

Friendly Faces: Weakly supervised character identification

Matthew Marter, Simon Hadfield, and Richard Bowden

CVSSP, University of Surrey, UK
{m.marter, s.hadfield, r.bowden} @surrey.ac.uk

Abstract. This paper demonstrates a novel method for automatically discovering and recognising characters in video without any labelled examples or user intervention. Instead weak supervision is obtained via a rough script-to-subtitle alignment. The technique uses pose invariant features, extracted from detected faces and clustered to form groups of co-occurring characters. Results show that with 9 characters, 29% of the closest exemplars are correctly identified, increasing to 50% as additional exemplars are considered.

1 Introduction

This paper presents an approach to weakly supervised character identification from video footage using subtitle and script information as weak supervision in a facial clustering process. This is challenging as the the annotation gained from scripts and subtitles is a weak verbal commentary and the pose, expression and lighting of characters is highly variable. To overcome these problems we present



Fig. 1: Example faces from Friends opening sequence.

an approach which incorporates both subtitle and shot boundary detection into a script alignment process which gives accurate start and end times for each scene. The script also indicates which characters are present within each scene. By automatically detecting faces throughout the video, accurately regressing facial features and then using these facial features in an unsupervised clustering

process, individuals can be identified without user intervention. The script information is used as weak supervision in this clustering process to identify the likelihood of characters over subsets of frames. We demonstrate a completely unsupervised approach to character identification and show results on the popular American sitcom “Friends” (See Fig 1).

The remainder of this paper is structured as follows. In section 2 we discuss relevant related work in the areas of script alignment, character identification and shot boundary detection. In section 3 we present the proposed methodology with scene identification and timing described in section 3.1, facial feature detection and extraction described in section 3.2 and the facial clustering process described in section 3.3. Section 4 describes results and conclusions and future work are discussed in section 5.

2 Related work

Computer vision has seen the use of video footage increase in previous years and areas such as action recognition have moved away from staged datasets such as KTH [29] and Weizmann [2], which are generally considered “solved” with many techniques reporting performance of 95% or more. As a result, recent action recognition datasets have been concerned with action recognition “in the wild”, dealing with unconstrained variations in location, lighting, camera angle, actor and action style. Alternative means for obtaining data have been explored, including videos obtained from youtube [17] and clips extracted from commercially available films (e.g. “Hollywood” [15], “Hollywood 2” [19] and “Hollywood 3D” [12]).

The size of the Hollywood 2 dataset was facilitated by the automated extraction process employed by Marszalek *et al.* [19]. Such techniques offer ease in data collection and rely on the extraction of time stamped subtitles from the film. These subtitles are then aligned with film scripts, which describe the behaviour of the actors in addition to dialogue.

Perhaps one of the earliest attempts to use script information was the work of Everingham *et al.* [8][9]. Everingham used the subtitles to provide timing information by matching it with the dialogue in the scripts. When there are inconsistencies with the script and subtitle information, dynamic time warping is used to find the best alignment. When subtitle information is either not available or there are scenes without dialogue, other methods need to be used to provide alignment information. Sankar *et al.* [28] use a combination of location recognition, facial recognition and speech-to-text to align scripts when subtitles are not available. However, the approach requires manual labelling of characters and location information was hard-coded to repetitive use of stock footage. Subtitles have also been used as weak supervision in learning sign [6, 4, 26]

Subtitles only provide timing information for dialogue. However, identifying shots within a video can help to identify when a scene containing dialogue might start or end. A shot in a video sequence is a group of continuous frames between edits. The edits can take various forms such as fades or abrupt transitions. To be able to learn about scenes, a video must first be divided into shots. Hanjalic [13] defines the problem of Shot Boundary Detection (SBD) as finding discontinuities

in the feature distance between frames. Many different feature types have been used for SBD including: Pixel differences[35], Colour histograms[20], tracking of features[10] and mutual information[5]. Yuan et al.[34] try to provide a formal framework for SBD and review many existing techniques while suggesting optimal criteria for the different sections of the framework. To take advantage of existing information, Patel et al.[25] use the motion vectors from the video compression to detect cuts. Due to the large number of approaches available, there have been many survey papers such as Boreczky and Rowe[3] and Smeaton et al.[30]. Boeczky and Rowe find that, in general, the approaches based on region-based comparisons and motion vectors worked better and simpler algorithms out performed more complex ones. Smeaton et al. found that there was usually very little difference in performance between the top approaches and shot cut performance is still a lot higher than gradual transitions.

