Improving Speaker Diarization of TV Series using Talking-Face Detection and Clustering

Hervé Bredin
LIMSI, CNRS
Université Paris-Saclay
F-91405 Orsay, France
bredin@limsi.fr

Grégory Gelly
LIMSI, CNRS, Univ. Paris-Sud
Université Paris-Saclay
F-91405 Orsay, France
gelly@limsi.fr

ABSTRACT

While successful on broadcast news, meetings or telephone conversation, state-of-the-art speaker diarization techniques tend to perform poorly on TV series or movies. In this paper, we propose to rely on state-of-the-art face clustering techniques to guide acoustic speaker diarization. Two approaches are tested and evaluated on the first season of *Game Of Thrones* TV series. The second (better) approach relies on a novel talking-face detection module based on bi-directional long short-term memory recurrent neural network. Both audio-visual approaches outperform the audio-only baseline. A detailed study of the behavior of these approaches is also provided and paves the way to future improvements.

Keywords

speaker diarization; face clustering; talking-face detection

1. INTRODUCTION

Speaker diarization is the task of partitioning an audio stream into homogeneous temporal segments according to the identity of the speaker. Followed by a supervised or unsupervised speaker identification step [15], it allows to answer the question "who speaks when?". Automatic speech transcription also benefits from speaker diarization to address the question "who speaks what?". Resulting augmented (or "rich") transcription can be very useful for multimedia documents structuring and indexing.

Speaker diarization has been successfully applied to various types of content: radio or TV broadcast news (mostly prepared speech and clean audio), telephone conversation (limited number of speakers) and meetings (spontaneous speech and lower audio quality). It is usually tackled using three cascading modules (speech activity detection, speaker change detection and unsupervised speech turns clustering) followed by an optional resegmentation step [2].

This paper investigates the application of speaker diarization on TV series content. As a matter of fact, it has been shown that current state-of-the-art approaches do not perform well when applied to TV series or movies [5, 9]. Several reasons might explain why this is the case. First, the number of speakers may vary a lot from one TV series to another (and even from one episode to another). For instance, *Game of Thrones* episodes range from 28 to 48 speakers, while *The Big Bang Theory* episodes only vary between 9 and 12 speakers. Then, the quality of the speech signal is often degraded because of background music or noise. Finally, though it is acted speech, one can classify speech interactions between TV series characters as spontaneous – as opposed to prepared or read speech in broadcast news, for instance. This leads to a lot of short speech turns for which the usual *2 seconds or more* assumption made by state-of-the-art speaker diarization system no longer holds. For instance, median speech turn duration is only 1.6 second for *Game Of Thrones* TV series.

In this paper, we aim at improving the clustering step by taking advantage of recent advances in deep learning for face recognition and clustering [20] combined with a novel talking-face detection module based on bi-directional long short-term memory (BLSTM) recurrent neural networks. Section 2 compares the approach we propose with existing work from the literature. Our novel approach to talking-face detection using recurrent neural networks is described in Section 3. The complete pipeline is described in details in Section 4, while Section 5 summarizes the first set of experiments applied to *Game Of Thrones* TV series. Finally, we conclude the paper in Section 6.

