"Who are you?" – Learning person specific classifiers from video

Josef Sivic[1], Mark Everingham[2] and Andrew Zisserman[3]

[1]INRIA, WILLOW Project, Laboratoire d'Informatique de l'Ecole Normale Superieure, Paris, France

[2]School of Computing, University of Leeds, UK

[3]Department of Engineering Science, University of Oxford, UK

Abstract

We investigate the problem of automatically labelling faces of characters in TV or movie material with their names, using only weak supervision from automatically-aligned subtitle and script text. Our previous work (Everingham et al. [8]) demonstrated promising results on the task, but the coverage of the method (proportion of video labelled) and generalization was limited by a restriction to frontal faces and nearest neighbour classification.

In this paper we build on that method, extending the coverage greatly by the detection and recognition of characters in profile views. In addition, we make the following contributions: (i) seamless tracking, integration and recognition of profile and frontal detections, and (ii) a character specific multiple kernel classifier which is able to learn the features best able to discriminate between the characters.

We report results on seven episodes of the TV series "Buffy the Vampire Slayer", demonstrating significantly increased coverage and performance with respect to previous methods on this material.

1. Introduction

We are interested here in the automatic labelling of people in TV or film material with their identity. This is a tremendously challenging problem due to the huge variation in imaged appearance of each character and the weakness and ambiguity of available annotation. The 'faces in the wild' project [4] set the scene for this problem by using weak annotation from news web pages to label the faces in images on the page. The equivalent challenge in video was taken up in Everingham *et al.* [8], where we showed that by using the weak supervisory information available from transcripts (temporally aligned with subtitles) tracked faces could be labelled. Common to both projects is that they are limited to *frontal* faces, and that the classifiers are not learnt discriminatively.

Our objectives in this paper are two-fold: (i) to improve *coverage* (the number of characters that can be identified and the number of frames over which they are tracked), and (ii) to improve *accuracy* (correctly identify the characters). To address these objectives our first step is to incorporate profile views with comparable success to frontal views (detection, tracking, facial features, speaker detection) and with seamless integration of profile and frontal tracks, *i.e*. promoting profiles to first class status.

Profiles have been detected and tracked in uncontrolled video (TV, movies) at the level of difficulty considered here [14, 16], but these papers have not dealt with the problem of *recognizing* faces in profile. In fact, moving from the restricted case of frontal detection to also include profiles in the same track potentially makes the problem much harder as combining descriptors for frontal and profile views into the same recognition framework is not straightforward. Our second step addresses this problem by defining a kernel for each descriptor, and learning a discriminative classifier using a linear combination of these kernels.

The strengths of this approach are (i) an integrated treatment of multiple detectors enables learning *across* viewpoints *e.g*. (weakly) labelled profile views contribute to learning frontal appearance via tracking. For instance, if there is supervisory information available for a profile view and this profile is connected to a frontal view (*e.g*. the character turns their face) then the supervision can be transferred to the frontal view, harvesting additional labelled faces; (ii) the multiple kernel combination enables us to seamlessly combine diverse descriptors dependent on viewpoint; and (iii) this approach is naturally extensible to other descriptors which are computed on a per-instance basis and have missing values, *e.g*. pose-specific descriptors or descriptors which are affected by occlusion.

There has been considerable work on discriminative classification of faces. In the case of images from web pages (at the level of difficulty of faces in the wild) recent work has shown the benefit of the discriminative approach both for identity [15], and in the Facetracer project [13] for other attributes such as age, ethnic origin, and gender. Indeed, others [2, 9, 12] that have improved on the classification per-

formance of our previous work [8] have used discriminative classification, though none have considered profile views. The new element we bring here is using a linear combination of kernels for the classification, and in particular using Multiple Kernel Learning (MKL) [3, 19] to determine the combination of features used. This has the advantage that an optimal combination of features is learnt – others have considered multiple features, *e.g*. [13], but only by a greedy learning algorithm. Here we discriminatively learn a *character specific* combination of kernels. This means that the features (*e.g*. eye, a spatial region of the hair, etc) that best discriminate one character from another can be learnt.