This work integrates shot cut detection into the script-subtitle alignment process to provide accurate scene level annotation. We then use character information from the scripts as weak supervision to automatically identify characters by clustering their facial descriptors.

3 Methodology

Figure 2 gives an overview of the approach. Firstly the subtitles are extracted from the video along with the shot boundaries which are detected in the video footage. Fuzzy text matching is used to match the dialogue in the script with that of the subtitles and scene boundaries are constrained to only occur on shot boundaries. This extends the dialogue timing to give the start and end times of scenes. Using the time aligned script, all scenes for each character are identified along with the presence of other characters in the same scenes. This is passed to the clustering process along with the face descriptors for the corresponding frames. Face descriptors are extracted from detected faces by regressing the facial pose and extracting SIFT features around the contour of the face. Unsupervised clustering then provides sets of visually similar face descriptors which are assigned identities from the script.

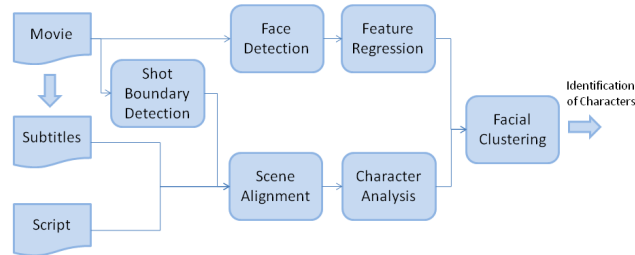


Fig. 2: Character identification framework.

3.1 Scene Identification

Television and film scripts contain the details for the scenes within a video sequence. There is a standardised format for scripts and their contents. This

format specifies the location in which the scene occurs, editorial information for the shots, a description of the activities and actions occurring within the scene and the dialogue spoken by the characters. The location information is contained within the “slug lines” which state if the scene is exterior or interior, where it is and the time of day it occurs. The script Γ contains text elements t , $\Gamma = t_0 \dots t_n$. The scenes within a script are a subset of the script. The scene boundaries A are defined as $A \subset \Gamma$. Equation 1 defines the text scene Σ^T as the text between the scene boundaries in the script,

$$\Sigma_j^T = \{j | A_{k-1} < j < A_k\}. \quad (1)$$

The script does not, however, contain any information about the timing of the scenes within the video. To use the scripts as annotation, the contents of the video and the script need to be aligned.

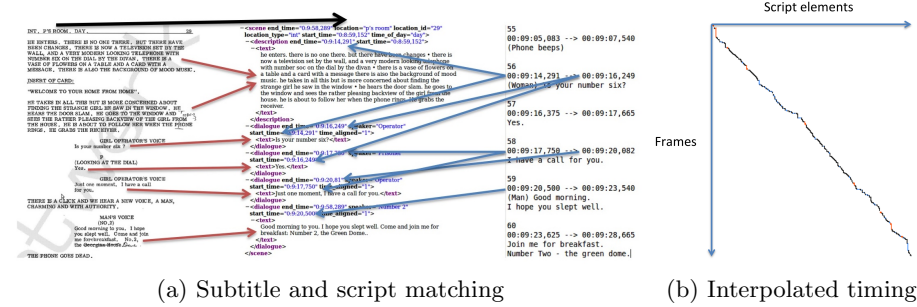


Fig. 3: Alignment between the text elements in the script and subtitles

Initial alignment can be performed by matching the video subtitles with the dialogue in the scripts. The subtitles contain text and timing information. Figure 3 shows the alignment between the script and subtitles and an example of how the script and subtitles are matched. Black elements are well aligned, blue elements are interpolated and orange elements are when multiple script elements occur simultaneously. The subtitles sub have text, $\text{sub}^t = t$, as well as start frames, sub^s , and end frames, sub^e . Fuzzy matching between subtitle text and script text is used to assign start and end frames to script text as in equation 2. This gives us the images that belong to a scene Σ^I ,

$$\Sigma_j^I = \{j | \text{argmin}(\text{sub}^s) < j < \text{argmax}(\text{sub}^e) \text{ and } \text{sub}^t \in \Sigma_j^T\}. \quad (2)$$

Scene Boundary Refinement The frames which occur between subtitles are ill-defined. A scene can contain many shots and a scene cannot change during a shot. Our method requires that we divide the video into shots so that they can be matched to scenes. We use a shot boundary detection method based on calculating the homography from one frame to the next. We trigger a shot boundary when the homography between shots is invalid.

