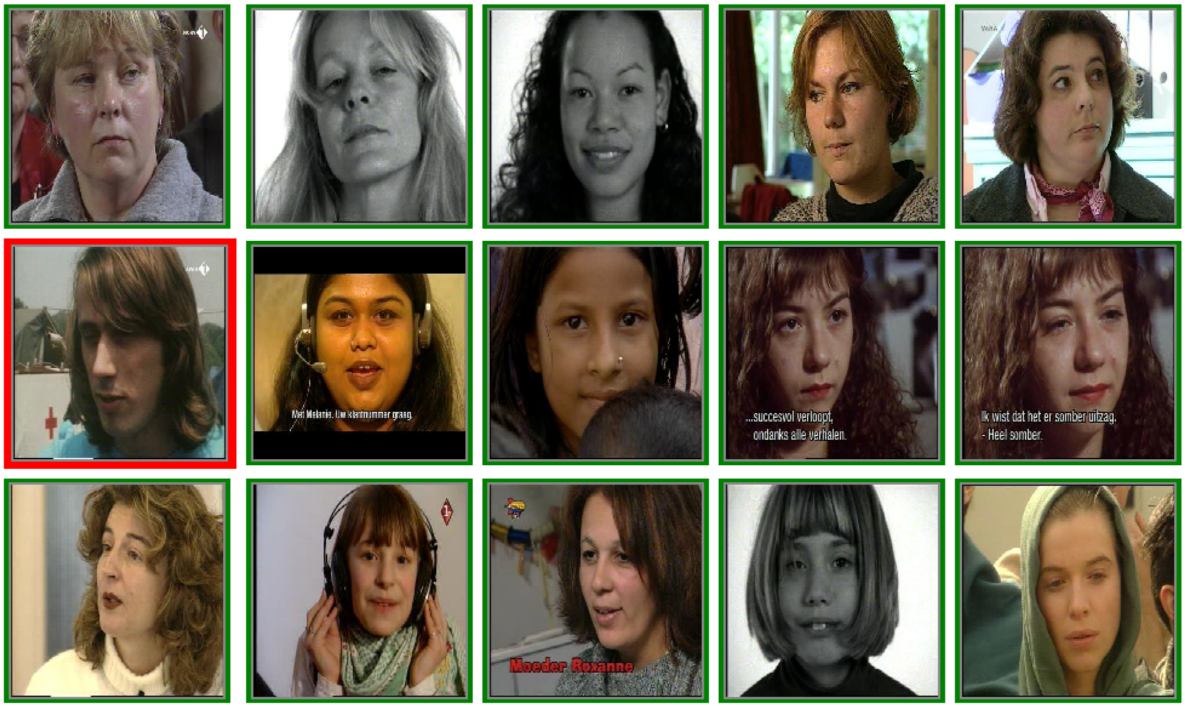

Figure 6: Top 15 shots from the TEST set (shown by keyframe) using a SVM Classifier on PHOW features for the Female-human-face-closeup category