SMART VIDEO BUDDY – CONTENT-BASED LIVE RECOMMENDATION

Damian Borth[1], *Christian Kofler*[2], *Adrian Ulges*[2]

[1]University of Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence, Kaiserslautern, Germany

1. INTRODUCTION

In this technical demonstration we present Smart Video Buddy[1], a content-based live recommender that links video streams with related information. The system performs a real-time analysis of video content by feeding a scenes' color and local patches to statistical learners, which detect semantic concepts in the video stream. Based on the provided recognition results, recommendations for news, products and links are made and automatically adapted to what is being watched in real-time. This way, passive video consumption is turned into an interactive experience.

2. SYSTEM OVERVIEW

The system (Fig. 1) consists of an off-line trained concept detection system, external data repositories and – most importantly – a real-time tagging and recommendation pipeline. To provide recommendations for an unknown video stream, frames are sampled and for each such frame, features are extracted and classified by multiple concept detectors. The resulting scores are fused, leading to an assignment of a single tag to the processed frame. This tag serves as input for a recommender module which selects appropriate pieces of information from external data repositories. The resulting live recommendations are displayed with the video stream in an integrated user interface. For example, if the user is watching a soccer match, Smart Video Buddy offers news, ads, and links related to soccer.

Automatic Tagging The key component of Smart Video Buddy is a video annotation (or *tagging*) engine that automatically infers the presence of semantic concepts (currently 16 sport concepts) from the video stream. Smart Video Buddy employs two automatic annotation systems (or *taggers*) based on RGB color histograms and on visual words [1, 2] using the SiftGPU Library (http://www.cs.unc.edu/~ccwu/siftgpu/). For each concept, a two-class problem of discriminating concept presence from absence is formulated by employing Passive-Aggressive Online Learning (PAMIR) [3].

Fusion The fusion engine combines tagging scores (computed for each sampled frame and feature modality) in two steps: first, for each frame, tag posteriors are combined in a weighted sum fusion and – second – integrated over consecutive frames to smooth temporary uncertainties. For example, the system classifies soccer even while the audience is shown for a moment.

Recommender Based on the final decision of tagging, the recommendation engine provides the user with related information. The current prototype of Smart Video Buddy follows a straightforward approach: news related to the detected concept are aggregated from defined feeds, related videos from YouTube are retrieved, and static lists with related links and a fixed set of advertisements are provided.

Summarizing, instead of passive TV consumption, the system offers interaction with the video content within the respective visual context.

[1]Demo Video available at:
http://madm.dfki.de/smartvideobuddy

3. ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG), project MOONVID (BR 2517/1-1). Further, we would like to thank Markus Koch for substantial contributions to Smart Video Buddy.

4. REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: Efficient Visual Search of Videos," in *Toward Category-Level Object Recognition*, 2006.

[2] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[3] D. Grangier and S. Bengio, "A Discriminative Kernel-based Model to Rank Images from Text Queries," *PAMI*, vol. 30, pp. 1371–1384, 2008.

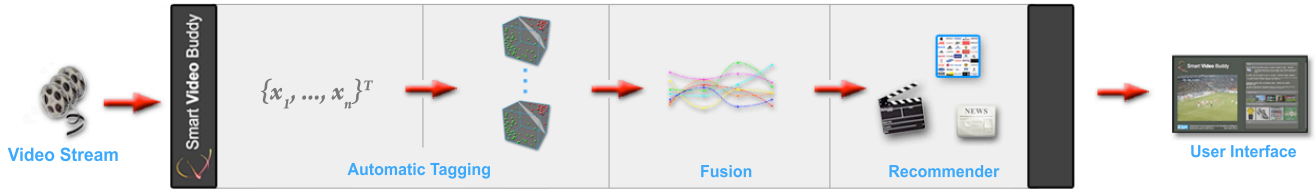

Fig. 1. System overview of Smart Video Buddy: frames are sampled from a video stream and analyzed by the system. Multiple concept detectors provide scores indicating the presence of semantic concepts. These scores are fused into a single detection serving as the input for a recommender module, which selects related information for display in the user interface.