AutomEditor: Video blooper recognition and localization for automatic monologue video editing

CARLOS TOXTLI, West Virginia University
NASSER M NASRABADI, West Virginia University
SAIPH SAVAGE, West Virginia University

Video blogs are every time more popular because of online streaming platforms. Anyone can post content no matter their video editing skills. Novice video bloggers have had to acquire these skills to publish quality content. Video editing is usually a time-consuming task that discourages users to publish periodic content. The most common format for individual video bloggers is the monologue. Monologues have fixed conditions, such as one person at a time and a fixed camera position. Monologues are a perfect setting for automatic video editing (AVE). In this paper, we present AutomEditor, a system that automates monologue video editing. AutomEditor uses multimodal video action recognition techniques to detect video bloopers. AutomEditor extracts body skeleton, face, emotions, and audio features from video clips. Our model implements early feature fusion over recurrent neural networks and multi-layer perceptrons. The model was trained and evaluated by using the BlooperDB, a manually collected and annotated dataset. Our model got a 100% accuracy in the validation set and a 90% in the test set. We propose a blooper localization algorithm for untrimmed videos, based in the predictions frequency. We implement a web interface to visualize the blooper fragments. AutomEditor was able to successfully locate and visualize all the bloopers on test untrimmed videos. We conclude presenting implications for design.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Carlos Toxtli, Nasser M Nasrabadi, and Saiph Savage. 2018. AutomEditor: Video blooper recognition and localization for automatic monologue video editing. 1, 1 (May 2018), 13 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

A blooper is a short clip from a film or video production, containing a mistake made by a person on screen. Detecting bloopers is not a trivial task even with a sophisticated tool. Existing video editing systems, such as Abode Premiere, are a great help for editing video, but the task is still a tedious and time-consuming requiring significant editing skills and an aesthetic sense. In this paper, we present a system that automates monologue video editing. By automated video editing (AVE), we refer to a process which automatically selects suitable or desirable segments from an original video source to create an edited video segment. Generally watching a long unedited video requires a great deal of patience

Authors' addresses: Carlos Toxtli, West Virginia University, nasser.nasrabadi@mail.wvu.edu; Nasser M Nasrabadi, West Virginia University, nasser.nasrabadi@mail.wvu.edu; Saiph Savage, West Virginia University, saiph.savage@mail.wvu.edu.

and time. An effective way to attract a viewer is to present a video that is as compact as possible, yet preserves the most critical features and has no evident errors.

The proper detection of actions in videos relies on long-term contextual information modeling and cross-modality analysis. Since gestures, emotions, and voice tone levels normally change gradually under the same context, analyzing the long-term dependency of these factors will stabilize the overall predictions. Meanwhile, humans perceive others' mistakes by combining information across multiple modalities simultaneously. Combining different modalities will yield better recognition with more human-like computational models [26].

In developing our multimodal system, we have been inspired by many previous works, such as combining visual and audio features [15], as well as speech content [26, 45]. People have also combined physiological signals for emotion recognition tasks [33]. Methods of combining cues from each modality can be categorized into early or late fusion. For early fusion, features from different modalities are projected into the same joint feature space before being fed into the classifier [32, 35]. For late fusion, classifications are made on each modality and their decisions or predictions are later merged together, e.g. by taking the mean or other linear combination [12, 18]. Some works [23, 31] even implemented a hybrid fusion strategy to utilize both the advantages of late fusion and early fusion.

In this paper, we investigated the use of a number of feature extraction, classification and fusion methods. Our final quadmodal method aggregates face, body, audio and emotion features for a single-shot video clip level classification using early fusion. To verify the effectiveness of multimodal fusion, we compared it with ten unimodal methods. Our proposed multimodal approach outperformed the unimodal ones as well as the baseline methods, achieving validation set accuracy of 1.0, and test accuracy of 0.9.

Video localization has been broadly studied for unimodal methods, mainly from image only analysis [24]. Video action localization in multimodal methods is challenging especially when there are temporal and non-temporal features mixed [9]. This is why we proposed a localization method based on the analysis of prediction sequences. The method was effective to localize all the bloopers inserted from the test set in untrimmed videos.

We deployed AutomEditor as a web interface where users are able to submit their videos and visualize the fragments that contain bloopers. In this paper, we describe the whole process of enabling a video action recognition and localization system, from the database creation to the deployment of a web application.

2 RELATED WORK

A research problem closely related to AVE is video summarization. Numerous contributions to this topic have been reported. One of the most straightforward approaches is to compress the original video by speeding up the playback [28]. However, the abstract factor in this approach is limited by the playback speed in order to keep the speech comprehensible. The InforMedia system [39] generates short synopsis of video by integrating audio, video and textual cues.

Another approach to generating semantically meaningful summaries is event-oriented abstraction scheme, such as that presented in [27]. More sophisticated techniques have also been proposed. For example, the trajectories of moving objects were used in [43]. The linear dynamical system theory is applied in [29]. In [19], the authors use singular value decomposition to summarize video content. Generally, summarization requires a semantic understanding of the video content. Our approach is content agnostic, so many of these techniques does not totally fit.

Another technique that is useful for AVE is video action recognition [44]. Video action recognition has been broadly studied for all kind of activities, there are datasets that contain more than a hundred activities [42]. There are an important amount of approaches to detect actions from image sequences [22]. Multimodal approaches have demonstrated to be effective by adding extra features such as 3D skeleton [25]. Multimodal approaches that integrate audio are often used for activities that are related to emotions [34]. The implementation of multimodal video action recognition to AVE has been used for meeting recording editing [8, 20, 21]. The automatic video editing system proposed in this paper uses video action recognition from multimodal features to perform the editing process.

