Unsupervised Metric Learning for Face Identification in TV Video

Ramazan Gokberk Cinbis, Jakob Verbeek, Cordelia Schmid

Unsupervised Metric Learning for Face Identification in TV Video

Ramazan Gokberk Cinbis, Jakob Verbeek and Cordelia Schmid
LEAR, INRIA Grenoble Laboratoire Jean Kuntzmann
firstname.lastname@inrialpes.fr

Abstract

The goal of face identification is to decide whether two faces depict the same person or not. This paper addresses the identification problem for face-tracks that are automatically collected from uncontrolled TV video data. Face-track identification is an important component in systems that automatically label characters in TV series or movies based on subtitles and/or scripts: it enables effective transfer of the sparse text-based supervision to other faces. We show that, without manually labeling any examples, metric learning can be effectively used to address this problem. This is possible by using pairs of faces within a track as positive examples, while negative training examples can be generated from pairs of face tracks of different people that appear together in a video frame. In this manner we can learn a cast-specific metric, adapted to the people appearing in a particular video, without using any supervision. Identification performance can be further improved using semi-supervised learning where we also include labels for some of the face tracks. We show that our cast-specific metrics not only improve identification, but also recognition and clustering.

1. Introduction

Face identification is the problem of determining whether two faces are of the same person or not, *i.e*. it is a binary classification task over pairs of examples, where the positive class corresponds to face pairs of the same person. This contrasts with face recognition, where a face should be recognized as one of a set of known individuals, or potentially rejected as being none of those, which is a multi-class classification problem over single examples. Generally, the identification confidence score can be interpreted as a similarity measure between faces: faces are more similar as they are more likely to be classified as a positive pair. Face identification is extremely challenging since the appearance variability of a single person may be very large compared to inter-person variations. Subtle inter-person appearance variations are easily obscured by big intra-person appearance variations due to photometric factors such as lighting, scale, and viewpoint, or due to changes in expression, hair style, or occlusions. In this work we address face identification in videos where, instead of a single image per face, we have a sequence of face images collected using a tracker.

Face identification for still images in difficult uncontrolled settings has recently received considerable interest following the release of the Labeled Faces in the Wild (LFW) data set [10]. This data set contains around 13.000 face images collected from the web, with large intra-person variations. Since its release in 2008 the best results have improved from around 28% error-rate in 2007 to around 11% for the current state-of-the-art [8, 17]. Face-track recognition has been studied before in controlled settings, see *e.g*. [2], but there has been little work on uncontrolled video.

Other recent work studies face recognition without using labeled examples. Instead, incomplete or ambiguous forms of supervision are used. For example, [1, 13] consider recognition of people in captioned images taken from *Yahoo!News* by automatically linking faces in the image with names in the caption. They do so based on correlations between name occurrence and face appearance that can be detected in large data collections. Others have worked on uncontrolled video material such as TV series [5] or movies [4], where scripts and subtitles can be used to obtain cues as to which characters are present when. These weak cues for character presence are then combined with facial similarities to perform character recognition.

While [1, 4, 5, 13] differ in how they associate names and faces, they all rely on face representations that are sensitive to the intra-person appearance variations. As shown in [7, 9] in the context of recognition from captioned news images, learned similarity metrics can significantly improve recognition performance. In this paper we explore whether metric learning can also be exploited in uncontrolled video. As opposed to [7, 9] which learn a generic metric from labeled faces of thousands of individuals, we are interested in learning similarity metrics adapted to the characters appear-

ing in a specific video given none or a few labeled faces.

Our first contribution is to show that such cast-specific metrics lead to significantly better performance than generic metrics trained on faces of many other people. Our second contribution is to show that cast-specific metrics can be learned without any supervision. Given face tracks, we exploit the fact that all faces in a given track are of one person, and that two tracks that appear in the same video frame contain faces of different people. In this manner we automatically collect positive and negative face pairs to train a cast-specific metric. We refer to this approach as "unsupervised" metric learning throughout the paper. Note that it can also be considered as a "self-supervised" learning approach.

We experimentally compare our unsupervised cast-specific metric to a cast-specific metric learned from labeled face tracks as well as to generic ones. As generic metrics we use the L2 distance over the face descriptors and a metric learned on the LFW data set. Experimental results show that our completely unsupervised cast-specific metric significantly outperforms generic metrics. Furthermore, using a small number of labeled face tracks in addition to the automatically generated training pairs further reduces the error rates to around half the error of the generic metrics.

