PREDICTING STARTUP INVESTMENTS FROM FACIAL EXPRESSIONS OF ENTREPRENEURS USING RECURRENT NEURAL NETWORKS

PRIYANKA VINOD

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

WORD COUNT

8754

STUDENT NUMBER

2065580

COMMITTEE

prof. dr. Itır Önal Ertuğrul
prof. dr. Merel Jung

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 14, 2022

DATA SOURCE AND ETHICS STATEMENT

The work on this research did not involve collecting data from human participants or animals by the author. The data used for this thesis was previously collected, unpublished data[1]. The original owner of the data and the code used for this thesis retains ownership of the data and the code during and after the completion of this thesis. The images used in the thesis, when not produced by the author, were licensed under CC-BY-4.0. The code used for the purposes of the study is publicly available [2].

[1] (W. J. Liebregts, Urbig, & Jung, 2018-2021)

[2] https://github.com/priyankavinod/Thesis

ACKNOWLEDGEMENTS

Thank you for taking the time to read this thesis. Writing a master's thesis during a pandemic was definitely a challenging experience. This work could not have been completed without the help and support of certain people. First, I would like to express my sincere gratitude to my supervisor, prof. dr. Itır Önal, for her invaluable support and guidance in the completion of this work. I highly appreciate her constructive remarks in improving the quality of my research. I would also like to thank prof. dr. Merel Jung for taking the time to review my thesis. Special thanks to prof. dr. Werner Liebregts for providing me the data required for this work. Finally, I would like to thank my loving husband, Atul Nair, for the unwavering support and for believing in me.

CONTENTS

PREDICTING STARTUP INVESTMENTS FROM FACIAL EXPRESSIONS OF ENTREPRENEURS USING RECURRENT NEURAL NETWORKS

PRIYANKA VINOD

Abstract

Startups require key resources in terms of investments from external stakeholders to flourish their business. The decision to invest in a startup is highly indecisive for investors. They rely on verbal and non-verbal cues exhibited by the pitchers during pitching sessions. Facial expressions are an important means of non-verbal social communication, influencing investor funding decisions. This research explores how facial expressions during startup pitches could be used to predict startups securing investments. This is achieved by analysing facial expressions of entrepreneurs from pitching videos. The expressions are analysed by means of Facial Action Coding System (FACS), which describe facial expressions in a subjective manner in terms of action units (AUs). Two variants of recurrent neural networks (RNNs), the LSTM and GRU models were used for this purpose due to the sequential nature of data. Facial AUs were extracted using two software, OpenFace and FaceReader and their performance, along with the performance of various choices of AU inputs were also compared. The performance were compared using (macro) F1 scores. The study found that the GRU model performed the best, in combination with OpenFace inputs.

1 INTRODUCTION

This research aims to predict whether a startup secures investments or not from external stakeholders, by means of analysing the facial expressions of entrepreneurs in their startup pitching videos, using recurrent neural networks.

A startup is a young business is in its nascent stages of development. In recent years, there has been a discernible increase in the popularity of startup companies. The Netherlands for example, is a vibrant startup ecosystem, due to its favorable economic conditions and several initiatives by the government aiding to set up a business. Such a company is usually funded by its founders in the beginning. However, for their business venture to flourish, at some point in the development of a startup, the entrepreneurs require vital resources from external stakeholders (Zott & Huy, 2007). Impressing investors and securing resources are crucial for the success of a startup (Nagy, Pollack, Rutherford, & Lohrke, 2012). These resources are investments from stakeholders in the form of time, advice, human capital (e.g., employees, customers) or financial capital (Nagy et al., 2012; Zott & Huy, 2007).

