Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Predicting emotion in movies:

Recurrent and convolutional models applied to videos

Degree's Thesis
Audiovisual Systems Engineering

Author: Oriol Bernal Poch
Advisors: Xavier Giró-i-Nieto, Maia Zaharieva

Universitat Politècnica de Catalunya (UPC)
2016 - 2017

Abstract

This Thesis explores different approaches using deep learning techniques to predict emotions in videos.

Working with videos implies a huge amount of data including visual frames and acoustic samples. The first step of the project is basically to extract features to represent the videos in small sets of arrays. This procedure is done using pre-trained models based on Convolutional Networks, the state of the art in visual recognition. Firstly, visual features are extracted using 3D convolutions and acoustic features are extracted using VGG19, a pre-trained convolutional model for images fine-tuned to accept the audio inputs.

Later, these features are fed into a Recurrent model capable of exploiting the temporal information.

Emotions are measured in terms of valence and arousal, values between [-1, 1]. Additionally, the same techniques are also used to attempt to predict fear scenes. In consequence, this thesis deals with both regression and classification problems.

Several architectures and different parameters have been tested in order to achieve the best performance. Finally, the results will be published in the MediaEval 2017 Challenge and compared to the state-of-the-art solutions.

Resum

Aquesta tesi explora diferents tècniques d'aprenentatge profund (deep learning) amb la finalitat de predir emocions en vídeos.

Treballar amb vídeos implica grans quantitats de dades incloent tant els frames dels vídeos com les mostres dels àudios. El primer pas del projecte és, bàsicament, extreure característiques per representar els vídeos en petits grups de vectors. Aquest procediment es duu a terme mitjançant models pre-entrenats basats en Xarxes Convolucionals, models punters en la detecció i reconeixement d'objectes. Primer les característiques visuals s'extreuen utilitzant convolucions 3D i les característiques acústiques utilitzant VGG19, un model convolucional pre-entrenat per imatges i tunejat per suportar entrades d'àudio.

Després, aquestes característiques alimentaran un model recurrent que serà capaç d'explotar la informació temporal.

Les emocions són mesurades en termes de valence i arousal, valors compresos entre [-1, 1]. A més a més, de forma addicional, també s'utilitzaran les mateixes tècniques per intentar predir escenes de por. Per aquesta raó, el present treball tracta amb problemes de regressió i classificació.

Diverses arquitectures i diferents paràmetres han sigut provats amb la finalitat d'aconseguir el millor model. Finalment, els resultats seran publicats al repte de MediaEval 2017 i seran comparats amb les solucions dels últims models amb millor rendiment.

Resumen

Esta tesis explora diferentes enfoques usando técnicas de aprendizaje profundo (deep learning) con el fin de predecir emociones en videos.

Trabajar con videos implica grandes cantidades de datos incluyendo tanto los frames de los videos como las muestras de audio. El primer paso del proyecto es, básicamente, extraer características para representar dichos videos en pequeños grupos de vectores. Este procedimiento se lleva a cabo usando modelos pre-entrenados basados en Redes Convolucionales, modelos de vanguardia en el reconocimiento visual. Primero las características visuales se extraen usando convoluciones 3D y las características acústicas usando VGG19, un modelo convolucional pre-entrenado para imágenes y tuneado para soportar como entrada los audios.

Después, estas características alimentarán un modelo recurrente que será capaz de explotar la información temporal.

Las emociones son medidas en términos de valence y arousal, valores comprendidos entre [-1, 1]. Además, de forma adicional, también se usarán las mismas técnicas para intentar predecir escenas de miedo. Por esta razón, la presente tesis trata con problemas de regresión y clasificación.

Varias arquitecturas y diferentes parámetros han sido probados con el fin de conseguir el mejor modelo. Finalmente, los resultados se publicarán en el reto de MediaEval 2017 y serán comparados con las soluciones de los últimos modelos de hoy en día.

Acknowledgements

First of all, I want to thank my tutors, Xavier Giro-i-Nieto and Maia Zaharieva, for helping me every time I needed and making my collaboration in this work possible. I also appreciate their patience for teaching me and advising me week after week.

I would also like to thank Alberto Montes for his advices and his work on Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks. This project wouldn't have been possible without it.

Revision history and approval record

Revision	Date	Purpose
0	05/06/2017	Document creation
1	30/06/2017	Document revision
3	01/07/2017	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Oriol Bernal Poch	oriol.bernal@alu-etsetb.upc.edu
Xavier Giró i Nieto	xavier.giro@upc.edu
Maia Zaharieva	maia.zaharieva@tuwien.ac.at

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	8/07/2015	**Date**	10/07/2015	**Date**	10/07/2015
Name	Oriol Bernal Poch	**Name**	Xavier Giró i Nieto	**Name**	Maia Zaharieva
Position	Project Author	**Position**	Project Supervisor	**Position**	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of purpose

We are continuously consuming audiovisual content. This content is now part of our daily life and therefore, every year more and more of this content is created. Video indexing, recommendations, and summarization are examples of tasks that required people working on them to get reliable results. Nowadays, because of this huge volume of data, new algorithms are being used to help humans to achieve this task.

Analysing the emotions that a video can elicit to people is an interesting task to extract the content-related information from audiovisual signals. With this information better summaries and overviews can be achieved. Algorithms capable of locating unpleasant scenes for the children such as fearful or violent are needed. Another trend fashion is to recommend the content to individual users based on the content they have consumed before, and using the affective information appears to be the right way to go.

The elicited emotions in humans are time dependent. This means that what we feel is related with what we have felt in the previous minutes. As a clear example, it is not the same to watch a funny movie scene after a scary one than watching the same funny scene after another funny scene.

This thesis studies the problem of emotion classification in videos by using convolutional neural networks and recurrent neural networks to exploit both spatial and temporal information. As inputs, not only the visual frames of the video are used, but also the audio associated with the video is processed and fed into the system.

The project has the next objectives:

- Explore a dataset and learn to deal with the main problems of deep learning.
- Train a model to predict emotions in videos.
- Trace strategies to improve the performance of the model.
- Compare the model to the state-of-the-art.

This project will exploit the emotional content of videos and will participate in an Open Research Workshop (carried by MediaEval) participating in the shared-learning with the MediaEval community.

1.2 Requirements and specifications

The MediaEval 2017 Emotional Impact of Movies Task had a few requirements that have been added to the project.

The requirements of the project are the followings:

- Use deep learning techniques to process videos and understand human emotions.
- In solving the task, participants are expected to exploit the provided resources. Use of external resources (e.g. Internet data) will be however allowed as specific runs.
- This task allows to each participant to submit a few runs for each subtask.
    - Valence/Arousal predictions: 5 runs can be submitted up.
    - Fear predictions: 5 runs can be submitted up.
- Long movies are considered and the emotional impact must be predicted for consecutive 10 seconds sliding over the whole movie with a shift of 5 seconds.

The specifications of the project are the followings:

- Use of python as the main coding language and use of Keras[1] as the main framework to develop deep learning.
- The dataset[2] consist on 30 professionally made amateur movies that are provided with the continuous annotations according to fear, valence and arousal for consecutive 10 seconds with a shift of 5 seconds.
- 10 additional movies will be provided later as the test set.
- The evaluation criteria consist on the Mean Square Error (MSE) and the Pearson's Correlation Coefficient when predicting valence and arousal, and the MAP criteria when predicting fear scenes.

1.3 Methods and procedures

This project deals both with audio and video and it is mainly divided into two parts: features extraction and emotion classification. To extract features two different networks are used, one for the visual features and the other for the acoustic features.

To classify this features LSTMs are used. With this kind of Networks, temporal information can be exploited in order to predict the emotions considering the past saw by the network. This project has studied different ways to feed the LSTM Network, using only acoustic features, only visual features, fusing both features by concatenation or even using the raw audio samples.

[1] https://keras.io/

[2] http://liris-accede.ec-lyon.fr/

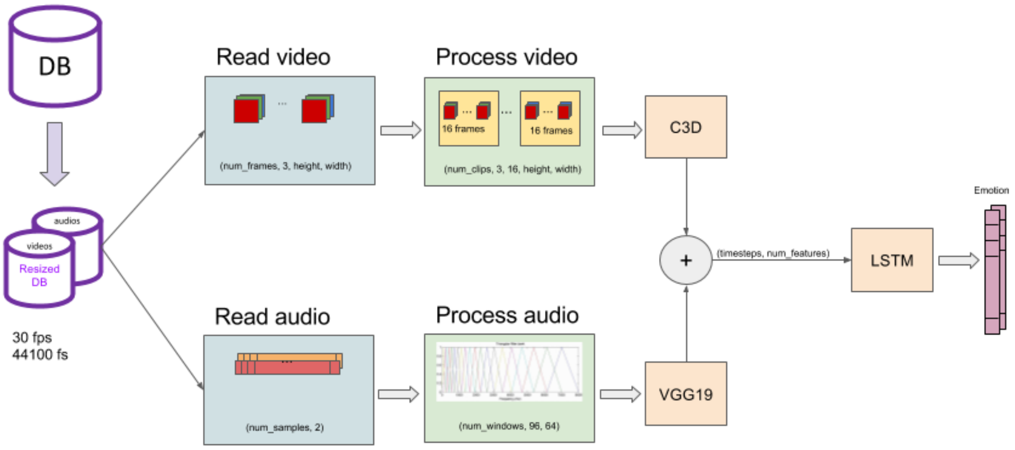


Figure 1.1: Global architecture of the solution presented in the Thesis

To extract visual features a C3D network used to train Sports1M is used. This pre-trained model extract features every 16 frames. To avoid problems, we coded all the videos of our dataset into 30 fps.

