MultiAffect: Reproducible Research Framework for Multimodal Video Classification and Regression Tasks

Carlos Toxtli, Salvador Medina, Saiph Savage

West Virginia University
{carlos.toxtli,saiph.savage}@mail.wvu.edu
Carnegie Mellon University
salvadom@cs.cmu.edu
http://humancomputerinteraction.wvu.edu

Abstract. The access to information and distributed computing resources are becoming more and more open, but its accessibility has not advanced towards the solution of the reproducibility crisis for computational studies and their experimentation. Researchers over the years have investigated the factors that affect reproducibility in data science related studies. Some common findings point that non-reproducible studies lack information or access to the dataset in its original form and order, the software environment used, randomization control, and the actual implementation of the proposed techniques. In addition to that, some studies require a large number of computational resources that not everybody can afford. This work explores how to overcome some of the main challenges in reproducible research with a focus on multimodal video action recognition. We present MultiAffect, an inclusive reproducible research framework that standardizes feature extraction techniques, training and evaluation methods, and research document formatting for multimodal video action recognition tasks in an online environment. The proposed framework is designed to use a simple vanilla version of popular algorithms as a baseline, with the flexibility to plug-in state-of-the-art algorithms into the workflow with ease for further research. We tested the framework on two different video analysis approaches: video action recognition and affect recognition. MultiAffect was able to perform both tasks by only setting up the proper configuration. The results produced by MultiAffect were competitive in regard to published studies, and was deployed in Google Colaboratory (http://bit.ly/multiaffect), validating its inclusiveness as we are able to reproduce experiments with no client requirements (online), no-configuration, and free-of-charge. We aim that inclusive reproducible research frameworks for complex and highly demanding tasks can reduce the barrier to entry of video analysis and boost the progress in this area.

1 Introduction

The affective computing term was coined by Rosalind Picard in 1995 [34] to study the role that affect plays in intelligent behavior, as affect are the bodily manifestations of an experienced emotion. Affective computing explores ways to recognize human affect automatically and how to generate corresponding responses by a computer. Performing automatic affect recognition falls into the list of expected capabilities of an AI, as it is for the auditory and visual perception [15]. Machine learning approaches enable computers to learn directly from examples instead of providing explicit models. Humans intuitively recognize affect from visual and auditory cues, therefore computers need to have a digital representation of these cues, which are usually represented in high dimensional vectors. This poses affective computing to a problem known as the *curse of dimensionality*, which is a phenomenon where the higher the number of dimensions the data has, the less effective conventional computational and statistical methods become [14]. A common solution to this problem is to project the high-dimensional data into a lower-dimensional space through approaches such as feature selection.

Machine learning algorithms with so-called *shallow* architectures, such as kernel methods and single-layer neural networks often rely on handcrafted features based upon heuristics of the target problem. These approaches may not always be the most efficient way to solve some of the most challenging problems such as affect recognition [15]. Since 2010 researchers started to explore the application of *deep learning* architectures for affect recognition, by following traditional machine learning insights for affect recognition. It is important to notice that between 2010 and 2017, around 950 studies were published in this area [36] showing a significant interest by the community. In this work, we propose an easy to use deep learning approach for affect recognition.

Instantaneous emotion categorization is performed in short-term videos that usually last a couple of seconds [42, 43]. These short clips use short utterances instead of natural utterances [26, 24]. An utterance is a continuous piece of speech beginning and ending with a clear pause, and it represents the smallest unit of speech. People have the capacity to adjust their internal emotion representation to a newly perceived emotional expression instantaneously, and use it to obtain a greater understanding of the emotional behavior of another person. This mechanism is known as a developmental learning process. Algorithms that aim to understand emotions as humans, should be able to process and understand emotions from long-term natural utterances.

The notion of categorical affective states had origin in Charles Darwin's research on the evolution of affect, this theory is controversial and the discussion in literature still remains [30]. The six universal emotions categorization scheme was originally proposed by Paul Ekman and was based on facial expressions [22]. It includes Disgust, Fear, Happiness, Surprise, Sadness, and Anger. Humans usually express themselves in different ways, sometimes even combining one or more characteristics of these so-called universal emotions. These paradigm allows us to represent emotions in a six-dimensional space. Dimensional models aim to avoid the restrictiveness of discrete states and allow a more flexible definition of affective states as a continuous multidimensional space. The most commonly used example is Russell's circumplex model of affect [37], which consists of the two dimensions valence and arousal. For example, in our work we used the OMG Emotion dataset [13] which is formed by relatively long emotion videos with an average length of 1 minute, where each video includes categorical and dimensional annotations. Each utterance contains arousal and valence values, as well as labels of seven discrete emotions. Arousal has a continuous value from 0 (calm) to 1 (excited), while valence has a value from -1 (negative) to +1 (positive). Two metrics are used to evaluate the arousal or valence estimation over this dataset: mean squared error (MSE) and concordance correlation coefficients (CCC). Our work uses categorical and dimensional annotations to build the models.

Humans rely on multiple modalities when expressing and sensing affective states in social interactions. It seems natural that computers could benefit from the same variety of sensorial inputs [31]. In fact, there has been an increased interest to design such multimodal systems [20], and it is generally accepted that audiovisual sensor fusion can increase model robustness and accuracy. This is commonly known as joint multimodal feature learning, whose challenge is to determine how and what stage or stages to fuse from the multiple modalities. Fusion can be performed at early stages of the model closer to the raw sensor data, or at a later stage combining independent models. In early or feature-level fusion, features are extracted independently and then concatenated for further learning of a joint feature representation; this method allows the model to capture correlations between the modalities. Late or decision-level fusion aggregates the results of independent recognition models. Our work uses five modalities extracted from video frames, audio, and text. Some features are spatiotemporal frame-level sequences, while others are aggregated utterance-level features. The used features are: face deep features, face handcrafted features, body skeleton joint angles and body embedded features, word MPQA opinion lexicon, word Bing Liu lexicon, audio features per utterance and per fragments, and instantaneous emotions per utterance and per fragments. For each feature, we performed unimodal and multimodal tests. We implemented LSTM (Long short-term memory) for spatiotemporal features and multilayer perceptron (MLP) to build aggregated features. We performed early and late fusions in the multimodal tests.