In the following section we discuss the related work in more detail. In Section 3 we present our face identification approach, as well as the extraction of face tracks and facial features. In Section 4 we present our experimental results based on three episodes of the TV series "Buffy the vampire slayer". Finally, we present our conclusions in Section 5.

2. Related Work

Our goal is to exploit unlabeled face tracks to learn metrics that are robust to intra-person appearance variations. By using unlabeled tracks, we can learn a metric from the same faces that need to be recognized at a later stage. Closely related to our work, [9] learns metrics from captioned news images in a multiple-instance learning setting where bags of examples (faces in an image) come with bags of labels (names in the caption). An alternating optimization procedure learns a metric based on names-faces associations, and then updates the name-face associations given the metric. In our work we go one step further by not requiring any labels at all; instead we rely on the structure of the face tracks.

Recently, there has been considerable interest in face recognition without using labeled examples [1, 4, 5, 6, 13, 14]. Instead, ambiguous and incomplete supervision from image captions, or subtitles and scripts for video, are used in combination with facial similarity to perform recognition. In contrast to our work, default or non-optimized metrics are used to define face similarities. We show that this is suboptimal, as these similarities can be sensitive to intra-person appearance changes due to nuisance factors such as lighting, scale, and viewpoint changes, or due to changes in expression, hair style, or occlusions.

In [1] a large data set of captioned news images collected from *Yahoo!News* was introduced, with the goal to automatically label the faces in the images without using manual labels. The face appearance of each person is modeled with a Gaussian distribution, and the names in the caption are used to enforce that each face can only be assigned to the Gaussians that correspond to the names in the caption. A similar approach was used in [14], but here the faces are first clustered based on appearance, and then they learn a multinomial distribution over the cluster indices for each name. In [13] interest points detections are matched across face pairs to compute a matching score by averaging the Euclidean distance between matched SIFT descriptors. Using the distances between faces that all have a particular name in the caption, clusters of highly similar faces are found by computing the densest component in a graph over the faces with edge weights given by the matching scores.

Others have addressed the same problem in the context of TV series and feature films [4, 5, 6]. Here, instead of image captions, the recognition is based on subtitles and possibly scripts, and individual face detections are grouped using low-level feature tracking. In [5, 6] scripts are temporally aligned with the video using the timed-stamped subtitles. Speaker detection makes it possible to label a number of face tracks with high accuracy: [5] reports 90%. These automatically labeled face tracks are then used to classify the remaining ones based on the minimum frame-to-frame L2 distances between the face descriptors, either using a nearest neighbors classifier in [5], or using SVMs with RBF kernels in [6]. In [4] only subtitles are used, exploiting first, second, and third person references therein. Several distances between tracks are defined, including the minimum face-to-face L2 distance between PCA projections of the faces, and χ^2-distances between color histograms computed over the faces. On a short temporal scale, a cost is computed for all possible clusterings of faces based on these distances. The final grouping is only determined at a later stage when the subtitle-based supervision is also taken into account.

The idea to exploit tracking to obtain training data has been explored by others before in the context of supervised classifier training [3, 19, 11]. In [3] unlabeled face tracks were used to complement manually labeled static face images to learn facial attributes in a semi-supervised manner. Starting from a classifier learned from hand-labeled data, iteratively examples are added from tracks that contain frames classified with high confidence. Since facial attributes, such as gender or age, are unchanged over the face track, all examples from these tracks may be added to the training set. In [19] track information is used to improve learning of person-specific classifiers. In addition to supervised training data, within-track face pairs are used to define a penalty for classifying them as different people, and face

pairs from temporally overlapping tracks are used to define penalties for classifying those as the same person. Similarly in [11], same-person and different-person constraints are included into a Gaussian Process (GP) classifier. These constraints guide the inference procedure for prediction and active learning tasks. Unlike our work, these approaches require a minimum of hand labeled examples. In addition, the domain-specific metrics we learn can be used to define a better kernel for these approaches.

3. Unsupervised face metric learning

In this section we describe our processing pipeline to extract face-tracks, and facial-features in Section 3.1, see Figure 1 for an overview. We continue in Section 3.2 to present how we learn metrics for face identification from the extracted face tracks, and how we used them for track identification in Section 3.3.

