Fitting video clips to music based on Mood Analysis of both video and audio

Yun

Abstract

The modern world sees popular use of videos paired with a background music track, on social media, organized events et cetera. Production and editing of such videos take time and effort. However, a large portion of which is spent on pairing video clips to audio clips, cutting video clips to the beat, and redoing the previous two. This paper aims to reduce time spent on the video assembly process by automating it, it aims to achieve this via a combination of video mood analysis (VMA) and audio mood analysis (AMA).

Audio mood analysis is performed with a convoluted neural network(CNN) that takes in the audio segments in the form of mel-spectrograms, subsequently, it finds slots(segments) for video insertion based on a beat/tempo detector.

Video mood analysis is performed first by splitting the video based on detected visual segmentations, and each clip is analyzed with a combination of face recognition (OpenCV), mood classification of facial expressions (resnet34), and VA calculation based on extracted video features.

Lastly, attempts to pair video clips with audio segments based on their computed mood, to produce a single video timeline (combined video clips/images).

This paper aims to leverage time from the video assembly process, so editors could use the time for creative work such as VFX or sound design, or simply to save time when in a pinch.

This article was rewritten in LaTeX on 20 Sep 2024 without modifications to content.

I. Introduction

FROM 2012 to 2020, YouTube saw its user grow from 0.8 billion to 2.3 billion. Undeniably, many humans consume a lot of media content every day, and many likewise take the job of creating media content. Multimedia plays a significant role in our modern 21st-century world, and video is a huge part of it.

Video editing is the manipulation and arrangement of video shots. It is used to structure and present all video information, including films and television shows, video advertisements, and video essays. But video editing is time-consuming.

Assembly is an early step of video editing, where the editor sets the in-out points of the source footage and places them in the video timeline. This creates a rough cut before finetuning. Assembly can be fast or time-consuming based on the amount of source footage because the editor has to consume the source footage, find suitable footage for insertion, and go through multiple stages of trial-and-error.

This paper tries to automate the process of video assembly by matching found video clips to audio clips by closest mood. With the main focus placed on a continuous audio track. In the following, I describe my approach.

II. Method

Both video and audio analysis are based on the dynamic valence-arousal model [1], assuming the 2 dimension number is enough to encapsulate the emotions presented. *(See appendix)*

Valence	Arousal
how pleasant an emotion is	how exciting and arousing an emotion is

A. Audio Analysis [2]

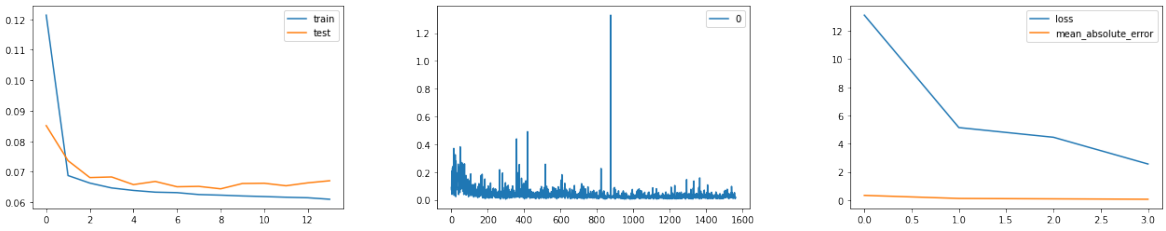
Every song is segmented into audio clips of fixed length for dynamic regression. Three methods were used and were trained on the 1000 song dataset or the PMEmo2019 dataset [3].

First method [4]: 88 features are extracted with **OpenSMILE** [5] for each 0.5s segment with eGeMAPSv02. They are used to form 10 samples groups (overlapping), normalized and fed into the CNN(input=(10,88)). In total, 38k groups of 10 samples is generated for training. Early stopping of 5 epoch without improvement is used, and after 13 epochs, the mean absolute error MAE has been reduced to 0.067.

Second method: Dividing audio spectrum data into chunks of 0.5s or 22050 samples (assuming 44.1k sample rate), and are feed into a **regression network**, with hidden nodes (1000, 500, 10). The model is very large and has a MAE of 0.15 after 1600 cycles.

Third method: use a **CRNN** (Convolutional Recurrent Neural Networks) with 4 time distributed convoluted layers of 32 filters & (32,1) kernel size, followed by two Gated Recurrent Unit (GRU) each with a 0.4 dropout. Finally a dense sigmoid layer with 2 outputs. 3000 audio spectrum points is feed into the AI with each time. This method produced a MAE of 0.10 after 4 epochs.