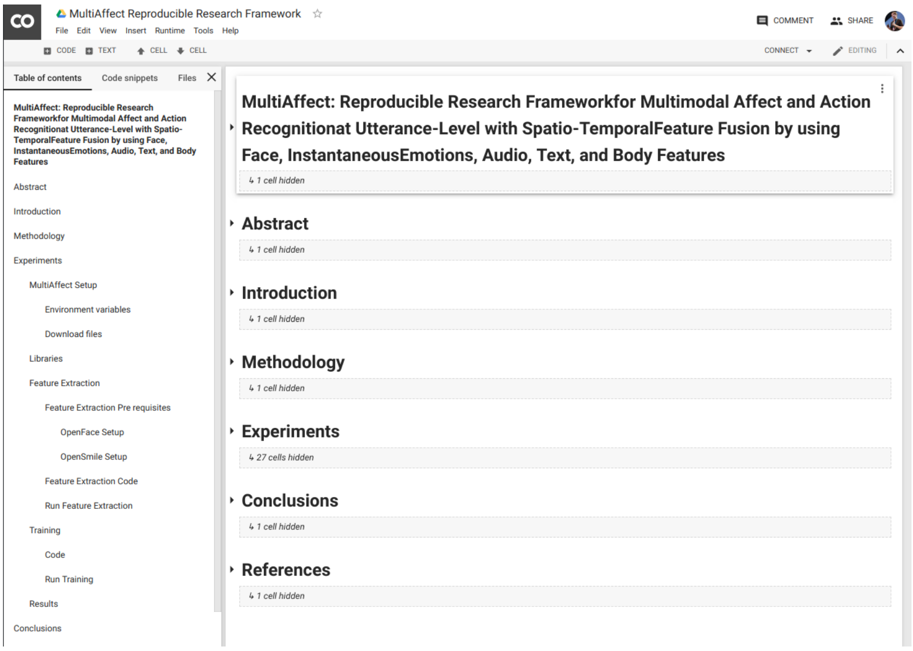
Building a video pipeline is a complex task and reproducing the results can become a laborious task. According to [10] there is a general perception of a reproducibility crisis within the research community. Their studies show that around 70% of researchers have tried and failed to reproduce another scientist's experiment. A manifesto for reproducible science states that improving the reliability and efficiency of scientific research will increase the credibility of the published scientific literature and accelerate discovery [29]. There have been important advances in promoting platforms to reproduce computational workflows such as the Jupyter digital notebook [27]. Reproducing multimodal affect recognition is a particularly complex task, since it requires the configuration of multiple tools and many resources in terms of data storage and processing. With that focus in mind, we structured a dynamic version of this chapter that can be found at http://bit.ly/multiaffect [1]. The dynamic version is designed as a framework that defines guidelines for feature extraction and training processes.

Our framework takes advantage of a powerful online platform such as Google Colaboratory to set up the virtual machine, load the data, extract the features, and train the model, all of these from the same notebook.

The main contribution of this work is to show design principles for reproducible research frameworks for video analysis, and how this work can be expanded to other fields. We did not perform all the possible feature combinations in order to allow the possibility of others to keep on exploring ideas on top of this work.

2 MultiAffect

MultiAffect is a reproducible research framework for computational workflows based on multimodal video classification and regression tasks. The main goal of MultiAffect is to give guidance on how to reproduce research experiments in a fixed setting. In order to achieve that goal, the MultiAffect framework is formed by five main components: (1) Research Paper Template: Defines the minimum set of sections and mandatory citations; (2) Platform Setup: Ensures that the machine is properly configured; (3) Feature Extractor: Monitors the feature extraction and manage the extracted features; (4) Model Trainer: Defines, trains, and fine-tunes the model; (5) Evaluator: Calculates and reports the performance metrics. Figure 1 shows our interface.


CO
MultiAffect Reproducible Research Framework
File Edit View Insert Runtime Tools Help
CODE TEXT CELL CELL
COMMENT SHARE
CONNECT EDITING
Table of contents Code snippets Files
MultiAffect: Reproducible Research Frameworkfor Multimodal Affect and Action Recognitionat Utterance-Level with Spatio-TemporalFeature Fusion by using Face, InstantaneousEmotions, Audio, Text, and Body Features
Abstract
Introduction
Methodology
Experiments
MultiAffect Setup
Environment variables
Download files
Libraries
Feature Extraction
Feature Extraction Pre requisites
OpenFace Setup
OpenSmile Setup
Feature Extraction Code
Run Feature Extraction
Training
Code
Run Training
Results
Conclusions

MultiAffect: Reproducible Research Frameworkfor Multimodal Affect and Action Recognitionat Utterance-Level with Spatio-TemporalFeature Fusion by using Face, InstantaneousEmotions, Audio, Text, and Body Features

↳ 1 cell hidden

Abstract

↳ 1 cell hidden

Introduction

↳ 1 cell hidden

Methodology

↳ 1 cell hidden

Experiments

↳ 27 cells hidden

Conclusions

↳ 1 cell hidden

References

↳ 1 cell hidden

Fig. 1. MultiAffect interface

2.1 Research Paper Template

Reproducible research papers must be formed by an explanatory document and an available infrastructure attached to the document to reproduce the results. Interactive notebooks are ideal for achieving this goal, and Jupyter has become one of the most used platforms for this purpose. Jupyter is an open-source online platform that enables users to create and share documents that contain executable code, renders equations, allows data visualizations, and displays narrative text. The notebooks created in Jupyter can be styled through a Markdown syntax. Markdown is a light-weight markup language with a simple text syntax. The goal of Markdown is to enable people to write formatted text using an easy-to-read and easy-to-write syntax. Research paper formats are usually written using LaTeX syntax. LaTeX is a popular typesetting system designed for the creation of technical and scientific documents. LaTeX differs from Markdown, as it needs to be compiled to build a formatted human-readable version of the document. These tasks are time-consuming and do not add value to the research work. MultiAffect removes the burden of all these formatting tasks, helping researchers to focus on the experimentation and the dissemination of their work.

