

Automated Video Face Labelling for Films and TV Material

Omkar M. Parkhi^{id}, Esa Rahtu, Qiong Cao^{id}, and Andrew Zisserman^{id}

Abstract—The objective of this work is automatic labelling of characters in TV video and movies, given weak supervisory information provided by an aligned transcript. We make five contributions: (i) a new strategy for obtaining stronger supervisory information from aligned transcripts; (ii) an explicit model for classifying background characters, based on their face-tracks; (iii) employing new ConvNet based face features, and (iv) a novel approach for labelling all face tracks jointly using linear programming. Each of these contributions delivers a boost in performance, and we demonstrate this on standard benchmarks using tracks provided by authors of prior work. As a fifth contribution, we also investigate the generalisation and strength of the features and classifiers by applying them “in the raw” on new video material where no supervisory information is used. In particular, to provide high quality tracks on those material, we propose efficient track classifiers to remove false positive tracks by the face tracker. Overall we achieve a dramatic improvement over the state of the art on both TV series and film datasets, and almost saturate performance on some benchmarks.

Index Terms—Automatic face labelling, face tracking, deep learning

1 INTRODUCTION

OUR goal in this work is the automated identification of characters in TV series and movies. This topic has received plenty of attention in the community and papers have investigated the problem under various assumptions on the available information [2], [3], [6], [7], [8], [9], [11], [12], [19], [25], [26], [30], [33], [37]. In this work, we are particularly interested in the weakly- and un-supervised settings, where the video stream is either accompanied by subtitles and transcripts, or used without any additional information.

There is much commercial interest in this area with companies such as Amazon and Google starting to provide a limited service where certain videos can be paused and information about actors etc accessed (e.g., using Amazon X-ray and Google Play). It is limited at the moment as not every actor visible is labelled, and the available information (who is labelled) can change throughout a shot.

The original paper in this area by Everingham et al. [11] introduced three ideas that have been adopted by all others working on this problem: (i) associating faces in a shot using tracking by detection, so that a face-track is the ‘unit’ to be labelled; (ii) the use of aligned transcripts with subtitles to provide supervisory information for character labels; and (iii) visual speaker detection to strengthen the supervision

(if a person is speaking then their identity is known from the aligned transcript).

A significant extension in the use of supervisory information was introduced by Cour et al. [8] who cast the problem as one of ambiguous labelling. The key innovations were: (i) that supervisory information could be used from every shot (not just where a person is speaking); and (ii) a convex learning formulation for multi-class labelling under these partially supervised conditions. The idea of formulating the problem as one of multiple labels [9] is very appropriate for this scenario, given the ambiguous supervision available. An alternative approach to ambiguous supervision is to use Multiple Instance Learning (MIL), as employed by [2], [14], [19], [37], [39]. In particular the work of Bojanowski et al. [2] proposed an elegant convex relaxation of the problem. Further important improvements have been contributed to the form and learning of the visual descriptors [23] (e.g., by unsupervised and partially-supervised metric learning) [6], [13]; to the range of face viewpoints used (e.g., adding profile face tracks in addition to the original near-frontal face tracks) [12], [30]; and to obtaining an episode wide consistent labelling [33] (by using a graph formulation and other visual cues). Finally, Haurilet et al. [14] study the problem using a MIL framework in a setup where only subtitles are available for supervision. This is different from our setup, where both subtitles and transcripts are used (transcripts provide speaker names).

In this paper, we adopt a MIL approach, and make the following five novel contributions: *first*, Section 3, we propose a way of obtaining stronger supervisory information for the *principal* characters. For the purposes of this paper, a principal character is one who speak at least three times according to the subtitles and transcript. The change for the positive training data is quite subtle—it involves tight alignment with the subtitles rather than the shot or scene. For

- O.M. Parkhi, Q. Cao and A. Zisserman are with the Visual Geometry Group, Department of Engineering Science, University of Oxford, Oxford OX1 2JD, United Kingdom. E-mail: {omkar, qiong, az}@robots.ox.ac.uk.
- E. Rahtu is with the Department of Signal Processing, Tampere University of Technology, Tampere 33720, Finland. E-mail: esa.rahtu@tut.fi.

Manuscript received 28 Dec. 2016; revised 31 Mar. 2018; accepted 12 Apr. 2018. Date of publication 27 Dec. 2018; date of current version 4 Mar. 2020. (Corresponding author: Omkar Parkhi.)

Recommended for acceptance by E. G. Learned-Miller.

Digital Object Identifier no. 10.1109/TPAMI.2018.2889831

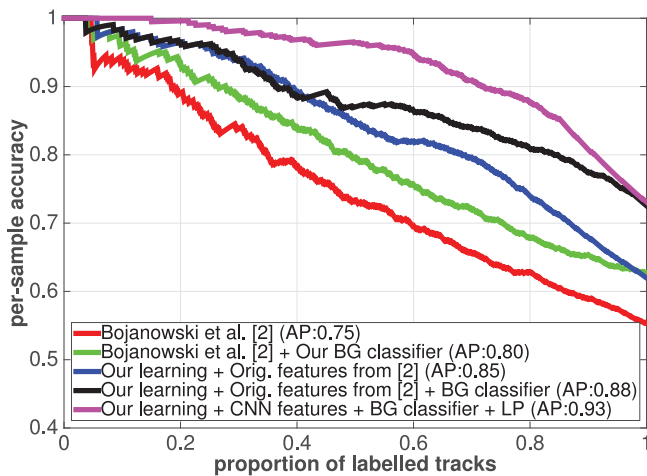


Fig. 1. Results of the automatic character naming using the “Casablanca” dataset provided by Bojanowski et al. [2]. In addition to the baseline curve of [2] the other curves are: (i) baseline [2] with our explicit background character modelling; (ii) using features of [2] with our learning framework and bag formation; (iii) adding explicit modelling of background characters, and (iv) changing the features to our ConvNet (CNN) based face descriptors and adding linear programming label selection.

negative training we are able to obtain, somewhat surprisingly, far more than has been used in previous work; *second*, Section 4, we introduce an explicit classifier for *background characters* by using their face-track information. These are the actors that have nonspeaking parts, and are usually in the background (for example, in an audience or busy street scene). These background characters are a rich source of negative training data when identifying the principal characters and positive training data for identifying the non-principal characters; *third*, we introduce state-of-the-art face-track feature vectors based on training a ConvNet architecture similar to [29]; *fourth*, Section 6, we propose a novel approach for labelling all face tracks jointly using linear programming; *fifth*, Section 9, we provide high quality tracks on new video material to investigate the generalisation and the robustness of the classifiers and features. To this end, we improve on the track classifiers of [18] by incorporating ConvNet based face/non-face classifiers and additional features to handle dark video material.

To show the improvement in performance brought by each contribution, in Section 7 we compare to previous methods using the face-tracks of the original publications [2], [11], [30], [33], so that the performance is assessed on exactly the same training and test data and face feature, and the improvements due to each contribution can be isolated. For example, the stronger supervisory information alone brings a significant improvement in performance (see Fig. 1). Where code or datasets are not available [8], [37] we implement previous methods so that the strengths and weaknesses of previous approaches can be compared.

We evaluate these contributions over *all* publicly available datasets: a number of TV sitcoms and a feature film: Buffy [30]; Big Bang Theory [33]; and Casablanca [2]. Overall, we demonstrate a substantial improvement over the previous state of the art in all cases—as can be seen for example in Fig. 1. We also investigate the generalization of the classifiers by labelling new episodes *without* the use of supervisory information from the transcript, and compare this performance to using weak and strong supervision.

Having evaluated on publicly available face-track sets, in Section 8, we also compare using our own face detector, tracker and track classifier on the film Casablanca (for backwards compatibility) and on episodes of the TV series “Sherlock”. Our tracker yields more tracks and thus improves on the *coverage* of the video material—which is an important step towards automatically labelling every occurrence of the principal characters. Also, the track classifier successfully eliminates non-face tracks. The tracker software is released at http://www.robots.ox.ac.uk/~vgg/software/face_tracker/.

The learning and inference formulation is described in Section 5, and implementation details of all the methods used are given in Section 9.