3.1. Face detection, tracking, and features

In order to build face tracks in videos, we first use a face detector on individual video frames and then link the obtained detections. Such a detection-based approach for object tracking has been shown effective in uncontrolled videos [5, 12, 16].

We use the Viola-Jones [18] face detector to get an initial set of detections. In order to link the detections into face tracks, we employ the approach of [12], which is a variant of the tracking method proposed in [5]. A Kanade-Lucas-Tomasi (KLT) tracker [15] is applied forwards and backwards in time, which provides point tracks across detection bounding boxes. Each detection pair is assigned a connectivity score according to the number of shared point tracks. The tracks are formed using agglomerative clustering on the detections using the connectivity scores, which results in tracks.

Many of the false positives of the face detector do not have temporal support. Therefore, such false detections are easily eliminated by forming face tracks only from detections with a sufficiently large number of shared KLT point-tracks, and then discarding very short tracks. Similarly, there are sometimes temporal gaps in the true face tracks. Such missed detections are recovered by filling in these gaps using a least-squares estimation technique [12]. Using the bounding-box coordinates of the detections in a track, the coordinates of the missing detections are estimated by minimizing the distances to the coordinates of neighboring detections. The same estimation method is also used for temporal smoothing of the already existing detection bounding boxes.

We use facial features to encode the appearance of the face detections in each track. First, using the publicly available code of [5], we localize nine features on the face: the corners of the eyes and mouth, and three points on the nose, see Figure 1. We then extract SIFT descriptors at these nine locations at three different scales, which we concatenate to form a feature vector $\boldsymbol{f} \in \mathbb{R}^D$ of dimension $D = 3 \times 9 \times 128 = 3456$. As the descriptors are computed at facial feature points, it is robust to pose and expression changes. Using the SIFT descriptor makes it also robust to small errors in localization.

Figure 1. An overview of our processing pipeline. (a) A face detector is applied to each video frame. (b) Face tracks are created by associating face detections. (c) Facial points are localized. (d) Locally SIFT appearance descriptors are extracted on the facial features, and concatenated to form the final face descriptor.

3.2. Metric learning from face tracks

Given a set of face tracks we can extract face pairs from them to learn a metric over the face descriptors in an unsupervised manner. Let $T_i = \{\boldsymbol{f}_{i1}, \ldots, \boldsymbol{f}_{in_i}\}$ denote the i-th track of length n_i. We generate a set of positive training pairs P_u by collecting all within-frame face pairs:

$$P_u = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{il})\}. \tag{1}$$

Similarly, using all pairs of tracks that appear together in a video frame, we generate a set of negative training pairs N_u by collecting all between-track face pairs:

$$N_u = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : o_{ij} = 1\}, \tag{2}$$

where $o_{ij} = 1$ if two tracks appear in the same video frame, and $o_{ij} = 0$ otherwise.

If for some of the face tracks T_i the character label l_i is available, then we use these to generate supervised training pairs in a similar manner as above. Positive pairs are collected from tracks of the same character:

$$P_s = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : l_i = l_j\}, \tag{3}$$

and tracks of different people provide negative pairs:

$$N_s = \{(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}) : l_i \neq l_j\}. \tag{4}$$

In practice a large number of training pairs can be generated without using any supervision: the 327 tracks in our test set generate roughly 1.4 million positive pairs, and the 79 pairs of distinct tracks that occur at the same time yield approximately 600.000 negative training pairs. This large

number of training pairs obtained in this manner, however, have some biases. The positive within-track pairs occur nearby in time, which means that they show less appearance variations, *e.g*. lighting and pose will vary less within a track than across different tracks. The negative tracks can be biased: if there are some characters that co-occur much more often than others, the metric learning will focus on distinguishing these characters.