See appendix for CRNN model

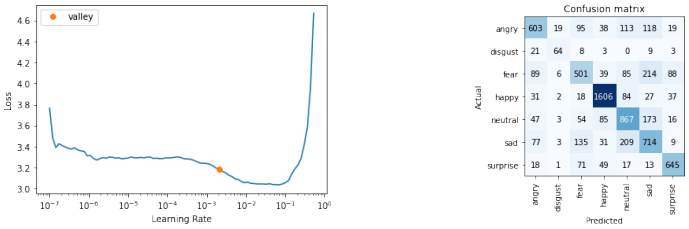


(a) CNN w/ OpenSmile (b) MLP Regressor (c) CRNN

B. Video Analysis

1) Facial Expression: A resnet34 model is trained on 36k images of expressions: angry, disgust, fear, happy, neutral, happy and surprise. Images are split into 80% for training and 20% for testing to prevent overfitting. Images are preprocessed with RandomResizedCrop and aug_transforms provided by fastai.

Optimal learning rate is discovered to be 0.002. The dataset used was found to be very dirty, manual cleaning was found to improve the model's accuracy. 5 cycles are trained for the headers first, and 5 more cycles after unfreezing the model.



(a) Learning Rate Discovery (b) Final Result

OpenCV is used to for face detection, and aforementioned model is used to determine the facial expression for the cropped face. The facial expression is then used to calculate the current frame's VA by averaging the corresponding VA value of each face.

(a) Demo (b) Arbitrary Mood Position

2) Static feature calculation: Done by calculating the hue of the dominant color, brightness lightness and chroma for each frame. These values will be used to calculate VA of each frame via an self-made arbitrary equation with parameters based on correlation values found by *How Color Properties Can Be Used to Elicit Emotions in Video Games* [6].

$$Valence = \arctan(-0.85 \times brightness + 0.21 \times hue - 0.85 \times chroma - 0.80 \times lightness)$$
$$Arousal = \arctan(+0.39 \times brightness - 0.04 \times hue - 0.43 \times chroma - 0.33 \times lightness)$$

This value is combined with the facial VA if faces were found.

C. Combination

The input videos are all processed into individual usable clips and cached. The input music is passed through beat detector to find cutting points, has its features extracted at .5s intervals.

The program loops through all the audio beat positions, and decide on the clip to insert.

- If clips are found with greater length than the audio beat gap, those clips are sorted by the Euclidean distance between their VA value and the audio gap's VA value.
  - An randomized bias is added to the sorting process to add variety
- If no such clips are found, the closest emotion clip will attempt to fuse with a clip neighboring itself, and jump back to the previous step.
- Timeline of clips are serialized with OpenTimelineIO, and exported to Final Cut Pro XML file.

Fig. 4: Examples images for static feature calculation

1) Image Montage: For images, each image is treated as an clip of arbitrary length, remaining operations are similar. Image's VA values are calculated as following. Hue used is the dominant color.

$$\text{Valence} = \arctan(A_v H_v - hue + B_v \times saturation + C_v \times chroma + D_v \times range + E_v \times variance)$$
$$\text{Arousal} = \arctan(A_a H_a - hue + B_a \times saturation + C_a \times chroma + D_a \times range + E_a \times variance)$$

H_a is red(0) and H_v is green (0.4). Other parameters are manual tuned to obtain the desired VA value range and spread for each batch of video clips/images.

III. Results & Discussion

The pipeline is successful in producing a usable project file that uses the source video clips. It can avoid showing cuts in clips (only cuts present are on sync) as long as the minimal clip length set was sufficient. It has achieved the baseline goal of making a usable montage/AMV.

The current method can determine the visual VA value accurately but is computationally expensive and time-costing to run on a massive amount of high-definition footage. Visual emotion detection is also a lot more complicated than just faces and hues, more data could be collected to suit this task directly, and more studies could be made on changing colors and effects on mood.

Audio emotion detection can produce a good result on paper but does not provide temporal consistency. This causes the VA value to fluctuate significantly when using different parts of the same sample. AI model also could've been trained on the spectrogram instead.

Major beats and minor beats are indifferent to the beat detector. It would be preferable to find the difference between them and put cuts on major beats. It may also be useful to analyze the song's lyrics as sad songs can be upbeat as well.