There are some challenges related to the research paper template replication in interactive notebooks, such as the wide variety of formats and the number of columns. Although two-column research paper formats are commonly used for some artificial intelligence proceedings, there is currently challenging for an interactive notebook to replicate that format. Markdown syntax is mainly focused on a one-column format. The standard version of Jupyter notebook does not include graphical tools to split texts into two columns. Even though some HTML tricks can be done to structure in an equivalent format, it loses their responsiveness across devices. For the initial version of MultiAffect framework, we proposed a basic version and a set of rules that can help to expand its capabilities. The proposed solution replicates a one-column ACM (Association for Computing Machinery) manuscript format. The reason behind this decision is the number of HCI (Human-Computer Interaction) proceedings available and its impact on affective computing.

The MultiAffect template contains six main sections that are: (1) Abstract: Overall description of a video action recognition or affect regression paper; (2) Introduction: Extended description of the approach that implements fused multimodal features; (3) Methodology: Explain the feature extraction and training models applied; (4) Results: Generic explanatory text, a benchmark table, and performance plots; (5) Conclusions: Give a general conclusion of how fused multimodal models are useful for action recognition; and (6) References: It already contains the references to the used feature extraction methods, datasets, and algorithms.

It is important to mention that although the framework provides a ready to use and publish template, we do not encourage users to submit slightly edited versions to proceedings. Incremental research best practices should be considered before submitting a derived version from this framework.

2.2 Platform Setup

Preparing a host machine to replicate machine learning research is usually challenging, time consuming, and expensive. One of the reasons is that most of the models available today require a large scale dataset for training. Hence, multimedia datasets have a high storage requirement. In machine learning tasks, the feature extraction step helps algorithms to reduce the dimensionality of the data and aids the model to focus on their most significant or discriminative parameters. However, extracting features from multimedia samples is a highly demanding task in terms of computation. Another reason of why it is Some of the tools that are required to perform the data extraction need to be compiled for the host operating system. Scientific tools are commonly built from multiple libraries and sometimes depend on specific versions of certain libraries for certain operating systems; this makes them prone to throw compilation errors. Sometimes the code is not given, and there is an extra effort to code the instructions described in the publication. Even if the code is available, sometimes the code is not ready to reproduce, and important efforts should be performed to make it work when works.

The software challenges can be mitigated by using virtual machines or containers. Virtual machines and containers give a base operating system that can contain the proper configuration built-in. These approaches can run in the top of the host operating system or in online infrastructure. The hardware challenges can be overcome by investing in powerful enough architecture in-site or by using online on-demand infrastructure. Conventional research paper replication depends on multiple factors as we have explored.

The MultiAffect framework uses Google Colaboratory to publish the interactive notebook and to perform the computation in the attached virtual machine. Google Colaboratory is a free research tool that enables users with a Google account to host and run code over Google's infrastructure. Google Colaboratory offers users the ability to execute their code segments in CPUs, GPUs, and TPUs (an AI accelerator application-specific integrated circuit). By the time this work is published, Google Colaboratory offers a virtual machine with a Tesla K80 GPU, 12 GB of RAM, and 350 GB of storage. This platform provides enough resources to perform the computation required for multimodal analysis of video, and the storage required for storing the 8.4 Gb required for the OMG-Emotion dataset used for showcasing this work.

Our framework configures Google Colaboratory's built-in virtual machine with the packages that are required to perform the feature extraction. This platform includes a Debian based operating setting, so the provided instructions are platform-specific. Local replication of our framework requires an Ubuntu 18.04 operating system in order to install all the libraries successfully. Our platform is agnostic to the Python version, all the code executed in the notebook is written in Python, and it can be executed in the versions 2 or 3 of the interpreter. Our framework is able to set up and run the experiment from the online platform, enabling users to deploy and execute the code in a free of charge environment and without special requirements in the client-side.

Some of the pre-requisites are libraries that are already available in package managers. The operating systems packages were installed by using the apt-get package manager and the Python libraries by using the pip tool. The required packages and libraries were installed from a single line command containing all the package or library names separated by a space. The tools that were downloaded and processed in the host (compiled or adapted) were: OpenSMILE, an audio toolkit that needs to be compiled; OpenFace a toolkit for FER (Face Expression Recognition) applications that needs to be compiled; and VGG-Face, a tool to extract deep features from face images, it needs to be edited to support the latest Keras (a deep learning framework) version. Additional work was needed to prepare the Google Collaboratory environment to be ready to compile the tools and run the code. It was needed to upgrade the CMake tool version, a tool that commands the compilation and building processes. The OpenCV (a popular computer vision library) was needed to be upgraded to the latest version.

Other libraries that were used to extract certain features such as poses or instantaneous emotions were loaded as pre-trained models. We implemented Caffe and Keras pre-trained models. The Caffe models were implemented by using OpenCV DNN as an interface that loads a Caffe model and a prototxt files. The Keras models were loaded as checkpoints (HDF5 format) instead of only weights (h5 format). The main difference between both approaches is that checkpoint files include the network architecture and the weights, and the weights file requires the architecture to be coded. We implemented OpenPose (a pose extractor toolkit) as a Caffe model and five different emotion recognition models in Keras checkpoint format.

The setup process was conducted in three steps: (1) Initial setup: The first functional version; (2) Packing components: Uploading components in batches to cloud storage; and (3) Optimal setup: A version that loads faster.