A video consists of a set of images, $I = i_{1...n}$. The set of shot boundaries (SB) is where the homography ($h_{j,j+1}$) between consecutive images (i_j and i_{j+1}) leads to a transformed image with an area smaller than a threshold (t),

$$SB = \left\{ j \mid \frac{|h_{j,j+1}(i_j)|}{|i_j|} < t \right\}. \quad (3)$$

A particular shot (S_k) is defined as the set of frames which between 2 consecutive shot boundaries (SB_k and SB_{k+1}),

$$S_k = \{i_j | SB_k < j < SB_{k+1}\}. \quad (4)$$

We define the scenes Σ^S using the shots that belong to them,

$$\Sigma^S_k = \left\{ \bigcup_j S_j \mid S_j \cap \Sigma^I_k \neq \emptyset \right\}. \quad (5)$$

This gives us all the shots belonging to a scene so that frames which occur after or before a subtitle are also included. This avoids missing face candidates that occur without dialogue.

3.2 Face extraction



Fig. 4: Facial Feature Regression

Cootes suggested learning a simple linear mapping between facial image intensity difference and the pose of an Active Appearance Model (AAM) in 1998 [7]. Similar ideas have since been applied to tracking rigid objects [14][21] as well as using more complex, non-linear mappings in regression tracking [32]. Zimmerman et al. proposed the idea of a sequential linear predictor and this idea was applied to facial feature regression by Ong and Bowden [23][24], whose sequential and hierarchical linear predictors can be trained using Monte-Carlo sampling to accurately track facial features in real-time. The same principle was adopted by Xiong and De la Torre [33] to regress from SIFT [18] features rather than image intensities. Both approaches use a simple cascade of sequential linear mappings but SIFT gives better person independence in the regression process. Another difference between Ong and Xiong is the latter regress all feature points

in a single regressor which captures the motion dependency of the features unlike Ong who treat each facial feature as independent. We combine these ideas to train a cascade of linear predictors that regress facial position from SIFT features extracted at each facial landmark.

Given a set of face images $\mathbf{d}_i \in \mathbf{D}$ with associated facial landmarks $\mathbf{x}_i \in \mathbf{X}$, the objective is to find a mapping \mathbf{H} that can predict the displacement of the landmarks $\delta\mathbf{x}$ such that $\delta\mathbf{x} = \mathbf{H}\phi(\mathbf{x})$. The image observation, $\phi(\mathbf{x}) \in \mathbb{R}^{nm+1}$, is a 6733 dimensional concatenated SIFT vector where $n = 99$ is the dimension of a SIFT descriptor after projection through PCA, $m = |\mathbf{x}|/2$ is the number of facial landmarks (in our case 68) and a single dimension is added for the bias term in linear regression.

To learn \mathbf{H} , a training set of random displacements $\mathbf{T} \in \mathbb{R}^{(|\mathbf{x}| \times r | \mathbf{X}|)}$ is extracted by randomly offsetting the model from the true face location and recording the displacements. For each displacement the image observations are compiled into $\Phi \in \mathbb{R}^{(|\phi| \times r | \mathbf{X}|)}$ where $r = 10$ is the number of random offsets per image.

\mathbf{H} is then calculated as the least squares solution $\mathbf{H} = \mathbf{T}(\Phi\Phi^T)^{-1}$. As a single linear mapping is insufficient to model the complexities of a highly deformable object like a face, a sequence of regressors is used where $\delta\mathbf{x}^i = H^i\phi(\mathbf{x}_0 + \delta\mathbf{x}^{i-1})$.

We train our facial regressor using 5691 example images taken from the *300 Faces in-the-wild* challenge dataset (300-W) [27]. These images are annotated using the Multi-PIE [11] 68 point mark up shown in Figure 4a. The full dataset consists of consistent re-annotations for LFPW [1], AFW [36], HELEN [16] and XM2VTS [22]. We refrain from using the additional 135 IBUG images and the testing subsets of the aforementioned datasets which are retained for internal validation e.g. regressor convergence during training.