The final serialized output of the program is limited due to OpenTimelineIO's limited support of various functions, and a lack of documentation in pymiere. It also currently lacks a GUI for general users.

Since all video analysis data are cached, repeatedly running the program is cheap in terms of processing power and time, this can be used alongside clip-locking to find the desired output via trial and error.

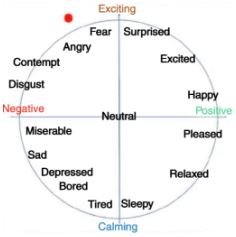
An external program is also required to render the video.

The demo videos implements **OpenSMILE+CNN** method for audio analysis and without facial recognition due to the lack of strong GPU. (See appendix)

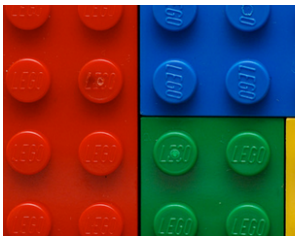
IV. Conclusion

This paper discussed and compared various methods of finding **valence&arousal** values for both video and audio data. It discussed a method of pairing audio clips to video clips based on their emotional value, focused on music-videos with continuous soundtrack. The ultimate idea is a autonomous music video generator. It has fulfilled the baseline goal of generating AMVs.

APPENDIX A
FIGURES



(a) VA model



(b) "Why not use the average?"

(c) "Why not use the average?"

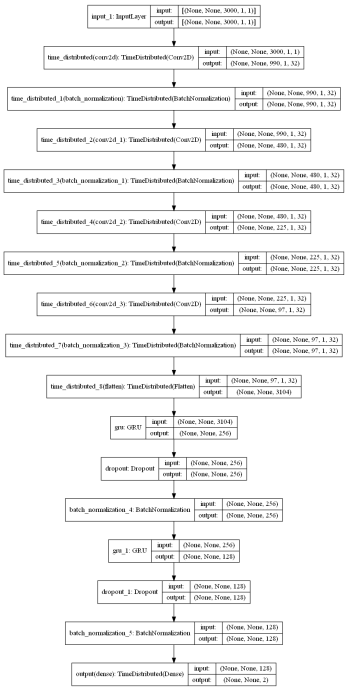
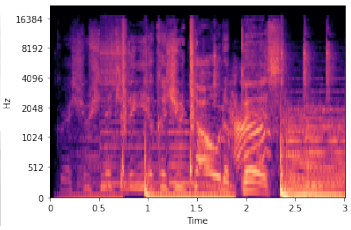
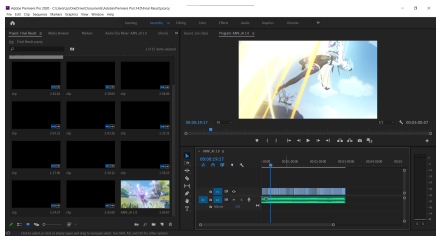


Fig. 6: Audio Audio CRNN Model w/ 2,830,978 trainable params

(a) Faces used



(b) Spectrogram of Audio Clip



(c) Generated Video Clip Sequence

APPENDIX B
EXPORTED DEMO

Videos can be found here: Youtube Playlist https://youtube.com/playlist?list=PLDnmTChqytlMiicOIJrj07ZQ7ya0pA5pM

ACKNOWLEDGMENT

The author would like to thank Da Mo Shu Shu and Genshin Impact for footages used the demo videos.

REFERENCES

[1] A. Toisoul, J. Kossaifi, A. Bulat, G. Tzimiropoulos, and M. Pantic, "Estimation of continuous valence and arousal levels from faces in naturalistic conditions," *Nature Machine Intelligence*, vol. 3, 01 2021.
[2] M. Soleymani, A. Aljanaki, y.-h. Yang, M. Caro, F. Eyben, K. Markov, B. Schuller, R. Veltkamp, F. Weninger, and F. Wiering, "Emotional analysis of music: A comparison of methods," 11 2014.
[3] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "The pmemo dataset for music emotion recognition," 06 2018, pp. 135–142.
[4] F. Weninger, F. Eyben, and B. Schuller, "The tum approach to the mediaeval music emotion task using generic affective audio features," *CEUR Workshop Proceedings*, vol. 1043, 01 2013.
[5] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1459–1462.
[6] E. Geslin, L. Jégou, and D. Beaudoin, "How color properties can be used to elicit emotions in video games," *International Journal of Computer Games Technology*, vol. 2016, no. 1, p. 5182768, 2016.