Entrepreneurs secure investments by pitching their business ideas or presenting product prototypes to potential investors, such as venture capitalists, angel investors or lending specialists such as bankers (Nagy et al., 2012). Investors face considerable indecisiveness in determining whether to invest in a startup or not (Nagy et al., 2012). This is due to several factors, for instance, the lack of operating experience of a startup, and thus a proven successful performance history (Nagy et al., 2012; Zott & Huy, 2007). The stakeholder finds it difficult to judge the quality and long-term viability of the business and whether it is worth to commit their resources to the startup (Hellmann, 2005). Apart from the evident factors which are crucial in making such decisions to invest, which include namely the quality of the business proposal, novelty of the product idea and the passion of the entrepreneurs to name a few, there are several unseen factors that further influence the decision making of investors (W. Liebregts, Darnihamedani, Postma, & Atzmueller, 2020). Research has shown that aside from verbal cues, non-verbal cues also play a significant role in this process (A. Hu & Ma, 2021; W. Liebregts et al., 2020). Some examples of such cues are posture, eye gaze and facial expressions of the presenter (A. Hu & Ma, 2021; W. Liebregts et al., 2020). This study aims to analyse the videos of entrepreneur pitching sessions, specifically, the features of their facial expressions, in order to predict startups securing investments.

A common methodology used for the analysis of facial expressions is the Facial Action Coding System (FACS) (Ekman & Friesen, 1978), made of atomic units called Action Units (AUs). This research utilises AUs generated from videos of entrepreneur pitching sessions as feature variables. There are several software available which can do this feature extraction automatically. Two popular ones are OpenFace (Baltrusaitis, Zadeh, Lim, & Morency, 2018a) and FaceReader (Noldus, 2014). OpenFace is an open-source software, developed for academic research purposes. It is freely available and thus widely used. FaceReader on the contrary, is a commercial software package and is expensive. Recent years have seen an increase in interest in using FaceReader for academic research purposes (Lewinski, den Uyl, & Butler, 2014). There is not much literature available that compares the two software, as FaceReader is not as widely used as OpenFace. The research of Namba, Sato, and Yoshikawa (2021) recognises this gap in literature and attempts to compare their performance. They found that OpenFace performed better than FaceReader in detecting certain AUs.

The facial AUs extracted by the software represent data of sequential nature, as they are extracted from videos. Previous research work into the analysis of facial expressions and videos make use of deep learning techniques. The sequential nature of extracted features (AUs) demand the use of recurrent neural networks (RNNs) or convolutional neural networks (CNNs) (Ebrahimi Kahou, Michalski, Konda, Memisevic, & Pal, 2015). While these works look into the performance of different architectures, such as RNN (Graves, Mayer, Wimmer, Schmidhuber, & Radig, 2008), or combinations of RNNs with CNNs (Ebrahimi Kahou et al., 2015), there is a gap in literature in comparing the performance of two popular RNN architectures, LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014), especially in entrepreneurial decision-making analyses. Hence, this study attempts to make such performance comparisons and bridge the gap in literature by analysing facial expressions of entrepreneurs in pitching videos. Comparisons of OpenFace and FaceReader are also made, as there are not much studies comparing their performance. Additionally, since it is not known which choice of AUs are best for the prediction problem, different feature sets of facial AUs common to OpenFace and FaceReader are used as inputs for the models and their performance are compared.

The findings of this study has potential societal benefits, to both the entrepreneur as well as the investor. The work of Peleckis, Peleckienė, and Polajeva (2016) states that in business communications, a good negotiator must be both aware of, as well as be in control of the non-verbal communication they present to the opponent. If the entrepreneur has the knowledge of how their facial expressions, besides other factors, play a role in the outcome of their efforts to secure funding, it would be positive to their cause. This has greater application outside of entrepreneurial contexts, such as, it could be useful in other business scenarios of interaction, for instance, marketing a product to a target audience. For the investors, this knowledge is beneficial as they could better assess the viability of a presented idea, without being largely affected by biases formed from non-verbal cues.