Video localization is a common task for solutions that implement video action recognition [7]. Most of the localization methods are built into the model such as in multi-stage CNN [38]. These techniques are more suitable to image domain features that are benefited from convolutions [37]. Multimodal video action localization techniques have been used for indexing purposes [14, 40]. Localization is a complex task for built-in methods that combine multimodal features that have mixed temporal and non-temporal features. In our system, we implement a localization mechanism based on the frequency of prediction sequences.

3 AUTOMEDITOR

AutomEditor is an end-to-end solution that automates monologue video editing. Figure 1 shows the six stages of the solution that are: 1) Blooper DB: a Bloopers dataset; 2) Feature Extractor: Extract the features from the video clips; 3) Learner: Trains and evaluates the model; 4) Predictor: Retrieves a sequence of predictions; 5) Locator: Localizes the blooper clips in untrimmed videos; 6) Server: Exposes a web service and shows the results in a web interface.

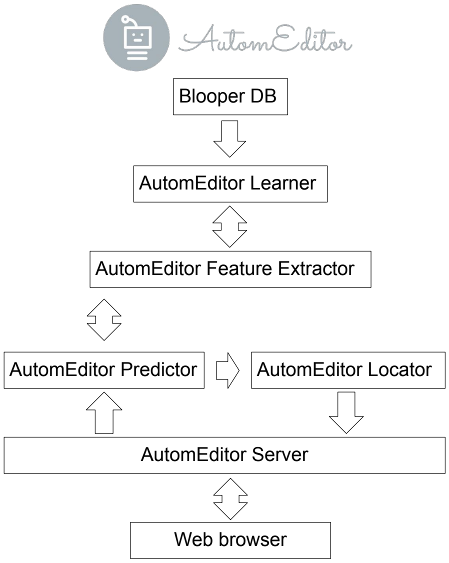


Fig. 1. System components

3.1 Blooper DB

The lack of datasets focused on bloopers force us to build our own database, we titled it Blooper DB. The Blooper DB is a long-term multimodal corpus for blooper recognition. It is constructed by picking out the videos that contain video bloopers from Youtube videos using keywords like ‘bloopers’, ‘green screen’, etc. The videos have multiple resolutions and multiple languages. The dataset is split into training, validation and testing sets. There are 464 videos in the training set, 66 videos in the validation set, and 66 videos in the testing set. Each video clip is annotated by categorical labels, 0 (no blooper) and 1 (blooper). Each video clip lasts between 1 to 3 seconds. The dataset is stratified and has an equal number of samples per each category. Figure 2 shows some examples of the dataset.

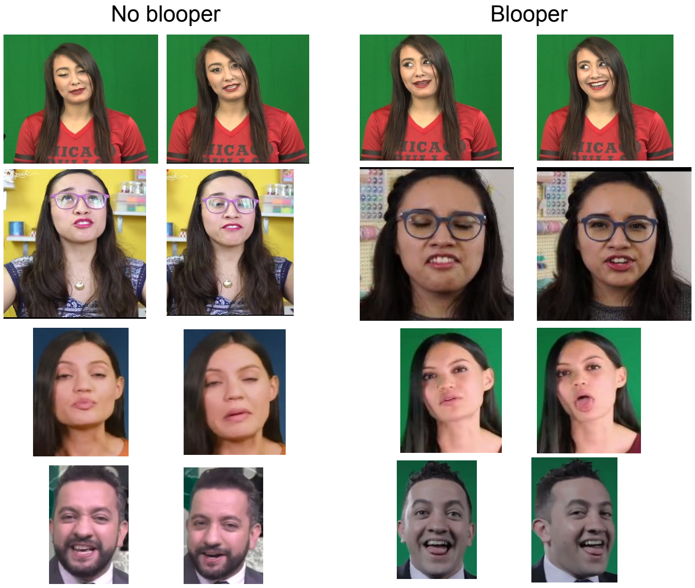


Fig. 2. Bloopers DB samples

Different criteria were taken into account to select the videos. The videos were monologues where only one person was on screen, the camera is fixed, the shoulders are visible, and there are no face pictures in the background. We split long bloopers (more than 2 seconds) into two video clips, a non-blooper (before the mistake) and blooper (the clip that contains the mistake). For short bloopers (1 to 2 seconds), we looked for other video clips of non-bloopers from the same video of about the same length. The video clips do not start or end in truncated phrases. We tried to avoid as much as possible cases where bloopers were green-screen and non-bloopers not.

3.2 Feature Extractor

The main aim of our feature extraction process is to maintain as invariant factors the person descriptors (i.e. gender, age, etc), scale, position, background, and language. Our approach considers four different types of features, that are features from face, body, audio, and emotions.

1) Face features: Visual features consist of OpenFace [11] estimators on the whole frames, and VGG face representation [30] on facial regions. For OpenFace features, we use OpenFace toolkit to extract the estimated 68 facial landmarks in both 2D and 3D world coordinates, eye gaze direction vector in 3D, head pose, rigid head shape, and Facial Action Units intensity [16] indicating the facial muscle movements. The detailed feature descriptions are seen in [2]. Those visual descriptors are regarded as strong indicators of human emotions and sentiments [33, 41]. For the VGG face representation, the facial region in each frame is cropped and aligned using a 3D Constrained Local Model described in [10]. We zero out the background according to the face contour indicated by the facial landmarks. Then, the cropped faces are resized to 224x224x3 and fed into a VGG Face model pre-trained on a large face dataset. We take the 4096 dimensional feature vectors in the fc6 layer and concatenate them with the visual features extracted by OpenFace. The total dimension of the concatenated features is 4805. Specifically, 20 frames are uniformly sampled from each video clip and fed into the network for training and testing. In the case of a shorter length of a video clip, we duplicated the last frame to fill the gap.