1.1. Background – overview of Everingham *et al.* [8]

In our previous work [8] we automatically label cast members in video using only the video stream and textual information in the form of sub-titles and aligned transcripts. The method consists of three stages: (i) visual processing to obtain face tracks for individual people in the video; (ii) speaker detection to determine if a name proposed by the aligned transcript should be used to label a face track; and (iii) classification, where the unlabelled face tracks are labelled by a classifier trained on the labelled tracks. We briefly summarize these three stages which form the starting point for this paper.

Face tracks and their descriptors. Frontal faces are detected in every frame of the video and tracked throughout a shot, such that each track is of a single character. The aim of the subsequent algorithm is to label (associate names with) each *track*. Each face in the track is represented by a feature vector, computed from the image regions around 13 facial features (based on the eyes, nose and mouth). A track is represented by this set of feature vectors.

Speaker identification and exemplars. A transcript is aligned with the subtitles using dynamic time warping, so that speaker names appearing in the transcript are associated with a time interval of the video. This is weak supervision because the person speaking may not be visible or detected, and there may be other faces in the scene. To strengthen the supervision, names are only associated with tracks where the face is detected as speaking. Nevertheless, this is still noisy supervision. Speaker detection is based only on visual information. The outcome of this stage is that some of the tracks are labelled (these are referred to as exemplars). Note that our aims differ from related work which has made more extensive use of script text [6] in that we aim to build models which allow labelling of faces with *no* associated text, rather than selecting from candidate names obtained from a full transcript.

Classification of unlabelled tracks. The exemplar tracks are used to label the remaining tracks using their visual descriptors. This is formulated as a nearest neighbour classifier, with tracks being labelled by their nearest exemplar based on the min-min distance between their sets of feature vectors.

2. Face detection, tracking and alignment

In this section we describe our method of generating tracks containing both frontal *and* profile faces, which is our first improvement over [8].

2.1. Face detection

Two independent face detectors are used: one for approximately frontal faces, and one for approximately "3/4 view" to full left profile; right profiles are detected by running on a mirrored input image. Detectors for each view were implemented using a multi-scale sliding-window classifier. Histogram of Oriented Gradients (HOG) feature extraction is used and a linear support vector machine (SVM) [7]. Training data came from web images and is disjoint to all test data used here. Details are given in [11].

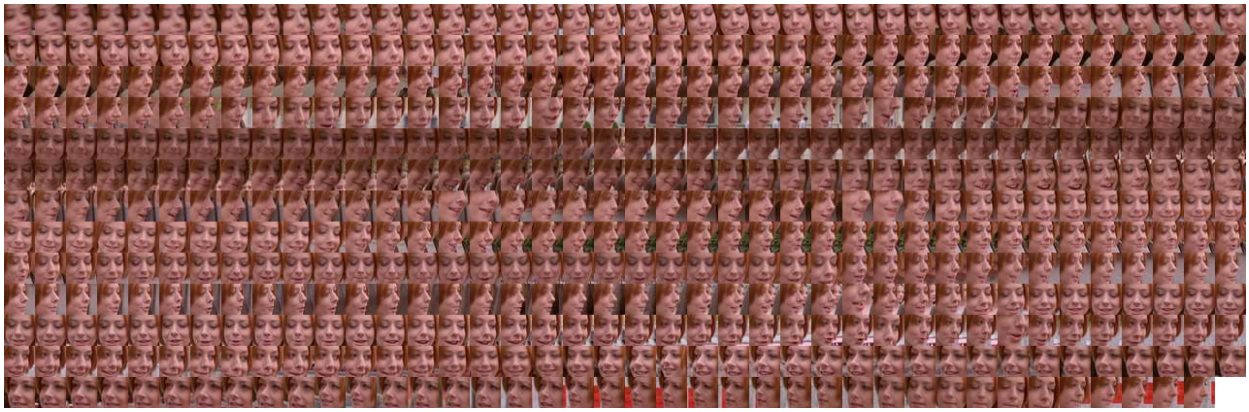
Merging frontal and profile detections. As is common with sliding-window detectors the face detector outputs multiple responses at nearby locations and scales, and faces between frontal and profile views are sometimes detected by both detectors. Two stage processing is applied to remove multiple detections and merge ambiguous frontal/profile detections. First, conventional non-maximum suppression is applied to the individual detectors. Second we merge frontal/profile detections which are of the same face using agglomerative clustering based on the overlap of the detections.