The goal of this research is to predict the likelihood of a startup to secure investments by analysing the facial expressions of the entrepreneur (pitcher), when presenting (pitching) to potential investors. This is done by means of analysing video recordings of the pitcher during startup pitches and the rankings given by the investors to the pitch, indicating the probability whether they would invest in the startup or not. For this purpose, features are extracted from the videos in term of different AUs, using two software, OpenFace and FaceReader. The extracted features are then fed into the two RNN variants, namely, the LSTM and the GRU models. This is done in order to model the temporal information in the videos and predict the probability of investment.

To this end, the main research question of this study is formulated as follows:

> *To what extent can the likelihood of startup investment be predicted by analysing the facial expressions of entrepreneurs during pitches using Recurrent Neural Networks?*

In answering this research question, the following questions arise:

1. *How do LSTM and GRU models trained with facial action units compare in performance in predicting the probability of startups securing investments?*
2. *How do the facial action units from OpenFace and FaceReader influence the performance of the LSTM and GRU models?*
3. *How do different choices of facial action units influence the performance of the LSTM and GRU models?*

In order to answer the formulated research questions, several experiments were conducted. It was found that the GRU classifier, combined with OpenFace features of individual facial AUs corresponding to basic emotions (discussed in Section 2) gave the best predictive performance. Furthermore, it was found that GRU models performed better than LSTM models for this classification task. OpenFace AUs performed better than FaceReader AUs, specifically, with the individual AUs that correspond to the four basic emotions chosen as input.

The remainder of this paper is structured as follows: Section 2 provides an overview of the previous work done related to the topic and provides a context for this research. Section 3 summarises the theoretical background of the algorithms and evaluation method used for the study. Section 4 details the experimental setup, which includes description of the software and the dataset used, data preprocessing and the undertaken experimental procedures. Section 5 presents the results of the experiments. Section 6 discusses the findings of the experiments, limitation of the research and scope for future work. Section 7 concludes the paper.

2 RELATED WORK

The objective of this study, which is predicting whether a startup would secure investments from external stakeholders or not, falls under the broader domain of entrepreneurial research, specifically, decision-making in entrepreneurial contexts involving social interactions. There is a multitude of studies that focus on decision-making in entrepreneurial scenarios where human interaction is involved. The studies by W. Liebregts et al. (2020) and A. Hu and Ma (2021) for instance, particularly focuses on this topic. In their analysis of behavioural cues that influence entrepreneurial decisions, the authors propose that investors make decisions regarding startup funding on the basis of both verbal as well as non-verbal communication of the entrepreneurs during their pitches. They state that in entrepreneurial scenarios, non-verbal cues have a direct influence on decision-making. Non-verbal behaviour consists of cues such as gestures, posture, eye-gaze patterns, vocal behaviour and facial expressions of the person involved (A. Hu & Ma, 2021; W. Liebregts et al., 2020; Warnick, Davis, Allison, & Anglin, 2021). Amongst these cues, facial expressions are the most important, as emotions expressed through expressions of face are prominent means of social communication (Keltner, Sauter, Tracy, & Cowen, 2019; Lee & Anderson, 2016). According to Lee and Anderson (2016), facial expressions are particularly influential in scenarios of visual presentations, such as entrepreneurial pitching sessions. The work of Mehrabian (2017) finds that up to 55% of human communications is represented by facial expressions, which further empha-sises its significance.

The most common protocol used for the analysis of facial expressions is the Facial Action Coding System (FACS) (Ekman & Friesen, 1978). It is made of basic units called action units (AUs). AUs represent the smallest visually discernible facial muscle movement (Namba et al., 2021; Prince, Martin, Messinger, & Allen, 2015; Savran, Sankur, & Bilge, 2012). AUs are coded numerically, for instance, AU1 corresponds to raising the inner brow. AUs can represent facial activity, direction of eye gaze and head orientations; in this research only facial AUs are considered. The codes of frequently used AUs and their action descriptors are shown in Table 6 (Appendix

A, page 34). Combinations of AUs correspond to emotions, for instance, happiness and sadness. The list of emotions that can be characterised by combining different AUs in shown in Table 2 (Section 4.4) .