Initial setup In this step, the libraries were downloaded and compiled directly from the notebook by running shell commands from the notebook cells. Prerequisites, missing dependencies, and additional packages were installed in the same notebook. The dataset and the pre-trained models were downloaded from their original sources to the virtual machine. The feature extraction, training, and evaluation code were directly inserted into the notebook in separate cells. The first version was tested until it successfully extracted the features, trained, and evaluated the models from the notebook. A backup of this notebook was documented and set as the initial version.

Packing components Each individual compiled library was packaged into a zip file that contains the binary files as well as the configuration files. The pretrained models that were individually downloaded from their original sources were packed together into a single file. Sometimes the latency is reduced by downloading a single large file from a high-speed source and increased when downloading multiple large files from different bandwidths. The outcome of this task is a collection of zip files that were uploaded to a Google Drive account. The files were shared with public access to be able to be downloaded in Google Colaboratory notebooks logged with different accounts. In case that the dataset license allows users to store it in cloud infrastructure, it should be considered to reduce the bandwidth load, especially for large video datasets.

Optimal setup After packaging and storing the files from the initial setup to the cloud, we started a branch of the initial setup that loads these files. The optimal setup notebook was a simplified version of the initial notebook, instead of having a long section documenting the setup process, it was replaced with a download pre-requisites section. The files were downloaded by using a Python tool called GDown that is already installed in Google Colabory. It is important to mention that the virtual machine attached to the Google Colaboratory notebooks has already an Ubuntu distribution with the most common machine learning tools and libraries already installed. This optimal version is tailored to Google Colaboratory only. Other Ubuntu environments should consider the initial setup since it installs all the required libraries.

Per each of the libraries installed, we measured the time that takes to install the pre-requisites plus the compilation time. In average, the overall setup of each library was five times slower than downloading and extracting a previously compiled and zipped version of the library. It is important to mention that some datasets requires explicit permission to be downloaded and stored, and must be deleted after using them. This is the case of the OMG-Emotion dataset that should be deleted once it was used. The total setup time for the Google Colaboratory environment was reduced from 43 minutes to 6 minutes after implementing the pre-compiled tools strategy and by downloading the files from the same Google infrastructure.

2.3 Feature Extractor

MultiAffect includes a feature extraction module as an independent component. Multimodal feature extraction is often a highly demanding task, as it requires a certain pre-processing of the videos before being able to extract features. Some common pre-processing tasks are: separating the audio, extracting frames, identifying faces, cropping faces, removing the background, among many other procedures. Our feature extraction methodology is based on the common ground found in submissions [40, 33, 25, 44, 18, 17] to the OMG-Emotion Challenge [3].

Our feature extraction process aims to maintain as invariant factors features such as the person descriptors (i.e., gender, age), scale, position, background, and language. Our approach considers ten features from five different modalities: face, body, audio, text, and emotions.

1) Face features: Visual features are extracted using OpenFace [12] estimators over the whole frames, while using VGGFace to extract embedding representations [32] of the facial regions. We use OpenFace toolkit to extract 68 facial landmarks in both 2D and 3D world coordinates, head pose, rigid head shape, eye gaze direction vector in 3D, and Facial Action Units intensity [21] which abstract the facial muscle movements. The detailed feature descriptions can be found on their website [4]. These visual descriptors are regarded as strong indicators of human emotions and sentiments [35, 39]. For the VGGFace representation, the facial region in each frame is aligned and cropped using a 3D Constrained Local Model described in [11]. We zero out the background respectively to the face contour indicated by the face landmarks. Then, the resulting cropped faces are resized to $224 \times 224 \times 3$ and fed into the pre-trained VGGFace model. To fuse the facial features, We take the 4096-dimensional feature vector from VGGFace's FC6 layer and concatenate it with the visual features previously extracted by OpenFace. The total dimension of the resulting concatenated features is 4805. Twenty frames are uniformly sampled from each video clip, and these are fed into the network for training and testing. In the case of shorter length video clips, we duplicated the last frame to fill the gap.

2) Body features: We use from the pose estimator framework OpenPose [16], the Body-25 model, which extracts 25 joint locations of the person's skeleton found in the image. We compute the joint angles from the detected nose, neck, arms, and shoulders, as well as a binary flag per joint, to indicate if the joint was visible in the image or not. In total, we extract and normalize 11 handcrafted features from the OpenPose estimation. We only used joints from the arms, ears, eyes, neck, shoulders, and nose, as these body parts are usually visible in single person videos. From the filtered joints, we drew a skeleton centered by the neck joint and normalized dimensions. The normalized skeleton is inserted in a frame of size 224×224 with a black background. That visual representation is processed by the VGG16 net to extract 4096 features. We take the 4096-feature vector from the FC6 layer and concatenate the 11 features computed from the OpenPose joints. The total dimension of all the concatenated features is 4107. Twenty frames were uniformly sampled per clip.

3) Emotion features: We use EmoPy [8], a machine learning toolkit for emotional expression to extract the score of seven basic emotions (sadness, fear, disgust, happiness, contempt, and anger) which are typically used for Facial Expression Recognition (FER). The same features were extracted from other four emotion recognition models [7, 6, 2]. We concatenated the predictions from the models into a 35-dimensional feature vector. Since we used the same twenty faces while computing the face features, we decided to summarize the temporal features into a single feature vector through a normalized sum of the frame feature vectors, resulting also in a 35-dimensional vector.

4) Audio features: Audio features are extracted using OpenSMILE toolkit [23], and we use the same feature set as suggested in the INTERSPEECH 2010 para-linguistics challenge [38]. The set contains multiple features such as Mel Frequency Cepstral Coefficients (MFCCs), MFCC, loudness, pitch, jitter, among others. [5]. These set of 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. The audio feature extractor delivers a feature vector for the whole video clip, as well as a set of 20 feature vectors from uniformly sampled audio clips from the video clip.