Face detector coverage. Example face detections can be seen in Figure 1 and throughout. For a 40 minute video we obtain around 65,000 raw detections after non-maximum suppression. The merging process reduces this set to around 45,000 detections without discarding valid faces. Of these around 42,500 are true positives, with approximately 30% being profile views, indicating the importance of moving beyond frontal face detection. At this high detection rate the false positive (non-face) rate is quite high: around 6% in total, with the non-frontal detector accounting for around 80% of the false positives. The majority of these can be removed by tracking (Section 2.2) and removing intermittent detections (short tracks), and by measuring confidence in facial feature localization (Section 2.3). We return to the issue of coverage in Section 6, where we assess coverage in terms of what proportion of all appearances of a character are detected.

2.2. Connecting faces by tracking

Faces detected in different frames of the same *shot* are connected by tracking. This means that the subsequent recognition effort is correctly targetted on a character (with temporal longevity), rather than on individual detections. For recognition, the principal advantage of grouping faces

(a) Face detections and a sample of point tracks over 519 frames



(b) Extracted faces

Figure 1. Example results of face tracking by point feature tracking. (a) shows a subset of frames for a track of 519 frames (20 seconds) in length. The characters walk through a scene such that the background is continuously changing (compare first frame to last). For each face detection a subset of point tracks connecting the face to the previous detection shown are drawn in yellow. (b) shows the extracted faces. Note that the tracker seamlessly connects frontal and profile face detections.

into tracks based on temporal continuity is that we gain additional examples of the facial appearance 'for free'. Matching *tracks* rather than individual faces is more effective since, for example, only the *closest* pair of faces spanning a pair of tracks needs to be matched – this is exploited in the 'min-min' kernels discussed in Section 5.

In our case, where we address detection and recognition of both frontal and profile views, tracking has an additional importance: by successfully tracking between frontal and profile views we can link frontal/profile appearances which could never be matched directly. The key here is that the temporal coherence provides the link between the images. As discussed in Section 3, this is essential in mining example images of characters by automatic speaker detection: by recognizing instances of frontal speaking and tracking to profile we extract profile exemplars, and vice-versa.

We thus require our tracker to be robust enough to track between frontal and profile views but not to 'drift' onto non-face regions. Some promising results starting with frontal-only detections have been shown in [16] using color-histogram matching. We found that such an approach works poorly for video such as Buffy where lighting is challenging, with scenes often set at night and fast motion – the success in previous work [16] may be due to the sitcom genre investigated ('Friends'), where scenes are well-lit and contain little camera or subject motion.

The method we propose here is an extension of [8] and combines point tracking with clustering to obtain very robust tracks. The Kanade-Lucas-Tomasi feature tracker [18] is applied to find evidence for connections between faces detected by the face detector. In [8] the feature tracker was applied independently of the face detector. However, this can result in few features on faces which are either small or in images with strong texture in the background (given a finite budget of features per frame). To avoid this problem, here we seed the feature tracker with features on every face detection, tracking between detections, thus ensuring that each face is covered by a dense set of tracks. The tracker is run from the first to last frame of each shot, and symmetrically from the last to first frame. In this way, we ensure that features which are strong in faces appearing later in the shot also contribute connections between faces.