2. OVERVIEW AND RELATED WORKS

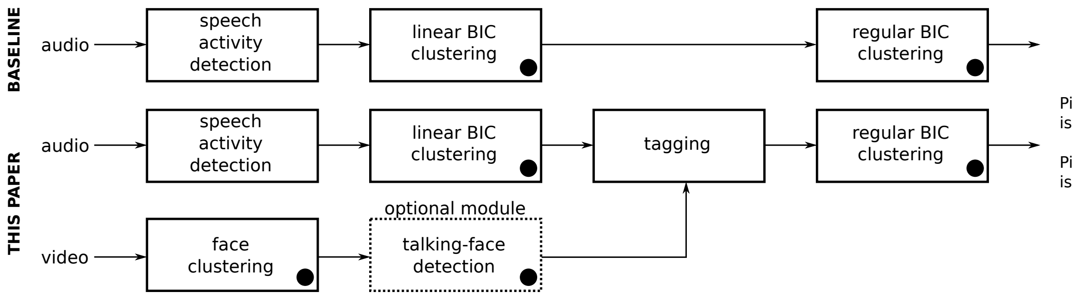
Figure 1 provides an overview of the approaches proposed in this paper. The main intuition is that, when a character is speaking, it is very likely that their face is visible and their voice is audible at the same time. Therefore, it should be possible to achieve (or at least guide) speaker diarization using *face clustering*. Obviously, the reciprocal is not true: the face of a character may be visible even if they do not speak. This is why we added the optional module dedicated to *talking-face detection* – which can be used to avoid tagging speech turns with non-speaking face labels. More details are available in Sections 3 and 4.

While a large number of papers addressed the problem of face recognition or clustering in TV series [12, 13, 23], only a handful of papers focused on speaker identification or diarization in TV series [7, 5]. [7] assumes that acoustic speaker models are available to perform supervised speaker identification. [5] is closer to our work as it aims at improv-

MM '16, October 15 - 19, 2016, Amsterdam, Netherlands

ACM ISBN 978-1-4503-3603-1/16/10. . . $15.00

DOI: http://dx.doi.org/10.1145/2964284.2967202



Pipeline without the optional module is denoted **FACE** in the paper.

Pipeline with the optional module is denoted **TALKING-FACE**.

Figure 1: Overview of the proposed approaches. Every bullet ● corresponds to one hyper-parameter.

ing unsupervised acoustic speaker diarization using dialogue scenes detected automatically from repeating shot/reverse-shot patterns. Focusing on those dialogue scenes (which represents only half of the episode duration), they report a diarization error rate (DER) of 49.3% on one episode of *Game of Thrones*, even though they rely on manual speech turn segmentation. Other scenes are not processed nor part of the evaluation – DER of the full episode is therefore expected to be much higher.

Among face recognition papers referenced earlier, most of them rely on some kind of talking-face detection module to map subtitles and/or transcript to the current speaker. [12] computes the minimum pixelwise total difference between regions around the mouth determined via block matching, average it over the duration of the subtitle, and dual-threshold the resulting value into three classes: speaking, not sure, not speaking. They report a precision of around 94% and recall of 18% on this subtitle-to-face matching task. [13, 23] compute the spatial distance between lower and upper lips, apply band-pass filtering at the expected lip motion frequency (4-8Hz), accumulate it over the duration of the subtitles, and finally threshold it to perform the same task. Our talking-face detection module differs from these previous works in several ways: we do not assume that subtitles are available (and instead rely on detected acoustic speech regions) and we model this task as a sequence labeling task based on audio-visual features (instead of a binary classification task based on one single visual features).

3. TALKING-FACE DETECTION

Recurrent neural networks (RNN) and in particular those based on long short-term memory (LSTM-RNN) have been successfully applied to a wide range of classification tasks for which the discriminative information is embedded in a sequence. In particular, for (acoustic-only) voice activity detection (VAD), [14] showed that LSTM-RNN can outperform other VAD techniques. This section focuses on our first contribution: adapting the modified bidirectional long short-term memory recurrent neural networks (BLSTM-RNN) introduced in [14] to the talking-face detection problem.

3.1 Feature extraction

Each face track (defined in Section 4.2) is described by a sequence of audio-visual features, made of the concatenation of acoustic and visual features. Acoustic features are extracted every 20ms on a 32ms window. We concatenate 6 Mel-Frequency Cepstral Coefficients (MFCC), their first derivatives, their second derivatives, and the first and second derivative of the energy – leading to 20-dimensional acoustic features. Visual features are extracted every 40ms (videos are encoded at 25 frames per second) and interpolated linearly to reach a period of 20ms. We concatenate the normalized (x, y) coordinates of 18 facial landmarks [16] (those located around the lips), their first derivatives, their second derivatives, and the normalized mouth width with its first and second derivatives – leading to 111-dimensional visual features. The final 131-dimensional audio-visual features are used as input of the RNN described in the next paragraph.