To learn face identification metrics we use the Logistic Discriminant Metric Learning (LDML) approach of [8], which achieved state-of-the-art results on the LFW benchmark. LDML learns a Mahalanobis distance defined by a semi-positive definite matrix $\boldsymbol{M} \in \mathbb{R}^{D \times D}$:

$$d(\boldsymbol{f}_i, \boldsymbol{f}_j) = (\boldsymbol{f}_i - \boldsymbol{f}_j)^\top \boldsymbol{M} (\boldsymbol{f}_i - \boldsymbol{f}_j). \quad (5)$$

The Mahalanobis distance is mapped to a classification probability using a logistic discriminant model:

$$p(y_{ij} = +1) = \frac{1}{1 + \exp\left(d(\boldsymbol{f}_i, \boldsymbol{f}_j) - b\right)}. \quad (6)$$

The matrix $\boldsymbol{M}$ and bias b are learned by maximizing the log-likelihood over training pairs $(\boldsymbol{f}_i, \boldsymbol{f}_j)$ labeled as either positive ($y_{ij} = +1$, same person) or negative ($y_{ij} = -1$, different people).

Since we have very high dimensional feature vectors, learning a full matrix $\boldsymbol{M}$ would lead to overfitting: a symmetric 3456×3456 matrix has 5.973.696 unique elements. To avoid overfitting, we use a low-rank constraint on $\boldsymbol{M}$ by defining it as $\boldsymbol{M} = \boldsymbol{L}^\top \boldsymbol{L}$, where $\boldsymbol{L}$ is a $d \times D$ matrix [7, 9]. In practice we set $d = 35$ which results in optimization over 60.480 parameters.

3.3. Metrics for identification and recognition

Once a metric is learned we can use it to define a distance between tracks for identification and recognition. A common approach [4, 5] is to take the min-min distance over the faces in each track:

$$d_{mm}(t_i, t_j) = \min_{k,l} d(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}). \quad (7)$$

The motivation for the min-min distance is that it will be robust against pose and expression changes, since it only compares the most similar appearances.

When we use metrics specifically learned to suppress intra-person appearance variations, ideally, all faces of the same person should be close and not only the ones with the same expression or pose. Therefore, we could also use the average face-to-face distance

$$d_{av}(t_i, t_j) = \frac{1}{n_i \times n_j} \sum_{k,l} d(\boldsymbol{f}_{ik}, \boldsymbol{f}_{jl}). \quad (8)$$

A potential advantage of the average distance is that it is based on more face comparisons and might therefore be less sensitive to outliers: a pair of faces of different people that have, erroneously, a small distance. We will compare these two track distances for identification in our experiments.

In our recognition experiments we use a set of labeled face tracks to classify unlabeled tracks in the test set. We compare nearest neighbor classifiers based on the track-to-track distances with a multi-class kernelized logistic discriminant classifier. We use an exponential RBF kernel defined as: $k(t_i, t_j) = \exp(-\frac{1}{\sigma^2} d(t_i, t_j))$, where we set σ^2 as the average track-to-track distance (which can be either min-min or average) among the training tracks.

4. Experimental evaluation

We first describe the data set we use in our experiments, before presenting our experimental results in Section 4.2.

4.1. Dataset

Our data set consists of tracks from episodes 9, 21 and 45 of the TV series "Buffy the vampire slayer", where each episode belongs to a different season of the series. We manually annotated 639 of the automatically extracted face tracks, which in total encompass around 45.000 face detections. In our annotations, we use nine categories, where eight of them represent the main characters and the remaining one is used for other characters.

We split the data set into 312 training and 327 test tracks, with the number of training and test tracks being approximately equal for each character. There are 85 training and 71 testing tracks assigned to the "other" category. When separating the data into training and test set, we use temporally continuous parts, the length of which vary depending on the distribution of occurrence of a character. The tracks in the training set are used for supervised learning, and the ones in the test set to evaluate performance. The tracks in the test set are also used to gather unsupervised examples for metric learning. However, we never use the category labels of the test tracks for training.

In the experiments involving supervised and semi-supervised learning, we provide tracks only from the eight main characters as the supervised examples. In contrast, for the unsupervised and semi-supervised scenarios, unsupervised learning is performed on the tracks both from the main characters and the ones labeled as "other". This provides a realistic setting where the unsupervised learning includes faces of many other people, *e.g*. in the background, that are not the main characters in the video. Considering that the "other" category constitutes approximately 25% of the test tracks, its presence significantly increases the difficulty of unsupervised learning.

Both training and test sets do not include false positive face tracks. We manually removed false positive face tracks, although most can be eliminated automatically using various simple post-processing methods.