2) Body features: We used from OpenPose [13] (a pose estimator framework) the Body-25 model that extracts 25 joints of the person's skeleton from an image. We computed the joint angles from shoulders, arms, neck, and nose, as well as a binary flag per each joint to indicate if it was present. In total, we extracted and normalized 11 handcrafted features from the OpenPose output. We only used joints from the nose, ears, eyes, neck, shoulders, and arms, that are the parts of the body that are usually visible in monologues. From the filtered joints we draw a skeleton with the neck joint fixed at the center, the dimensions normalized. and this skeleton is inserted in a frame of size 224x224 with a black background. This visual representation is passed through a VGG16 net to extract 4096 features. We take the 4096 deep features vector from the fc6 layer and concatenate the 11 features computed from the OpenPose joints. The total dimension of the concatenated features is 4107. 20 frames were uniformly sampled per clip.

3) Emotion features: We used EmoPy [6] a machine learning toolkit for emotional expression to extract the score of each of the seven basic emotions (anger, fear, disgust, happiness, sadness, and contempt) typically used for Facial Expression Recognition (FER). The same seven features were extracted from other four emotion recognition models [1, 4, 5]. We concatenated all the predictions from the models into a 35 features vector. The same 20 faces used for computing the face features were used to compute the emotion temporal features. We condensed the temporal features to a general feature vector by performing a normalized sum to each prediction segment resulting in a vector of 35 elements.

4) Audio features: Audio features are extracted using openSMILE toolkit [17], and we use the same feature set as suggested in the INTERSPEECH 2010 paralinguistics challenge [36]. The set contains Mel Frequency Cepstral Coefficients (MFCCs), âĹĘMFCC, loudness, pitch, jitter, etc. [3]. These features describe the prosodic pattern of different speakers and are consistent signs of their states. For each video clip, we extract 1582 dimensional features from the audio signal. We processed the general features from the whole video clip audio, and the temporal features from the analysis of 20 fragments of the video clip audio.

3.3 Learner

System Architecture Figure 3 shows the architecture of our proposed model. Our deep neural network model consists of three parts: (1) the sub-networks for every single modality; (2) the early fusion layer which concatenates four unimodal representations together; and (3) the final decision layer that estimates the sentiment.

1) Sub-networks: There are 6 subnetworks, two from the face, one with handcrafted features and the other with deep features, all the features are in a sequence of 20 samples. The same case with body features. Emotions are the result of predictions of existing models and are also in a sequence of 20 samples. The audio handcrafted features were taken from one sample per video clip.

2) Early fusion layer: This part concatenates four unimodal representations together, in specific face related features and body related features are joined. The concatenated features from a single video clip are further fed into an LSTM layer with 64 hidden units followed by a dense layer with 256 hidden neurons for temporal modeling. The audio and emotion features are then fed into a fully connected layer with 256 units.

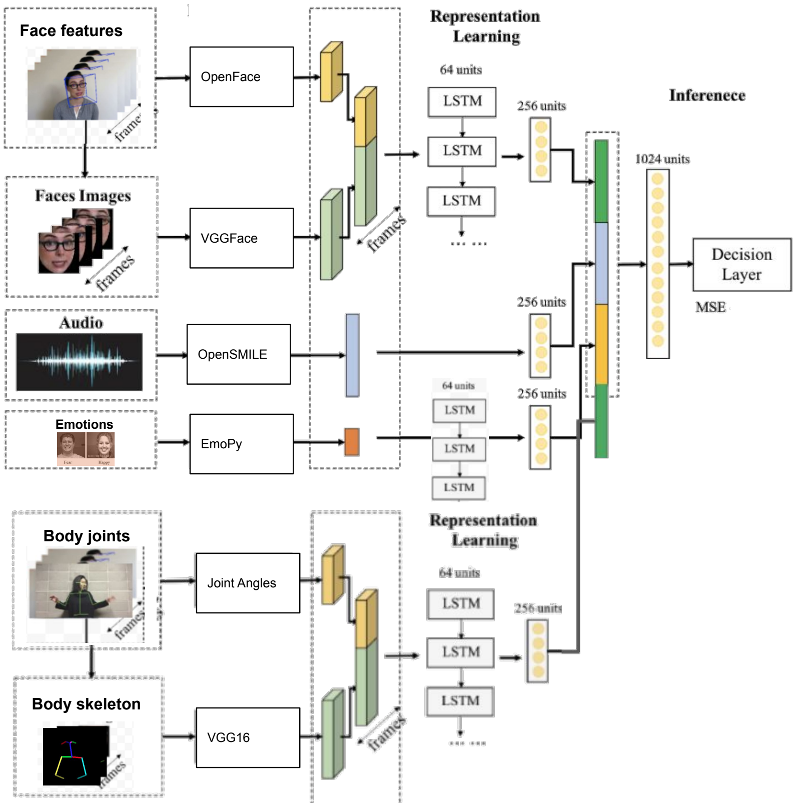


Fig. 3. Architecture model

3) Fusion and Decision Layers: We combine cues from the four modalities using early fusion strategy. The aggregated feature vector is fully connected to a two-layer neural network with 1024 hidden units and a single output neuron, activated by softmax. We use MSE as the loss function for joint training.