Given a set of feature point tracks the support for connecting a pair of faces is measured by the number of features which are in common to both faces *i.e.* those which intersect the bounding boxes of both faces, and normalized by the number of feature tracks which pass through either but not both faces (*i.e.* an *overlap* score). Face pairs for which this ratio exceeds 0.5 are considered candidates for connection. Agglomerative clustering is applied as in the

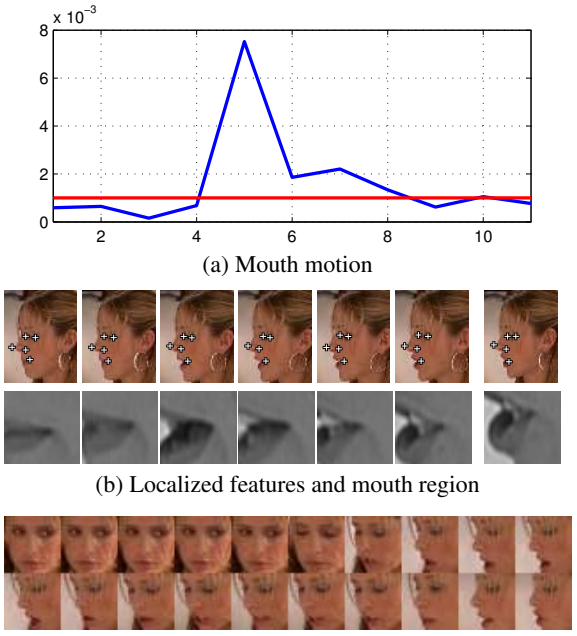
Figure 2. Facial feature localization in profile views. Note the accurate feature localization despite variations in pose between '3/4' view to full profile and challenging lighting.

intra-frame merging method (Section 2.1) but with no pose bias. With no tuning of parameters the method gives extremely robust results, connecting *e.g.* 45,000 face detections into around 2,000 tracks with *zero* incorrect merges. Figure 1 shows an example of tracked faces over a long shot with significant camera and subject motion and variation in pose of the face between frontal and profile. By delegating the face tracking problem to point tracking the method deals effortlessly with variation in face pose from frontal to full profile and copes with missing detections. Since the tracking task is formulated as one of connecting detections, the method exhibits none of the drift or failure to terminate tracks associated with conventional online tracking, and has no critical parameters to be specified manually.

2.3. Facial feature localization

The output of the face detector exhibits some noise over location and scale, particularly for views which are between frontal and profile. Facial feature (eye, mouth, etc.) localization is therefore useful as a means to better align pairs of faces, and subsequently extract descriptors based on the facial features after a viewpoint normalization. We extend our approach [8] to profile views. The method combines a discriminative model of feature appearance in the form of boosted classifiers using Haar-like features [20] with a generative model of feature locations. The location model uses a mixture of Gaussians, where each mixture component has a tree-structured covariance such that efficient inference for the MAP locations can be performed using the generalized distance transform [10].

Training images for the profile feature detector were obtained from a movie disjoint to all test data here and hand-annotated with the position of five features: the corners of the eye, the tip of the nose, the junction between nose and face, and the corner of the mouth. We also marked points on the ear where visible, but found that because the ear is often occluded by hair this feature, while not causing failure in the model, could not be reliably localized. Figure 2



(a) Mouth motion

(b) Localized features and mouth region

(c) Face track joining frontal and profile faces

Figure 3. Speaker identification for profile faces. (a) Inter-frame differences for 11 frames of a face track. The horizontal line indicates the 'speaking' threshold. (b) Above: Extracted face detections with facial feature points overlaid for frames 3–9. Below: Corresponding extracted mouth regions. (c) Original face track from frontal to profile views. A label is proposed for the automatically identified profile speaker detection. Note the character does not speak while in a frontal pose.

shows some results of automatic feature localization in profile views. The method gives quite accurate localization over the range of pose ('3/4' to profile) detected by the profile face detector. We found that the use of features such as the tip of the nose which are close to the boundary of the face (such that the feature window covers both face and background) did not cause significant problems, and as can be seen the method can cope with quite extreme lighting, low resolution images (small faces), and moderate motion blur.

3. Speaker detection

As noted in Section 1.1, speaker detection is the key to extracting useful supervisory information from aligned subtitles and scripts [8] – while the knowledge from text that a character is speaking gives a very weak cue that they are actually visible, an on-screen character speaking gives a very strong cue to their identity. Indeed, from the text alone it cannot be known if the character speaking is even visible in the shot (they may be off-camera), or if there are other non-speaking characters visible.