According to the basic emotion theory of Ekman and Friesen (1978), there are six basic emotions, which was found could be condensed into four distinguishable emotions, namely happiness, anger, fear and sadness (Jack, Garrod, & Schyns, 2014; Warnick et al., 2021). The facial expressions of the pitcher during entrepreneurial pitching scenarios could refer to either positive or to negative emotions (Warnick et al., 2021). A positive emotion associated with pitcher-investor interactions is that of happiness, whereas negative emotions associated with it include anger, fear or sadness (Warnick et al., 2021). The works of A. Hu and Ma (2021) and Warnick et al. (2021) finds that display of positive emotions may positively influence investor decisions regarding funding and vice versa for negative emotions.

AUs can be manually coded from still images or videos by certified FACS coders (Prince et al., 2015). However, this requires extensive professional knowledge. Currently, several specialised software are available to extract features from image or video sources. These software can be open source or commercial. Two popular automatic facial feature detection software are OpenFace (Baltrusaitis et al., 2018a) and FaceReader (Noldus, 2014). OpenFace is an open-source software whereas FaceReader is a proprietary software. These software provide their output in terms of several features, which also includes facial AUs. 18 different AUs can be obtained from OpenFace (1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 20, 23, 25, 26, 28, 45) compared to the 20 different AUs obtained from FaceReader (1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 18, 20, 23, 24, 25, 26, 27, 43) (Namba et al., 2021). OpenFace and FaceReader extract different facial features, with only 16 facial AUs (1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 20, 23, 25, 26) common to both of them (Fortin-Côté, Beaudin-Gagnon, Campeau-Lecours, Tremblay, & Jackson, 2019). Since OpenFace is freely available, it is widely used. Due to lack of transparency of their algorithms and training data (Baltrusaitis, Zadeh, Lim, & Morency, 2018b) and being expensive, FaceReader is not as widely used as OpenFace. Consequently, there are not many studies that compare the performance of the two software. The work of Namba et al. (2021) found OpenFace performed better than FaceReader in the detection of facial AUs. Lewinski et al. (2014) found in their analysis of two image datasets that FaceReader detected some AUs better than others.

Facial expressions could be analysed by looking into either static data (images) or dynamic data (videos) (Dhall, Ramana Murthy, Goecke, Joshi,

& Gedeon, 2015). The static nature of images does not capture how emotions of subjects vary over time, whereas video data captures the spatio-temporal progression of facial features, which is important in understanding the behaviour of subjects in videos (Deng, Chen, Zhou, & Shi, 2020; Ebrahimi Kahou et al., 2015). Facial AUs extracted from videos of pitchers represent data which is sequential in nature. In recent years, deep learning architectures have been seen to exhibit high performance in a variety of vital tasks related to facial and video analyses (Ebrahimi Kahou et al., 2015; B. Hu, Guo, Yang, Liu, & Xu, 2021). Use cases of such tasks include face recognition, human activity recognition and emotion recognition (Ebrahimi Kahou et al., 2015). Earlier, CNNs were used for the such analyses. However, these networks used by themselves rely on temporal averaging for the aggregation of visual features in the video data (Ebrahimi Kahou et al., 2015). It is in scenarios like these, which require the analysis of sequential data, that RNNs produce best results, as they allow the modelling of spatio-temporal evolution of features in video data (Ebrahimi Kahou et al., 2015). Unlike CNNs or other feed-forward network architectures, RNN architectures have one or more network layers connected to themselves (Abedi, Sadiq, & Nadher, 2020; Graves et al., 2008). These networks are able to make flexible use of temporal contexts, as the connections to their self (Figure 1) allow the network to construct internal representation of past events, thus learning long-term dependencies (Abedi et al., 2020; Gao & Glowacka, 2016; Graves et al., 2008).