5) Text features: We use two opinion lexicons to analyze the patterns in the language. The first one is Bing Liu opinion Lexicon [19] with 2006 positive words and 4783 negative words. The second one is MPQA Subjectivity Lexicon [41] with 4913 negative words and 2718 positive words. For each utterance, we compute the frequency of negative and positive words according to the lexicons, as well as the total word count in the whole utterance. For utterances without a transcript, we replicate the same transcript of the nearest utterance in time. We also extract the word frequencies over the entire video and incorporate them as features for all utterances in the same video. The total dimension of the word feature is 10, including utterance-level and video-level word frequency from the two lexicons and the total word counts.

2.4 Model Trainer

The MultiAffect models use different deep learning models to recognize affect. Among them we find RNNs (Recurrent Neural Networks), CNNs (Convolutional Neural Networks), and MLPs (Multilayer Perceptrons). The implemented models are based on the work proposed by Deng et al. [18] to analyze affect from a multimodal perspective. MultiAffect is an incremental work as we add the body pose and the instantaneous emotions as two extra modalities.

Figure 2 shows the architecture of the proposed model. Our neural based model is formed by three main components: (1) the sub-networks for each modality; (2) the early fusion layer which concatenates the unimodal representations together; and (3) the final decision layer that estimates the sentiment.

1) Sub-Networks : MultiAffect is formed by ten sub-networks in total, some of them are RNNs to process the spatiotemporal samples, while some others are DNNs which output different vector sizes. As aforementioned, the features extracted from the face and the body joints are both handcrafted and deep embedding representations; both of them based on DNN approaches. Differently, instantaneous emotions and audio features are extracted from a DNN and a RNN, respetively, while the two text sub-networks follow a DNN approach.

2) Early Fusion Layers : This component concatenates per frame four unimodal representations together. Namely, the two face related features are concatenated into a single face vectorial representation, and the two body pose features are fused in the same manner. Then, a sequence of fused features from the face and body of a full video clip are further fed into two separate LSTMs with a hidden size of 64 units, followed by a dense layer of 256 hidden neurons for temporal modelling of each modality. The emotion features are temporally modelled as the face and body modalities, and the audio features are fed into a fully connected layer with 256 units.

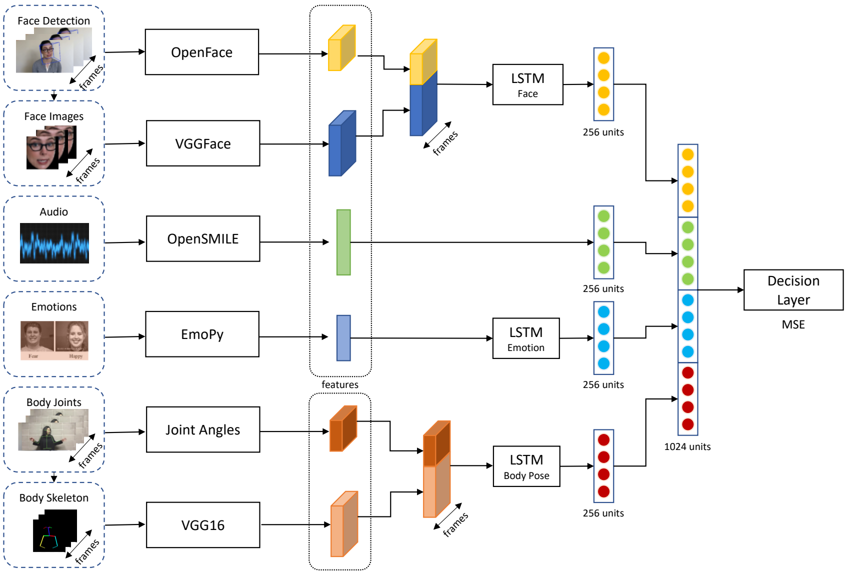


Fig. 2. MultiAffect Architecture

3) Late Fusion and Decision Layers : In this component we fuse the representations obtained from the four main modalities: face, audio, emotion and body pose. This late fusion stage concatenates the vector from each modality into one single vector. This aggregated feature vector is processed through a two-layer fully connected network. The hidden size of the network are both 1024 hidden units and have a single unit output, activated by a softmax layer for the classification task. For the regression task, we use MSE as the loss function for joint training and a concordance correlation coefficients (CCC) loss for further refinement.

2.5 Evaluator

The MultiAffect framework is designed to perform classification and regression tasks. Depending on the performed task, the platform is adjusted to display meaningful evaluations. The classification task gives accuracy metrics for the training, validation, and testing sets. In the case of a regression task, the framework computes the CCC that describes how well a new test or measurement reproduces a gold standard test [28]. This metric is described in eq. 1.

$$\rho_c = \frac{2\rho\sigma_{Gnd}\sigma_{Pred}}{\sigma^2_{Gnd} + \sigma^2_{Pred} + (\mu_{Gnd} - \mu_{Pred})} \tag{1}$$

The greatest advantage of making MultiAffect a reproducible framework on Google's Colab notebooks is the ease for displaying the experimental results graphs and plots within the document. For this reason, we provide our code in this form, to allow the interested parties in evaluating the baseline model, while at the same displaying the obtained results, and being able to make modifications without struggling with the environment setup.

The results obtained from our reproducible framework for the classification task are two plots, one to visualize the accuracy while training and one for the training and testing loss; and a confusion matrix obtained while evaluating the model on the test data. On the other hand, for the regression tasks the results are displayed in a scatter plot that shows the correlation between the predicted and gold standard labels. As a reminder, in a correlation plot, if the points are highly concentrated over a 45-degree line, it means a correlation exists. On the other hand, if the concentration is found over a 135-degree line, it means a negative correlation exists. However, if the points are scattered as a cloud, this means there is no correlation and therefore the model did not learn correctly how to regress the data.