To extract acoustic features VGG19 is used. This network is a Convolutional Networks and it was designed for the ImageNet Challenge (taken the first and second places in localisation and classification tasks respectively). Applying CNNs architectures to audio have shown promising results in their ability to classify videos. The input of the network are the Mel-scaled frequencies of the windowed signal, making sure our dataset was coded in 44100 fs and using window sizes of 23520 samples.

As it will be explained in the methodology, the fed inputs that achieved the best performance were inputs made of acoustic and visual features. The main structure of the project can be seen in the Figure 1.1.

1.4 Work Plan

This project has followed the established work plan, with a few exceptions and modifications explained in the section 1.5.

1.4.1 Work Packages

- WP 1: Documentation.
- WP 2: State-of-the-art
- WP 3: Design thinking

- WP 4: Coding
- WP 5: Testing
- WP 6: MediaEval submissions
- WP 7: Oral presentation

1.4.2 Gantt Diagram

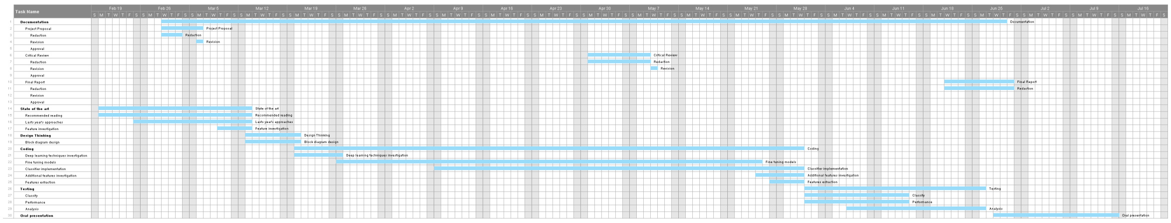


Figure 1.2: Gantt Diagram of the Degree Thesis

1.5 Incidents and Modification

During the project some packages were modified in order to keep the timetable. At the beginning we thought that would be interesting to solve the task with some machine learning old techniques such as SVM. This was an interesting thing to do in order to have reference numbers to outperform, to play with the database and with their features, and to demonstrate the power of deep-learning. However, due to some installation incidences in the beginning of the project that delayed the project per se, we decided to avoid this work package and go directly to the deep learning.

Another modification that we made was to make a major focus on the fine tuning part of the project. This consist on using other approaches and adapt them to your work so they can solve the task. Fine tuning is a very powerful tool and therefore, we decided to use it not only with the visual data, but also with the acoustic data.

Chapter 2

State of the art

2.1 Deep-learning

Deep learning is a new area of Machine learning that can exploit data in high-level terms. It is based on a biological model of the brain proposed by Nobel laureates Hubel and Wiesel in 1959. It is formed by layers and neurones that learn from the lower layers to the higher ones with the ability to extract abstract concepts.

Sometimes it can be very difficult to extract high-level abstract features from raw data. Deep-learning techniques solve this problem by expressing these sophisticated data in terms of simpler representations (it builds complex concepts based on the simpler ones, such us the contours or the edges of an image). This behaviour can be seen in Figure 2.1.

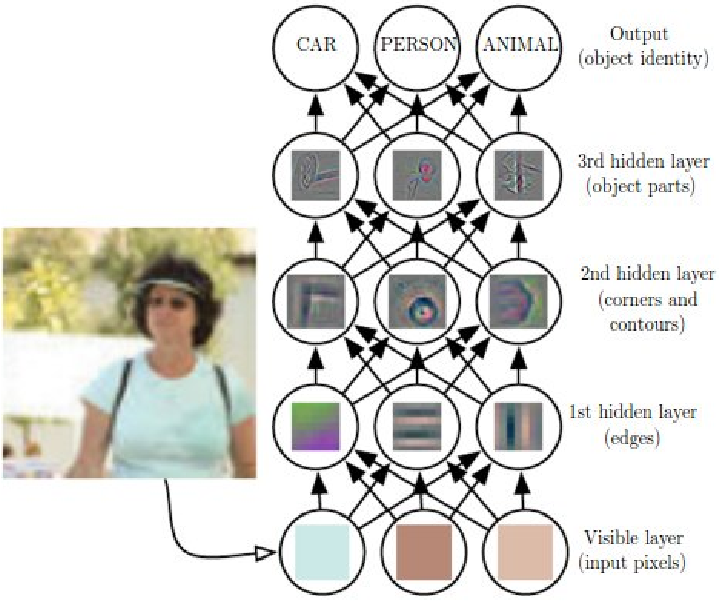


Figure 2.1: Understanding of the presence of a person in an image by computing, first, low-level features, and then more complex features based on the lower ones, as it is explained in [5]

Due to the computational advances and the big amount of available data today, big models with high accuracy and complexity can be built. One of the biggest points of deep-learning that differentiate it from machine-learning algorithms is that the more date they see, the better performance they can achieve. Old machine-learning algorithms, reach a plateau in performance at a certain amount of data.

In addition, Deep-learning models can have the ability to perform automatic feature extraction from raw data (also called feature learning). This tool is very powerful since these deep-models are supposed to learn the features that contain the most discriminant information to solve the specific task. In this project, some well known deep architectures will be used to solve the problem of recognising emotions.

2.2 Emotion in videos

Humans are very complex and emotions are a part of us that plays an important role in our daily decisions. However, can a machine predict those irrational things? With the pass of the years, more and more improves in Artificial Intelligence and Machine Learning have been achieved. Recently, with the new deep learning techniques, a lot of people have been working on predicting emotional or other irrational aspects of the human being such as Interestingness, Popularity, Memorability, etc.

Nevertheless, the emotions we feel are due to the things we are surrounded by, by the things we see or hear, and by how we perceive these things. When we watch a film, we can feel one emotion or another depending on the music, the colours of the scene, its motion, etc. However, the way we perceive the things is very complex and it is related with how we understand those things (the contextual information or the context).

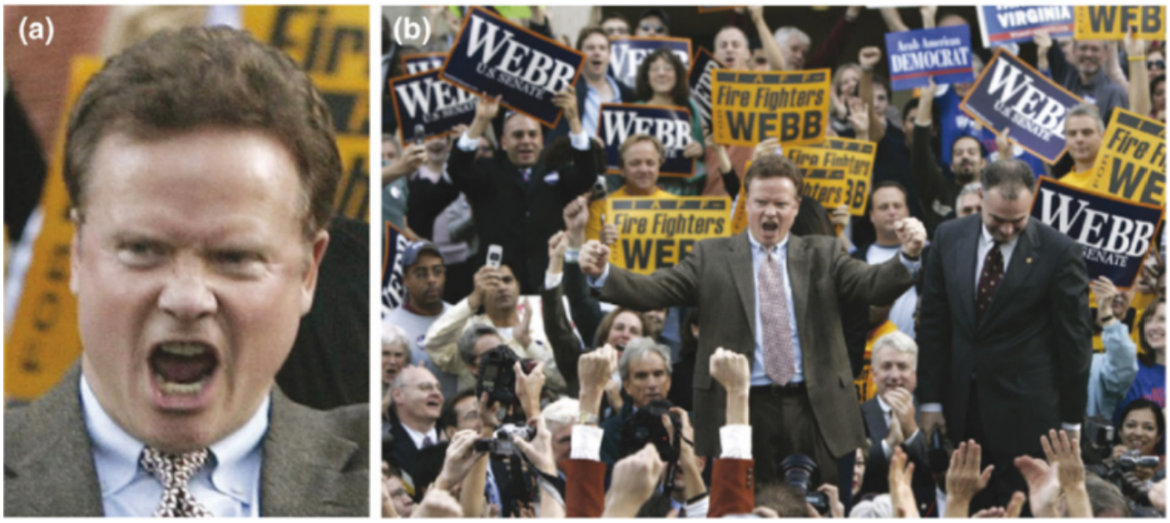


Figure 2.2: The role of context in emotion perception. When we watch a) we can think about an angry Senator Jim Webb, but when we watch at b) he appears to be more happy and excited rather than angry

Figure 2.2 shows, (as it is explained in [17]) a clear example about that. This figure shows how emotions are influenced by not only what we see or hear but also by the context of it.

Emotions are subjective and difficult to measure. Their magnitude can be measured to a certain degree by monitoring your physical response (brainwaves for example) but, in general, it is difficult to tell the difference between one emotion or another. One way to monitor the emotions is considering the valence and arousal scores, introduced by Feldman in 1995. The arousal is related to the level or amount of physical response (intensity) while the valence measures the emotional direction (good or bad emotions). This can be seen in Figure 2.3, where the two axes representing valence and arousal are plotted with some emotions inside it. The valence scores can distinguish between annoyed (emotion we would consider negative) and content (positive) for example. Arousal, instead. can distinguish between furious or exited (intense emotions) and relaxed or bored (weak emotions).