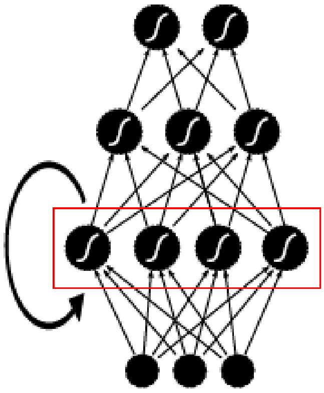


Figure 1: RNN architecture. Source: (Graves et al., 2008)

The classic RNN architecture however, cannot always be used to train on long-term sequences, as they suffer complications due to exploding and vanishing gradients (Abedi et al., 2020). This is solved by modifying the basic RNN architecture by including gated units. Two popular refined RNN architectures which makes use of such gated units are the LSTM (Hochreiter & Schmidhuber, 1997) and the GRU (Cho et al., 2014) networks

(Abedi et al., 2020; Gao & Glowacka, 2016; Graves et al., 2008). In the researches that employ analysis of facial expressions through video data, such as for the detection of pain (Rodriguez et al., 2017) or depression (B. Hu et al., 2021) and other works of facial video analysis, it was seen that the analyses were done using the RNN architectures; either LSTM or GRU, or combination of these architecture with CNN architectures (Abedi et al., 2020; Dhall et al., 2015; Donahue et al., 2015; Graves et al., 2008; Hans & Rao, 2021; B. Hu et al., 2021; Yüksel & Skarbek, 2019). There were however, no studies found that compared the performance of LSTM and GRU models for analysis of facial videos in entrepreneurial contexts. A search of related works that compared the performance of LSTM and GRU proved few interesting. The research of Huang, Fukuda, and Nishida (2019) compared performance of LSTM and GRU which were trained on facial AUs. They found that GRU outperformed LSTM when trained on a small dataset of 16 subjects. The comparison study of LSTM and GRU performance by Yang, Yu, and Zhou (2020) on a Yelp review dataset showed that GRU outperformed LSTM only on small datasets, whereas LSTM outperformed GRU on large datasets. The works of Yang et al. (2020) and Rana (2016) also found that GRU models were much faster LSTM, while maintaining comparable performance.

Several valuable insights were derived, in light of all the literature reviewed above, based on the which the research questions of this study were formulated. From the literature reviewed on investor funding decisions, it was concluded that facial expressions play a major role in interactive communications, and their importance in deciding the verdict of startup investments was recognised. It was understood that video data of pitching sessions provided better understanding of the progression of emotions rather than using static image data. In this respect, facial expressions of entrepreneurs were decided to be analysed by means of extracting AUs from pitching videos using OpenFace and FaceReader. The gap in literature in performance comparisons of OpenFace and FaceReader was identified and a research question was formulated towards that end. 16 AUs were found to be common to OpenFace and FaceReader software, but there was no study which indicated which AUs gave good performance for video data. Since combinations of AUs represented certain emotions pivotal to investment decisions (positive emotion of happiness and negative emotions corresponding to sadness, anger and fear), it was thought to be prudent to explore different feature sets of AUs, both individual and combined, to be used as inputs to the models being analysed to compare performance. This was another research question of the study. The final research question formulated was to compare the performance of the two RNN models, LSTM and GRU. The choice of these algorithms for this research was justified by their good performance on sequential data as suggested by reviewed literature, as well as to bridge the gap in studies comparing the models in the context of investor funding videos.

This work of this research has both societal and scientific relevance, of which the former is already discussed in Section 1. The scientific novelty of this research as compared to the reviewed studies is the comparisons made in terms of algorithms, data sources and choice of features in the context of entrepreneurial decision-making scenarios.

3 METHOD

This section provides a theoretical background of the modelling algorithms and the evaluation method used for this research. A brief overview of LSTM and GRU models is provided. This is followed by a summary of nested cross-validation.