Since one of our principal aims is to improve coverage, we tackle the task of detecting when a character is speaking in profile as well as frontal views. We found that the sim-

ple method proposed in [8] could be generalized to profile views effectively. Figure 3 outlines the method. The new model for feature localization in profile views enables a region around the mouth to be extracted. For profile views in particular we limit the size of this region such that it does not contain too much background outside the face (Figure 3b). Given a detection in one frame a local search is carried out in the previous frame for the corresponding mouth region, and the image distance to the closest matching region measured (Figure 3a). Applying a simple threshold to this distance gives a reliable, if sparse, detection of speech. Faces detected as speaking in a given frame can then be assigned a name by transfer from subtitles co-occurring in time.

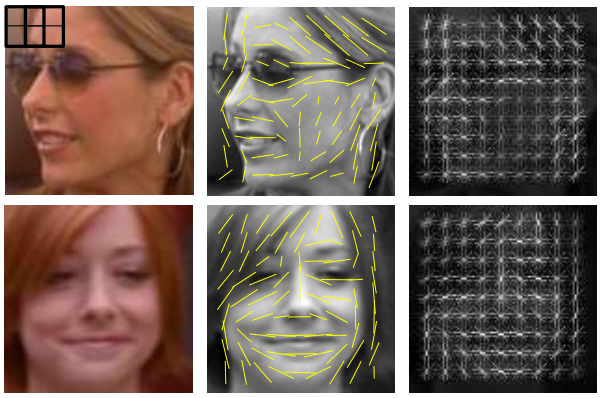
In addition to improving coverage by automatically obtaining some training labels for profile faces, detecting speaking in profile also improves the amount of training data available for *frontal* faces. Figure 3c shows an example. In this sequence (a subset of frames are shown) the character *only* speaks while in profile. By correctly detecting this, then exploiting the face tracks, labels can automatically be assigned to the connected frontal faces.

4. Representing face appearance

The goal is to obtain a descriptor of the face robust to variations in pose, lighting and facial expressions, and also suitable for the multiple kernel framework (Section 5).

Detected facial features are used to estimate a similarity transformation between the image and a rectified frame compensating mainly for the noise in the output scale of the face detector and in-plane rotation of the face. A HOG descriptor [7] is then extracted from the normalized frame, as illustrated in Figure 4. The HOG descriptor measures histograms of image gradient orientation at a coarse grid of spatial locations, and is thus robust to misalignment errors and illumination variation. Here we extract a HOG descriptor consisting of a 9×9 grid of overlapping blocks, where each block contains 4 cells, with a 6 bin orientation histogram within each cell, resulting in a $81 \times 4 \times 6 = 1,944$ dimensional descriptor. Note that for extracting descriptors (and recognition as outlined in Section 5), frontal and profile faces are treated separately.

We also compare performance of the HOG descriptor to the baseline patch-based descriptor of [8] available at [1]. A pixel-wise descriptor of the local appearance around a facial feature is extracted by taking the vector of pixels in the patch and normalizing (so that the intensity has zero mean and unit variance) to obtain local photometric invariance. The descriptor for the face is then formed by concatenating descriptors extracted around the 13 frontal facial feature locations, *e.g.* the corners and centers of the eyes, nose and mouth.



(a) (b) (c)

Figure 4. Representing facial appearance using HOG descriptors: profile face (top) and frontal face (bottom). The HOG descriptor has 81 overlapping blocks on a 9×9 grid. Each block contains 4 cells, with adjacent blocks overlapping by two cells (a, top). The descriptor is extracted from a slightly larger region then the actual face detection to capture the outline of the face and appearance of the hair. (a) the original image; (b) the dominant orientation within each block; (c) orientation histograms within each block.

5. Classification by multiple kernels

Our aim is to learn a SVM classifier [17] to discriminate the tracks of one person from the tracks of others. Rather than using a single pre-specified kernel, the kernel is a linear combination of given base kernels [3, 19], with each base kernel corresponding to a different feature. For two tracks, i and j the composite kernel has the form

$$K(i,j) = \sum_f b_f K_f(i,j) \tag{1}$$

where $K_f(i,j)$ is the kernel corresponding to feature f between tracks i and j. The intuition is that this combination behaves as an "or" of the base kernels, so that any features that match well will contribute strongly. The linear combination has similarities to the type of classifer learnt in boosting, but here the individual classifiers need not be weak, and the learning (see below) is not greedy.