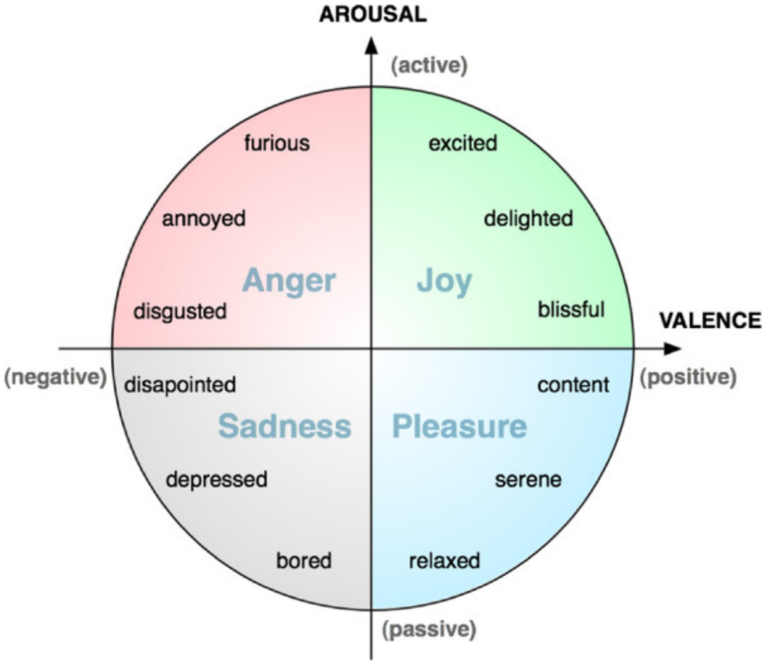


Figure 2.3: Emotions represented as terms of Valence and Arousal

2.3 Other approaches

The problem of predicting emotions have been solved with different techniques in the recent yeas. These solutions include not only deep-learning but also typical machine learning techniques. In the last past years, The MediaEval benchmark has opened this problem for the scientist to compare the state-of-the-art solutions.

Methods using variants of well-known classifiers, such as Bayesian networks, support vector machines, neural networks, and (hierarchical) hidden Markov models (HMMs), have been employed widely in the field of cognitive content analysis for object, person, scene, and event detection and classification [6]. The main problem lies in the fact that the variety of content that can appear each emotion (happy, sad, or even bored or relaxed) is practically unlimited. We can describe a goal scored in an important football match as exciting, but a parachute jump or even a simple conversation in a movie can also be labelled as exciting.

2.3.1 Features and Modalities for affective content

Trying to extract the mood in videos can be a difficult task since a lot of data has to be considered. The emotional information does not only resides in the video itself (sequence of frames) but also in the audio, for example. Motion is also an interesting modality to process because the difference between the shots in a film can evoke to the viewer different emotions. Those modalities can be processed by extracting features of them. The number of features that can be extracted from these modalities is very high and depend on the task itself. It is known that the main colours of a scene can evoke one emotion or another, but also the presence of a face and its face-expression or the pitch and tone of a speaker can participate in our way of perceiving the emotions. [7]

Colour: The colour dominance in a scene and its influence in art images on the human affective state has been investigated [3]. In general, although the humans' emotions are very complex, sometimes simple features such as the main colour can be decisive in our perception. For example, the colour red is assumed to communicate happiness, dynamism, and power. Orange is thought to resemble glory; green should elicit calmness and relaxation; and blue may suggest gentleness, fairness, faithfulness, and virtue. Purple, on the other hand, sometimes communicates fear, while brown is often used as the background colour for generating relaxing scenes. Colours and their combinations in scenes can be a relevant information for emotion description.

Motion: Experts have shown that motion can be a decisive psycho-physiological when evoking strong emotional responses in viewers. The directors of films usually use short shots to evoke stress to the viewer or to accent moments. Larger shots, in contrast, are typically used to deaccentuate an action.

Audio: Audio is also a very important modality which is decisive in our emotion perception. The changes in affect intensity seem to be correlated with pitch range, loudness, spectral energy in higher frequencies, and speech rate (e.g., faster for fear or joy and slower for disgust or romance). However, other speech features such us inflection, rhythm, duration of the last syllable of a sentence (short in anger and long in joy and tenderness), and voice quality (more resonant for joy and tenderness, and breathy for anger and sorrow) have shown a high correlation with the evoked emotion in the listener. Besides, this research about the study of audio features related with emotions is still far to be finished.

2.3.2 Last year approaches

As this problem has been organised by MediaEval in the past years, the state-of-the-art solutions from the last year approaches have been studied.

The different techniques used are very different from each other. Some teams decided to extract features manually and then connect those features to machine-learning models such as Support Vector Machines, Random Forest, or linear regression. All the teams used different modalities from the video (Sequence of frames), Audio, Motion, or a combination of those. Others, used more modern techniques such as Long Short Term Memory Networks (LSTM).

In [1] they extract local features (from patches of the frames or overlapping windows of the sound) and global features (from the entire frame of the video). Then those features are normalised using well-known techniques like Bag-of-Words or Fisher Vectors. Finally, the features are combined to feed to a Support Vector Regression.

In [4] the three Modalities are combined and the features extracted are fed in a Support Vector Regression and to a Random Forest. They studied which features were the most useful to predict emotions.

In [11] they downsample the raw video and create an array of features with it. Then they used Neural Networks and Support Vector Regression to compare the results. In addition, some dimensionality reduction techniques are applied to decrease the number of features before training the model. They show how in the dimensionality reduction, the learned subspace keeps the discriminant information giving good results.

In [12] They combine audio features with image features extracted from a Convolutional Neural Network and compare the performance of LSTM and BLSTM with different layers and configurations.

Finally, in [9] they extract audio features and make a Linear Regression studying possible smoothing techniques to remove the high-frequency noise irrelevant to the emotion information. However, their final results showed poor performance in predicting the Valence.

Chapter 3

Methodology

This chapter describes the pipeline of the project and all the followed steps done in order to train the final network that predicts emotions.

3.1 Objective

The main objective of this project is to build a model using deep learning able to solve the problems described. Because of the main nature of emotions and how they flow through time, the model has to exploit the temporal information of the videos to make the predictions. The main architecture will be composed of two networks, one for predicting fear scenes (classification problem) and another to predict emotion scores (regression problem), but both of them will follow a similar structure.

3.2 The dataset

The data used for training the model (supervised learning) consist of a set of films with their specific scores for valence, arousal and fear every 5 seconds. This dataset named Liris-Accede (as it is explained in [2]), is provided by the of the MediaEval task organisers. It is a dataset made of long videos and specially created to draw on the affective content of those films. The development set is composed of 30 films, while the test set (the one used for releasing the final results) is composed of 10 films.

Before starting to work on the models that make predictions, it is always recommended to understand and be aware of what kind of data are we working with. A previous stage analysing the dataset is always a good manner to start the project.

Looking to our ground truth data is interesting to see the nature of those scores and what can we do or not do with it. Figures 3.1 and 3.2 show how these labels vary in time for a specific film (After the Rain). As we can see, valence and arousal are continuous scores which can flow between -1 and 1. The fear, instead, is a binary score, where 0 means that those 5 seconds of the film do not induce fear to the viewer, while 1 means the opposite.

If we want to predict the valence and arousal scores we are dealing with a regression problem. This means that when predicting, our model will try to find out the best relationship that represents the dataset. However, if we want to predict the fear scenes we are dealing with a classification problem. The model will try to predict and classify the video scenes into 2 classes.

As we can see in Figures 3.3 and 3.4. The provided dataset is composed of movies which duration can vary between 200 seconds and 1 hour and a half, and these movies have been coded in a different number of frames per second, 24, 25 and 30 fps.

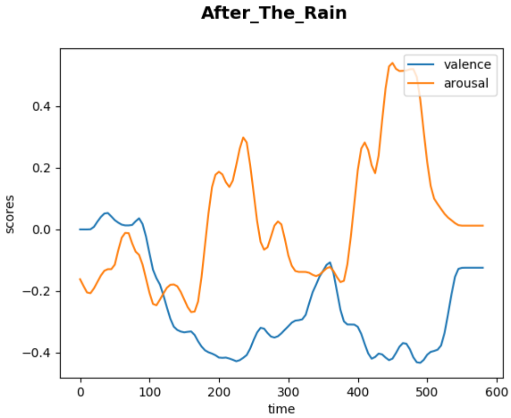


Figure 3.1: How the valence and arousal scores flow during the film After the Rain.mp4

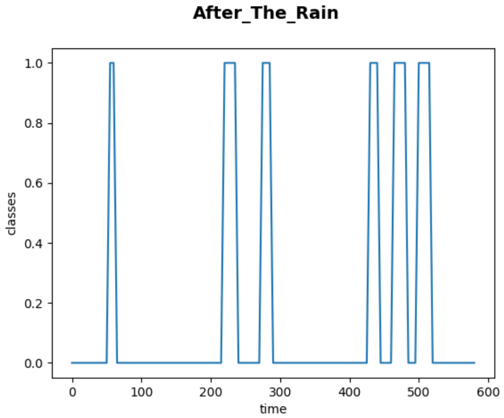


Figure 3.2: Fear scenes during the film After the Rain.mp4.