3 Experiments

Our aim with MultiAffect framework is to generate reproducible research for complex video analysis tasks. In order to test its generalizability, we performed experiments on two main tasks: affect recognition and video action recognition. The video action and affect recognition tasks are attacked through the training and testing of classification and regression models, respectively. One of the main goals of the proposed framework is to be able to perform both actions by only configuring a new set of variable without performing any change to the code. Another goal was to deliver results comparable to existing work. Since we are implementing a generic vanilla version of the algorithms, we do not expect to have outstanding results. Our framework focuses on simplicity, without abandoning the idea of flexibility which would allow the incorporation of more complex state-of-the-art algorithms.

3.1 Video Action Recognition

The selected video action recognition task to test the framework was a blooper recognition model of monologue videos. A blooper is a clip from a video that contains a mistake made by a person on the screen. Monologue videos by definition have constraints such as one person at a time and a fixed camera position. Detecting bloopers is not always a trivial task despite the sophistication of the tools. Nonetheless, our attempt to build a video blooper recognizer is through a video action classifier which considers features from different modalities found in the videos. Bloopers cannot be identified from instantaneous shots, so a long-term video analysis is needed. The conditions under which this classification task is performed are an ideal use case scenario of the MultiAffect framework.

The dataset used to perform this task was BlooperDB [9]. The BlooperDB is a long-term multimodal corpus for blooper recognition, which was gathered by selecting videos from YouTube that contain video bloopers. The videos were obtained by searching on YouTube using keywords like *bloopers*, *green screen*, *monologue*, among others. The videos obtained have multiple resolutions and the hosts speak different languages. The dataset is split into training, testing, and validation sets. The corpus is formed by a total of 596 video clips, 464 for training, 66 for validating, and 66 videos are found in the test set. Each video clip is annotated by two categorical labels, 0 (no blooper) and 1 (blooper). Each video clip lasts between one to three seconds. The dataset is stratified and contains an equal number of samples per each category. In Figure 3, we can see some examples from the BlooperDB dataset.

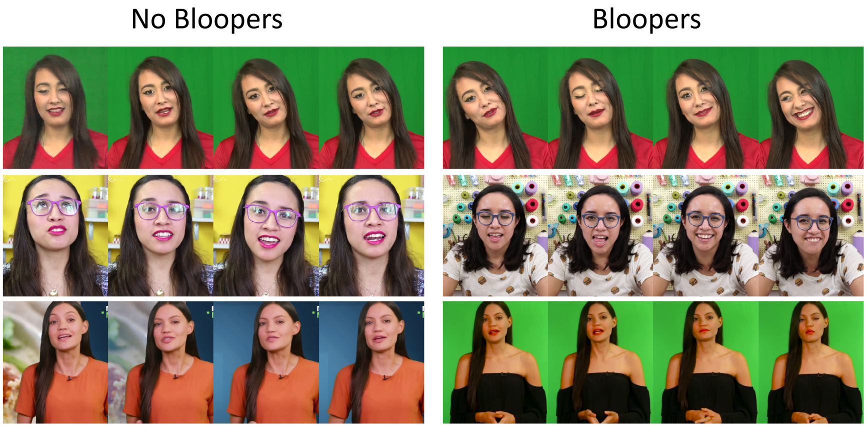


Fig. 3. Bloopers DB samples

We trained and evaluated our proposed framework on the BlooperDB. The multimodal model was trained under a maximum of 300 epochs. In order to prevent overfitting, an early-stopping policy was applied with a patience of 20 epochs. This policy stops the training if the validation loss does not drop after a certain number of epochs. A dropout strategy with a ratio of 0.5 for each fully connected layer was in place, and the learning rate was set to 1e-3.

Unimodal Approach We first evaluated the performance of all the models trained under a single modality. All the available framework modalities were used except for text. We decided not to evaluate text since our approach attempts to be language agnostic. Table 1 shows the results of the unimodal approach.

Multimodal Approach In this experiment, we titled the fusion of all the evaluated modalities as Quadmodal network. The Quadmodal contains the face embeddings, handcrafted face features, body pose embeddings, handcrafted body features, temporal emotion features, and general audio features. We trained

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 video action recognition task.

the Quadmodal network by concatenating all the multimodal features. Figure 4 shows the training performance metrics for different fusion strategies. We compare the early and late fusion strategies in Table 2. As we can see, the results demonstrate the learning benefits from early fused representation. Table 1 compares the performance of the unimodal and multimodal models. We also compare the fusion of the top three accuracy unimodal models against the Quadmodal network. The Quadmodal outperformed the model that contains face and audio features only, and performed better than the rest of the unimodal and multimodal models.

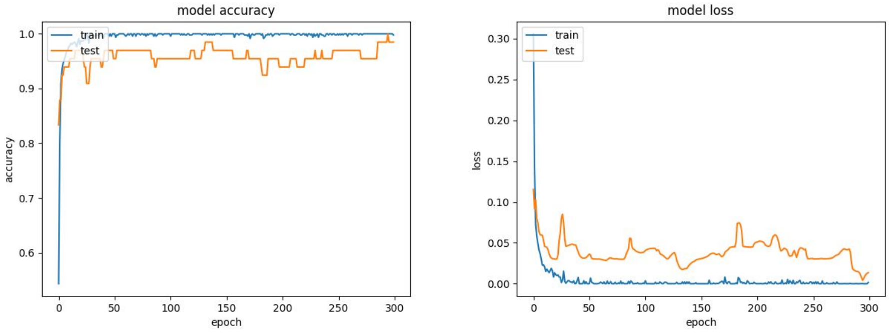


Fig. 4. Accuracy and loss plots of the Quadmodal model

Then we evaluate the confusion matrices for the quadmodal to analyze how was the performance per each category over the different sets. Figure 5 shows the confusion matrices of each set.