3.2 Bidirectional long short-term memory recurrent neural networks

We use the modified version of the LSTM cell introduced in [14] and train the RNN using Quantum Particle Swarm Optimization (QPSO) [21] and Back-Propagation Through Time (BPTT) [24]. QPSO loss function is a weighted frame error rate defined as follows:

$$\text{loss} = (1-\alpha)\sum_{s \in S}(1-z_s) + \alpha \sum_{n \in N} z_n \qquad (1)$$

where S is the set of talking-face frames, N is the set of non-talking-face frames, z is the binary output of the BLSTM-RNN classifier for each frame and α sets the relative importance between errors on the talking-face frames and errors on the non-talking-face frames. For BPTT, we used a similarly weighted version of the maximum likelihood loss function for a binary classifier. All details can be found in [14].

4. FACE-DRIVEN SPEAKER DIARIZATION

This section focuses on our second contribution: using face clustering (and optionally talking-face detection) to improve speaker diarization.

4.1 Speech detection and BIC clustering

Because we focused our work on clustering, we used reference speech activity detection as the first module of all approaches. It is obtained automatically by force-aligning manual speech transcripts with the audio stream. This first step results in a sequence of very short speech segments (one per spoken word): 230ms on average.

The same feature extraction step is used for both the *linear BIC clustering* and the *regular BIC clustering* modules: 12 MFCC coefficients and the energy are extracted every 16ms on a 32ms window. Both modules consist in the application of iterative agglomerative clustering using the Bayesian Information Criterion (BIC) as similarity measure [8]. Each iteration consists in three steps: merge the

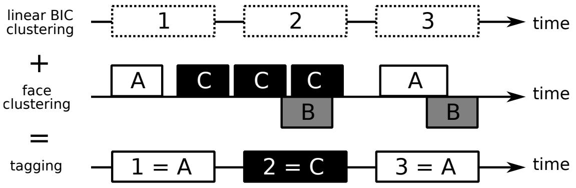


Figure 2: Each speech turn is tagged with the most co-occurring face cluster (e.g. C for turn #2).

two most similar clusters according to the Bayesian Information Criterion (BIC) [8], compute the resulting cluster model, and update the BIC criterion to all other clusters. This process is iterated until the BIC criterion is negative.

The main difference between *linear* and *regular BIC clustering* is that the former process the audio stream in chronological order and can only merge adjacent speech segments coming from the speech activity detection module. The second difference is that the BIC criterion is computed from Gaussian with diagonal covariance matrix for *linear BIC clustering*, while *regular BIC clustering* relies on full covariance matrix. Each module relies on its own penalty coefficient hyper-parameter (λ_{linear} and λ_{regular}) that controls how similar clusters must be to be mergeable.

4.2 Face tracking and clustering

The *face clustering* module in Figure 1 is actually built upon four submodules. First, shot boundaries are detected using optical flow and *displaced frame difference* [25]. Then, face tracking-by-detection is applied within each shot using a detector based on histogram of oriented gradients [10] and the correlation tracker proposed by *Danelljan et al.* [11]. Each face track is then described by its average *FaceNet* embedding and compared with all the others using Euclidean distance [20]. Finally, starting with one cluster per face track, we iteratively merge the two most similar clusters, compute the average *FaceNet* embedding of the resulting cluster and update the Euclidean distance to the other clusters. This process is repeated until the minimum distance between two clusters is higher than a threshold. Additionally, this hierarchical agglomerative clustering process is constrained not to merge co-occurring face track. Source code for this module is available in *pyannote-video* [6].