3.4 Predictor

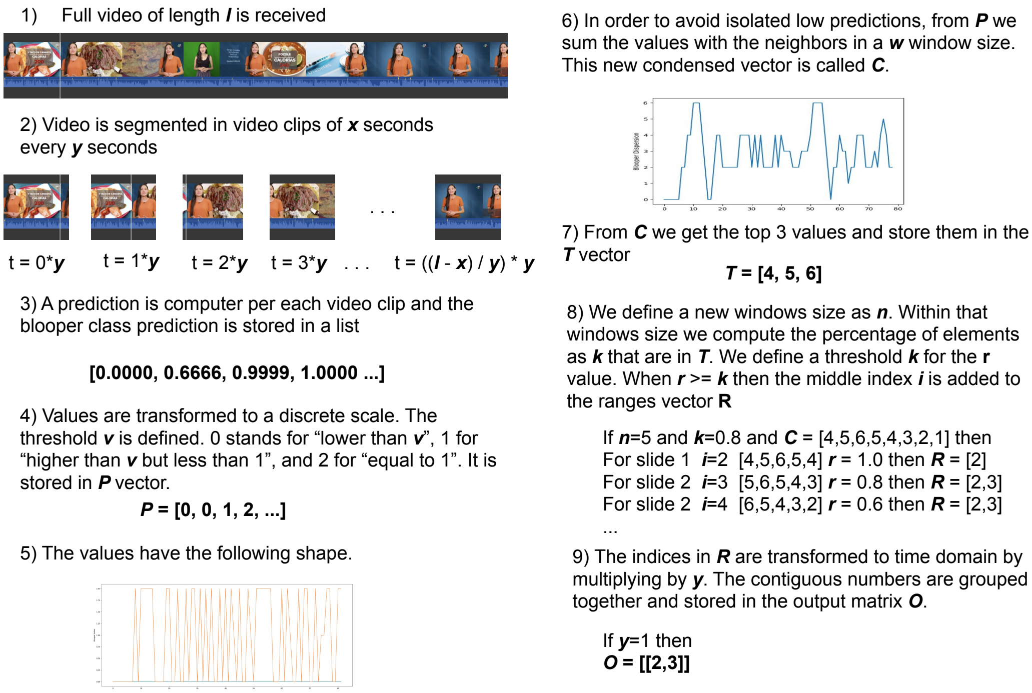
This module is able to get an untrimmed video and split it into fragments of the same length (2 seconds was the default). The precision of our algorithm depends on the separation of each clip. This parameter can be set into the platform and it is defined in milliseconds (500 milliseconds was the default). Once the full-length video was divided into multiple overlapped video clips, then we proceeded to extract features per each video clip and predict the blooper score. We defined the blooper score as the value retrieved by the model for the blooper category.

The predictor output is a sequence of predictions per the configured time-lapse between predictions. The scores are further analyzed in the Locator component to find the ranges where bloopers occur.

3.5 Locator

This module takes the sequence of previously computed blooper scores. Instead of using a 0 to 1 scale, It uses the 0, 1, and 2 values. 0 stands for blooper score = 0, 1 for intermediate values in a threshold, and 2 for blooper score = 1.

The main goal of the Locator module is to find the range of high valued numbers that are together (range). To facilitate this task we first pre-process the prediction sequence. First, we define a sliding window of a configurable size that adds the values of their neighbors to condense the scores and to keep some context of each point. We called this new list of added values as the condensed list. From the condensed list, it computes the three highest values and stores them in the top 3 list. Then the Locator defines another sliding window (of configurable size) and calculates the percentage of elements that are in the top 3 values (within the window). It uses a percentage threshold to add the index of the window to a range. The grouped range index positions correspond to the times of the video that contain the bloopers. The localization steps are explained in Figure 4.



1) Full video of length ***l*** is received

2) Video is segmented in video clips of ***x*** seconds every ***y*** seconds

3) A prediction is computer per each video clip and the blooper class prediction is stored in a list

[0.0000, 0.6666, 0.9999, 1.0000 ...]

4) Values are transformed to a discrete scale. The threshold ***v*** is defined. 0 stands for “lower than ***v***”, 1 for “higher than ***v*** but less than 1”, and 2 for “equal to 1”. It is stored in ***P*** vector.

***P* = [0, 0, 1, 2, ...]**

5) The values have the following shape.

6) In order to avoid isolated low predictions, from ***P*** we sum the values with the neighbors in a ***w*** window size. This new condensed vector is called ***C***.

7) From ***C*** we get the top 3 values and store them in the ***T*** vector

***T* = [4, 5, 6]**

8) We define a new windows size as ***n***. Within that windows size we compute the percentage of elements as ***k*** that are in ***T***. We define a threshold ***k*** for the **r** value. When ***r*** >= ***k*** then the middle index ***i*** is added to the ranges vector **R**

If ***n***=5 and ***k***=0.8 and ***C*** = [4,5,6,5,4,3,2,1] then
For slide 1 ***i***=2 [4,5,6,5,4] ***r*** = 1.0 then ***R*** = [2]
For slide 2 ***i***=3 [5,6,5,4,3] ***r*** = 0.8 then ***R*** = [2,3]
For slide 2 ***i***=4 [6,5,4,3,2] ***r*** = 0.6 then ***R*** = [2,3]
...