Figure 4a shows the 68 landmark points used in our regression and Figure 4b the result of applying the learnt regressor to a sample static image. At runtime, a Viola Jones boosted face detector [31] is applied to each frame. Following non maximal suppression, the regressor is independently applied at each positive detection using the mean face $\mathbf{x}_0 = \frac{1}{|\mathbf{x}|} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathbf{x}_i$ as the initial estimate as was used during training. The initial face estimate is scaled and translated to the detected face region and the SIFT features are set at one tenth of the face scale which translates to roughly half the size of an eye. The regressor typically converges onto the face using 4-5 linear predictors in the sequential cascade.

Figure 4c shows the process of regression for each regressor in the cascade starting from a mean face (dark) to final regression (yellow). An example of regressing multiple faces from a frame taken from Friends is shown in Figure 1.

3.3 Face Clustering

To allow us to name the faces we have extracted, we can cluster the face descriptors and label them with character names from the script. To avoid clustering all of the face descriptors at once, which could cause less pure clusters, we divide the descriptors into subsets that contains all occurrences of a certain character. For each scene, the characters Π_k are defined using the characters z present in Σ^S_k . The set of all the characters in the episode, Z , is defined by $Z = \bigcup_k \Pi_k$.

With the script aligned to the video, we can use the scene boundaries to work out the frames and face descriptors belonging to these scenes. This gives us a matrix of images with character co-occurrence,

$$\mathbf{E}_{xy} = \{i | i \in \Sigma_z^S \text{ and } z_x \in \Pi_z \text{ and } z_y \in \Pi_z\}. \quad (6)$$

We can then normalise by the number of frames each character is in,

$$\bar{\mathbf{E}}_{jk} = \frac{|\mathbf{E}_{jk}|}{\sum_l |\mathbf{E}_{jl}|}. \quad (7)$$

For each character z_k we perform k-means on the descriptors for all face candidates Υ_k . The co-occurrence of characters is used to reduce the number of characters present in the input to the k-means. The face candidates are obtained by,

$$\Upsilon_k = \{v^F | v^F \in \Sigma_j^S \text{ and } z_k \in \Pi_j\}. \quad (8)$$

The number of clusters p is calculated from the total number of characters present in the scenes using,

$$p_i = |\{\mathbf{E}_{jk} \neq 0\}|. \quad (9)$$

We expect that the face descriptors will be clustered into the characters. The cluster centres C are found by the k-means algorithm,

$$C_k = \text{kmeans}(\Upsilon_k; p_k). \quad (10)$$

We now have the the clusters of face candidates but we don't know which characters belong to each cluster. To overcome this problem we create a histogram of the number of face descriptors belonging to each cluster. The histogram M_{jk} is calculated by,

$$M_{jk} = \frac{\left| \left\{ v | v \in \Upsilon_j \text{ where } \arg \min_l (|v - C_l|^2) = k \right\} \right|}{|\Upsilon_j|}. \quad (11)$$

To match the labels from the co-occurrence histogram to the membership histogram, we re-order the membership histogram to find the minimum χ^2 distance. We expect the largest cluster to correspond to the character themselves as they would be present the most often within their scenes. This relies on characters appearing separately from other characters in different scenes.

4 Results

We evaluated the proposed technique on a dataset of approximately 35,000 images, obtained from the TV sitcom "Friends". This program is ideal for examining the approach, as the large cast of "main" (i.e. re-occurring and named) characters, makes the task especially challenging. In addition, scripts are easily

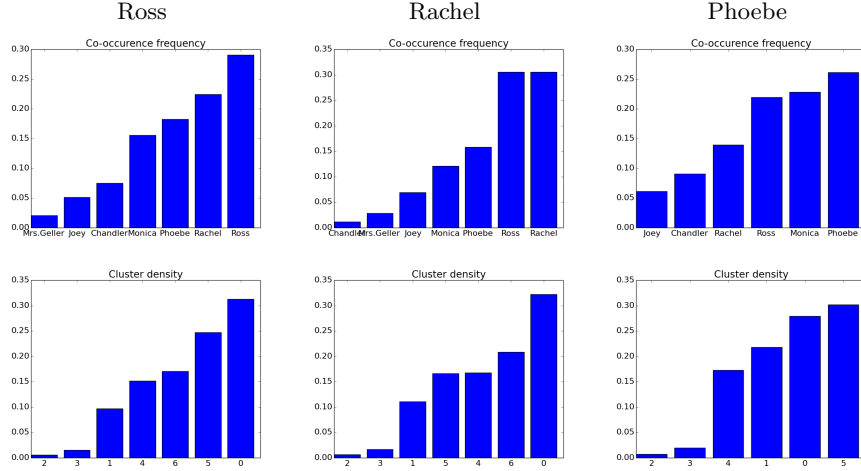


Fig. 5: Comparison of cluster density and co-occurrence frequency after identity assignment, for the 3 most common characters.