4.3 Tagging

Figure 2 illustrates what the tagging module does. In a nutshell, we propagate the labels resulting from the *face clustering* step onto co-occurring speech turns resulting from the *linear BIC clustering* step. As depicted by the dashed *talking-face detection* module in Figure 1, we tried two variants of this approach – with or without talking-face filtering. Without talking-face filtering, any face label can be propagated to the current speech turn. With talking-face filtering, only labels of talking faces are propagated.

5. EXPERIMENTS

5.1 Dataset

For this set of experiments, we relied on the reproducible corpus TVD [19] and focused on the first season of *Game Of Thrones* TV series. It contains ten episodes of approximately 55 minutes each. In particular, TVD provides the *"who speaks when?"* annotation obtained by forced alignement of the manual transcript with the audio track. Additionally, we relied on the manual face track annotation provided in [22] to train and evaluate the face clustering and talking-face detection modules.

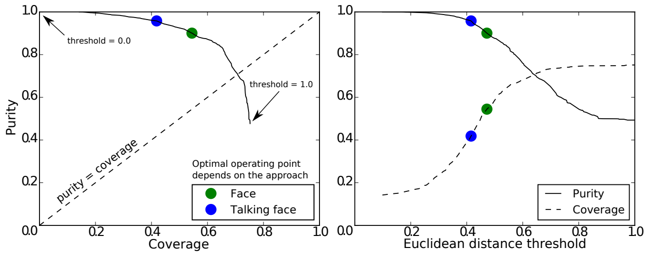


Figure 3: Performance of face track clustering.

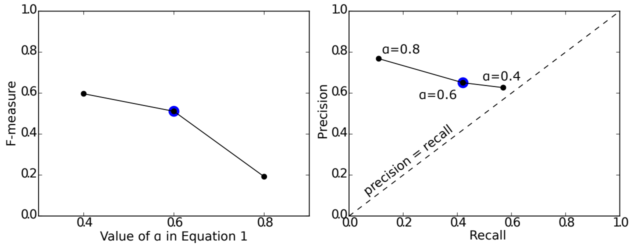


Figure 4: Performance of talking-face detection.

For all experiments and unless otherwise stated, episodes 6 to 10 are used for training and reported evaluation metrics are averaged over episodes 1 to 5.

5.2 Face tracking and clustering

Figure 3 summarizes the performance obtained by the face clustering module. Pratically, we relied on *dlib* machine learning toolkit [17] for face detection [10] and tracking [11], and on *Openface* [1] for *FaceNet* embeddings [20]. Manual face track identities provided in [22] were used for evaluation.

The purity reported in Figure 3 is computed as the ratio of the accumulated total duration of the dominant class in each cluster over the accumulated total duration of each cluster. Complementarily, the coverage is the same as purity after (automatic) clusters and (manual) classes exchanged their role. A perfect clustering would lead to both purity and coverage equal to one. Setting the distance threshold to a value close to 0.0 would result in high purity and low coverage. The fact that coverage never reaches one (even with high distance threshold) is explained by the introduction of the constraint preventing co-occurring face tracks from being merged.

5.3 Talking-face detection

Figure 4 reports the performance obtained by the talking-face detection module. For each face track, the task is to decide when (if ever) the face is actually speaking. Groundtruth was generated by combining manual face track identities from [22] with *"who speaks when?"* annotations from [19]. We relied on *dlib* machine learning toolkit [17] for facial landmarks detection and *Yaafe* [18] for MFCC extraction.

Precision is computed as the ratio of the total duration

Table 1: Results (● is number of hyper-parameters).