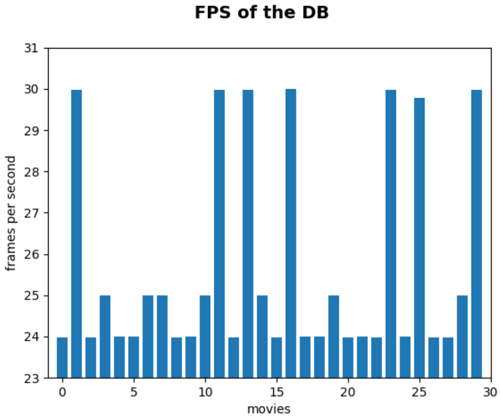


Figure 3.3: Bar chart of the number of frame per seconds of each of the movie

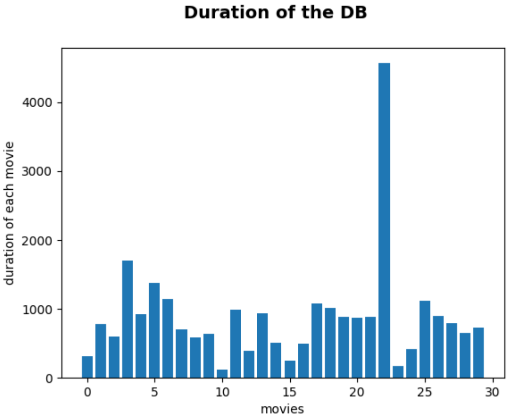


Figure 3.4: Bar chart of the durations of all the movies in the train set.

The Figure 3.5 shows the distribution of these ground truth data along the 30 movies of the development set. As we can see, the highest and lowest values of valence and arousal tend to appear less than the neutral ones. This has sense because not all the scenes of a movie evoke intense feelings to the viewers, but just a few of them. In addition, respect to the fear histogram we can see how unbalanced our dataset is. This means that we have many more samples from one class (non-fear scenes) than from the other (fear scenes). This is an event that must be taken into account when training.

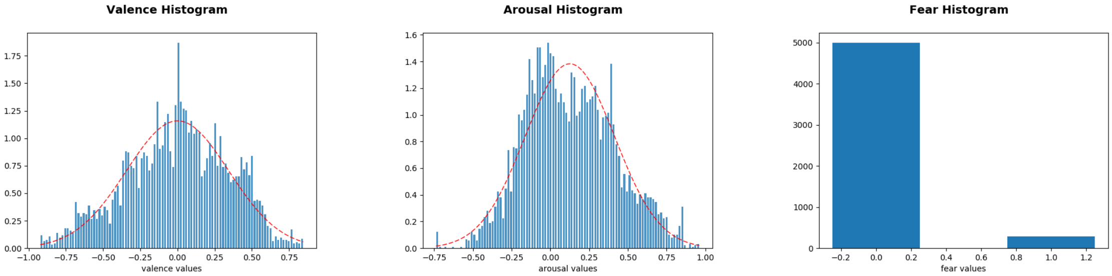


Figure 3.5: Valence, Arousal and Fear histograms

3.3 Deep learning techniques

In this project, several deep-learning techniques have been used in order to solve the problem. As it has been explained in Section 2.1, deep-learning are modern useful techniques to exploit complex and abstract structures of a dataset. In this project, some deep architectures such as Convolutional Neural Networks and Recurrent Neural Networks are used to extract spatiotemporal information along the different films of the dataset. These techniques have shown promising results in the last years in similar tasks and are the state-of-the-art in several fields.

3.3.1 Convolutional Neural Networks

CNNs are a type of Neural Networks that can exploit spatial information. These networks use convolutions to take advantage of the 2D structure of the input and have been proved to achieve great results in fields such as object recognition and detection.

CNNs can have multiple convolutional layers with its subsequent subsampling layers optionally followed by some fully connected layers. The convolutional layers have K filters or kernels that produce K feature maps. Then the pooling layer is the one in charge of reducing these feature maps. The following fully-connected layers give the network the ability to predict something.

Convolution results always depend on the filter that has been used, but in practice, these Networks are supposed to learn the value of these filters during its training. In general, the more number of filters we have, the more image features we extract and the better our network becomes.

Spatial Pooling (also called downsampling or subsampling) reduces the dimensionality of each feature map but retains the most important information. Thanks to this function, we can reduce the feature dimensions reducing the number of parameters and computations in the network. It also makes the network invariant to small transformations or distortions.

Convolution and pooling act as feature extractor from the input image while Fully connected layer acts as a classifier.

3.3.2 Recurrent Neural Neyworks

Recurrent Neural Networks are a type of neural network that handles with time-dependent sequences. In a traditional Neural Network, all inputs are assumed to be independent of each other. However, in some task, like predicting the next word in a sentence, taking into account the previous samples to predict can increase considerably the final scores. This can be understood as networks with an “internal memory” that is able to store and compute the information that has been captured so far.

Figure 3.6 shows an unfolded or unrolled RNN. In this figure X correspond to the inputs of the Network. For example, if we were trying to predict the next word in a sentence, X0 would be an input related to the first word of the sentence, X1 with the second word, and so on. O is the output of the Network. This output, such as the input, is a vector and each sample corresponds to one timestep. If we continue with the same example, O1 would be the predictions of the

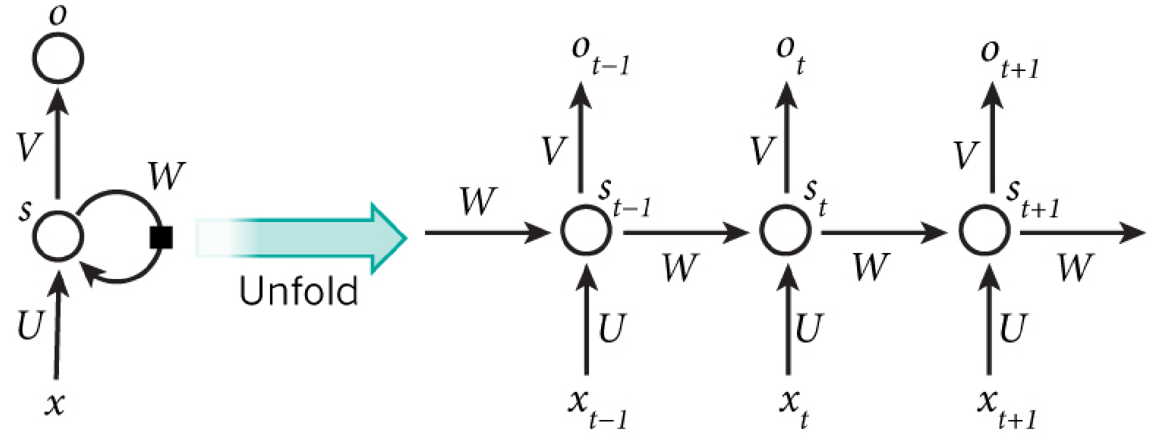


Figure 3.6: Unrolled RNN

second word of the sentence taken into account I0, O2 would be the predictions of the third word considering I1 but also I0. S are referred to the hidden states and the sub-index t is the index for the timestep. RNNs have achieved excellent results in tasks such as speech recognition, translation, language modelling, etc.

3.3.3 Long Short Term Memory Networks

RNNs are networks that predict using past information, but how long can this time dependency be? Dealing with long dependencies can be difficult in computational terms since it can be very expensive and therefore, it can increase the training time considerably.

Long Short Term Memory Networks are a type of RNN designed to solve this problem and deal with long-term dependencies. This is done thanks to some internal gates that let information through or not.

The core of LSTMs is the called cell state. This is the information that flows between the unfolded Network. This cell state contains the main information, and with the help of some gates, LSTMs are capable of exploiting the time when predicting.

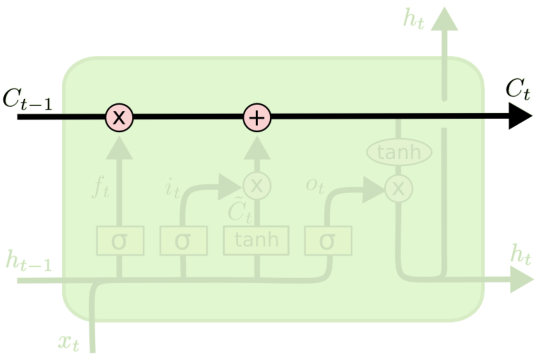


Figure 3.7: Cell sate flowing in a LSTM

Simplifying, there are the three main gates that, basically, transform this information that flows in the cell state so the final output considered the time. This idea is the one that makes LSTM capable of exploiting longer timesteps.