The features used may be any of those defined in Section 4. Given that each face track is represented by sets of facial descriptors $F^f = \{\mathbf{F}_m^f\}$ for different features f, a base kernel between two face tracks, i and j is then defined as

$$K_f(i,j) = \exp(-\gamma_f d(F_i^f, F_j^f)^2), \tag{2}$$

where γ_f is a parameter of the kernel and $d(.,.)$ is the min-min distance between sets of descriptors F_i^f and F_j^f, given by

$$d(F_i^f, F_j^f) = \min_{\mathbf{F}_k \in F_i^f} \min_{\mathbf{F}_l \in F_j^f} ||\mathbf{F}_k - \mathbf{F}_l||. \tag{3}$$

Note that kernels formed using the min-min distance given by (3) are not necessarily positive definite. However, we

found that in practice this does not cause problems during learning.

We form one base kernel for each of the 81 *blocks* of the HOG descriptor, resulting in a total of 162 base kernels, treating frontal and profile faces separately. We investigate first the canonical kernel combination, denoted SUM in the following experiments, where weights are set uniformly for all base kernels, *i.e.* $b_f = 1/N_f$, where N_f is the number of base kernels. Although here the same set of weights are used for each person, the canonical kernel combination together with the exponential form of the kernel, given in (2), robustly combines information from different blocks of the HOG descriptor – a large distance between two faces at a particular block of the HOG descriptor, *e.g.* due to occlusion, will only give a very limited contribution to the kernel sum.

Following the method of [5], a different set of weights $\{b_f\}$ can also be learnt *for each person*. Such weights can emphasize more discriminative features for a person or ignore features that are not discriminative by setting b_f to zero. We learn the weights $\{b_f\}$ for each person using the cost function and optimization method for Multiple Kernel Learning (MKL) given in [19].

Finally, we also investigate using the entire HOG descriptor as a single feature for comparing faces, resulting in two base kernels – one for profile and one for frontal faces – which are then combined using uniform weights. This method is referred to as CAT.

Note that in all the above methods the kernel combination also provides a natural way to aggregate information from profile and frontal views as a single face track may contain both frontal and profile face detections. In all cases, one SVM classifier is learnt for each character using a 1-vs-all scheme. Each test track is then labelled according to the classifier with the maximum score.

6. Experimental results

The proposed methods were tested on seven episodes from season 5 of "Buffy the Vampire Slayer": episodes 1–6 and 13. Table 1 shows statistics of the number and type of faces detected and tracked for these episodes.

How much coverage? Almost all face recognition papers which consider TV and film material only report results on the faces actually detected – not on the actual appearances of the characters (whether they are detected or not). One of our main aims in this work is to increase the *coverage* of the method *i.e.* how much of the video is labelled. Here we quantify how successful we are in capturing *all* appearances. For episode 2 we have labelled every 10th frame with ground truth as to which of the 11 principal characters are present, and where present we have specified their pose. Each character present is labelled as one of: frontal, profile, or 'other'. The label 'other' is assigned where the character faces away from the camera, the face covers only a few pixels or *e.g.* only a shoulder or piece of clothing is visible *i.e.* all cases insufficient to identify the character without additional contextual information. For this episode 42% of the appearances are frontal, 21% profile, and 37% 'other'.

The coverage is significantly improved by adding profiles: using the tracks from our previous method [8] (which does include some 3/4 views that count towards profiles) 54.5% of all possible frontal and profile faces are recovered; with the integrated frontal and profile tracking introduced here, this increases to 78.7%. Note that if we consider all appearances of a character including the 'other' class, the potential recall is just under 50% – this suggests that future work on increasing coverage must go beyond the use of face detection as a first step.

Labelling performance: As shown in Table 1, the speaker detection labels between 123 and 215 face tracks depending on the episode. Of these between 80% and 90% are correct. The face tracks labelled by speaker detection form the training data for each episode. From the rest of the tracks we consider those with more than 10 face detections for labelling. Note that as training data is obtained automatically using speaker detection, no manual annotation of any data was performed other than to evaluate the method (ground truth label for each face track).