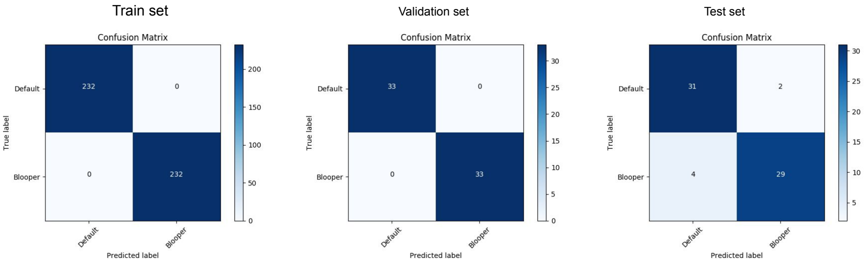


Fig. 5. Confusion matrices

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.

The results obtained were outstanding, achieving perfect scores in the validation scenario. We validate that MultiAffect is a viable framework for performing action recognition tasks.

3.2 Affect Recognition

In order to test how a regression task can be performed by using MultiAffect framework, we chose an affect recognition task that predicts valence and arousal as dimensional values. We chose to test our framework by performing the task required for an emotion recognition challenge. The 2018 One-Minute GradualEmotion Recognition (OMG-Emotion) challenge, which was held in conjunction with the IEEE World Congress on Computational Intelligence, encouraged participants to address long-term emotion recognition by integrating cues from multiple modalities, including facial expression, audio, and language. Intuitively, a multimodal inference network should be able to leverage information from each modality and their correlations and from there improve recognition over that achievable by a single modality network.

The challenge provided the One-Minute Gradual-Emotional Behavior dataset [13] that contains utterance-level videos from long-term emotional behaviors such as monologues, auditions, dialogues, and emotional scenes. The dataset consists of 5288 (train: 2442, validation: 617, test: 2229) segments from YouTube videos of about 1-minute each. We used the OMG Emotion dataset to recognize emotions in long-term natural utterances. Figure 6 shows some dataset samples.

For this task, we configured the framework similar to the video action recognition approach. The only parameters that were changed were the database source, label files, and the task type was switched from category to arousal or valence. We performed unimodal and multimodal approaches to evaluate the framework performance.

Fig. 6. OMG-Emotion dataset samples

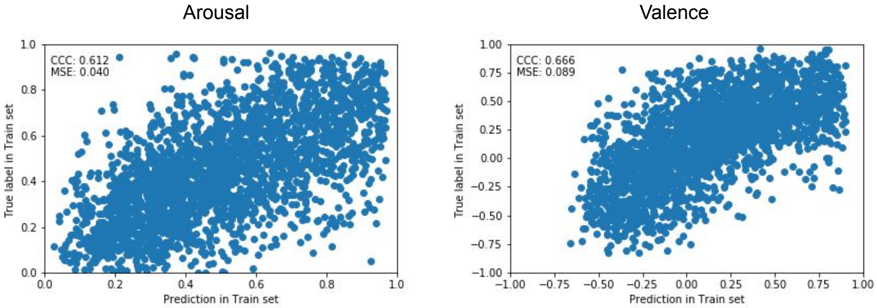


Fig. 7. Concordance Correlation Coefficient plots

Unimodal Approach The modalities used for these tasks were the same as the previously used during the action recognition task. Table 3 shows the results obtained for this task through an unimodal approach.

Multimodal Approach For this task we applied the same multimodal approach as we did for the action recognition task. In Table 4 we can see the results on the affective recognition task through a multimodal approach. Moreover, in Figure 7 we can see an example of how the evaluation data is represented in a plot.

The results obtained through our multimodal approach were satisfactory according to the OMG-Emotion challenge leaderboard [3], as it achieved a score similar to the majority of the submissions. This outcome was expected, since we are using common and general algorithms in each stage. If we were to explore for models more suitable for this task, it is expected to show an improvement in the performance of MultiAffect on this challenge. Thus, these results prove that our framework in an affective recognition scenario can satisfactorily perform a regression task and obtain competitive results off-the-shelf.

Features	ccc_arousal	ccc_valence
Emotion General	0.03	0.17
Emotion Temporal	0.04	0.06
Body Handcrafted	0.04	0.06
Body Deep	0.05	0.04
Body Fusion	0.05	0.02
Face Handcrafted	0.88	0.01
Face Deep	0.12	0.08
Face Fusion	0.09	0.08
Audio Temporal	0.14	0.27
Audio General	0.25	0.18
Text Feature	0.05	0.20
Text Fusion	0.10	0.28

Table 3. Unimodal results for the affect recognition task.

Features	ccc_arousal	ccc_valence
Trimodal (FAT)	0.22	0.17
Quadmodal (FATB)	0.17	0.17
Pentamodal (FATBE)	0.15	0.21

Table 4. Multimodal early fusion results. F=Face A=Audio T=Text B=Body E=Emotions

4 Conclusion

We presented MultiAffect, a reproducible research framework for multimodal affect and video action recognition. To achieve this goal, we designed a multipurpose interactive notebook capable of properly configuring an environment that can extract features, train, and test a multimodal model from annotated long-term videos. The framework allows users to perform these tasks from different modalities such as text, audio, instant emotions, face images, and body pose. Unimodal and multimodal approaches with different fusion strategy can be easily configured and processed. Classification and regression tasks over videos can be trained and evaluated with our proposed framework. We tested the framework in video action classification and affect recognition tasks. The tool was able to obtain results on both tasks by only adjusting the configuration. The results of the vanilla algorithms which conform our framework were satisfactory in the regression task and outstanding in the classification task. The framework is readily available online, it can be executed with no requirements from the client-side and is free of charge. We expect this framework contributes to the increase of reproducible research in this area by demonstrating the design principles followed in this work.