Prior Approaches to Background Characters. In previous work [2], [12], [30], [33], [37] the task of labelling background characters has been hardly discussed, though a distinction has been made between principal characters and ‘others’, with ‘others’ generally being considered as a single class. Cour et al. [9] pruned all face tracks from the data that did not contain one of the target (principal) characters. In [11], [30], [37], [33] the background characters were not pruned, but no classifier was learned for this category. In [33] a character was labelled as background class if none of the principal characters obtained high score. For [11], [30], [37], any track depicting a background character was guaranteed to be misclassified. Since the test material in these papers consisted of TV-series, where only a few background characters appear, this problem was not clearly visible in the results. In still images, [32] show approaches to find important people using image cues.

To our knowledge, [2] is the only one that attempts to explicitly classify the background characters. However, their use of the term background refers more to an ‘other’ character (i.e., not one of the principal characters) rather than the role of a background character introduced in our work. [2] uses 300 random samples from the “Labelled Faces in the Wild” dataset as positive samples of the background class, but does not make use of face-track information (e.g., size, position) as we do here. As a result, their prediction accuracy for the background class is significantly lower than for any of the principal characters and also over 20 percent less than our performance (see Fig. 5 in Section 7).

Prior Approaches to Label Selection. In most previous works the final character labels are selected by considering each face track independently and choosing the option corresponding to the largest score. In [9] the set of possible labels was further limited to the label bag mined from the transcripts.

To our knowledge [33] is only previous work that proposes an approach for optimizing character labels jointly for all face tracks. They model the labelling problem as a Markov Random Field which combines individual track scores and constraints that prevents overlapping tracks to be assigned to the same label. Unfortunately this results in an energy function that is not possible to minimize globally, but requires numerical optimisation.

On the Content. This submission is an extended version of the workshop paper [22]. The extensions include: the joint optimization over all labels using linear programming; results on new datasets (Sherlock) and previously used

films (Casablanca) using the higher yield face tracker and track classifier (the previous publication [22] only evaluated on the publically available face tracks); an analysis of the types of errors remaining; and, a full description of the face tracker and track classifier pipeline used here.

2 PROBLEM SPECIFICATION

The problem is the following: given a set of face tracks and a label set \mathcal{Y} of characters, the objective is to predict a label $\hat{y}_i \in \mathcal{Y}$ for each face track indicating that this track depicts the target character \hat{y}_i .

In this paper, we formulate the character naming problem as Multiple-Instance Learning [1], [13], [37], [39]. This means that for each character we group the face tracks into sets (referred to as bags) and label them as positive if they contain at least one sample of the target character and negative otherwise. Then we learn a classification function that forces a maximum margin between the best scoring track in the positive bags and all the tracks in the negative bags. The initial bags are formed using text-based information obtained from the subtitles and transcripts.

In more detail, every bag, \mathcal{B} , is associated with one label for each target character y . The label l^y can take one of the values $\{1, -1, 0\}$, where 1 indicates that at least one track in \mathcal{B} depicts characters y , -1 indicates that none of the tracks in \mathcal{B} belong to the character y , and 0 indicates that we have no information if the tracks in \mathcal{B} depict the character y or not. These cases are referred to as positive, negative, and ignore bags, respectively. Note, a face track can appear in multiple bags (as it can be positive for the target character, but negative for others).

An optimal positive bag would be small, but still contains at least one track of the target character. Conversely, the negative bags should contain as many tracks as possible, but optimally not a single track of the target character. In practice, it is not possible to construct perfect bags and it turns out that the quality of these bags has a significant impact on the final classification performance (as discussed in Section 7).

In this paper, we apply very different approaches for generating the bags for the principal characters and background characters. For the principal characters (Section 3), the supervisory information is obtained from the aligned subtitles and transcripts [11], which provides a speaker label $sp_k \in \mathcal{Y}$ and the corresponding time interval for each subtitle k , and the bags are based on this. For the background characters (Section 4), the supervisory information is obtained from a background character classifier.

In our work, as in most others, we construct a separate classifier for each principal character and one additional classifier for the others (secondary and background). Conventionally, the final character labels are obtained by evaluating all classifiers and selecting the highest scoring class label for each face track independently. As discussed in the introduction, the problem in such approach is that the labels are selected without considering any additional constraints. For instance, tracks that overlap in time, may claim the same class label or one classifier may claim all face tracks.

In this paper we introduce a new approach for assigning the final character labels based on linear programming. The

proposed approach (presented in Section 6) is computationally very efficient and it provides an easy way to include many additional constraints to the labelling. Our formulation has some similarities with the classical assignment problem due to similar nature of the problems. (note that in our case, many tracks can take the same label).

3 SUPERVISION FOR THE PRINCIPAL CHARACTERS

A positive bag is constructed for each subtitle k as follows: consider all tracks that overlap with the corresponding frame interval by more than v_p frames (we use $v_p = 5$). If any such tracks exist, collect them into a bag \mathcal{B} and set the label $l^y = 1$ for $y = sp_k$ and zero for all the others. Furthermore, if any of these tracks are detected to be speaking, remove all other tracks from the bag. This is done only if the speaker detection is used. Note, using the subtitle interval provides a very ‘tight’ bag, as opposed to using bags defined by multiple shots or a scene level grouping. We return to this point in section 9.3. Clearly, this bag formation may contain errors, for example in a reaction shot where the person speaking the line is not shown. However, the MIL-SVM learning framework is tolerant to such noisy supervision.

A negative bag is constructed for each face-track i by finding all the subtitles k that overlap with the extended face track interval $\{f_i^{start} - v_n, f_i^{end} + v_n\}$, where the frame indices $f_i = \{f_i^{start}, f_i^{end}\}$ refer to the first and last frame of the track. The magnitude v_n of the extension is 200 frames if the track i is not detected to be speaking, and zero otherwise. If overlapping subtitles are found, take the corresponding speaker names and label the track i as negative sample for all the other characters. For example, if the overlapping subtitles for the track t_4 in Fig. 2 indicate that characters “Laszlo” and “Renault” are speaking, use t_4 as a negative sample for all other characters but “Laszlo” and “Renault”. After going through all tracks, we pool all face tracks that are marked as negative for a particular character into a single negative bag.

In Section 9.3 we review how the supervisory information (e.g., the positive/negative bags) was generated from aligned transcripts in the prior work of [2], [9], [11], [19], [30], [37]. We also describe our implementation of these methods as used in the experimental comparisons of different bag formation strategies in Section 7.

4 LEARNING A CLASSIFIER FOR THE BACKGROUND CHARACTERS

As described in the introduction, movies and TV-series usually contain two types of characters, the principal characters and the background characters. The principal characters appear and speak often during the film, whereas, the background characters may appear just once and they often don’t speak at all. A sample of background characters are shown in Fig. 3.

The number of background characters depends very much on the material. For example, in the TV sitcom the Big Bang Theory which mainly centres on the principal characters only, there are very few scenes containing background actors. However, in other sitcoms like Friends, there are many background characters in the cafe scenes; and in a feature film like Casablanca, where there are many scenes in a bar or outside