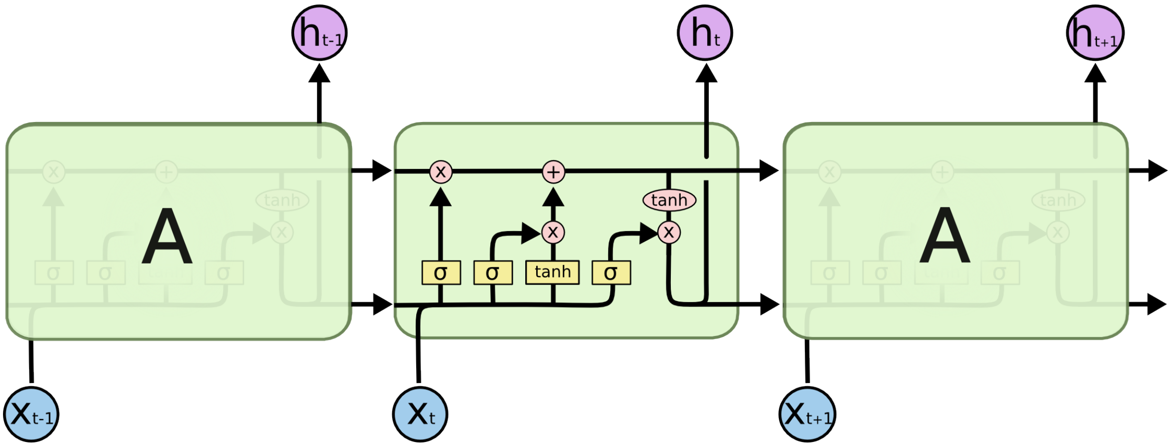


Figure 3.8: How gates modify the Cell State

The forget gate layer decides how much information is the network going to let through. Input gate layer decides which new information is the network going to consider. The third gate computes the output based on the cell state.

3.4 Baseline

The main baseline of the project is made of 3 main steps: pre-processing, feature extraction, and regression/classification.

The pre-processing step consists of extracting the audio from the video, coding both, the video frames and audio samples in the right rates (frames per second and sampling frequency) and processing them to have the correct shapes before feeding the feature extractors.

Feature extraction is done by two networks, one for video frames and another for the audio samples. The visual features extraction has been done using 3D convolutions with a deep network called C3D. This was chosen as feature extractor due to the good results achieved in other works [13, 16]. The acoustic features extracted with the audio samples have been done using VGG19, a convolutional network trained with images. This is done by pre-processing the audio input first in order to be accepted by the network. VGG19 is one of the possible solutions for fine-tuning convolutional networks with audio proposed by [8]

Finally, the regression and classification steps are achieved using an LSTM, a recurrent neural network which is able to exploit long-term temporal information.

With this architecture, we expect to exploit spatiotemporal information to predict the affective content of the movies. While the C3D exploits short-time and spatial information of the frames, the LSTM is able to exploit large temporal information. C3D features are merged with the VGG19 ones in order to predict using information from audio and video at the same time.

3.4.1 Preparing the data

The C3D network used to extract features deals with clips of 16 frames long. This means that the input of the network are groups of 16 frames as it will be explained in the next section (3.4.2).

One of the main objectives of the project is to send the predicted values to the MediaEval task organisers in order to participate in the workshop. For doing that, it is required to send a prediction every 10 seconds with a shift of 5 seconds.

As it has been explained in section 3.2 our dataset is not regular and the movies are coded with different frames per second rates. When we window the signal in groups of 16 frames we are obtaining different windows with different time length depending on those rates. Therefore, when we compute the features and later we predict the emotions, we would have a different lengths depending on the rate of the video.

To solve this problem, we decided to convert first all our dataset into videos of 30 fps and dimensions of 112x112 of width and height. This is done using open-cv and ffmpeg over python.

Our audios extracted from the videos are all coded with the same sampling frequency rate: 44100. However, we decided to prepare some code to change to that rate in case we used another dataset in the future.

3.4.2 Extracting visual features

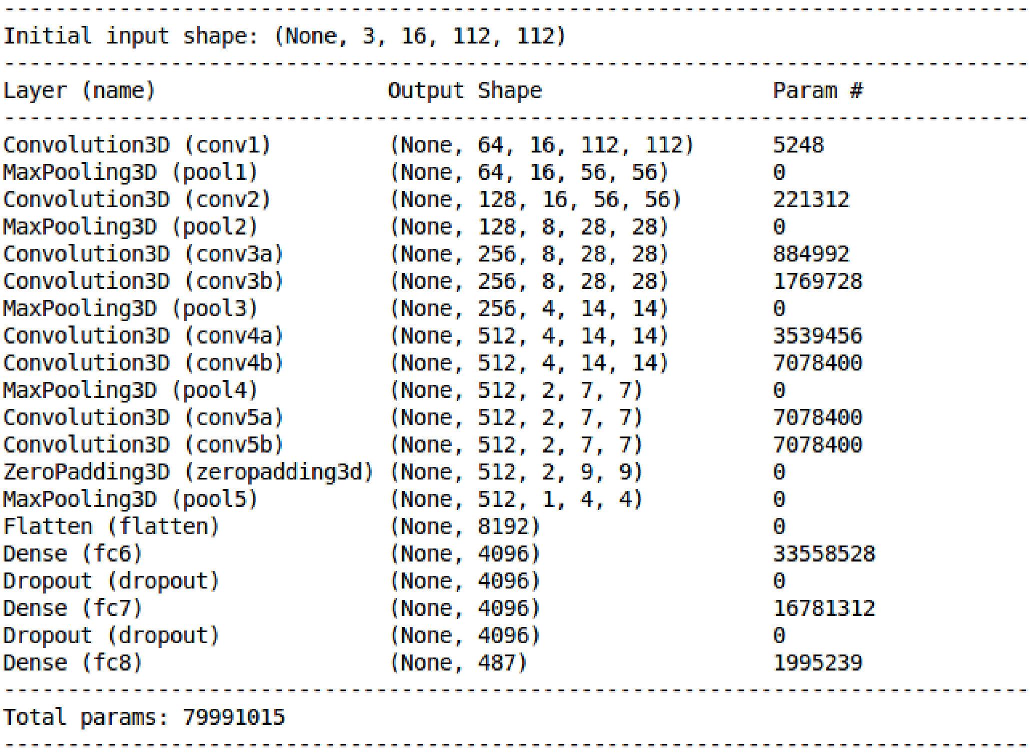
Videos are read frame by frame and stored in a matrix. This means that for every video we are dealing with a huge amount of data. To reduce this big amount, some features from each video are extracted using convolutional networks. In concrete, we used the C3D model trained with the activityDataset as Alberto Montes explains in his report [13].

The network is composed of 8 convolutional layers and 2 fully-connected layers and it can act as a feature extractor. The features have a length of 4096 samples. Before connecting the videos to the network, they have to be prepared adequately. The network makes a prediction for every 16 frames of the input video. This means that each video has to be reshaped into groups or clips of 16 frames. The width and height of each of the 3 RGB channels are of 112x112. Hence, each video we fed into the network needs to follow the next organisation: (num-clips, 3, 16, 112, 112).

This grouping of 16 frames allows the network to exploit short-term temporal information for each video. Once the data is prepared, each video can be fed into the network and a set of features will be extracted by it with the shape (num-clips, 4096).

3.4.3 Extracting acoustic features

To extract features from audios the network VGG19 [15] is used. This is a convolutional network composed of 3x3 convolutions and 2x2 poolings. This model is available inside Keras to be fine-tuned.



```
--------------------------------------------------------------------------------
Initial input shape: (None, 3, 16, 112, 112)
--------------------------------------------------------------------------------
Layer (name)                  Output Shape                  Param #
--------------------------------------------------------------------------------
Convolution3D (conv1)         (None, 64, 16, 112, 112)      5248
MaxPooling3D (pool1)          (None, 64, 16, 56, 56)        0
Convolution3D (conv2)         (None, 128, 16, 56, 56)       221312
MaxPooling3D (pool2)          (None, 128, 8, 28, 28)        0
Convolution3D (conv3a)        (None, 256, 8, 28, 28)        884992
Convolution3D (conv3b)        (None, 256, 8, 28, 28)        1769728
MaxPooling3D (pool3)          (None, 256, 4, 14, 14)        0
Convolution3D (conv4a)        (None, 512, 4, 14, 14)        3539456
Convolution3D (conv4b)        (None, 512, 4, 14, 14)        7078400
MaxPooling3D (pool4)          (None, 512, 2, 7, 7)          0
Convolution3D (conv5a)        (None, 512, 2, 7, 7)          7078400
Convolution3D (conv5b)        (None, 512, 2, 7, 7)          7078400
ZeroPadding3D (zeropadding3d) (None, 512, 2, 9, 9)          0
MaxPooling3D (pool5)          (None, 512, 1, 4, 4)          0
Flatten (flatten)             (None, 8192)                  0
Dense (fc6)                   (None, 4096)                  33558528
Dropout (dropout)             (None, 4096)                  0
Dense (fc7)                   (None, 4096)                  16781312
Dropout (dropout)             (None, 4096)                  0
Dense (fc8)                   (None, 487)                   1995239
--------------------------------------------------------------------------------
Total params: 79991015
--------------------------------------------------------------------------------
```

Figure 3.9: Architecture of the C3D model

VGG-19 was trained on a subset of the ImageNet database [10], and was used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [14]. The training set was composed with more than a million images and it was designed to classify images into 1000 object categories. Keyboard, mouse, pencil, and some animals are examples of the final output of the network. As a result, it can be said that this model has learned rich feature representations for a wide range of images. The main architecture of VGG-19 can be studied in Figure 3.10.