References

1. Multiaffect reproducible research framework - colaboratory. `https://bit.ly/multiaffect`. (Accessed on 05/30/2019)
2. 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)
3. Omg-emotion challenge. `https://www2.informatik.uni-hamburg.de/wtm/OMG-EmotionChallenge/`. (Accessed on 05/29/2019)
4. Openface. `https://cmusatyalab.github.io/openface/`. (Accessed on 04/26/2019)
5. opensmile/emobase2010.conf at master naxingyu/opensmile. `https://github.com/naxingyu/opensmile/blob/master/config/emobase2010.conf`. (Accessed on 04/26/2019)
6. 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)
7. priya-dwivedi/face_and_emotion_detection. `https://github.com/priya-dwivedi/face_and_emotion_detection`. (Accessed on 04/28/2019)
8. 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)
9. Video blooper dataset for automatic video editing — kaggle. `https://www.kaggle.com/toxtli/video-blooper-dataset-for-automatic-video-editing`. (Accessed on 05/31/2019)
10. Baker, M.: 1,500 scientists lift the lid on reproducibility. Nature News **533**(7604), 452 (2016)
11. Baltrušaitis, T., Robinson, P., Morency, L.P.: 3d constrained local model for rigid and non-rigid facial tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2610–2617. IEEE (2012)
12. Baltrušaitis, T., Robinson, P., Morency, L.P.: Openface: an open source facial behavior analysis toolkit. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10. IEEE (2016)
13. Barros, P., Churamani, N., Lakomkin, E., Siqueira, H., Sutherland, A., Wermter, S.: The omg-emotion behavior dataset. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2018)
14. Bengio, Y., Delalleau, O., Le Roux, N.: The curse of dimensionality for local kernel machines. Techn. Rep **1258** (2005)
15. Bengio, Y., LeCun, Y., et al.: Scaling learning algorithms towards ai. Large-scale kernel machines **34**(5), 1–41 (2007)
16. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008 (2018)
17. Delbrouck, J.B.: Transformer for emotion recognition. arXiv preprint arXiv:1805.02489 (2018)
18. Deng, D., Zhou, Y., Pi, J., Shi, B.E.: Multimodal utterance-level affect analysis using visual, audio and text features. arXiv preprint arXiv:1805.00625 (2018)
19. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 2008 international conference on web search and data mining, pp. 231–240. ACM (2008)

20. D'mello, S.K., Kory, J.: A review and meta-analysis of multimodal affect detection systems. ACM Computing Surveys (CSUR) **47**(3), 43 (2015)
21. Ekman, P.: Wallance. v. friesen." facial action coding system. ConSultingPsychologists PreSSInc (1978)
22. Ekman, P., Friesen, W.V.: The repertoire of nonverbal behavior: Categories, origins, usage, and coding. semiotica **1**(1), 49–98 (1969)
23. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM international conference on Multimedia, pp. 1459–1462. ACM (2010)
24. Fayek, H.M., Lech, M., Cavedon, L.: Towards real-time speech emotion recognition using deep neural networks. In: 2015 9th international conference on signal processing and communication systems (ICSPCS), pp. 1–5. IEEE (2015)
25. Ferreira, P.M., Pernes, D., Fernandes, K., Rebelo, A., Cardoso, J.S.: Dimensional emotion recognition using visual and textual cues. arXiv preprint arXiv:1805.01416 (2018)
26. Huang, Z., Dong, M., Mao, Q., Zhan, Y.: Speech emotion recognition using cnn. In: Proceedings of the 22nd ACM international conference on Multimedia, pp. 801–804. ACM (2014)
27. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B.E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J.B., Grout, J., Corlay, S., et al.: Jupyter notebooks-a publishing format for reproducible computational workflows. In: ELPUB, pp. 87–90 (2016)
28. Lawrence, I., Lin, K.: Assay validation using the concordance correlation coefficient. Biometrics pp. 599–604 (1992)
29. Munafò, M.R., Nosek, B.A., Bishop, D.V., Button, K.S., Chambers, C.D., Du Sert, N.P., Simonsohn, U., Wagenmakers, E.J., Ware, J.J., Ioannidis, J.P.: A manifesto for reproducible science. Nature human behaviour **1**(1), 0021 (2017)
30. Ortony, A., Turner, T.J.: What's basic about basic emotions? Psychological review **97**(3), 315 (1990)
31. Pantic, M., Sebe, N., Cohn, J.F., Huang, T.: Affective multimodal human-computer interaction. In: Proceedings of the 13th annual ACM international conference on Multimedia, pp. 669–676. ACM (2005)
32. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: bmvc, vol. 1, p. 6 (2015)
33. Peng, S., Zhang, L., Ban, Y., Fang, M., Winkler, S.: A deep network for arousal-valence emotion prediction with acoustic-visual cues. arXiv preprint arXiv:1805.00638 (2018)
34. Picard, R.W.: A ective computing (1997)
35. Ranganathan, H., Chakraborty, S., Panchanathan, S.: Multimodal emotion recognition using deep learning architectures. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–9. IEEE (2016)
36. Rouast, P.V., Adam, M., Chiong, R.: Deep learning for human affect recognition: Insights and new developments. IEEE Transactions on Affective Computing (2019)
37. Russell, J.A.: A circumplex model of affect. Journal of personality and social psychology **39**(6), 1161 (1980)
38. Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., Narayanan, S.S.: The interspeech 2010 paralinguistic challenge. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
39. Soleymani, M., Pantic, M., Pun, T.: Multimodal emotion recognition in response to videos. IEEE transactions on affective computing **3**(2), 211–223 (2012)

40. Triantafyllopoulos, A., Sagha, H., Eyben, F., Schuller, B.: audeering's approach to the one-minute-gradual emotion challenge. arXiv preprint arXiv:1805.01222 (2018)
41. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005)
42. Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., Rigoll, G.: Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. Image and Vision Computing **31**(2), 153–163 (2013)
43. Zhang, S., Zhang, S., Huang, T., Gao, W., Tian, Q.: Learning affective features with a hybrid deep model for audio–visual emotion recognition. IEEE Transactions on Circuits and Systems for Video Technology **28**(10), 3030–3043 (2018)
44. Zheng, Z., Cao, C., Chen, X., Xu, G.: Multimodal emotion recognition for one-minute-gradual emotion challenge. arXiv preprint arXiv:1805.01060 (2018)