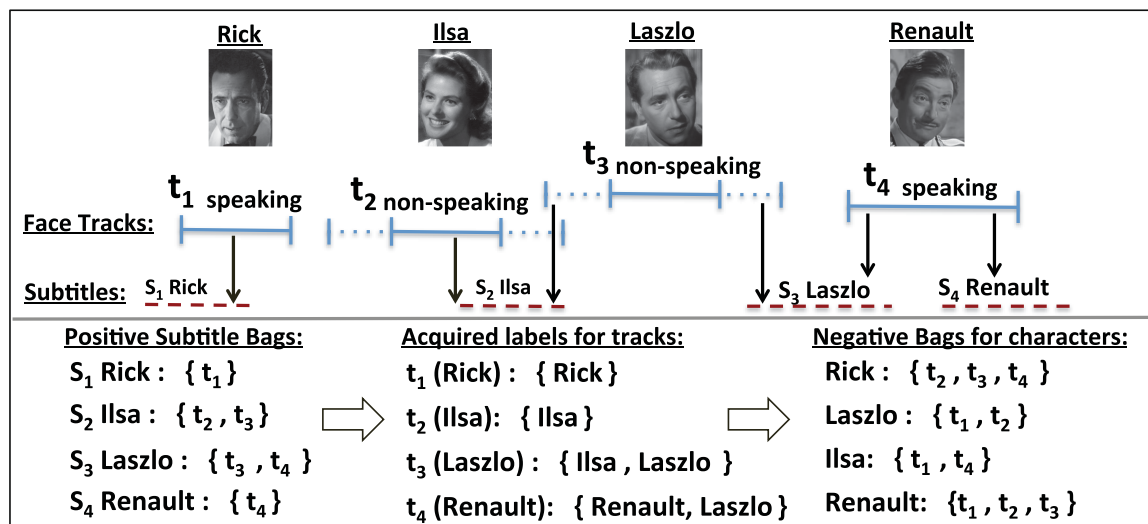


Fig. 2. Label bag formation for the principal characters: Top part: face tracks (t_i , Blue Lines) and their ground truth (Names and Pictures) and associated speaking/non-speaking label. The boundaries of non-speaking tracks are extended to provide better coverage. The aligned subtitles (S_i , Dotted Red Lines) are used to form the bags as follows: (a) each subtitle forms a bag containing all overlapping tracks (“Positive Subtitle Bags”). (b) These bags determine the possible labels for each track (“Acquired labels”). (c) Track with given possible labels forms negative example for all other track not containing those labels in their bags (“Negative bags”).

on the streets, then there can be tens to hundreds of background actors. Table 1 reports the number of principal and background characters for the datasets used in this paper. In the “Casablanca” dataset, for example, as many as 30 percent of the total tracks belong to the background category. There are websites devoted to background characters, for example spotting bizarre behaviour or bad acting.

In most works, including ours, the aim is to learn a separate classifier for each principal character and a single classifier for the remainder. The issue is that it is very difficult to obtain supervisory information for background characters as they do not appear in the transcripts (as they don’t speak) and this can lead to labelling confusions with the principal characters. To this end, we propose next a novel classification scheme to automatically classify tracks as background or not, using features formed from the face-tracks, and generate bags for them in a manner that takes proper account of the available information.

4.1 Background/Non-Background Track Classifier

The classifier is based on simple features obtained from track level statistics together with a linear SVM classifier. It is inspired by the false positive track classifier of [18]. We

observe that background characters appear at particular locations, often away from the central action in a frame. This makes the location of the detections of a track, their sizes and their motion in a shot, key elements in making a decision. More specifically, a feature vector consists of: the track length; mean and standard deviation of three properties (i) detection sizes, (ii) location of the face detection center, and (iii) facial landmark detection scores; giving a seven dimensional feature vector per track. A linear SVM classifier is trained using ground truth obtained from the television series “Scrubs” (the dataset is described in Section 8.1). This learnt classifier is then used to rank the tracks in a given video. This simple scheme achieves an Average Precision (AP) of 75 percent (Fig. 4).

4.2 Bag Construction

The background-classifier is applied to all face tracks, and tracks are then ranked according to their classification score. This results in an ordering $\{t_1, t_2, t_3, \dots, t_n\}$, where the face track t_1 has the highest likelihood to depict a background character. At this point, there are several possible options for forming the bags. One option is to put each track into its own bag. This approach yields the maximum number of



Fig. 3. Examples of background characters: Each row shows three frames of a track. One can observe that background characters have typical characteristics: the scale of the face, the motion through the shot, and their location. These characteristics assist in identifying them.

TABLE 1
Dataset Statistics: The Total Number of Tracks in Each Video (The Number of Background Character Tracks Shown in Brackets)

Dataset	Episodes					
	E1	E2	E3	E4	E5	E6
BBT	622 (8)	565 (2)	613 (87)	581 (41)	558 (82)	820 (195)
Buffy	592 (2)	756 (3)	822 (9)	652 (32)	604 (31)	
Scrubs	608 (104)	518 (101)	449 (68)	430 (55)	454 (49)	
Casablanca			1278 (404)			
Hannah			2002 (736)			

Note that Casablanca and Hannah have a high proportion of background characters.

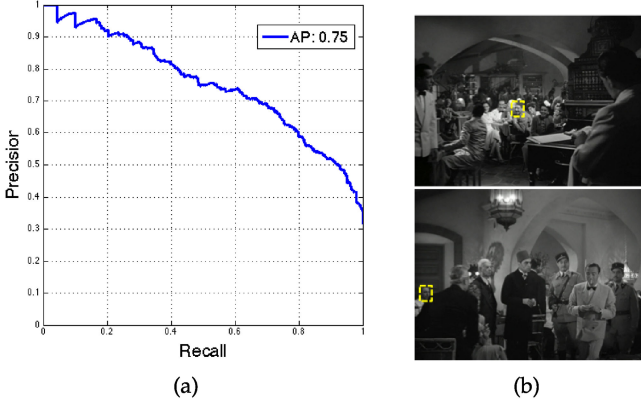


Fig. 4. *Background classifier performance*: (a) The precision-recall curve obtained by training on the Scrubs dataset and testing on the Casablanca dataset, and (b) Examples of high scoring detections from the test set.

bags, but since the background-classifier could make a mistake, some of the bags may not contain a positive sample; something which is undesirable for multiple instance learning. On the other hand, one giant bag can be formed out of all the tracks which avoids the problem of occasional classifier mistakes, but yields only one positive sample. There are multiple possibilities between these two extremes. We select one plausible solution amongst them and construct positive bags for the background characters as follows.

A bag is formed from the face tracks t_1 and t_2 , and labelled as positive for the background character (and ignore for the principal characters). Next, a bag is formed from tracks t_2 and t_3 , and labelled in the same way. We continue in this manner until the background-classification score falls below zero. Including two tracks increases the probability that any positive bag actually contains at least one track depicting a background character. In fact, we could take even more than two consecutive tracks, but two was found to be a good compromise between bag size, bag number and accuracy.

The scores given by the background classifier are also used to construct a negative bag for the background characters. In this bag, we take the N_n lowest scoring tracks (i.e., those corresponding to the principal characters), where N_n is set to be 40 percent of the total number of face tracks. Here we assume that at least 40 percent of the tracks in a video depict the principal characters. On the other hand, when there are more than 40 percent of principal characters, including 40 percent in the negative bag still gives a reasonable number of negative samples. Although the precision of our classifier is good (see below), there may be a few background characters included, but our MIL framework does not require all bags to be extremely pure.

5 LEARNING FORMULATION AND OPTIMISATION

The definition of our learning objective roughly follows the standard MIL-SVM formulation [1], where the idea is to force a maximum margin between the best scoring samples in positive bags and all the samples in the negative bags. The difference to [1] is that we use one-versus-all multi-class setup and have separate slack variable for each negative face track sample. This is equivalent to using a separate bag for every negative sample in the framework of [1].

We learn a linear classifier $\{w_y, b_y\}$ for each character y by minimising the following objective:

$$\min_{w_y, b_y, \xi} \frac{1}{2} \|w_y\|^2 + C \left(\sum_I \xi_I + \sum_{I,i} \xi_{Ii} \right) \quad (1)$$

$$\begin{aligned} \text{s.t.} \quad & \langle w_y, x_{s(I)} \rangle + b_y \geq 1 - \xi_I \quad \forall I : l_I^y = 1, \\ & -\langle w_y, x_i \rangle - b_y \geq 1 - \xi_{Ii} \quad \forall I, i : i \in \mathcal{B}_I, \text{ and } l_I^y = -1, \\ & \xi_I \geq 0, \text{ and } \xi_{Ii} \geq 0 \quad \forall I, i, \end{aligned} \quad (2)$$

where bags are indexed by I and tracks by i , $\langle \cdot \rangle$ denotes the inner product, C is a constant (set to $C = 0.6$ in all experiments), and

$$s(I) = \operatorname{argmax}_{i \in \mathcal{B}_I} \langle w_y, x_i \rangle + b_y, \quad (3)$$

is a selector variable denoting the highest scoring sample in the bag \mathcal{B}_I . The slack variables ξ are introduced to explicitly model the noise in the bag labels. That is to say that the classifier is allowed to misclassify training samples with a certain additional cost. This is important if e.g., a bag \mathcal{B}_I with label $l_I^y = 1$ actually does not contain any track of the character y .