3.1 *Long Short-Term Memory*

The LSTM model was first introduced in by Hochreiter and Schmidhuber (1997). LSTM is an advanced architecture of RNN (Gao & Glowacka, 2016) The flow of information in LSTM cells is similar to that of a RNN. However, the operations occurring within the cells are different. Such operations decide whether the LSTM retains or forgets the information. The key concept of LSTM is the cell state and the various gates present. Cell state acts as memory of the network, transferring the relative information down the sequence chain. This is achieved through a mechanism of adding or removing of information from cell state by means of gates. Gates learn what information is relevant to keep or forget during training. Two different activation functions that appear in LSTM are tanh and sigmoid.

The flow of information is regulated by means of gates in LSTM cell. LSTM consists of three gates:

- Forget gate
- Input gate
- Output gate

The forget gate decides which information should be discarded and which should be retained. This is achieved by providing information from the previous hidden state and information from the current input to the sigmoid function. The output would be between 0 and 1, where closer to 1

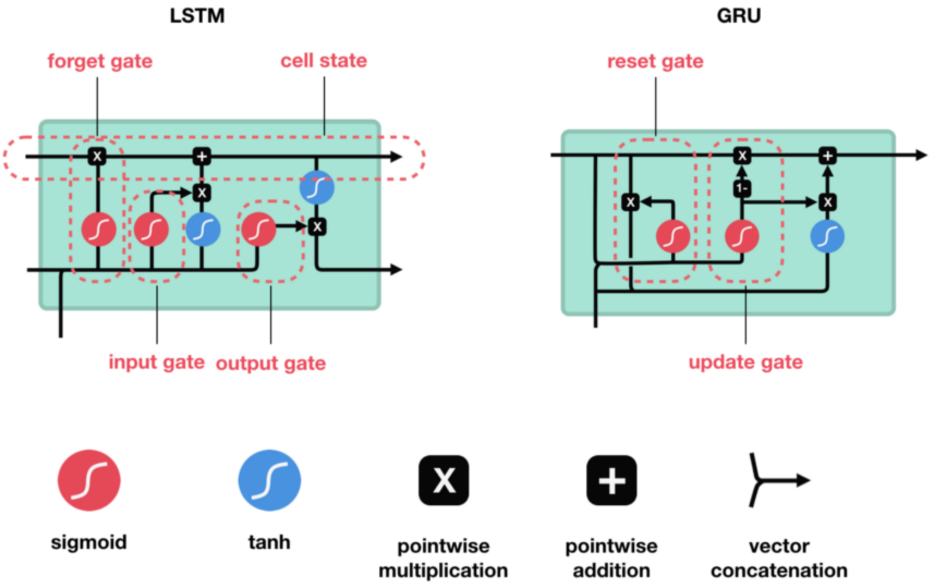


Figure 2: LSTM and GRU architecture. Source: Google (CC BY 4.0).

implies to retain and closer to 0 means to forget. Input gate updates the cell state. It has two activation functions. Previous hidden state and current input is passed to sigmoid function, which decides if the information has to be retained or discarded. They are also passed to tanh activation, which scales the values between -1 and 1. The output of both actuations are then multiplied. Cell state is calculated using outputs of forget and input gates. First the cell state is pointwise multiplied by the forget gate output. Then the output of input gate is pointwise added. Thus, only the important information is kept and we get the new cell state. The output gate decides what the next hidden state should be. Previous hidden state and the current input is passed to a sigmoid function, and the new cell state to a tanh function. These two are multiplied to get the new hidden state.

3.2 *Gated Recurrent Units*

The GRU model is newer than, but similar to the LSTM model. It was introduced by Cho et al. (2014). The difference between the two models is that, GRU has no cell state. It uses the hidden state to transfer information. GRU consists of two gates:

- Reset gate
- Update gate

The reset gate is another gate is used to decide how much past information to forget. The update gate acts similar to the forget and input gate of an LSTM by deciding decides what information to discard or retain. GRU's have fewer tensor operation than LSTM, which makes them faster to train.