9) The indices in ***R*** are transformed to time domain by multiplying by ***y***. The contiguous numbers are grouped together and stored in the output matrix ***O***.

If ***y***=1 then
***O* = [[2,3]]**

Fig. 4. Localization steps

3.6 Server

This component is a Python web server using Flask. The server retrieves to the browser a user interface where a video control is displayed with their controls, as well as an extra time bar and a file input control. Once a file is chosen the platform displays it locally in the web player and the user can proceed to send it. The interface shows a status bar of the current state of the file upload. Figure 5 shows some examples processed by the interface.

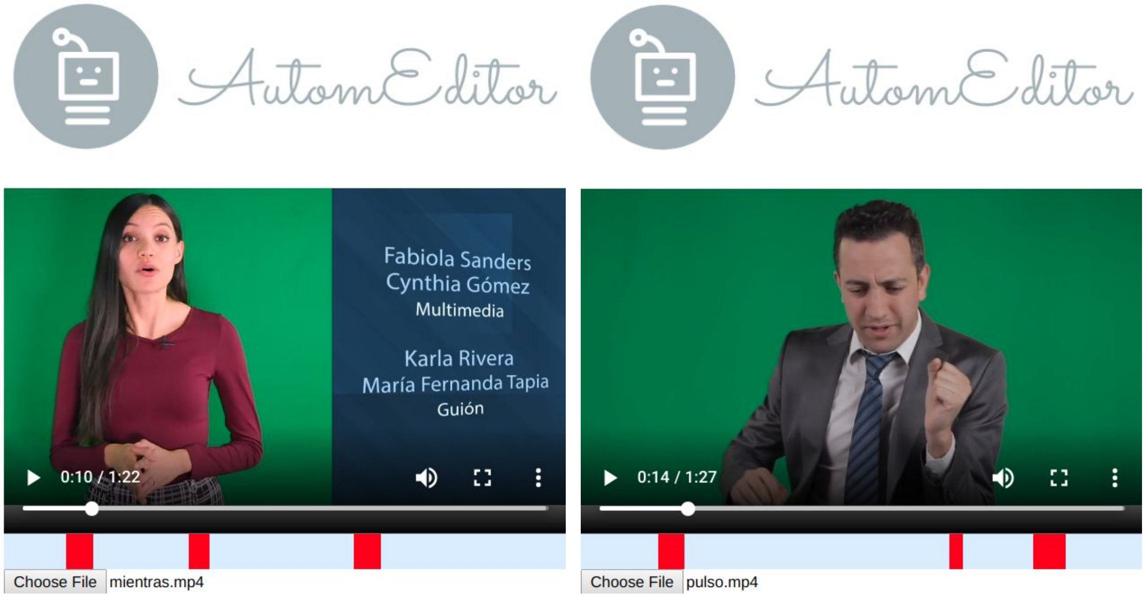


Fig. 5. Web interface

When a file was uploaded to the server via a multipart form-data HTTP request, it is stored temporally and renamed with a unique id (UUID) as the name of the file to prevent duplicates. The server implements size and length filters to prevent to exceed the processing and storage capabilities of the server. The server then invokes the Predictor followed by the Locator that gets the predicted sequence values. The Locator computes the blooper ranges and retrieves them in a JSON (JavaScript Object Notation) format.

Once the web client gets the serialized ranges, it deserializes and processes them to display in an extra timeline that is below the player. The timeline has the capability to navigate the video when is clicked on a blooper range. The tool is able to edit the video in case the user wants it.

4 EXPERIMENTS

We evaluated the blooper recognition and location modules of the tool.

4.1 Video blooper recognition

We trained and evaluated the multimodal network on Blooper DB. The model was trained for at most 300 epochs. To prevent overfitting, we applied an early-stopping policy with 20 epochs patience, which means to stop training after the validation loss does not drop for 20 epochs, and we deployed dropout strategy with ratio 0.5 for each fully connected layer. The learning rate was 1e-3.

A. Unimodal Approach

We first evaluated the performance of a model trained with a single modality. For each unimodal model, the same decision layer was deployed. For visual unimodal model, we investigated the effectiveness of VGG-face and OpenFace features separately in an ablation test. The comparison results are shown in Table I. Our results demonstrated that

VGG-face features outperformed OpenFace features under the same model architecture. The same behavior can be detected in the body features when comparing handcrafted and deep features.

Features	acc_val	acc_train	acc_test	f1_score	loss
Emotion General	0.59	0.86	0.59	0.60	0.28
Emotion Temporal	0.62	0.99	0.69	0.66	0.32
Body Handcrafted	0.63	0.92	0.54	0.72	0.27
Body Deep	0.68	0.99	0.65	0.72	0.22
Body Fusion	0.66	0.98	0.66	0.74	0.26
Face Handcrafted	0.84	0.99	0.87	0.89	0.12
Face Deep	0.89	1.00	0.81	0.92	0.12
Face Fusion	0.89	1.00	0.89	0.92	0.09
Audio Temporal	0.86	1.00	0.84	0.89	0.11
Audio General	0.95	1.00	0.90	0.96	0.03
Audio G. + Face F.	0.96	1.00	0.93	0.98	0.03
Quadmodal (All)	**1.00**	**1.00**	**0.90**	**1.00**	**0.01**

Table 1. Unimodal and multimodal results.