The minimisation of (1) is a non-convex problem and we solve it by alternating between the estimation of the selector variables (3) with fixed $\{w_y, b_y\}$, and the minimization of (1) with respect $\{w_y, b_y, \xi\}$ with fixed selectors $s(I)$. Once the selector variables are fixed, the optimization of (1) turns into a convex problem and it can be solved using standard techniques developed for SVMs.

The alternating optimization needs to be initialized by providing either the initial classifiers $\{w_y, b_y\}$ or the indices for the best scoring track in each positive bag. Such initialization can be obtained using e.g., [30], [9], or [2]. In the experiments we show that the proposed approach is not sensitive to the initialization.

Once the classifiers have been inferred, we compute the classification scores $s_i^y = w_y^T x_i + b_y$ for all face tracks i . The obtained values are further normalized by applying a sigmoid function $p_i^y = (1 + \exp(A_y s_i^y + B_y))^{-1}$ to map the original scores to corresponding posterior probabilities p_i^y . The parameters A_y and B_y are learned using 25 percent of the most confident positive and negative samples in the class y . If there are classes with less than 10 positive bags, we estimate shared normalization parameters for all of them.

6 TRACK LABELLING USING LINEAR PROGRAMMING

Once the classification scores are obtained, we make the final decision of the class labels. Conventionally, the labels are obtained by selecting the class with the maximum score as

$$\hat{y}_i = \operatorname{argmax}_y p_i^y, \quad (4)$$

where p_i^y denotes the normalised score of a character label y for a track i . Instead of labelling every track independently, we wish to label all tracks *jointly* and formulate this as an optimization problem

$$\max_x \sum_i \sum_y x_{i,y} p_i^y, \quad (5)$$

where $x_{i,y} \in \{0, 1\}$ is an assignment variable which allocates the label y to track i .

A solution to this optimization problem can be obtained as a linear program (LP) by relaxing the constraint that $x_{i,y}$ is a discrete variable, and instead letting $x_{i,y} \in \mathbf{R}$ subject to the constraints

$$\forall i \forall y \quad x_{i,y} \geq 0 \forall i \quad \sum_y x_{i,y} = 1. \quad (6)$$

The formulation allows fractional values for $x_{i,y}$, but if the resulting constraint matrix is totally unimodular, as it is here, the optimal solution is guaranteed to have only integer values (in this case $x \in \{0, 1\}$) [38].

However, this LP formulation would lead to the same solution as (4), but by adding more constraints we can include further information and obtain a different, and hopefully improved, solution. We illustrate this by three examples.

A particularly important case would be to prevent face tracks that overlap in time from claiming the same label. This is obtained by adding constraints

$$\forall k \forall y \setminus y^{bg} \quad \sum_{i \in T_k^y} x_{i,y} \leq 1, \quad (7)$$

where T_k^y is a k th subset of face tracks that appear at least once in the same frame and y^{bg} the label assigned to the background characters (note that two background characters can appear simultaneously).

Similarly, one can add constraints that force a particular character label to claim at least a certain number of tracks in a particular track set. For instance, the transcript may indicate that a particular character is present in a certain scene and we like to ensure that at least one track in this scene is labelled with the corresponding class. These constraints have the form

$$\forall k \quad \sum_{i \in T_k^y} x_{i,y} \geq n_k, \quad (8)$$

where T_k^y is a k th subset of face tracks that must contain the label y at least n_k times.

Finally, it is possible to limit the set of allowed labels for a given track (e.g., forbid some labels or restrict to those characters that are mentioned at the corresponding part of the transcript). Such limitation is obtained using constraints

$$\forall i \quad \sum_{y \in L_i} x_{i,y} = 1, \quad (9)$$

where L_i is a set of allowed character labels for a track i . Note that this constraint is equal to (6) for all tracks for which L_i includes all labels.

The additional constraints (7), (8), and (9) might not produce a totally unimodular constraint matrix and thus the optimal solution is not guaranteed to have integer values but $x \in [0, 1]$. In such cases the corresponding final labels can be selected according to largest indicator values

$$\hat{y}_i = \operatorname{argmax}_y x_{i,y}. \quad (10)$$

However, a non-integer solution was never obtained in our experiments. In the experimental section we apply the first of these constraints, and discuss the benefit it brings.

7 EXPERIMENTS USING PUBLIC BENCHMARK DATASETS

In this section we present results obtained entirely on publicly available benchmark datasets. We will compare different parts of our solution to the current state-of-the-art approaches and analyse the differences.

7.1 Public Benchmark Datasets

We evaluate on two television series, “Buffy” and “Big Bang Theory”, and one feature film “Casablanca”. For all of these datasets, we make use of face tracks and ground truth annotations provided by the authors of the original publications.

Buffy. This dataset based on the popular sitcom “Buffy–The Vampire Slayer” was first introduced in [11]. This was later extended in [30] to include face tracks using both frontal and profile face tracks. We used the first 5 episodes from Season 5 of the series. It has a good number of primary characters thus providing reliable training data. Also, due to the nature of the show, this dataset has a large variation in lighting conditions. We use face tracks obtained from the author’s webpage [30] (with both frontal and profile detections) for fair comparison. Speaker identification output is also provided with the dataset. Supervisory information was obtained by aligning subtitles with transcripts available from a fan website.

Big Bang Theory. This dataset consists of episodes of the popular American Sitcom “Big Bang Theory” (BBT). It was first used for person labelling in [33]. It consists of 6 episodes from Season 1 of the series. This is a more recent series than Buffy or Scrubs and, consequently, the picture quality of this series is much better as compared to the other series. However, this series has only a very limited number of principal characters and they appear in very restrictive environment. Even the clothing styles of the characters are very distinctive making it one of the easiest datasets to work on. We use the face tracks obtained from the website of the authors [33] and perform textual processing ourselves.

Casablanca. This movie was used in [2] for joint face and action labelling. It is a black and white film, a feature which is different from the other datasets used here. We use data entirely obtained from the author’s website [2] for evaluation.

Hannah and her Sisters. This movie was used in [21] for face track evaluation. We use the data provided at the website [21] for the same purpose.

In the following section, we first compare different bag formation strategies over episodes of Buffy, and then assess the contribution of different algorithmic components on the Casablanca dataset. We use Casablanca for this assessment rather than Buffy as the publicly available Buffy tracks only have a few background characters. We then compare to the state of the art for all three publicly available datasets (Buffy, BBT and Casablanca).

7.2 Results

Effect of Bag Formation. The bag formation strategy has a significant effect on the resulting classification performance. In Section 3 we described our approach, and in Section 9.3

TABLE 2

Comparison of Different Bag Formation Strategies: The Average Precision Values Obtained using Different Bag Formation Strategies, on the Same Set of Tracks and with the Same Features in Each Case

Method	Feat	Spk	Buffy						
			1	2	3	4	5	Mean	Median
[30]	FV	✓	0.85	0.63	0.65	0.77	0.78	0.74	0.77
[37]	FV	✓	0.79	0.53	0.52	0.68	0.78	0.66	0.68
[2]	FV	-	0.88	0.71	0.65	0.82	0.81	0.77	0.81
[9]	FV	✓	0.83	0.61	0.75	0.79	0.78	0.75	0.78
Ours	FV	✓	0.94	0.86	0.87	0.93	0.93	0.91	0.93
Ours	FV	-	0.94	0.85	0.82	0.91	0.89	0.88	0.89
[30]	CNN	✓	0.92	0.74	0.90	0.94	0.91	0.88	0.91
[37]	CNN	✓	0.90	0.65	0.73	0.81	0.87	0.79	0.81
[2]	CNN	-	0.98	0.88	0.86	0.93	0.96	0.92	0.93
[9]	CNN	✓	0.94	0.76	0.87	0.91	0.90	0.88	0.90
Ours	CNN	✓	0.98	0.96	0.95	0.95	0.97	0.96	0.96
Ours	CNN	-	0.98	0.90	0.91	0.94	0.95	0.94	0.94

The Feat and Spk columns indicates the face descriptor type (Fisher Vector or CNN) and if speaker detection is used. These results are computed using our implementations of the corresponding methods.

compare this to several previously proposed methods. In this experiment, we assess all these techniques using the same tracks and features from the Buffy dataset. The corresponding results, in Table 2, clearly indicate that our bag formation strategy obtains the best overall performance. It is also notable that our results did not change much even if the speaker detection was not applied. For curiosity, Table 2 contains also results obtained with Fisher Vector features [23]. In this case the differences between the approaches are even more notable than with CNN features. Note that the results in Table 2 are not comparable with those in [37] since they used older and smaller dataset from [11] that contains only frontal face detections.