3.3 *Nested Cross-Validation*

Nested cross-validation is a resampling technique used for model evaluation, which is bested suited for use when dealing with small datasets. This technique combines model hyperparameter optimisation along with model selection. Ideally, large datasets are suitable for modeling algorithms, as they yield unbiased estimate of the true model generalisation error. However, when datasets are small, the available data has to be used for both hyperparameter tuning and model selection. k-fold cross-validation could be used for problems as such. However, they may introduce bias into model performance estimates, as the same dataset is used for tuning and model selection (Raschka, 2018). Nested cross-validation solves this problem.

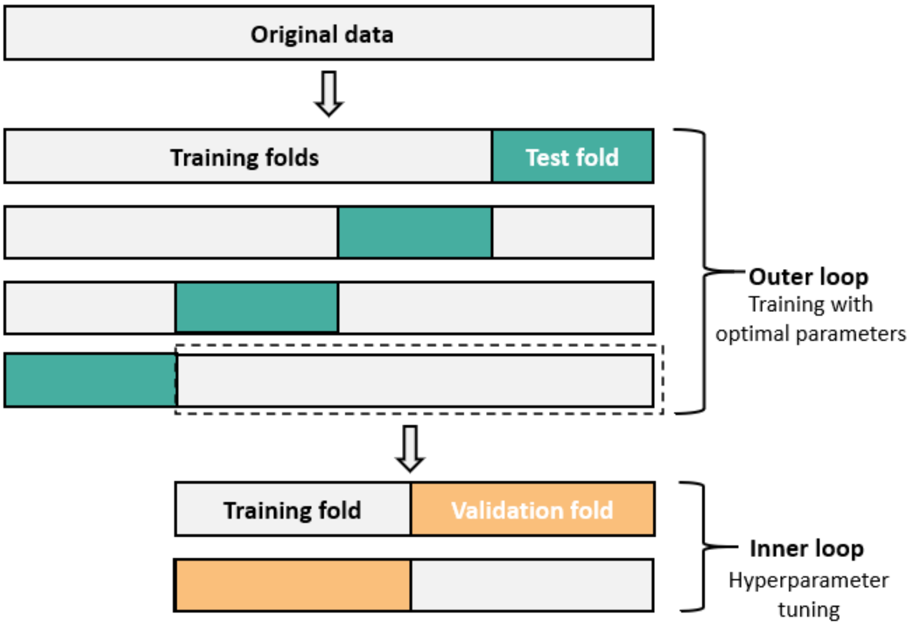


Figure 3: Illustration of nested cross-validation.

The working of nested cross-validation is illustrated in Figure 3. It can be seen that there are two loops; an outer loop and an inner loop, which are simply two k-fold cross-validation loops nested within each other. In the outer loop, the aggregate of all folds used for training the model is labeled as the training folds, and the holdout fold used for testing is called the test fold. For each iteration of outer loop, the training folds are further divided into training and validation folds as shown in the inner loop. Here, a hyperparameter tuning procedure such as grid search is used to find the optimal hyperparameters. The outer training folds are fitted to this settings and the performance is reported on the test fold. After all iterations of the outer loop are completed, the average score across all folds is reported as the performance.

4 EXPERIMENTAL SETUP

4.1 *Dataset*

The data used for this study is an unpublished dataset (W. J. Liebregts et al., 2018-2021), which was collected during the startup pitching competitions held at the Jheronimus Academy of Data Science (JADS). These startup competitions are held annually. The data consisted of video recordings as well as investor survey data collected during the event in the years between 2018 and 2020. There were three sets of pitches, with a combined total of 24 startup pitch recordings and their corresponding survey data across all the years considered. Facial features were extracted from the video data by means of both OpenFace (Baltrusaitis et al., 2018a), an open source software, as well as FaceReader (Noldus, 2014), a proprietary software. These data files were also available along with the video recordings and the investor survey data. All data were obtained from the data owner for the purpose of this research upon signing a non-disclosure agreement. In the following paragraphs, brief explanations are made of each of the different data sources obtained for this research.