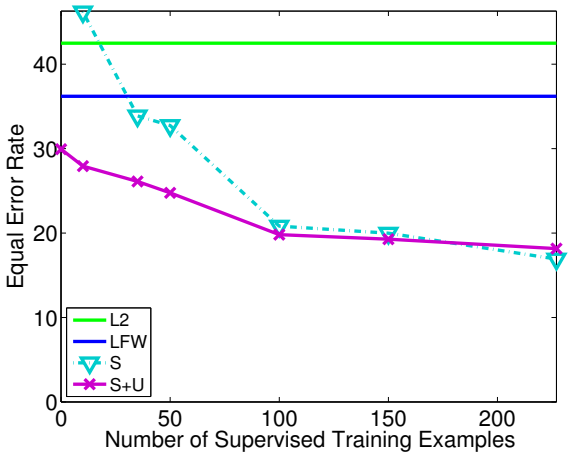


Figure 2. Equal error rate (EER) as a function of the number of training examples when using metrics learned from only supervised tracks (S, green) and using semi-supervised learning that also exploits unlabeled tracks to learn the metric (S+U, magenta). The performance of the L2 distance (red) and a metric learned on the LFW set (blue) are also shown for reference.

The resulting dataset is available at http://lear.inrialpes.fr/data.

4.2. Experimental results

Face track identification. In our first set of experiments we evaluate track identification performance using different metrics. Figure 2 shows the identification equal error rate (EER) as a function of the total number of supervised training tracks. The EER is computed by sorting all pairs of test tracks by their distance, then computing for all distance thresholds the false positive and false negative rate, and then reporting the point where both errors are equal. We compare the results obtained using only the supervised tracks to learn the metric, and when including the unlabeled tracks for metric learning. The left-most point on the semi-supervised curve (S+U) corresponds to only using unsupervised examples. When using supervised tracks, we choose an equal number of tracks of each character from the training set when possible, when all tracks of one character are exhausted we add more examples of other characters.

The results show that our cast-specific metrics perform much better than the generic L2 (42.5%) and LFW distances (36.2%). When a few labeled tracks are available (< 100), the unsupervised training examples improve performance significantly. In particular using no-supervised tracks we obtain a 30% EER, for which around 70 labeled tracks are needed if we do not use the unsupervised training pairs. Using only 10 labeled tracks the supervised metric is worse than the L2 and LFW metrics, probably due to overfitting. When using all labeled training tracks, adding the unsupervised tracks slightly degrades performance. This might be due to the biases in the unsupervised training pairs, as explained in Section 3.2.

Supervision:	0	10	35	50	100	150	227
S (avg)	—	46	34	33	21	20	17
S (min-min)	—	47	37	35	27	25	22
S+U (avg)	30	28	26	25	20	19	18
S+U (min-min)	33	32	30	29	26	24	23

Table 1. Comparison of supervised (S) and semi-supervised (S+U) training using average (avg) and min-min track distances. The EER is shown for several numbers of supervised training tracks.

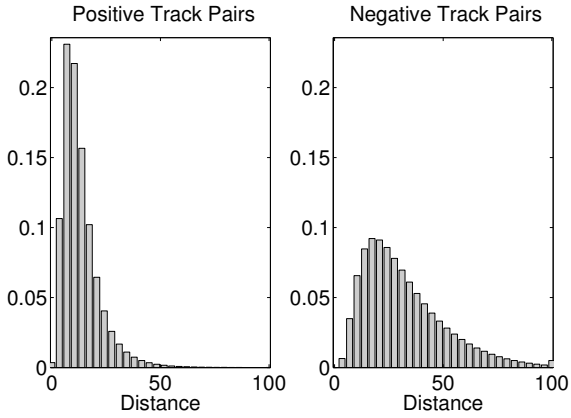
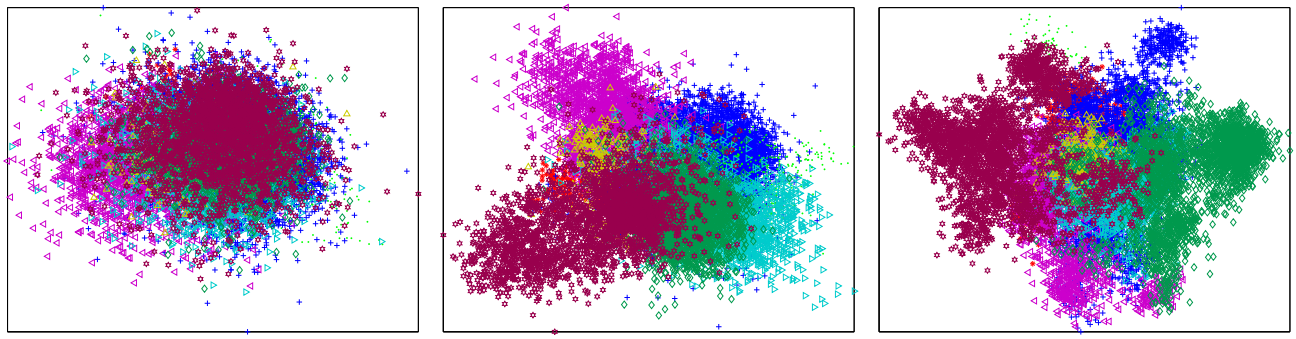