To enable further analysis, we calculated the bag characteristics for the first episode. In Table 3, we report the total number of positive bags, their average size, and the proportion that actually contains the corresponding character.

The approach of Sivic et al. [30] (row 1) generates the bags using the speaking character with unique speaker

TABLE 3

Evaluation of Bag Properties with Different Formation Strategies: For the Positive Bags, We Show the Total Number of Bags Generated over All Characters, the Average Size of an Individual Bag, and the Proportion of Bags That Actually Contain the Indicated Character

Method	Spk	Positive Bags			Negative Bags	
		Num	Size	Corr.	Num smpl	Corr.
[30]	✓	58	1.0	87.9%	522	98.7%
[37]	✓	90	53.6	92.2%	1928	94.5%
[2]	-	673	5.1	90.6%	-	-
Ours	-	581	2.1	86.9%	4157	97.4%
Ours	✓	581	1.4	79.0%	4426	97.3%

For negative bags, we report the total number of tracks in all negative bags (in this case the number of bags is irrelevant) and the proportion of those tracks that do not depict the corresponding target character. These results correspond to the first episode of the Buffy dataset. Spk column indicates if speaker detection is used.

TABLE 4

Contributions of the Different Components of the Algorithm on Casablanca Dataset: The Average Precision Results for the Casablanca Dataset Using Different Versions of Our Method and the Baseline [2] with and without Our Background Character Modelling

BG-classifier	Learning	Bags	Features	LP	AP
-	A	A	A	-	0.75
✓	A	A	A	-	0.80
-	O	O	A	-	0.85
✓	O	O	A	-	0.88
✓	O	O	O	-	0.93
✓	O	O	O	✓	0.93

The columns correspond to background character classifier (BG-classifier), learning algorithm, bags, features, and linear programming, indicating the source of these components. A denotes that these were used as available from authors of [2] while O indicates the use of our described method for obtaining them.

label. This strategy results in bags that contain exactly one sample, but it obtains a very limited number of them. Wohlhart et al. [37] (row 2) enhanced the previous method by adding scene bags. This increases the number of samples and their accuracy, but at the cost of substantially enlarging the average bag size (and the AP decreases). However, their strategy clearly improved the negative bags by almost tripling the number of tracks without compromising their accuracy. Bojanowski et al. [2] (row 3) generate the bags by considering a set of shots around each subtitle. This strategy demonstrates a clear improvement in the number of positive bags, without exploding the average bag size or losing their accuracy. This approach was only intended for generating positive bags.

Our approach constructs the bags using tracks around each subtitle time frame. In this way we obtain slightly fewer bags than [2], but the average bag size drops from 5.1 to 2.1, and further to 1.4 face tracks if the speaker detection is applied. This is a substantial improvement in reducing the ambiguity of the supervision. Moreover, our approach results in a considerable increase in the number of negative samples with almost no loss in their accuracy.

Effect of Initialisation on Learning Algorithm. As discussed in Section 5, the alternating minimization of our learning algorithm requires an initialization. This can be provided in the form of initial classifiers $\{w_y, b_y\}$ or the best scoring track indices $s(I)$. Such initialization can be obtained using the method of [30], [9], [2], or by randomly picking the selector variable $s(I)$ for each positive bag.

In order to demonstrate the robustness to the initialization, we evaluated our algorithm using the first five episodes, one at a time, of the Buffy dataset with all four different initialization methods mentioned above. This resulted in mean average precision 0.9 for initializing using [2], and 0.89 for all other initializations. It is notable that also the random initialization resulted in equal performance compared to the other methods. This is probably partially due to the tight bags obtained using our approach.

Effect of Algorithm Components. In Sections 3, 4, and 5, we describe different components of our algorithm. Fig. 1 and Table 4 show the contributions of each of these components. We compare our results with the strong baseline of [2] on the “Casablanca” dataset.

TABLE 5
Comparison with State of the Art (TV Series): The Average Precision (AP) Values for Each of the Tested Episode in the Buffy and Big Bang Theory Datasets

Data	Episode	[30]	[33]	Our	Our LP	Raw	Raw
						Our	GT
Buffy	1	0.90	-	0.98	0.99	0.92	0.98
	2	0.83	-	0.96	0.96	0.95	0.96
	3	0.70	-	0.95	0.95	0.91	0.98
	4	0.86	-	0.95	0.96	0.92	0.97
	5	0.85	-	0.97	0.97	0.93	0.96
	Avg	0.83	-	0.96	0.96	0.93	0.97
	Med	0.85	-	0.96	0.96	0.92	0.97
BBT	1	-	0.98	0.97	0.98	1.00	1.00
	2	-	0.99	0.98	0.99	0.99	0.99
	3	-	0.94	0.95	0.95	0.96	1.00
	4	-	0.98	0.96	0.97	0.92	0.98
	5	-	0.94	0.96	0.97	0.91	0.99
	6	-	0.98	0.88	0.91	0.87	0.97
	Avg	-	0.97	0.95	0.96	0.94	0.99
	Med	-	0.98	0.96	0.97	0.94	0.99

LP indicate if we are using a proposed linear programming approach in track labelling. In addition, the two right most columns show the average precision values obtained in the “raw” experiment. For these, each row corresponds to the case where the indicated episode is used as a test data and all others for training. The baseline results are obtained from the corresponding original paper. Note, the results of [33] are not directly comparable to ours, since in [33] the classifier is trained using manually labelled examples which overlap with the test data.

By changing the learning algorithm and bag formation, we demonstrate a considerable boost. This is largely due to the use of negative bags. Furthermore, by introducing the new approach for handling the background characters, we improve our results by an additional 3 percent and finally, by changing the track features, we obtain further 5 percent improvement, taking the result up to 0.93 (compared to 0.75 in [2]).

The proposed linear programming approach for track labelling using the additional constraint (7), has only a very small effect on the Casablanca dataset. However, one can see a greater difference in Table 5, which shows results for the two TV-series. The lack of a more significant performance gain is due to fact that these face track datasets contain relatively few cases where two similar looking tracks (that claim the same label) overlap in time.

7.3 Comparison with the State of the Art

In Table 5, we compare our approach to the previously known state of the art methods on two different TV datasets. Our method using linear SVMs clearly outperforms the

TABLE 6
Comparison with State of the Art (Movies): The Average Precision Values for Different Methods on the Casablanca Dataset

Sr. No.	Dataset	Method	AP
1	Casablanca	Sivic et al. [30] results by [2]	0.63
2		Cour et al. [9] results by [2]	0.63
3		Bojanowski et al. [2]	0.75
4		Our Method	0.93

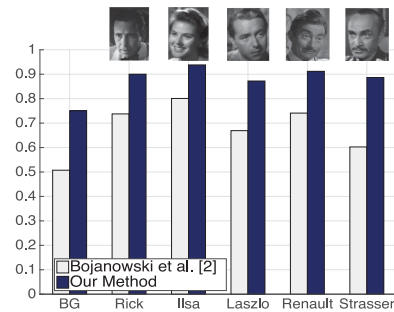


Fig. 5. Per character AP results on Casablanca using our best method and [2]. Characters are ordered according to their frequency of occurrence. Note the significant improvement for all categories, including “BG” (background).

complex 162 Non-linear MKL method of [30] on the Buffy dataset. While on the Big Bang Theory dataset we obtain 0.96 mean AP value.

In Table 6, we compare our method on the “Casablanca” dataset. Here we make use of our best working setting from the components described earlier and compare against results published in [2]. Our method also significantly outperforms all other methods on this dataset. Fig. 5 illustrates the results per character for our method and the best performing baseline [2].