In all experiments, the SVM misclassification penalty parameter C is set to 10. For each base kernel, the inverse kernel width parameter γ_f is set to $\bar{d}_f/5$, where $\bar{d}_f$ is the average distance between all face tracks within an episode for feature f. The relative weight for the L_1 regularizer on kernel weights b_f in the multiple kernel learning cost function [19] is set to 10^{-6} in all experiments. The parameters were coarsely tuned on episode 2 and left unchanged for the remaining episodes.

Performance is measured using precision and recall. The term "recall" is used here to mean the proportion of tracks which are assigned a name at a given confidence level of the classifier, and precision is the proportion of correctly labelled test tracks. Results across all seven episodes are summarized in Table 2 by the average precision (AP), which corresponds to the area under the precision recall curve. Precision-recall curves for three episodes are shown in Figure 5. The kernel sum with equal weights (SUM) gives almost the same performance as the learnt kernel combination (MKL). Hence, in this case the benefit of learning a sparse kernel combination is mainly computational as at test time fewer features need to be extracted and compared. However, kernel combination (either MKL or SUM) is in most cases beneficial compared to using a single kernel for the entire feature vector (CAT). This might be attributed to the robust feature combination performed by kernel combination methods as discussed in Section 5. To assess the effect of noise in automatically obtained training labels the mul-

	Episode						
	1	2	3	4	5	6	13
(a) frames	62,620	62,157	64,100	63,700	64,083	64,107	64,075
(b) face detections (frontal)	28,170	28,055	19,421	24,510	25,884	30,202	26,794
(c) face detections (profile)	8,315	14,327	13,931	12,996	8,103	11,685	8,449
(d) face detections (all)	36,485	42,382	33,352	37,506	33,987	41,887	35,243
(e) face tracks	1,506	2,088	2,140	1,985	1,532	2,020	1,548
(f) training tracks w/ spk. det.	202	198	200	182	162	123	215
(g) test tracks (longer than 10)	390	558	620	470	442	679	462
(h) main characters	14	17	13	14	14	19	14

Table 1. Statistics for episodes 1–6 and 13: (a) number of frames, (b)-(d) number of detected faces (excluding false positive detections), (e) number of face tracks, (f) number of labelled face tracks obtained with automatic speaker detection, (g) number of test tracks (with more than 10 face detections) considered for labelling, (h) number of principal characters automatically extracted from the script.

	Episode						
Method	1	2	3	4	5	6	13
(a) MKL	**0.90**	**0.83**	**0.70**	**0.86**	**0.85**	**0.70**	0.80
(b) SUM	0.89	**0.83**	0.68	0.82	**0.85**	0.69	0.78
(c) CAT	0.83	0.76	0.62	0.82	0.81	0.66	**0.81**
(d) MKLgt	0.94	0.91	0.96	0.91	0.84	0.86	0.94
(f) Baseline	0.74	0.60	0.46	0.60	0.62	0.53	0.65

Table 2. Average precision for different classification methods on episodes 1–6 and 13. Methods (a-c), and (f) are trained using noisy labels obtained automatically from text. For method (d) the noisy labels were manually corrected.

tiple kernel classifier is also trained using noiseless manually corrected labels (method MKLgt). For some episodes the SVM classifier is able to deal with some amount of noise in the training data [9]. However, for others (*e.g.* episodes 3 and 6) there is a significant difference in performance between using ground truth labels for the exemplars and using those obtained automatically. This difference is partly due to characters that have a small number of tracks in some episodes – errors in the exemplar labels for such cases cannot be overcome by the SVM because there is insufficient training data. The problem of limited training data can be addressed by combining training exemplars from all episodes (not shown in table 2), improving AP from 0.70 to 0.82 and from 0.70 to 0.79 for episodes 3 and 6, respectively.