For the audio network, we focused on studying the importance of temporal modeling in video clips. We implemented another LSTM-based network for audio modality. Specifically, we divided each audio file into audio frames equally spaced in time and extracted openSMILE features for every single frame. Those features are then fed into a 64 cells LSTM layer followed by the decision layer. We compared this LSTM-based model with our audio unimodal model described. The results show the model without LSTM performs better than the audio model with LSTM. The LSTM layer does not benefit the estimation. Table 1 shows a comparison of all the evaluated models.

B. Multimodal Approach

We titled the fusion of all the modalities as the quadmodal network. The quadmodel contains the Face Handcrafted, Face Deep. Body Handcrafted, Body Deep, Audio General, and Emotion Temporal features. We trained the quadmodal network by using the concatenated multimodal features. Figure 6 shows the training performance metrics. With respect to fusion strategies, We compared the early and late feature fusion strategies in Table 2. The results demonstrated that learning benefits more from early fused representation. Table 1 shows the comparison of the unimodal and multimodal performances. We also compared the fusion of the top 3 accuracy unimodal models with the quadmodal, but quadmodal outperformed the model containing face and audio features only. The quadmodal model has better performance than any of the tested unimodal and multimodal models.

Then we evaluated the confusion matrices for the quadmodel to evaluate how was the performance per each category over the different sets. Figure 7 shows the confusion matrices of each set.

Features	acc_val	acc_train	acc_test	f1_score	loss
All Late Fusion	0.96	1.00	0.93	0.96	0.06
All Early Fusion	1.00	1.00	0.90	1.00	0.01

Table 2. Early versus late fusion.

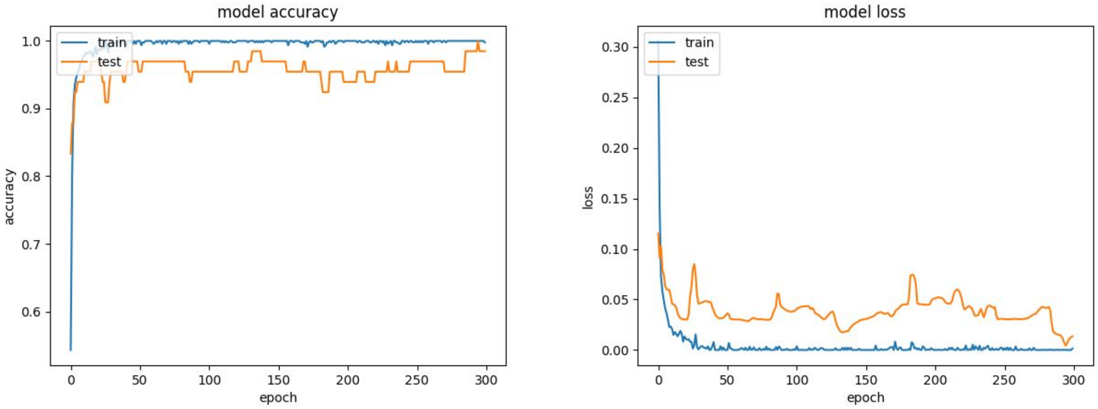


Fig. 6. Accuracy and validation plots

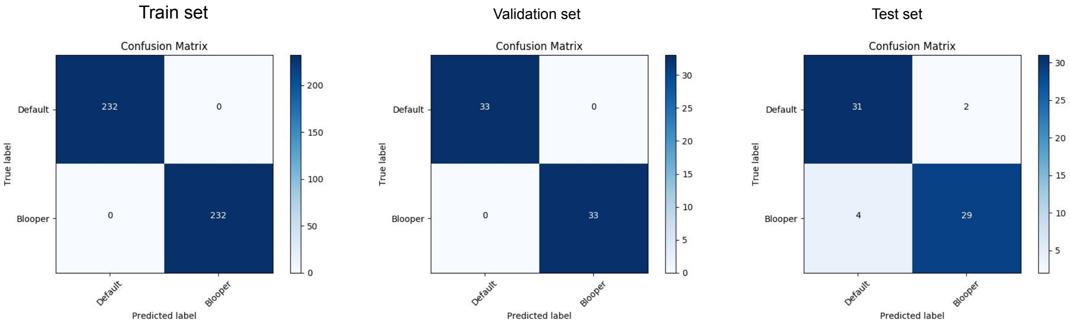


Fig. 7. Confusion matrices

4.2 Video blooper localization

In order to test if the localization algorithm was retrieving the correct time ranges we created two videos of 70 seconds length and inserted 6 videos (3 bloopers and 3 non-bloopers) from the test set in random positions as it is shown in Figure 8.

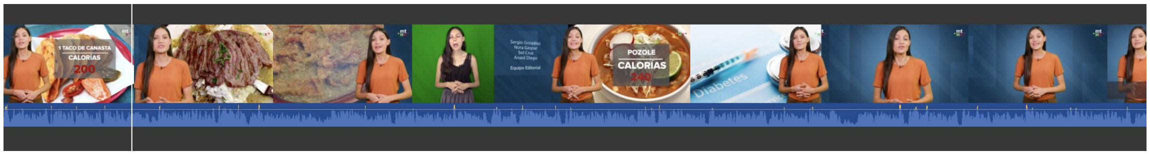


Fig. 8. Localization test videos

Then we ran the Locator to see how many ranges were detected and to verify if the bloopers were within the ranges. Figure 9 shows that the two videos retrieved three ranges each and these contained the video bloopers.