7.4 In the Raw Experiments

On TV data, we introduce a new measure of evaluating performance of the algorithms. Here we train our method using all other episodes of a series and test it using one episode disjoint from the training set. This is different from the usual practice of training and testing on the same episode. We evaluate in this way to assess the strength of the learned classifiers and to check if they are overfitting to the training data. As can be seen from the second last column of Table 5, our method indeed generalises well, sometimes exceeding the performance achieved by training and testing on an episode—presumably because more training data is available.

For comparison, we also performed the same experiment using the ground truth character labels in the training. The results are shown in the last two column of Table 5, and it is evident that for some episodes the raw results match the ground truth training, which is quite impressive. For other episodes there is room for improvement which requires further investigation.

7.5 What Is Missed?

Given the excellent AP results, we investigate what is still being missed. Fig. 6 illustrates a few examples of different error types. It turned out that most of the misclassified face tracks are due to false positive face detections included in the dataset (first row in Fig. 6). A few times the face detections did not include the actual speaker which could result in a faulty label bag (second row in Fig. 6). Although the model tolerates noisy labels, some of these resulted in misclassifications. Finally, some particularly difficult face tracks were not classified correctly (third row in Fig. 6).

8 EXPERIMENTS USING NEW DATASETS

The results of the previous section were on public dataset provided by the authors of the original publications. In this

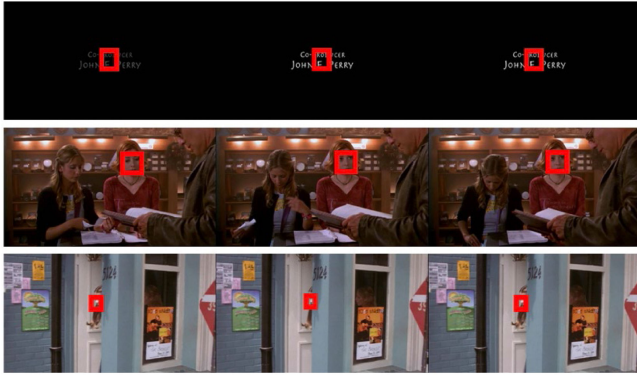


Fig. 6. Examples of the remaining errors in the Buffy dataset. The first row illustrates a false positive face detection included in the dataset. These were the most common cause for the errors. The second row shows a case, where the actual speaker (Giles) was not included in the face detections resulting in a faulty label bag. The third row, illustrates an example of a particularly difficult face track, which was misclassified by our approach.



Fig. 7. Examples of the highest scoring misclassified faces in the Sherlock episode 1. Each row shows one face track and in all cases the correct labelling is background character. The first row illustrates a difficult face track acquired in dark conditions. The second and third illustrate cases, where our background character does not work well since tracks appear similar to the main character tracks. In all cases, the classification certainty is really low.

section we use instead our own face tracker and track classifier in order to investigate the improvements brought by using higher quality tracks. We quantify the quality by measuring the *coverage* of the public tracks compared to our own on the same material—the film *Casablanca*. We also introduce a new evaluation dataset using episodes from “*Sherlock*”, and use episodes from “*Scrubs*” (i.e., a disjoint dataset) for parameter learning.

8.1 New Datasets

We introduce three new datasets. The face-tracks for these are obtained using the face tracker and track classifier described in Section 9.1, see Table 7 (row 2) for the dataset statistic. Shot boundary detection and subtitle processing is done using methods described in [31]. Alignment of subtitles to transcripts is obtained using the method of [12].

Sherlock. This dataset consists of 3 episodes from series 1 of the popular British-American crime drama “*Sherlock*”. It has a fair number of principal characters, and has a large variation in background and lighting.

Casablanca. This dataset contains 1921 face tracks. During tracking, the face detection is done using frames scaled up

TABLE 7

Dataset Statistics for Sherlock and Casablanca: “GT” Means Face Tracks Provided by the Tracker and Manual Removal of Non-Face Tracks; “Auto” Means Face Tracks Provided by the Tracker and Track Classifier

Dataset	Sherlock			Casablanca
	E1	E2	E3	
GT	1,824	1,762	1,752	1,927
Auto	1,766	1,783	1,755	1,921

by three times and thus many small faces are successfully detected and tracked. This dataset also contains a wide range of poses and cluttered scenes. Both factors indicate the challenge of the material.

Scrubs. This dataset formed of 6 episodes from Season 1 of the series “*Scrubs*” is introduced by us and used solely for parameter tuning. Its characteristics are very similar to that of the Buffy dataset. One of the key differences from Buffy is the number of background characters appearing in this series. Since the storyline is focussed on the life of employees of a hospital, there are many patients and staff appearing in the background. This provides us with the necessary data for tracking background characters and learning their track classifier. We use these episodes for parameter tuning and training of the background classifier.

8.2 Track Coverage

In this section, we evaluate the quality of the face tracks by considering their coverage. In particular, we measure the coverage by looking at how many people appeared in the video are covered by the face tracks. It is defined as the percentage of the number of persons that are tracked, compared to that of the actual persons appeared over time. Here, we use the “GT” face tracks which are produced by the tracker with further manual removal of the non-face tracks, see Table 7 (row 1) for the dataset statistic.

Table 8 reports the coverage rates on the *Sherlock* and *Casablanca* datasets. From Table 8 we can see our new *Casablanca* dataset achieves higher coverage rate than the public one, which shows the quality of the proposed dataset. Indeed, our proposed datasets are challenging since they contain very small faces, variations in poses and difficult lighting conditions.

8.3 Track Labelling Performance

Face Track Labelling. Fig. 8 illustrates the precision-recall curves for the new *Sherlock* and *Casablanca* datasets. For *Sherlock* tracks, we obtain perfect labelling up to approximately 70 percent recall for both Auto and GT tracks. Furthermore, the corresponding mean average precision over the episodes is 0.97 for both cases.

TABLE 8

Coverage Rates for the New Sherlock and Casablanca Datasets

Dataset	Sherlock			Casablanca	
	E1	E2	E3	Public	New
Coverage	0.82	0.79	0.69	0.69	0.81

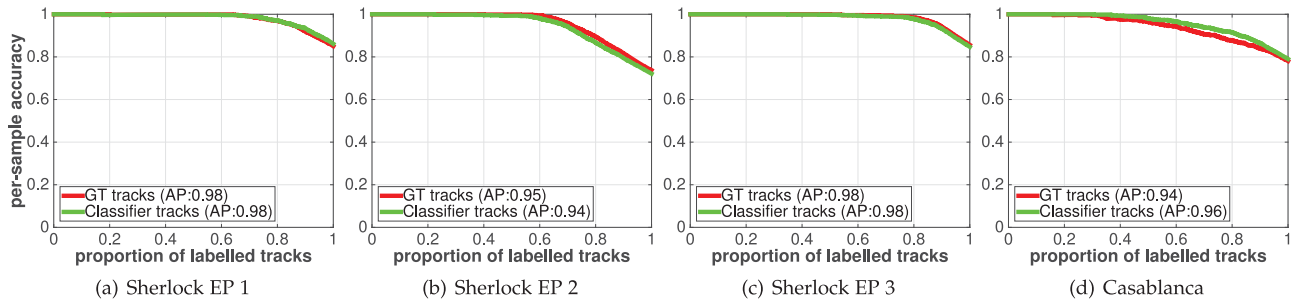


Fig. 8. Precision-recall curves for three Sherlock episodes and Casablanca. The average precision values are indicated in the corresponding legend.

Table 9 contains the average precision values for the “RAW” experiment where we use similar setup as in Section 7.4. The results indicate that the obtained classifiers generalize relatively well to new unseen episodes. In most cases the average precision values are only 2 percent less than those obtained using the transcripts of the corresponding episode (see Fig. 8).

8.4 What Is Missed?

The remaining errors are mainly due to rare background characters and dark unclear face images. Hardly any main character tracks are misclassified. Fig. 7 illustrates a few examples of the highest scoring classification errors. In all error cases, the classification scores remain really low, which makes possible to apply refuse to predict approach to avoid misclassification in these cases.