All proposed methods significantly improve performance compared to the baseline method, which trains a standard single kernel SVM classifier using patch descriptors [8] extracted only from frontal faces (see Section 4 for details). Of particular note is that the *recall* is significantly improved, by 20–30% over the baseline method (Figure 5). This is due to the fact that the proposed methods can label face tracks containing only profile faces, whereas the baseline method cannot. Examples of correctly detected and named characters obtained using the MKL method are shown in Figure 6.

7. Conclusions and future work

We have demonstrated that learning person specific classifiers by kernel combination can improve the accuracy of automatic naming of characters in TV video. In addition, we have shown that seamless integration of frontal and profile face detections throughout the face recognition pipeline can increase the proportion of video frames labelled, and appearances of characters in significantly non-frontal poses can be successfully detected and recognized. In future work, we plan to investigate incorporating other (non-facial) cues, such as hair and clothing, in the multiple kernel learning framework.

Acknowledgements: We are very grateful to Francis Bach and Manik Varma for discussion on multiple kernels. Financial support was provided by RCUK, EU Project CLASS, the Royal Academy of Engineering, Microsoft, and the MSR-INRIA laboratory. We would like to thank Alex Klaeser and Cordelia Schmid for providing the face detector.

References

[1] http://www.robots.ox.ac.uk/ vgg/research/nface/index.html.

[2] N. E. Apostoloff and A. Zisserman. Who are you? – real-time person identification. In *Proc. BMVC.*, 2007.

[3] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality and the SMO algorithm. In *International Conference on Machine Learning*, 2004.

[4] T. Berg, A. Berg, J. Edwards, M. Mair, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. CVPR*, 2004.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proc. CIVR*, 2007.

[6] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *Proc. ECCV*, 2008.

[7] N. Dalal and B. Triggs. Histogram of Oriented Gradients for human detection. In *Proc. CVPR*, 2005.

[8] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – Automatic naming of characters in TV video. In *Proc. BMVC.*, 2006.

[9] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5), 2009.

[10] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[11] A. Klaeser. Human detection and character recognition in TV-style movies. In *Informatiktage*, 2007.

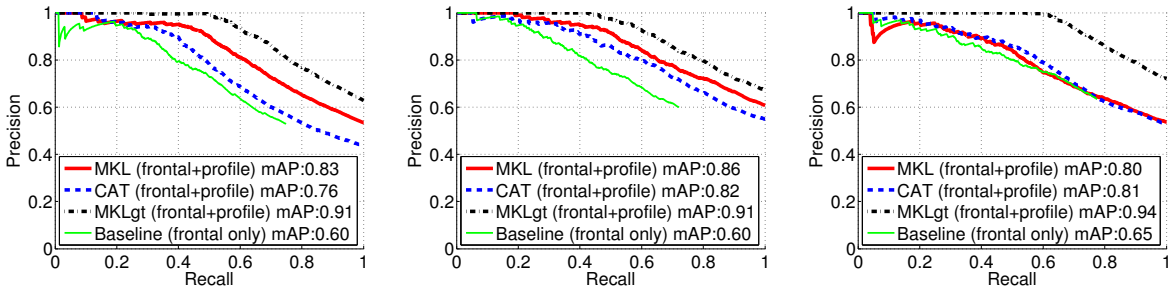


Figure 5. Precision/recall curves for episodes 2 (left), 4 (middle) and 13 (right). Recall refers to the proportion of test face tracks which are assigned labels by the different methods at a given confidence level. As performance of the MKL and SUM methods is similar, only the MKL precision-recall curve is shown.

Figure 6. Examples of correct detection and naming throughout episodes 2 and 5. Note that correct naming is achieved over a very wide range of scale, pose, facial expression and lighting.

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *ICCV*, 2007.

[13] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *Proc. ECCV*, 2008.

[14] P. Li, H. Ai, Y. Li, and C. Huang. Video parsing based on head tracking and face recognition. In *Proc. CIVR*, 2007.

[15] T. Mensink and J. Verbeek. Improving people search using query expansions: How friends help to find people. In *Proc. ECCV*, 2008.

[16] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *Proc. ICCV*, 2007.

[17] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[18] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, 1994.

[19] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proc. ICCV*, 2007.

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.