obtainable for the series, and the large amount of dialogue facilitates the accurate subtitle-to-script alignment. The total number of named characters present in the relevant scripts is 9, with co-occurrences spread across 14 different scenes.

From the dataset, around 20,000 face descriptors are extracted. After partitioning based on character co-occurrence as described in section 3.3, each character had on average of 6,830 face candidates.

4.1 Cluster feasibility

In order to assign identities, we assume that the density of the clusters relates to the frequency of character co-occurrences. In order to evaluate this assumption figure 5 compares histograms for the 3 most commonly occurring characters in the dataset.

It can be seen that, in general, cluster density correlates well with character co-occurrence frequency, meaning that the assumption used for identity assignment is likely a valid one. There are some inconsistencies, particularly with minor characters, as they tend to talk (and be visible) less often, even when they are technically present in the scene. Thus the cluster density of the smallest clusters is generally somewhat lower than what we would expect from the script.

4.2 Character identification

It is extremely time-consuming to manually annotate the identity of tens of thousands of face images. Avoiding this task is one of the primary motivations for our automatic data-driven approach. As such, we evaluate the technique in terms of “character exemplars”. For each cluster with an assigned identity, we extract the face candidate closest to the cluster centre, and record whether this exemplar belongs to the assigned character. These exemplars are shown in


















































Exemplar of character	Rachel									
	Ross									
	Chandler									
	Janice									
	Joey									
	Monica									
	Mr. Geller									
	Mrs. Geller									
	Phoebe									
		when co-occurring with character								
		Chandler	Janice	Joey	Monica	Mr. Geller	Mrs. Geller	Phoebe	Rachel	Ross

Fig. 6: Character co-occurrence matrix \mathbf{E} , represented using the assigned exemplars. Exemplars with a blue border are correctly identified. Gaps indicate pairs of characters which never co-occur within the dataset. Roughly one third of cluster exemplars have had the correct identity assigned.

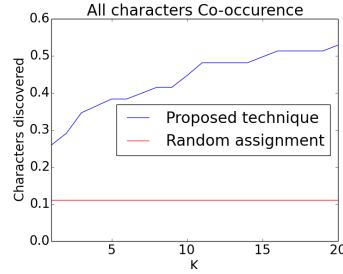


Fig. 7: Accuracy of identity assignment for the K-nearest exemplars.

figure 6 for the full co-occurrence matrix \mathbf{E} . Blue borders indicate exemplars with correctly assigned identities.

Overall 29% of the exemplars are assigned the correct identity. This is 2.5 times the accuracy achievable by random assignment, indicating that even the weak supervision provided by script co-occurrences, can be hugely beneficial.

We extend this evaluation to look at the the accuracy of the K-nearest exemplars, as shown in figure 7. If we examine up to the top 20 exemplars, we are able to identify more than 50% of the characters correctly.

5 Conclusions

In this paper, we have introduced an approach to identify characters in broadcast footage, without any manual intervention. Instead the technique uses weak supervision from script and subtitle alignments, to estimate character co-occurrences.

Face candidates were extracted, and described in terms of facial landmarks to mitigate pose variation. Modes in the face data were then estimated, and the density of these modes was used to assign character identities, by comparison to the expected co-occurrence rates. The technique was evaluated by examining the “exemplars” of these modes, 29% of which had the correct identity assigned (2.5 times the baseline from random assignment). Accuracy increased to over 50% when examining the top 20 exemplars per mode

In the future it would be interesting to investigate co-clustering techniques to collate character identity information across the rows of the co-occurrences. Using different clustering techniques that don’t assume spherical clusters such as GMMs should allow for better modelling of the distribution of characters as well as using more clusters than there are characters. The use of the actual speaking times from the subtitles could also be used to further improve performance.