	●	Purity	Coverage	**DER**
BASELINE	2	57.1 %	58.6 %	**54.5 %**
FACE	3	53.5 %	65.6 %	**53.1 %**
TALKING FACE	4	59.8 %	60.8 %	**51.8 %**

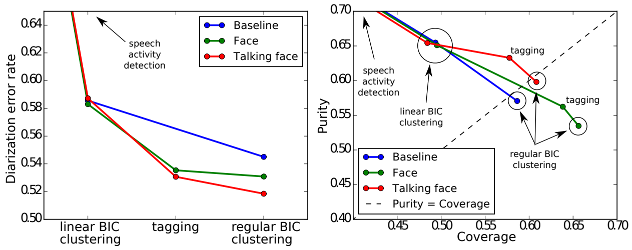


Figure 5: Behavior of tested approaches.

of face track correctly detected as speaking over the total duration of all face tracks. Recall is the ratio of the total duration of face track correctly detected as speaking over the total duration of speaking face tracks. Proper comparison of this approach with prior works on talking-face detection has yet to be done, as it was not the main focus of this paper. Note, however, that a simple baseline classifying all faces as talking would yield a precision of around 22% (i.e. the ratio of talking-faces in the corpus) and a F-measure of 36%.

5.4 Speaker diarization

Table 1 summarizes the performance obtained by the three tested speaker diarization approaches. BASELINE is the one from the upper part of Figure 1. FACE (respectively TALKING FACE) is the one from the lower part of Figure 1 without (resp. with) the *talking-face detection* module.

Diarization error rate (DER) is computed in two steps. First, the optimal one-to-one mapping between reference speaker labels and hypothesized speaker clusters is obtained using the Hungarian algorithm. Once hypothesized speaker clusters are mapped to reference speaker labels, the diarization error rate is the sum of three error rates: missed detection, false alarm and speaker confusion. Note that, in this paper, missed detection and false alarm are null because we rely on the reference speech/non-speech segmentation from [19]. Purity and coverage are defined in the same way as in Section 5.2, with the exception that speech regions replace face tracks.

For each approach, hyper-parameters were optimized jointly towards minimum diarization error rate on the training set, using tree-structured Parzen estimators [4]. Practically, we relied on the *hyperopt* hyper-parameters optimization toolkit [3]. The first observation is that the introduction of the face clustering module does improve the overall speaker diarization performance – whether non-talking faces are filtered (DER = 51.8%) or not (DER = 53.1%). Filtering out non-talking faces tends to improve both purity (+2.8%) and coverage (+2.2%); whereas keeping all faces seems to focus on improving coverage (+7.0%) at the expense of purity (−3.6%).

Figure 5 provides better insight at the internal behavior of each approach. The left (resp. right) part plots the value of the diarization error rate (resp. purity as a function of coverage) computed on the output of each module: *linear BIC clustering*, *tagging* and *regular BIC clustering*. It shows that the output of the face-driven *tagging* module in both *face* and *talking face* approaches is already better than the *baseline*, even before applying the final *regular BIC clustering* module. It also shows that *face* approach does not benefit from the final *regular BIC clustering* module as much as *talking face* does. This expected behavior can be explained by the fact that the *face tagging* module degrades purity a lot more than its *talking face tagging* counterpart – a state from which the subsequent *regular BIC clustering* step cannot recover.

Blue and green dots in Figure 3 correspond to the operating points selected through hyper-parameter optimization for the *face clustering* module of the TALKING FACE and FACE approaches respectively. The TALKING FACE approach "chooses" to stop face clustering earlier than the FACE approach does. Combined with a relatively precise *talking-face detection* module (blue dot in Figure 4), the resulting tagging module is able to increase coverage significantly (+9.1%) while leaving purity relatively stable (−3.1%).

The main limitation of the proposed approach is obviously the *linear BIC clustering* step. As a matter of fact, it appears that all three approaches converged towards the same hyper-parameter λ_{linear} that leads to a huge drop in purity (from 100% to 65%), from which subsequent clustering steps cannot (by design) recover. Future work will have to focus on this particular aspect.

6. CONCLUSIONS

In this paper, we aimed at improving audio speaker diarization applied to TV series using the visual stream. We used state-of-the-art face clustering based on neural network embeddings to initialize speaker diarization with cooccurring face labels, leading to improve diarization error rate (from 54.5% to 53.1%). Additionnaly, we designed a bi-directional long short-term memory recurrent neural network for talking-face detection and integrated it in the proposed framework to further improve the results (from 53.1% down to 51.8%).