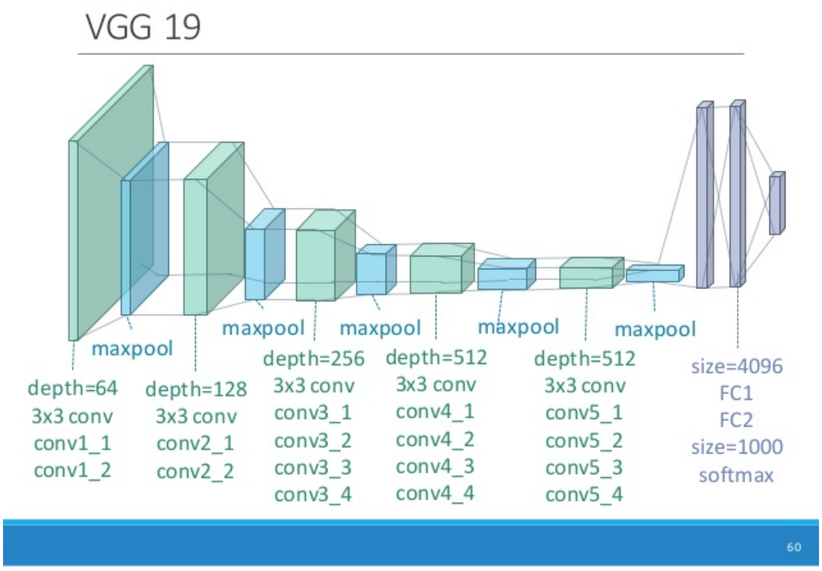


Figure 3.10: Architecture of the VGG-19 model

As it is explained in [8] the network needs to be modified and trained again with audio samples to fine-tune it. Audio samples need to be first pre-processed in order to be accepted by the network.

The audio is divided into non-overlapping 23520 samples windows. Each of these frames is decomposed using short-time Fourier transform using 10 ms windows every 5 ms. The resulting spectrogram is integrated into 64 mel-spaced frequency bins. This procedure results in a 98x64 patch shape for each window. Those patches mould the input of the classifier.

As it has been explained before, visual features were extracted every 16 frames with videos of 30 fps. This means that we have a feature vector every 0,533 seconds.

$$\frac{16}{30} = 0,533 \tag{3.1}$$

The reason for choosing audio windows of 23520 samples was to have exactly the same number of acoustic features (one feature vector every 0,533 seconds).

$$\frac{23520}{44100} = 0,533 \tag{3.2}$$

3.4.4 Fusing features

Features were fused by concatenation. This means that for every time-step of 0,533 seconds both arrays of features are connected resulting into a larger array.

3.5 Predicting emotion

Once the features are extracted, they can be used to try to predict the emotions. To do it, an LSTM based model is used. With this model, we try to predict the emotions every 0,53 seconds of a video. The main thing in this part is to make sure that these predictions take into account the past features from the same video. When the network has seen one video, their states need to be reset so the network will not try to predict using the information of another video. This can be done by defining a stateful network.

With the stateful model, all the states are propagated to the next batch. It means that the state of the sample i of Video Xj (Xji) will be used in the computation of the sample Xji+1, Xji+2, Xji+N in the next batch.

This year, MediaEval ask their participants to submit the emotions predictions every 10 seconds with a shift of 5 seconds. In our model, the predictions are outputted every 0,53 seconds. For this reason, some pre-processing and post-processing of the ground truth data need to be performed.

First, the ground truth data is expanded making a mean average to have values every 0,53 seconds. These values are used as the ground truth data seen by the model (in training). Once the model is trained, it is ready to make predictions. These predictions are compressed again, doing also the mean average to have values every 10 seconds with a shift of 5 seconds.

The MediaEval workshop of 2017 proposes two sub-tasks: predict emotion in terms of valence and arousal and predict fear scenes. Because of the nature of the valence and arousal scores, the first sub-task is related to a regression problem. The second sub-task, however, is related to a typical binary classification problem where 1 means the scene induces fear to the viewer and 0 that the scene does not induce it.

Although both tasks have different nature, the followed pipeline is the same and only the last layer of the LSTM changes.

The main architecture of the model is formed by:

- Input Layer:
- Batch normalization
- Drop-out
- LSTM Layer
- Drop-out.
- Dense Layer

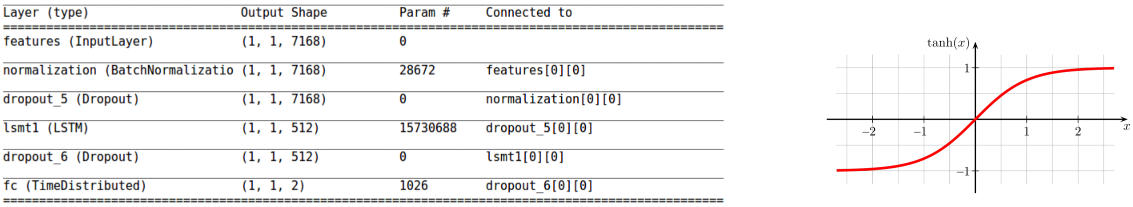
As we can see the main model is simple and is formed of only 6 layers. The input layer defines the shape of the input in terms of (batchsize, timesteps, features-dimension). This is important if the LSTM is defined as a stateful layer because it needs to know the shape of the batches in order to update the weights. The batch normalisation is in charge of normalising the input at every batch. Without it, it would be complicated for the network to learn from the fact that the distribution of each layer's inputs changes during training, slowing down this training. The batch normalisation is also very helpful since the inputs of the model are the result of a concatenation of two different features (the visual and the acoustic). Those features can have a huge variation in their range since they do not come from the same place and correspond to different modalities. The drop-out layer is very useful to avoid overfitting. Overfitting is an event that appears when the trained model performs very good on the training data but it performs really bad on new data unseen by the network. This can happen when your model is trained with poor data or when it is too complex for the amount of data that it receives. The main idea of the dropout is to randomly invalidate a number of neurones during the training, making the model less vulnerable to this overfitting and making a more generalised model. The LSTM layer is a kind of RNN (see 3.3.3) that, basically, uses its memory states and gives the model the ability to predict considering the past samples seen by the network. Finally, the Dense layer is the one in charge of making the classification. Depending on the kind of input of the model this Layer is designed to solve a regression or a classification problem (valence/arousal vs. fear).

3.5.1 The regression problem

In the regression problem, two scores are predicted. Those scores are valence and arousal and their values flow between [-1, 1]. To do so, the Dense Layer of the model is built with 2 neurones.

Each one of this neurone is in charge of predicting the valence and the arousal respectively. This layer is governed by a tanh function as its activation (Figure 3.11 b)). This function saturates the output between -1 and 1, making the network unable to output higher values.

When compiling the model, the mean squared error is used as the loss function that the model will try to minimise. This estimation is very used in regression problems and is one of the evaluation metrics used by MediaEval to compare the submitted results of the participants.



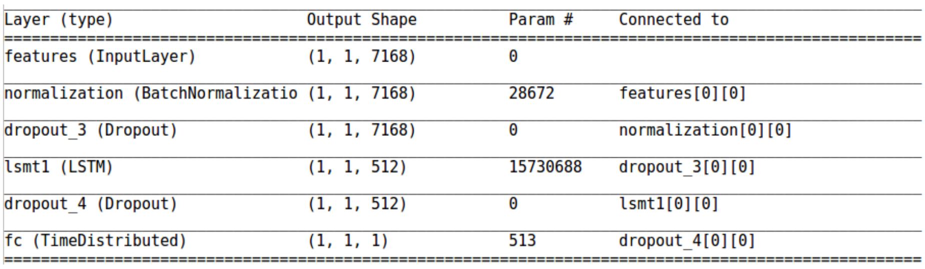
```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
features (InputLayer)            (1, 1, 7168)          0
____________________________________________________________________________________________________
normalization (BatchNormalizatio (1, 1, 7168)          28672       features[0][0]
____________________________________________________________________________________________________
dropout_5 (Dropout)              (1, 1, 7168)          0           normalization[0][0]
____________________________________________________________________________________________________
lsmt1 (LSTM)                     (1, 1, 512)           15730688    dropout_5[0][0]
____________________________________________________________________________________________________
dropout_6 (Dropout)              (1, 1, 512)           0           lsmt1[0][0]
____________________________________________________________________________________________________
fc (TimeDistributed)             (1, 1, 2)             1026        dropout_6[0][0]
====================================================================================================
```

Figure 3.11: a) Architecture of the LSTM model for valence and arousal predictions. b) Activation function for predicting scores: tanh

3.5.2 The classification problem

In the classification problem, two classes are predicted. Those classes are 1 for fear scenes and 0 for non-fear scenes. To do so, the Dense Layer of the model is built with 1 neurone. The output of this neurone shall be 1 or 0 depending on the classification. This layer is governed by a softmax function as its activation. This function calculates the probabilities of each target class over all possible target classes, being the class with higher probability the final decision of the classifier.

When compiling the model, the binary cross-entropy is used as the loss function that the model will try to minimise. This estimation is very used in binary classification problems.