Acknowledgements

This work was supported by the EPSRC project “Learning to Recognise Dynamic Visual Content from Broadcast Footage” (EP/I011811/1).

References

1. Belhumeur, P., Jacobs, D., Kriegman, D., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 545–552 (June 2011)
2. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proc. ICCV (2005)
3. Boreczky, J.S., Rowe, L.A.: Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging* (1996)
4. Buehler, P., Everingham, M., Zisserman, A.: Learning sign language by watching tv (using weakly aligned subtitles). In: In Computer Vision and Pattern Recognition (2009)
5. Cernekova, Z., Nikou, C., Pitas, I.: Shot detection in video sequences using entropy based metrics. In: Proc. ICIP (2002)
6. Cooper, H., Bowden, R.: Learning signs from subtitles: A weakly supervised approach to sign language recognition. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 2568–2574. IEEE (2009)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Proc. ECCV (1998)
8. Everingham, M., Sivic, J., Zisserman, A.: “Hello! My name is... Buffy” – automatic naming of characters in TV video. In: Proc. BMVC (2006)
9. Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing* (2009)

10. Gao, X., Li, J., Shi, Y.: A video shot boundary detection algorithm based on feature tracking. In: *Rough Sets and Knowledge Technology* (2006)
11. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. *Image Vision Comput.* 28(5), 807–813 (May 2010), <http://dx.doi.org/10.1016/j.imavis.2009.08.002>
12. Hadfield, S., Bowden, R.: Hollywood 3D: Recognizing actions in 3D natural scenes. In: *Proc. CVPR* (2013)
13. Hanjalic, A.: Shot-boundary detection: unraveled and resolved? *Circuits and Systems for Video Technology* (2002)
14. Jurie, F., Dhome, M.: Real time robust template matching. In: *BMVC* (2002)
15. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *Proc. CVPR* (2008)
16. Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.: Interactive facial feature localization. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *Computer Vision – ECCV 2012, Lecture Notes in Computer Science*, vol. 7574, pp. 679–692. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-33712-3_49
17. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos in the wild. In: *Proc. CVPR* (2009)
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* (2004)
19. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: *Proc. CVPR* (2009)
20. Mas, J., Fernandez, G.: Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003* (2003)
21. Matas, J., Zimmermann, K., Svoboda, T., Hilton, A.: Learning efficient linear predictors for motion estimation. In: *Comp. Vis. Graphics and Image Proc.* (2006)
22. Messer, K., Matas, J., Kittler, J., Lttin, J., Maitre, G.: Xm2vtsdb: The extended m2vts database. In: *In Second International Conference on Audio and Video-based Biometric Person Authentication*. pp. 72–77 (1999)
23. Ong, E.J., Bowden, R.: Robust facial feature tracking using shape-constrained multiresolution-selected linear predictors. *PAMI* (2011)
24. Ong, E.J., Lan, Y., Theobald, B., Harvey, R., Bowden, R.: Robust facial feature tracking using selected multi-resolution linear predictors. In: *computer vision, 2009 IEEE 12th international conference on*. pp. 1483–1490. IEEE (2009)
25. Patel, N.V., Sethi, I.K.: Compressed video processing for cut detection. *Proc. Vision, Image and Signal Processing* (1996)
26. Pfister, T., Charles, J., Zisserman, A.: Large-scale learning of sign language by watching tv (using co-occurrences). In: *British Machine Vision Conference (BMVC)* (2013)
27. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. pp. 397–403 (Dec 2013)
28. Sankar, P., Jawahar, C.V., Zisserman, A.: Subtitle-free movie to script alignment. In: *Proc. BMVC* (2009)
29. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: *Proc. ICPR* (2004)
30. Smeaton, A.F., Over, P., Doherty, A.R.: Video shot boundary detection: Seven years of trecvid activity. *CVIU* (2010)
31. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features (2001)

32. Williams, O., Blake, A., Cipolla, R.: A sparse probabilistic learning algorithm for real-time tracking. In: Proc. ICCV (2003)
33. Xuehan-Xiong, De la Torre, F.: Supervised descent method and its application to face alignment. In: Proc. CVPR (2013)
34. Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., Lin, F., Zhang, B.: A formal study of shot boundary detection. *Circuits and Systems for Video Technology* (2007)
35. Zhang, H., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. *Multimedia systems* (1993)
36. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. pp. 2879–2886 (June 2012)