9 IMPLEMENTATION DETAILS

We give implementation details on: (i) the face detector and tracker pipeline; (ii) the face track descriptors; and (iii) the supervisory information for prior methods.

9.1 Face Tracking and Face Track Classification

In this section, we first briefly describe the face detection and tracking approach. Then we describe in more detail the face track classifier that is used to remove non-face tracks.

Face Tracking. To track faces in videos, we apply the tracking-by-detection approach [12], [18] which has proved successful in uncontrolled videos. The face tracker consists of three steps: (i) shot boundary detection; (ii) face detection in every frame; (iii) face tracking and its post-processing in each shot. Faces are detected using a local version of the cascaded Deformable Part Model (DPM) [20] and then the Kanade-Lucas-Tomasi (KLT) tracker [27] is used to group the detections through consecutive frames.

Computational Cost. Face detection is the most expensive part of the entire method (including tracking, feature computation, track labelling, etc). This is because it has to be run on every frame, and the frames are large because they are scaled up in

size to allow small faces to be detected. Sherlock frames are doubled in size (four times the area) and Casablanca frames are scaled by a factor of three. The cost of running the DPM detector on a double size Sherlock frame (1152×2048 pixels) is about 8s per frame. In contrast, the cost of tracking is only about 3s per frame. Both timings are for a single core CPU.

Face Track Classification. Since the face detector is run on every frame of the video, background clutter can generate many false positives. Some of these survive to produce non-face tracks and these should be removed. Motivated by [18], we propose a simple and efficient classifier to distinguish the face tracks from non-face tracks using measures of the track statistics and face/non-face confidence.

In more detail, the classifier is a linear SVM, and the feature vector is 7-dimensional and consists of: the track length, and the mean and standard deviation over the track of (i) the face centre coordinates (normalized by the image size); (ii) grey level intensities for the face region; and (iii) the confidence scores of a face/non-face classifier. Pixel intensities are included since many true face tracks are in dark conditions. A face/non-face classifier is required because the SVM score from the DPM detector is not a reliable indicator of whether there is a face or not [16].

The face/non-face classifier is a ResNet-50 [15] CNN architecture pre-trained on the VGG Face Dataset [24], and fine-tuned for the face/non-face task. Two large datasets are employed for training the classifiers: the face/non-face classifier is trained on a class balanced dataset of 246,590 face and non-face image regions; and the face track classifier, is trained on a class balanced set of 5406 face/non-face tracks. Two track classifiers are trained, one for RGB videos and the other for B&W videos (by converting the RGB training material to B&W).

At test time, face tracks are classified and rejected by thresholding on the classifier scores. The threshold value is chosen as 96 percent recall on a validation set. Precision recall curves on the Sherlock and Casablanca test data are shown in Fig. 9. Clearly the performance is good.

Benchmark Tracker Evaluation. The tracker pipeline (face detection, tracking, and track classifier) is evaluated on the Hannah benchmark dataset of [21]. It achieves an object purity of 40.1, a tracker purity of 57.9, and an overall purity of 48.0 (the harmonic mean of object and tracker purity). This comfortably exceeds the best published figures of 22.5 (object purity), 50.6 (tracker purity), and 31.2 (overall purity) in [21].

9.2 Face Track Descriptors

We implement and compare two face-track descriptors, one based on Fisher Vectors (FV), the other based on ConvNets.

TABLE 9
Average Precision Values for the “RAW” Experiment
Using the New Sherlock Face Tracks

Dataset	Episodes			Mean	Median
	E1	E2	E3		
Sherlock	0.96	0.90	0.94	0.93	0.94

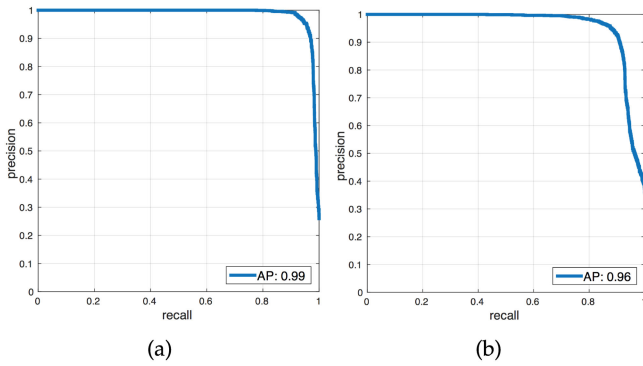


Fig. 9. Performance of the face track classifiers: The precision recall curve for (a) testing on the Sherlock dataset (RGB track classifier); (b) testing on the Casablanca dataset (B&W track classifier).

Fisher Vectors. This representation follows the approach of [23]. Dimensionality reduced SIFTs (64 dim.) are computed on a 2 pixel spaced grid on every face detection of the track, and aggregated into a single Fisher Vector for that track. With 512 GMM clusters, this generates a 67,584 dimensional descriptor. Discriminative dimensionality reduction (using the implementation of [28]) is then employed to reduce its dimensions to 1024.

ConvNet. This representation follows the approach of [24]. We use publicly available 19 layer model trained using MatConvNet [36]. A multi-scale feature vector is obtained for each face in the track as follows: the face detection is rescaled so that the lower of the height or width of the face matches three different sizes (256, 384 and 512 pixels). A 4096 dimensional descriptor (from the penultimate convolutional layer of the ConvNet) is then computed for each of 10 crops (corners, center with horizontal flips) at each scale of the face and sum-pooled. To obtain the track-level descriptor, face feature vectors are computed on 10 equally-spaced frames from the track, sum-pooled and L_2 normalised. The final descriptor is therefore 4096 dimensional.

9.3 Supervisory Information for Prior Methods

We describe here, briefly, how the supervisory information (e.g., the positive/negative bags) is generated from aligned transcripts in prior work [2], [9], [11], [19], [30], [37], and our implementation of it. We do this so that we can compare methods on the same data (tracks and features, and train/test splits) and with the same classifier. In the past methods have often been evaluated on different datasets with different features and different classifiers, so that it has not been possible to know which was really superior.

Method (A). Everingham et al. [11] and [30]. here a standard multi-class SVM formulation is used, which requires training samples with a unique class label. Tracks with a unique label are obtained by only selecting those speaking tracks where exactly one speaker name appears in the temporally overlapping aligned transcripts and subtitles. Such tracks generated a training sample which is labelled as positive for the speaking character and negative for all the others. Note, although, this results in strongly supervised training material, it utilizes only a small part of the available information as non-speaking tracks are completely ignored in the supervisory information.

Method (B). Wohlhart et al. [37] and [19]. this uses MIL approach for learning. Positive and negative bags are constructed as in Method (A) above, and then further negative information is added as follows: (i) make a bag from every non-speaking track and label it as negative for all characters appearing in temporally overlapping subtitles, (ii) make a bag from every track that overlaps temporally with positive bags and label it as negative to corresponding characters. Furthermore, a bag is made out of all face tracks that appear in one scene (a *scene bag*). The scene bag is labelled as positive for all characters that speak at least once during the scene according to subtitles and negative for all the others.

This method obtains more training data than (A), but the approach becomes heavily dependent on the speaker detection since every misclassified speaking track generates a wrong negative sample (e.g., t_2 in Fig. 2). Also, the scene bags are often large and do not provide much additional supervisory information. In contrast, our approach is highly tolerant to speaker detection errors (and indeed works even without speaker detection) and we utilise also non-speaking tracks without overlapping subtitles (e.g., t_3 in Fig. 2).

Method (C). Bojanowski et al. [2]. positive bags are formed by (i) assigning all speakers names from the aligned transcript to shots, and (ii) making one bag for each speaker name in each shot by collecting all face tracks corresponding to that shot and its neighbouring shots (a *shot bag*). This bag formation strategy succeeds in gathering many more positive training samples than (A) and (B), but there are no negative bags.