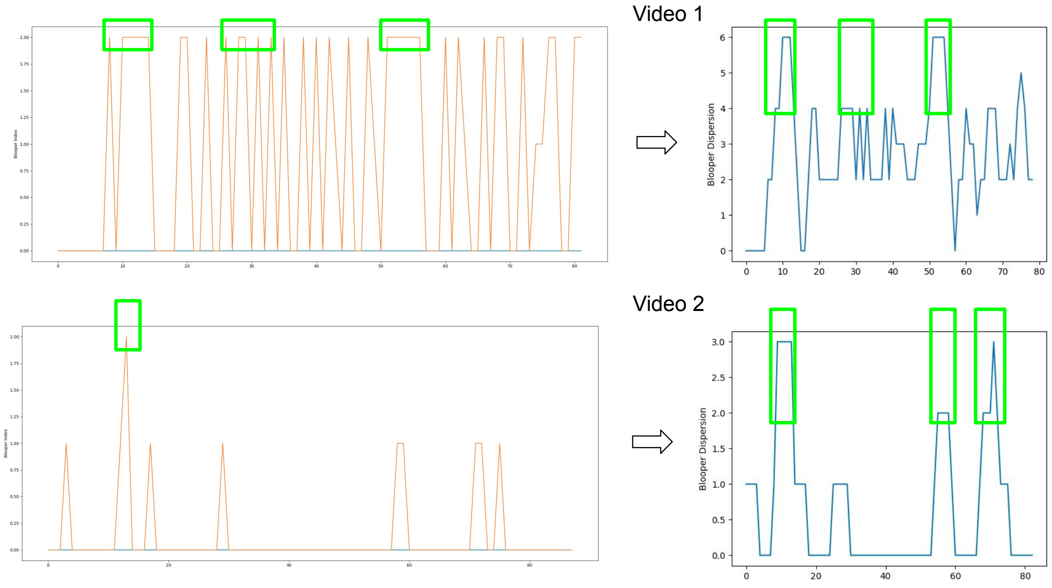


Fig. 9. Localization algorithm. The left plot shows the non grouped predictions. The right plot shows the condensed view.

5 LIMITATIONS AND FUTURE WORK

Our dataset is small in comparison with multi-action datasets that are designed for robust analysis. The limited amount of subjects makes it difficult to generalize to other people. Future work can explore the augmentation of the database by collecting more videos from people with different level of arousal. Artificial augmentation by generative techniques would be an interesting area of study.

The aim of this work was to show an end-to-end approach of a solution of this type. This work does not cover in depth all the blocks of the solution. Future work can explore one of these components in detail to understand why these performed in that way and to explore improvements.

Multimodal video action localization area can be benefited from having built-in mechanisms in the models to locate actions from mixed temporal and non-temporal features. The precision of this approach (500ms) is not enough for professional standards or for fully automated mechanisms. Future work can explore high precision localization methods for video bloopers.

The Human-Computer Interaction area can be benefited from exploring the expectations of users about automatic video interfaces. How novice and expert users want to be assisted in the editing process. Visualization only versus automatically applied editions would be an interesting question to have it solved before the massive implementation of AVEs.

6 CONCLUSION

In this paper, we present AutomEditor a system that is able to recognize and localize bloopers from monologue videos. To achieve that goal we created a dataset from online videos collected and annotated manually, we called it BlooperDB. We choose a multimodal approach to analyze features from face, body, audio, and emotions extracted from videos.

Early feature fusion strategy is deployed for combining the different modalities. Our multimodal models outperform the unimodal methods significantly. Our results show that multimodal information will greatly benefit the estimation of bloopers on videos. We also present a localization method based on the prediction sequence over sub clips that demonstrated to localize bloopers effectively. We deployed AutomEditor as a web application to help users to utilize this tool and to developers to easily deploy automatic video editing tools.

REFERENCES

[1] [n. d.]. oarriaga/face_classification: Real-time face detection and emotion/gender classification using fer2013/imdb datasets with a keras CNN model and openCV. https://github.com/oarriaga/face_classification. (Accessed on 04/28/2019).

[2] [n. d.]. OpenFace. https://cmusatyalab.github.io/openface/. (Accessed on 04/26/2019).

[3] [n. d.]. opensmile/emobase2010.conf at master Âů naxingyu/opensmile. https://github.com/naxingyu/opensmile/blob/master/config/emobase2010.conf. (Accessed on 04/26/2019).

[4] [n. d.]. petercunha/Emotion: Recognizes human faces and their corresponding emotions from a video or webcam feed. Powered by OpenCV and Deep Learning. https://github.com/petercunha/Emotion. (Accessed on 04/28/2019).

[5] [n. d.]. priya-dwivedi/face_and_emotion_detection. https://github.com/priya-dwivedi/face_and_emotion_detection. (Accessed on 04/28/2019).

[6] [n. d.]. thoughtworksarts/EmoPy: A deep neural net toolkit for emotion analysis via Facial Expression Recognition (FER). https://github.com/thoughtworksarts/EmoPy. (Accessed on 04/28/2019).

[7] Parvez Ahammad, Chuohao Yeo, Kannan Ramchandran, and S Shankar Sastry. 2011. High speed video action recognition and localization. US Patent 8,027,542.

[8] Marc Al-Hames, Benedikt Hörnler, Christoph Scheuermann, and Gerhard Rigoll. 2006. Using audio, visual, and lexical features in a multi-modal virtual meeting director. In *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 63–74.

[9] George Awad, Jonathan Fiscus, David Joy, Martial Michel, Alan Smeaton, Wessel Kraaij, Maria Eskevich, Robin Aly, Roeland Ordelman, Marc Ritter, et al. 2016. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *TREC Video Retrieval Evaluation (TRECVID)*.