Speaker diarization in TV series remains a scientific challenge, but achieving good results could lead to novel multimedia applications improving the audience experience. For instance, accessible subtitles as defined by BBC guidelines[1] could benefit greatly from such a multimodal speaker diarization output. For instance, one could use *"one color per speaker"*, *"horizontal positioning"* of subtitles closer to the current speaker's face thanks to the talking-face detection module, or even use mark *"single quotes for voice-over or out-of-vision speaker"* if no talking face is detected.

Future work will focus on improving the initial linear clustering (or speaker change detection) step, improving the talking-face detection module (and actually comparing it with existing approaches) and investigating joint audio-visual clustering techniques instead of the series of audio and visual clustering proposed in this paper.

Acknowledgements

This work benefited from the support of the CAMOMILE project (CHISTERA) and the MetaDaTV project (ANR).

[1] bbc.github.io/subtitle-guidelines

7. REFERENCES

[1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization: A review of recent research. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(2):356–370, 2012.

[3] J. Bergstra, D. Yamins, and D. D. Cox. Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. In *Proceedings of the 12th Python in Science Conference*, pages 13–20, 2013.

[4] J. Bergstra, D. Yamins, and D. D. Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 115–123, 2013.

[5] X. Bost and G. Linares. Constrained Speaker Diarization of TV Series based on Visual Patterns. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 390–395. IEEE, 2014.

[6] H. Bredin. pyannote-video: Face Detection, Tracking and Clustering in Videos. http://github.com/pyannote/pyannote-video. Accessed: 2016-07-04.

[7] H. Bredin, A. Roy, N. Pécheux, and A. Allauzen. “Sheldon speaking, bonjour!”: Leveraging Multilingual Tracks for (Weakly) Supervised Speaker Identification. In *ACM MM 2014, 22nd ACM International Conference on Multimedia*, Orlando, USA, November 2014.

[8] S. Chen and P. Gopalakrishnan. Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, volume 8, pages 127–132. Virginia, USA, 1998.

[9] P. Clément, T. Bazillon, and C. Fredouille. Speaker Diarization of Heterogeneous Web Video Files: a Preliminary Study. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4432–4435. IEEE, 2011.

[10] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[11] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Accurate Scale Estimation for Robust Visual Tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[12] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy” – Automatic Naming of Characters in TV Video. In *Proceedings of the British Machine Vision Conference*, 2006.

[13] M. Everingham, J. Sivic, and A. Zisserman. Taking the Bite out of Automatic Naming of Characters in TV Video. *Image and Vision Computing*, 27(5), 2009.

[14] G. Gelly and J.-L. Gauvain. Minimum Word Error Training of RNN-based Voice Activity Detection. *Interspeech*, 2015.

[15] V. Jousse, S. Petit-Renaud, S. Meignier, Y. Esteve, and C. Jacquin. Automatic named identification of speakers using diarization and asr systems. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4557–4560. IEEE, 2009.

[16] V. Kazemi and J. Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[17] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[18] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard. YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 441–446, Utrecht, The Netherlands, August 9-13 2010. http://ismir2010.ismir.net/proceedings/ismir2010-75.pdf.

[19] A. Roy, C. Guinaudeau, H. Bredin, and C. Barras. TVD: a Reproducible and Multiply Aligned TV Series Dataset. In *LREC 2014, 9th Language Resources and Evaluation Conference*, 2014.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: a Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[21] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu. Convergence Analysis and Improvements of Quantum-behaved Particle Swarm Optimization. *Information Sciences*, 193:81–103, 2012.

[22] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015.

[23] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Improved Weak Labels using Contextual Cues for Person Identification in Videos. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, May 2015.

[24] P. J. Werbos. Backpropagation Through Time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[25] Y. Yusoff, W. Christmas, and J. Kittler. A Study on Automatic Shot Change Detection. In *Multimedia Applications, Services and Techniques*, pages 177–189. Springer, 1998.