Figure 4. Normalized histogram of distances of face pairs sampled from positive (left) and negative (right) track pairs.

In Figure 3 we visualize the metric learning results by projecting the faces in the test set on the 2D principal subspace of the matrix $\boldsymbol{L}$ that has been learned. We can see that the different characters are completely mixed when using the LFW metric, while the cast-specific metrics yield much better separation. Note that using the completely unsupervised metric (Figure 3(c)), each person is represented in different clusters, while this is not the case using all 227 training tracks as supervision. This is explained by the training bias in the unsupervised case: groups of tracks of a single person might remain separated, if there are no positive training pairs that link different tracks.

In Figure 2 we used the average face-to-face distance $d_a(\cdot,\cdot)$ to define the track-to-track distance. In Table 1 we compare these EER rates to the ones obtained using the min-min distance with our cast-specific metrics. We see that the average distance consistently outperforms the min-min distance. To understand this, we plot in Figure 4 histograms of the face-to-face distances found among positive and negative track pairs using the fully supervised metric. While generally positive pairs have smaller distances, some negative face pairs also have small distances. Therefore, it is more robust to measure the track-to-track distances by av-



(a) (b) (c)

Figure 3. 2D projections of all face descriptors in the test set using LDML metrics trained on (a) all images in the LFW dataset, (b) the 227 supervised training tracks, and (c) using unsupervised training on the test tracks. The faces of the different people are color coded.

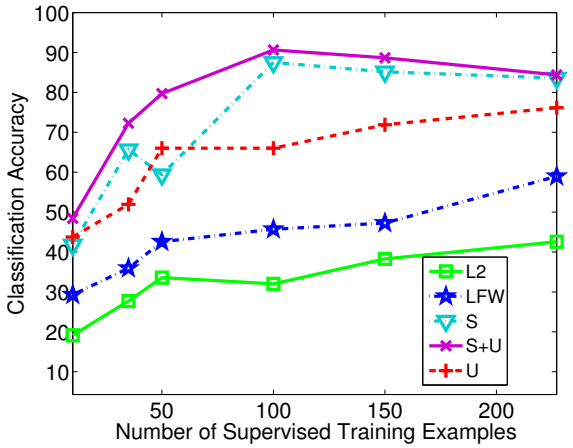


Figure 5. Nearest neighbor classification results.

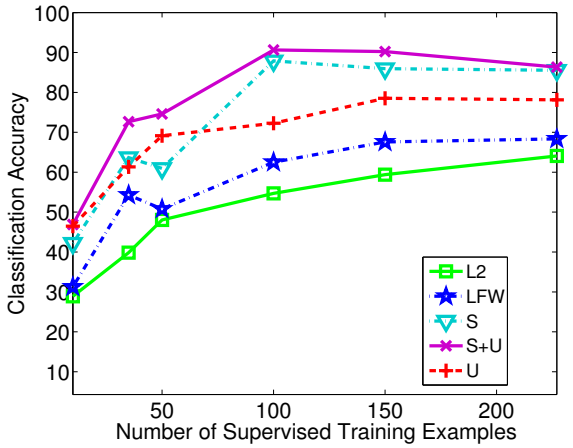


Figure 6. Multi-class logistic discriminant classification results.

eraging, so as to reduce the influence of a single face pair with a small distance. For the L2 and LFW metrics there is very little performance difference, they achieve 41.9% and 35.5% respectively using the min-min distance, compared to 42.5% and 36.2% using average distance.