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
features (InputLayer)            (1, 1, 7168)          0
____________________________________________________________________________________________________
normalization (BatchNormalizatio (1, 1, 7168)          28672       features[0][0]
____________________________________________________________________________________________________
dropout_3 (Dropout)              (1, 1, 7168)          0           normalization[0][0]
____________________________________________________________________________________________________
lsmt1 (LSTM)                     (1, 1, 512)           15730688    dropout_3[0][0]
____________________________________________________________________________________________________
dropout_4 (Dropout)              (1, 1, 512)           0           lsmt1[0][0]
____________________________________________________________________________________________________
fc (TimeDistributed)             (1, 1, 1)             513         dropout_4[0][0]
====================================================================================================
```

Figure 3.12: Architecture of the LSTM model for fear classification

Chapter 4

Results

This chapter will show some of the results obtained with the techniques explained in the methodology (3) which were done to improve the baseline system described in Section 3.4.

4.1 Experimental set-up

Our database is formed by:

- 30 films given as a training set with its correspondent ground truth data.
- 10 films given as a test set without ground truth data.

Due to our participation in the MediaEval workshop, it is not possible to use the 10 test films because of the lack of the ground truth data. Therefore, the training experiments will be done with only 30 films.

In the next experiments, some hiper-parameters will be studied, such as the contribution of our features and the importance of the stateful nature of the LSTM model.

In the next results, the 70% of the development set was used as training, the 20% was used as validation and the 10% as the test. However, for testing the model with mixed features (visual and acoustic) we used 27 films for training, 2 films for validation and 1 film for testing. We did that because our dataset is small (only 30 films) and we want our model to see the maximum data as possible in order to generalise. We compensated this reduction in the validation and test set by doing a cross-validation testing, where several runs are executed with different partitions, obtaining more reliable results.

4.2 Evaluation metric

The evaluation metric proposed by MediaEval is the Mean Squared Error (MSE) and the Pearson's Correlation Coefficient (PCC) for the regression problem, and the Mean Average Precision (MAP) for the classification problem.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{4.1}$$

The Mean Square error measures the distance between the predictions $\hat{\text{Y}}$ and the ground truth data Y, where n is number of samples contained in Y

$$PCC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.2}$$

PCC is a measure of the linear correlation between two variables X and Y. If the PCC value is one, there is a total positive linear correlation, if it is 0 there is a no correlation at all, and if it is -1 there is a total negative linear correlation. x̄ or ȳ is the mean of the array x and y, respectively.

$$MAP = \frac{1}{|C|}AP(c) \tag{4.3}$$

The mean average precision of the system is obtained by computing the mean of the average precisions (AP) of the C classes, as as presented in Equation 4.3.

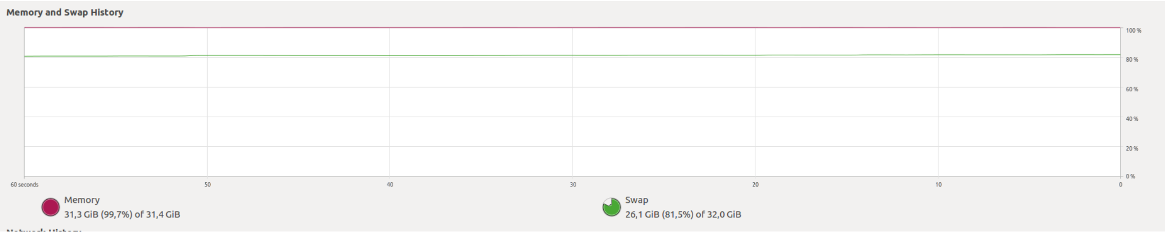
In table 4.1 the best results from last year are shown. RUC [4] team was the one that achieved them using features from video, audio and motion with a Random forest as machine-learning technique.

	Valence	**Arousal**
MSE	0.099	0.120
PCC	0.142	0.236

Table 4.1: Some of the Test results obtained after training the model with raw audio

4.3 Predictions with raw videos

At the beginning of this work, we tried to connect the raw RGB frames of the videos directly to the LSTM. The main problem we found was that the amount of ram memory required to do it was extremely high, and we didn't have enough resources to hold that load. Computers have a specific RAM memory. Once this memory is full, the swap technique can be used to help the machine. This technique fakes RAM memory and it helps computers with a small amount of RAM to achieve some tasks. However when the swap is activated the PC works very slow, making almost impossible to continue working with it until it ends. Even with the help of the swap, it was impossible to load in memory more than two films. As it was expected, using the raw data from the RGB frames was not an option.



Memory and Swap History

Memory
31,3 GiB (99,7%) of 31,4 GiB

Swap
26,1 GiB (81,5%) of 32,0 GiB

Figure 4.1: Using swap to load videos into memory

4.4 Predictions with raw audio

Before using the feature extractors, we also tried to predict the emotions using only the audio samples. Although it was still a big amount of data, it could be done by our machine. Different hiper-parameters were tried such as learning rate, batch-size or drop-out probability. In general, the results didn't perform very well in testing, and the validation curves were very irregular.

(bs, lr, dp)	min training	min validation	MSE Valence	MSE Arousal
(1632, 1e-3, 0.5)	0.1298	0.2985 /h	0.4201	0.3972
(1792, 1e-3, 0.5)	0.0745	0.1945 /h	0.4911	0.5070
(1376, 1e-3, 0.5)	0.0803	0.1027 /h	0.4343	0.3478

Table 4.2: Best results from last year.

Table 4.2 shows some of the best results. The first column shows the values used as batch-size (bs), learning-rate (lr) and drop-out probability (dp). The results were very poor when testing the model with an entire film unseen by the network. We tried more values but all the experiments showed similar results. 0.5 of dropout seemed always the best option and the learning rates of 1e-4 and 1e-3 showed the best curves (using Adam optimizer). However, with the change of the batch size, the results varied a lot.

4.5 Predictions with features

After extracting all the features a few experiments were done. We also compared the use of a stateful LSTM vs. a stateless LSTM (without considering timesteps). The results showed that the stateful LSTM outperforms the stateless ones in the test results.

However, on some occasions, the model seemed to work better (in training and validation) without taking into account the time and shuffling all the input features before training, but looking at the test results we realised that the stateful LSTM was doing better. These results were the expected since emotions are related in time.

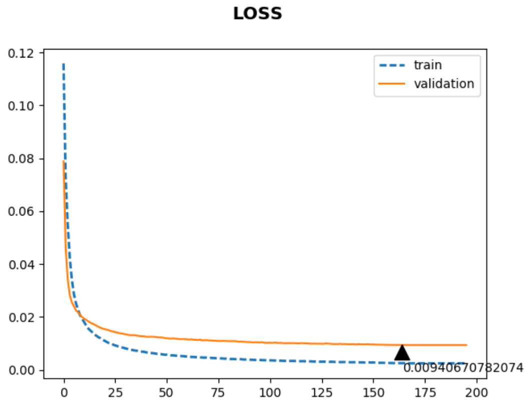


Figure 4.2: Batchsize=512, dropout=0.5, learning rate=1e-3

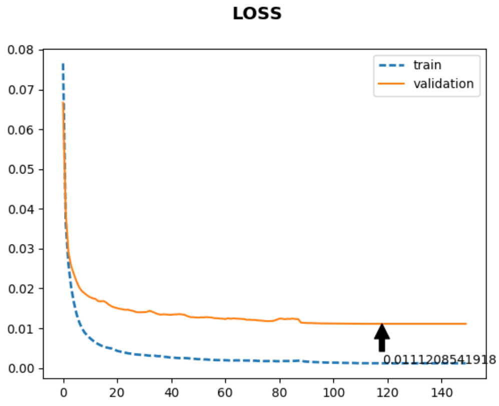


Figure 4.3: Batchsize=512, dropout=0.6, learning rate=1e-3

Figure	MSE Valence	MSE Arousal	PCC Valence	PCC Arousal
4.2	0.4998	0.4385	0.0049	0.0121
4.3	0.5045	0.4985	-0.0031	0.0072

Table 4.3: Some of the Test results obtained after training the model with a stateless model.

4.5.1 Audio features

One of the experiments we wanted to try was to compare the extracted features and see which ones could perform better. In the next table, the results of 2 configurations are shown. We can see that the audio features do not work very good. This can be due to the VGG-19 model used. This model has been fine-tuned with videos that may not have so much information about emotions. If this is the case, logically, the features extracted won't be correlated with the emotion. Using another database to fine-tune the model could be one solution to the problem, but this task would require more GPUs and more time that what we had. The main experiments were done changing some optimizers, changing the dropout values and the learning rates, but the best two configurations were using Adam and RMSprop.

(opt, lr, dp)	MSE Valence	MSE Arousal	PCC Valence	PCC Arousal
(Adam, 1e-7, 0.5)	0.3598	0.3985	0.0090	0.0111
(RMSprop, 1e-7, 0.5)	0.3745	0.4145	-0.0851	-0.0702

Table 4.4: Some of the Test results obtained after training the model with audio features

4.5.2 Video features

The two best results training with video features showed better performance than audio features. However, these test results were still far away from last year's best results. Adam seems to be the optimizer that fits better with our task since it always performs a little bit better than other optimizers such as RMSprop or Adadelta.

(opt, lr, dp)	MSE Valence	MSE Arousal	PCC Valence	PCC Arousal
(Adam, 1e-7, 0.5)	0.2298	0.2185	-0.0175	-0.0199
(Adam, 1e-7, 0.6)	0.2345	0.2145	-0.0156	-0.0209

Table 4.5: Some of the Test results obtained after training the model with video features

4.5.3 Mixed features

Figure 4.4 shows the resulting plots of two trainings using Batchsize=1, dropout=0.5, learning rate=1e-8. Adam was used as the optimizer and the test results can be seen in Table 4.6. For each one of these plots, different data was used as training, validation and test. These partitions are called folds and this technique is done to get a more reliable test result. As we can imagine, every video is very distinct. For that reason, when we train a model and we test on a single video or a few videos, it could be some case where the model performed very well but only for

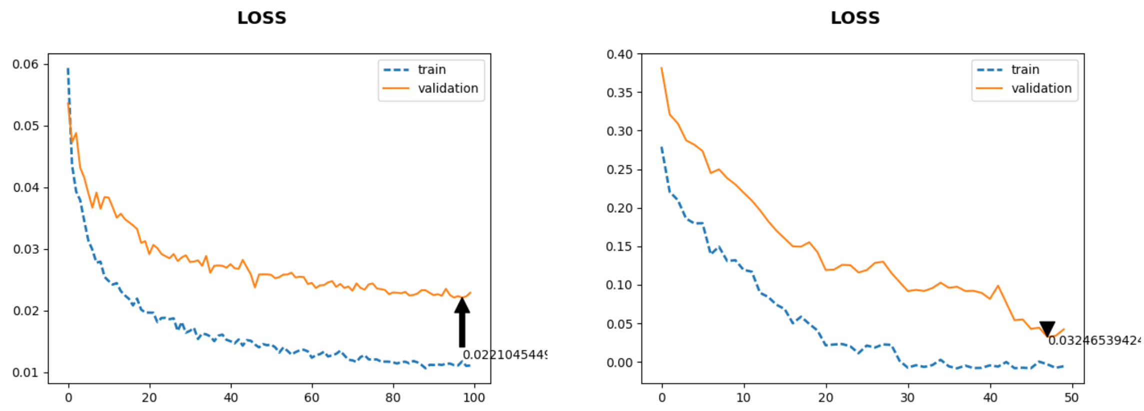