Method (D). Cour et al. [9]. this methods formulates the character naming task as an ambiguous label learning problem where the training material consists of face tracks associated with multiple character labels (i.e., a *trackbag*, that contains a single track and one of the multiple labels is correct, rather than shot or scene bags with multiple tracks but only one label). These labels are obtained by assigning all speaker names that appear in a scene to all tracks in the same scene. Only scenes containing up to three speaking characters are included. Additionally, if the face track is detected to be speaking then the labels are restricted to those obtained from the overlapping subtitles. There is an assumption that all characters appearing in the video are named in the transcript. In our implementation we form “bags” for each face track using temporally overlapping subtitles (if any exist) and assign all corresponding character names to this track. In addition, we make similar scene bags to Method (B), but use only those that contain up to three characters.

10 DISCUSSION AND EXTENSIONS

We have shown that stronger supervisory information can be obtained from an aligned transcript than that of previous methods. Moreover, from our implementation and comparison of bag properties over previous methods, we can conclude that a strategy of having tighter positive bags (i.e., fewer labels associated with the the bag) together with a greater quantity of negative data leads to a significant improvement in classification performance.

Furthermore, we have shown that explicit modelling and classification of background characters, also leads to a substantial improvement in classification of the principal characters. It is worth noting that the background classifier was trained on

one TV series (Scrubs) and then applied successfully both to another TV series (BBT) and a feature film—so it appears that it transfers well away from the original training source.

Taken together with the ConvNet based face-track descriptor, the stronger supervision leads to near saturation of performance on the Buffy benchmark, and also to learning classifiers that generalize well to unseen episodes. Finally, we have also shown that both our background classifiers and bag formation can benefit previous algorithms.

The contributions we have introduced are applicable to other tasks that obtain supervision from aligned scripts. For example, to action recognition [10]. We would expect similar performance improvements in these cases as well. Additional applications of this work span multiple active research topics: automatic labeling of this nature and accuracy can facilitate building datasets for movie based question answering [35], [40]; another interesting application is the alignment of books with movies [34]; or generating accurate ground truth for methods like [4].

As far as further improvements to the proposed method are concerned, the supervisory information can be made stronger still by using the temporal alignment of the audio and visual tracks, e.g., to determine who is speaking when given two subtitles for a shot, or for verifying a speaker detection using the correlation between the audio amplitude and lip movement [5]. Additionally, improving the face detection, as in [17], could further improve the track coverage.

ACKNOWLEDGMENTS

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract number 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon. Funding for this research was also provided by the EPSRC Programme Grant Seebibyte EP/M013774/1 and Academy of Finland project number 310325.

REFERENCES

- [1] S. Andrews, I. Tsochantaris, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. 15th Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 577–584.
- [2] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Finding actors and actions in movies," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 2280–2287.
- [3] M. Bauml, M. Tapaswi, and R. Stiefelhausen, "Semi-supervised learning with constraints for person identification in multimedia data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3602–3609.
- [4] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proc. Asian Conf. Comput. Vis.*, vol. 2, pp. 87–103, 2016.
- [5] J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in *Proc. Workshop Multi-View Lip-Reading, ACCV*, vol. 2, pp. 251–263, 2016.
- [6] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised metric learning for face identification in TV video," in *Proc. IEEE Conf. Comput. Vis.*, 2011, pp. 1559–1566.
- [7] T. Cour, B. Sapp, A. Nagle, and B. Taskar, "Talking pictures: Temporal grouping and dialog-supervised person recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1014–1021.
- [8] T. Cour, B. Sapp, and B. Taskar, "Learning from ambiguously labeled images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 919–926.
- [9] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *J. Mach. Learn. Res.*, vol. 12, pp. 1501–1536, 2011.
- [10] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *Proc. IEEE Conf. Comput. Vis.*, 2009, pp. 1491–1498.
- [11] M. Everingham, J. Sivic, and A. Zisserman, "'Hello! My name is... Buffy'—automatic naming of characters in TV video," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 899–908.
- [12] M. Everingham, J. Sivic, and A. Zisserman, "Taking the bite out of automatic naming of characters in TV video," *Image Vis. Comput.*, vol. 27, no. 5, pp. 545–559, 2009.
- [13] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 634–647.
- [14] M. Haurilet, M. Tapaswi, Z. Al-Halah, and R. Stiefelhausen, "Naming TV characters by watching and analyzing dialogs," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–9.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.
- [16] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar, "Has my algorithm succeeded? an evaluator for human pose estimators," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 114–128.
- [17] H. Jiang and E. G. Learned-Miller, "Face detection with the faster R-CNN," *CoRR*, 2016.
- [18] A. Kläser, M. Marszalek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *Proc. 11th Eur. Conf. Trends Topics Comput. Vis.*, 2010, pp. 219–233.
- [19] M. Köstinger, P. Wohlhart, P. Roth, and H. Bischof, "Learning to recognize faces from videos and weakly related information cues," in *Proc. 8th IEEE Int. Conf. Adv. Video Signal-Based Surveillance*, 2011, pp. 23–28.
- [20] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 720–735.
- [21] A. Ozerov, J.-R. Vigouroux, L. Chevallier, and P. Pérez, "On evaluating face tracks in movies," in *Proc. IEEE Int. Conf. Image Process.*, 2013, pp. 3003–3007.
- [22] O. M. Parkhi, E. Rahtu, and A. Zisserman, "It's in the bag: Stronger supervision for automated face labelling," in *Proc. ICCV Workshop: Describing Understanding Video Large Scale Movie Description Challenge*, 2015.
- [23] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "Compact and discriminative face track descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1693–1700.
- [24] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, Art. no. 6.
- [25] D. Ramanan, S. Baker, and S. Kanade, "Leveraging archival video for building face datasets," in *Proc. IEEE Conf. Comput. Vis.*, 2007, pp. 1–8.
- [26] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei, "Linking people with 'their' names using coreference resolution," in *Eur. Conf. Comput. Vis.*, 2014, pp. 95–110.
- [27] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.
- [28] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 8.1–8.12, <http://www.bmva.org/bmvc/2013/Papers/paper0008/index.html>
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.
- [30] J. Sivic, M. Everingham, and A. Zisserman, "'Who are you?'—learning person specific classifiers from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1145–1152.
- [31] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.
- [32] C. Solomon Mathialagan, A. C. Gallagher, and D. Batra, "VIP: Finding important people in images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4858–4866.

- [33] M. Tapaswi, M. Bauml, and R. Stiefelbogen, ““Knock! Knock! who is it?” Probabilistic person identification in TV series,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2658–2665.
- [34] M. Tapaswi, M. Bauml, and R. Stiefelbogen, “Book2Movie: Aligning video scenes with book chapters,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1827–1835.
- [35] M. Tapaswi, Y. Zhu, R. Stiefelbogen, A. Torralba, R. Urtasun, and S. Fidler, “MovieQA: Understanding stories in movies through question-answering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4631–4640.
- [36] A. Vedaldi and K. Lenc, “Matconvnet – convolutional neural networks for matlab,” *CoRR*, 2014.
- [37] P. Wohlhart, M. Köstinger, P. M. Roth, and H. Bischof, “Multiple instance boosting for face recognition in videos,” in *Proc. Joint Pattern Recognit. Symp.*, 2011, pp. 132–141.
- [38] L. A. Wolsey, *Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.
- [39] J. Yang, R. Yan, and A. G. Hauptmann, “Multiple instance learning for labeling faces in broadcasting news video,” in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 31–40.
- [40] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” *arXiv:1506.06724*, 2015.



Omkar M. Parkhi is a Research Scientist at Facebook’s Applied Machine Learning Group. Before that he was a Postdoctoral Researcher and a DPhil candidate in the Department of Engineering Science, at the University of Oxford.



Esa Rahtu received his PhD degree from the University of Oulu in 2007. Currently he is an Assistant Professor at Tampere University of Technology (TUT) in Finland. Prior to joining TUT, Rahtu was a senior researcher at the Center of Machine Vision research at the University of Oulu in Finland. In 2008, he was awarded a post-doctoral research fellow funding by the Academy of Finland. His main research interests are in computer vision and deep learning.



Qiong Cao is a Senior Researcher at Tencent YouTu Lab, Shenzhen. Before that, she was a Postdoctoral Researcher at the Department of Engineering Science, University of Oxford. She obtained her PhD in Computer Science from the University of Exeter.

Andrew Zisserman is the professor of computer vision engineering in the Department of Engineering Science at the University of Oxford.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**