Face-track recognition. In our next set of experiments we evaluate face recognition using the different metrics. In Figure 5 we use a nearest neighbor (NN) classifier to assign the test tracks to one of the eight characters, while in Figure 6 we use a kernelized multi-class logistic discriminant classifier. For both classifiers we use distances learned from (i) unsupervised examples, (ii) only supervised examples, and (iii) the semi-supervised combination of these. We use the same tracks to learn the (semi-) supervised metrics and the classifiers. For comparison, we also include results obtained using (iv) the L2 metric, and (v) a metric learned on the LFW data set.

We see that also for recognition, the cast-specific metrics yield much better performance than using the L2 or LFW metric. Using all 227 training tracks for recognition and the logistic discriminant classifier, the LFW metric yields a recognition rate of 68%, where the semi-supervised metric attains 86%. For small numbers of training examples, the unsupervised metrics perform comparable to the supervised ones, while for larger numbers of labeled samples it is advantageous to include the unsupervised examples. Perhaps surprisingly, we find both classifiers to give similar results.

Face-track clustering. In our last set of experiments we compare different metrics when used to perform hierarchical clustering of the face tracks in the test set. For evaluation we use the labeling cost of [8], and measure it over the complete range of numbers of clusters. For a given clustering the cost is defined as the number of clicks a user would need to correctly label all tracks. The user can use one button to label a complete cluster with a name, and another button to label a single track. See [8] for more details on

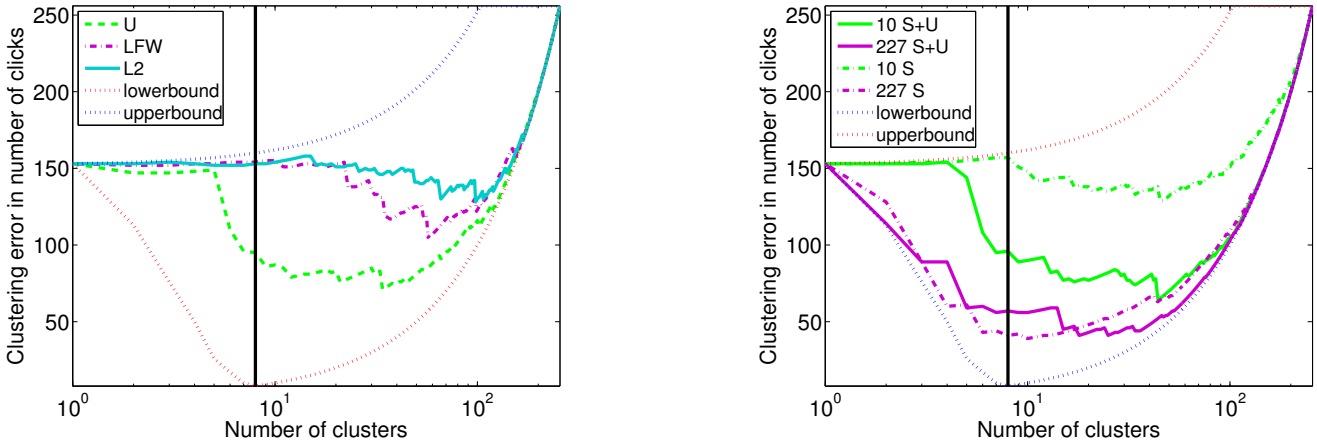


Figure 7. Evaluation of hierarchical clustering error based on different distance metrics, the true number of characters is eight.

10 S	10 S+U	227 S	227 S+U	LFW	L2	U	min	max
157	96	41	57	154	153	95	8	160

Table 2. Comparison of labeling cost using different metrics for eight clusters (equals the number of characters).

this cost, and the derivation of the maximum and minimum cost that can be obtained for a given number of clusters.

In the left plot of Figure 7 we give the labeling costs for unsupervised metrics: L2, learned from the LFW data set, and using unsupervised learning from the face tracks in the test set. We see that for up to 10 clusters, the L2 and LFW metric yield costs that are near the worst possible cost. By inspection, we find that this is because they generate one big cluster that contains almost all faces, and others with very few faces. In the right plot we compare (semi-) supervised metrics learned from 10 and 227 labeled training tracks. Using only 10 labeled tracks supervised-only learning performs about as badly as the L2 and LFW metrics, and in this case adding the unsupervised learning significantly improves the performance. Using all 227 training tracks to learn the metric allows to obtain much better results, and in this case including the unsupervised training examples from the test set has little effect on performance. In Table 2 we give the labeling cost obtained in the case of eight clusters, corresponding to the number of characters in the test set.