Figure 4.4: Results of the training with mixed features of two folds (different films used as training, validation and test)

those test videos. If we used the model for predicting with other unseen videos, the model would probably fail and get worse results. To get more reliable test results, normally people do 10 folds and then average the final results. We did not have that time to do so (15 hours * 10 folds for every configuration we wanted to try), so we used only 3 folds. This number is very poor but it gives an approximation of how our model could perform with unseen data. This technique is well known by the machine learning scientist by the name of cross-validation.

The plots also show a better learning. It is a little bit noisy but it can be seen how our model learns epoch by epoch in training and in validation.

In the next table some test results looking different optimizers are shown. All these configurations were used with learning-rate=1e-8 and 0.5 of dropout since it seemed the best values from previous experiments. Using SGD was extremely slow and we used only one fold. Those results are good but since only one fold have been used are not very reliable. The same happens with Adamax optimizer. When using Adam, Adadelta and RMSprop different number of cells in the LSTM were tried (256, 512, 1024). In most of the cases using 512 cells performed better.

(opt, cells, folds)	**MSE Valence**	**MSE Arousal**	**PCC Valence**	**PCC Arousal**
(Adam, 512, 3)	0.1598	0.1985	-0.0149	-0.0121
(Adadelta, 1024, 3)	0.2265	0.2401	0.0115	-0.0372
(RMSprop, 512, 3)	0.2202	0.2235	-0.0351	0.0072
(SGD, 512, 1)	0.1745	0.1245	-0.0022	-0.0041
(Adamax, 512, 1)	0.2745	0.2855	0.0041	0.0014

Table 4.6: Some of the Test results obtained after training the model with mixed features and using different configurations

Although we created a model which has learned to predict emotions, the final results obtained were not that good as we expected. These models were hard to train in computational terms since it took 15 hours to train 50 epoch (depending on the number of LSTM cells and the optimizer). For that reason, we could not do as many experiments as we would like. Another problem could be that we are using general features extracted with the C3D model and the VGG19. C3D model was trained with sports 1M Dataset. This means that possibly, the features extracted are more related to sports (more relevant for sports classification) than for emotion prediction. VGG19 was also trained with Youtube-8M, a general dataset which possibly would not be very related to the affective content. This can be solved by exploring other datasets which allow to exploit the emotional content inside them and fine-tune those models with that dataset. Regrettably, this was a task that required much more time that what we had and we decided to design the pipeline and consider it as future work. The more relevant the features are, the more information the LSTM is able to learn and the better the performance of the model is.

4.6 Predicting fear scenes

Here we dealt with a classification problem where our model had to predict whereas a feature vector corresponding 0,53 seconds induced fear to the viewer or not. We did not focus much in this part since all of our resources were destined to improve the valence and arousal predictions. Since valence and arousal can represent emotions, fear can be represented in terms of valence and arousal, so if the first model works perfectly, this problem could be solved by predicting valence and arousal and looking for the values that correspond to fear. The results we got here were very bad since our model almost every time predicted scenes as non-fear.This happened because our dataset was very unbalanced, which means that the model sees more examples from one class than from the other. Some techniques to avoid this problem could be to use data augmentation by flipping the frames of the videos.

Chapter 5

Budget

This project has been developed using resources provided by the Interactive Media Systems research group of the Faculty of Computer Science (TU Wien, Vienna). The main costs of this project do not consider the hardware used or the possible maintenance costs of it.

Hence, the main expenses of the project are due to the salary of the researchers and the amount of hours spent in it. I consider my position as a Junior engineer, but the position of my supervisors as Senior engineers, who have a higher salary rate. The total duration of the project has been of 21 weeks. In the next Table the cost of each engineer and the final cost of the project is depicted:

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior engineer	1	8,00 €/h	30 h/week	5.040 €
Senior engineer	1	20,00 €/h	3 h/week	2.520 €
			Total	7.560 €

Table 5.1: Budget of the project

Chapter 6

Conclusions

The main objective of this project was to build a pipeline that predicts emotion in videos using deep-learning techniques. Those predictions had to consider the spatial information presented with the pixels inside the frames, but also the temporal information presented between frames. It also had to use the audio extracted from the videos to help the system when predicting. In addition, we wanted to participate in the MediaEval workshop organised for 2017, even the results will be available after the deadline of this thesis. Thanks to this work I have been able to learn how to deal with audiovisual data using python and I have been able to introduce myself to the huge world of Deep-Learning, that day after day is bringing us incredible results in so many tasks. For this reason, I consider that the main goals of this project have been achieved and it has been decisive in my formation as a telecommunication engineer.

With this project we have taken advantages of other pre-trained deep models which have saved a long time, and in which, in some cases, it would not have been possible to train them because of our limited resources.

We have also compared different architectures, configurations and inputs when training our models, showing that our video features contained more relevant information (correlated with emotions) than the audio ones. However, mixing both modalities improved the final predictions showing how the audio contained possible information that the model could not extract only with the video features. We have also understood the difference between a stateful and stateless LSTM, showing that when the LSTM is considering the time to predict, those predictions achieved better results.

Even the final results were not that good or better compared with the last year results, the pipeline proposed can still be performed to improve. Due to the lack of time, some experiments could not be performed. This project uses two deep models that have been pre-trained with the Youtube-8M dataset. This dataset consists of random videos extracted from the web and may no be very useful when predicting emotion. If I had had more time, the next step would have been to try other datasets with affective content to train those models and then give better features to our model (more allied with emotional impact).

As a future work, aside from the aforementioned, more experiments changing the hiper-parameters could be done. Also trying to improve the models used to make them more complex or even adding more modalities to them, such as Motion, which in many cases can be correlated with the elicited emotion. In relation to the classification problem, few experiments were done compared with the regression one, but some kind of data augmentation for video could be tried in the feature to see if the model does not predict always 0s (non-fear scenes).

Chapter 7

Appendices

Bibliography

[1] Timoleon Anastasia and Hadjileontiadis Leontios. Auth-sgp in mediaeval 2016 emotional impact of movies task. 2016.

[2] Yoann Baveye, Emmanuel Dellandrea, Christel Chamaret, and Liming Chen. Liris-accede: A video database for affective content analysis. *IEEE Transactions on Affective Computing*, 6(1):43–55, 2015.

[3] Soraia Vanessa Meneses Alarcao Castelo. Emophoto: Identification of emotions in photos. 2014.

[4] Shizhe Chen and Qin Jin. Ruc at mediaeval 2016 emotional impact of movies task: Fusion of multimodal features. In *MediaEval*, 2016.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[6] Venkat N. Gudivada and Vijay V. Raghavan. Content-based image retrieval systems. *Computer*, 28(9):18–22, September 1995.

[7] Alan Hanjalic. Extracting moods from pictures and sounds: Towards truly personalized tv. *IEEE Signal Processing Magazine*, 23(2):90–100, 2006.

[8] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. *arXiv preprint arXiv:1609.09430*, 2016.

[9] Asim Jan, Yona Falinie Binti Abd Gaus, Hongying Meng, and Fan Zhang. Bul in mediaeval 2016 emotional impact of movies task. In *MediaEval*, 2016.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Yang Liu, Zhonglei Gu, Yu Zhang, and Yan Liu. Mining emotional features of movies. In *MediaEval*, 2016.

[12] Ye Ma, Zipeng Ye, and Mingxing Xu. Thu-hcsi at mediaeval 2016: Emotional impact of movies task. In *MediaEval*, 2016.

[13] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. In *1st NIPS Workshop on Large Scale Computer Vision Systems*, December 2016.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[17] Baohan Xu, Yanwei Fu, Yu-Gang Jiang, Boyang Li, and Leonid Sigal. Heterogeneous knowledge transfer in video emotion recognition, attribution and summarization. *IEEE Transactions on Affective Computing*, 2017.