[10] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 2012. 3D constrained local model for rigid and non-rigid facial tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2610–2617.

[11] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 2016. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1–10.

[12] Guoyong Cai and Binbin Xia. 2015. Convolutional neural networks for multimedia sentiment analysis. In *Natural Language Processing and Chinese Computing*. Springer, 159–167.

[13] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008* (2018).

[14] Xu Chen, Alfred O Hero III, and Silvio Savarese. 2012. Multimodal video indexing and retrieval using directed information. *IEEE Transactions on Multimedia* 14, 1 (2012), 3–16.

[15] Didan Deng, Yuqian Zhou, Jimin Pi, and Bertram E Shi. 2018. Multimodal Utterance-level Affect Analysis using Visual, Audio and Text Features. *arXiv preprint arXiv:1805.00625* (2018).

[16] Paul Ekman. 1978. Wallance. V. Friesen." Facial Action Coding System. *ConSultingPsychologists PreSSInc* (1978).

[17] Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 1459–1462.

[18] Michael Glodek, Stephan Reuter, Martin Schels, Klaus Dietmayer, and Friedhelm Schwenker. 2013. Kalman filter based classifier fusion for affective state recognition. In *International workshop on multiple classifier systems*. Springer, 85–94.

[19] Yihong Gong and Xin Liu. 2000. Video summarization using singular value decomposition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, Vol. 2. IEEE, 174–180.

[20] Benedikt Hornler, Dejan Arsic, Bjorn Schuller, and Gerhard Rigoll. 2009. Graphical models for multi-modal automatic video editing in meetings. In *2009 16th International Conference on Digital Signal Processing*. IEEE, 1–8.

[21] Alejandro Jaimes and Jun Miyazaki. 2005. Building a smart meeting room: from infrastructure to the video gap (research and open issues). In *21st International Conference on Data Engineering Workshops (ICDEW'05)*. IEEE, 1173–1173.

[22] Shian-Ru Ke, Hoang Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. 2013. A review on video-based human activity recognition. *Computers* 2, 2 (2013), 88–131.

[23] Loic Kessous, Ginevra Castellano, and George Caridakis. 2010. Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis. *Journal on Multimodal User Interfaces* 3, 1-2 (2010), 33–48.

[24] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman. 2010. Human focused action localization in video. In *European Conference on Computer Vision*. Springer, 219–233.

[25] An-An Liu, Ning Xu, Wei-Zhi Nie, Yu-Ting Su, Yongkang Wong, and Mohan Kankanhalli. 2017. Benchmarking a multimodal and multiview and interactive dataset for human action recognition. *IEEE Transactions on cybernetics* 47, 7 (2017), 1781–1794.

[26] Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. 2011. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*. ACM, 169–176.

[27] Jeho Nam and Ahmed H Tewfik. 1999. Dynamic video summarization and visualization. In *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*. ACM, 53–56.

[28] Nosa Omoigui, Liwei He, Anoop Gupta, Jonathan Grudin, and Elizabeth Sanocki. 1999. Time-compression: systems concerns, usage, and benefits. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 136–143.

[29] Xavier Orriols and Xavier Binefa. 2001. An EM algorithm for video summarization, generative model approach. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vol. 2. IEEE, 335–342.

[30] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. 2015. Deep face recognition.. In *bmvc*, Vol. 1. 6.

[31] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2539–2544.

[32] Soujanya Poria, Erik Cambria, Amir Hussain, and Guang-Bin Huang. 2015. Towards an intelligent framework for multimodal affective data analysis. *Neural Networks* 63 (2015), 104–116.

[33] Hiranmayi Ranganathan, Shayok Chakraborty, and Sethuraman Panchanathan. 2016. Multimodal emotion recognition using deep learning architectures. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1–9.

[34] Fabien Ringeval, Björn Schuller, Michel Valstar, Roddy Cowie, and Maja Pantic. 2018. Summary for AVEC 2018: Bipolar Disorder and Cross-Cultural Affect Recognition. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2111–2112.

[35] Verónica Pérez Rosas, Rada Mihalcea, and Louis-Philippe Morency. 2013. Multimodal sentiment analysis of spanish online videos. *IEEE Intelligent Systems* 28, 3 (2013), 38–45.

[36] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth S Narayanan. 2010. The INTERSPEECH 2010 paralinguistic challenge. In *Eleventh Annual Conference of the International Speech Communication Association*.

[37] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. 2017. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5734–5743.

[38] Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1049–1058.

[39] Michael A Smith and Takeo Kanade. 1997. Video skimming and characterization through the combination of image and language understanding techniques.

[40] Cees GM Snoek and Marcel Worring. 2005. Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications* 25, 1 (2005), 5–35.

[41] Mohammad Soleymani, Maja Pantic, and Thierry Pun. 2012. Multimodal emotion recognition in response to videos. *IEEE transactions on affective computing* 3, 2 (2012), 211–223.

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[43] Anthony Stefanidis, Panos Partsinevelos, Peggy Agouris, and Peter Doucette. 2000. Summarizing video datasets in the spatiotemporal domain. In *Proceedings 11th International Workshop on Database and Expert Systems Applications*. IEEE, 906–912.

[44] Min Sun, Ali Farhadi, and Steve Seitz. 2014. Ranking domain-specific highlights by analyzing edited videos. In *European conference on computer vision*. Springer, 787–802.

[45] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250* (2017).