In Figure 8 we illustrate several clusters, selected from the clustering with eight clusters, which equals the number of characters in the test set. We use the two best solutions selected from Table 2: the clustering obtained with the unsupervised cast-specific metric (top, cost 95), and the one obtained by supervised learning on all 227 training tracks (bottom, cost 41). For each cluster we show one face per track, with a maximum of eight. The clusters are sorted by size from top to bottom, and we do not display clusters which contain only a one or two tracks.

The clustering produced using the unsupervised metric is fair, but unbalanced. Although the first cluster is only 55% pure, the second cluster is 93% pure, and contains the same person under a wide range of poses, expressions, and lighting conditions. The last two clusters are pure, but contain only a few tracks. The fully supervised metric yields clusters that are much more balanced in size, and with a high degree of purity. We find this an encouraging result, since the clustering itself is completely unsupervised. It essentially shows that using cast-specific metrics we can group face tracks from uncontrolled video by identity with a high degree of accuracy.

5. Conclusion and future work

We have shown that learning a cast-specific metric is useful to improve results for identification, recognition, and clustering of face tracks automatically extracted from uncontrolled TV video. We have also shown that to some degree, such metrics can be learned in an unsupervised manner, by exploiting the temporal structure of the face tracks to sample training pairs for metric learning. A third conclusion is that face identification metrics learned on the Labeled Faces in the Wild data set do not offer a great advantage over using a simple L2 metric over the face descriptors. This can be explained by the differences between news images and TV video, *e.g.* lighting is generally good in news photographs, and very poor in TV video. Another difference is the amount of pose variation: while in news photography people tend to face the camera, in video a wide range of poses is observed as characters engage in conversation or other actions. Finally, in video one also has to cope with poor image quality due to motion blur.

Figure 8. Clustering results using an unsupervised metric (top), and a supervised metric (bottom). Each face image corresponds to unique track. The number of incorrect tracks shown (red) are proportional to the cluster purity. Figure is best viewed in color.

In future work we want to extend the our work to also exploit the profile faces contained in the tracks, and to learn metrics that are not only able to compare faces that are either both profile or frontal, but also to compare pairs of faces where one is frontal and the other profile. Another goal of future work is to evaluate our cast-specific metrics in the full subtitle and script based character recognition setting. Furthermore, we are interested in the applications of our approach to other instance verification problems in video where tracking can be exploited to drive unsupervised metric learning.

Acknowledgements. This work was partially supported by the EU project AXES.

References

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

[2] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, 2011.

[3] N. Cherniavsky, I. Laptev, J. Sivic, and A. Zisserman. Semi-supervised learning of facial attributes in video. In *The first international workshop on parts and attributes (in conjunction with ECCV 2010)*, 2010.

[4] T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking pictures: Temporal grouping and dialog-supervised person recognition. In *CVPR*, 2010.

[5] M. Everingham, J. Sivic, and A. Zisserman. 'Hello! My name is... Buffy' - automatic naming of characters in TV video. In *BMVC*, 2006.

[6] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5):545–559, 2009.

[7] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face recognition from caption-based supervision. *International Journal of Computer Vision*, 2011.

[8] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *ICCV*, 2009.

[9] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*, 2010.

[10] G. Huang, M. Jones, and E. Learned-Miller. LFW results using a combined Nowak plus MERL recognizer. In *Workshop on Faces Real-Life Images at ECCV*, 2008.

[11] A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker. Which faces to tag: Adding prior constraints into active learning. In *ICCV*, 2009.

[12] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *International Workshop on Sign, Gesture, and Activity (SGA) in Conjunction with ECCV*, 2010.

[13] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *CVPR*, pages 1477–1482, 2006.

[14] P. Pham, M. Moens, and T. Tuytelaars. Cross-media alignment of names and faces. *IEEE Transactions on Multimedia*, 12(1):pp.13–27, 2010.

[15] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593 –600, June 1994.

[16] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?": Learning person specific classifiers from video. In *CVPR*, 2009.

[17] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009.

[18] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[19] R. Yan, J. Zhang, J. Yang, and A. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *PAMI*, 28(4), 2006.