The videos are in-person recordings. Each video is a recording of a representative of the startup (the pitcher), facing and pitching their startup idea to a panel of three judges (the investors). The videos are recorded by means of a high-resolution camera focused on the pitcher. Only the pitcher appears in the videos, not the investors. The pitches were approximately 3 minutes in duration, which was followed by a question-answer session by the judges. The total duration of each original video recording was around 12 to 15 minutes, with both the sessions combined. Only the pitching part of the videos was of interest to this research. The videos were viewed to note down the time-stamps of the pitching session, so as to process the data only for those intervals. This is described in more detail in Section 4.2.

Once the entrepreneur finished their pitch, each individual judge proceeded to score the startup idea on the basis of several factors, such as the passion of the entrepreneur, their preparedness, originality of the startup idea and the investor's likelihood to invest, to mention a few. Amongst these, the last variable mentioned, which is the probability score of the judge indicating their likelihood to invest in the startup was of interest to this research. The probability was scored on a scale of 0 to 100, with 0 denoting least likely and 100 denoting most likely to invest. For the purpose of this research, the scores from the judges were later combined and re-coded to a single binary score, where 0 indicated low and 1 indicated high chances respectively of the startup to receive investments from external stakeholders. This re-coded score is the target variable of this research. The re-coding and the intuition behind it is explained in more detail in Section 4.2.

The facial feature data extracted from the videos using OpenFace and FaceReader consisted of several variables, not all of which were relevant to this research. Examples of some of the variables that were available include those measuring different aspects such as eye gaze directions, head orientations, and facial AUs to name a few. Out of these, only the variables measuring the intensity of facial AUs activated when the pitcher presents to the investors were of interest to this research. These AUs, along with their combinations representing different emotions (Table 2, Section 4.4), constitute the feature variables of this study (Table 1, Section 4.4). Thus, the facial AUs extracted from the videos of the pitcher using both OpenFace and FaceReader, and the scores of the investors transformed into a binary format constitute the data analysed for this research.

4.2 *Data Preprocessing*

Data preprocessing was required as the available raw data could not be used directly for the research in order to train or evaluate the RNN models. It was performed in R (Team et al., 2013) using the RStudio environment (RStudio Team, 2020) and in Python (Van Rossum & Drake Jr, 1995) using the Jupyter Notebook environment (Kluyver et al., 2016).

The first step undertaken was to prepare the independent variables of the research. This meant the extraction of only the required number of rows and columns from all the participant data files and combining them to produce two files, representing the required features from OpenFace and FaceReader respectively. As described in Section 4.1, for each of the total 24 videos, facial features of the participant extracted using both OpenFace and FaceReader were available. This meant that each participant had two sets of data files, resulting in a total of 48 files. Only the columns indicating the intensity of facial AUs, which were present in both the software, were extracted. This corresponded to 16 AUs, as mentioned earlier in Section 2.

The next task was to extract the required rows. Each row of data represented one frame. The extraction of rows was necessary as the original video recordings, and thus the individual data files, consisted of both the pitch as well as the question-answer session of the pitcher with the investors. Facial AUs of the pitcher for only the duration of the pitching session was required. For this purpose, each video was viewed to determine how long each pitching session lasted and time-stamps were noted. It was observed that on average, a pitch lasted for 3 minutes. In order to keep the number of samples comparable, equal number of rows (4500 frames) were extracted for both OpenFace and FaceReader data. Exploratory data analysis (EDA) was performed on both OpenFace and FaceReader AUs, which is discussed in Section 4.3.

The next step undertaken was to prepare the dependent variable of the research, which is a binary-coded score indicating the likelihood of the startup to secure investments from external stakeholders. As mentioned in Section 4.1, the three judges scored each startup based on several factors. Among these, only the probability score of the likelihood of the judge to invest in the startup was of importance to this research. To this end, the scores from the three judges were combined to produce a binary-coded score. This final score was coded on the intuition that if at least one of the judges scored the startup above or equal to 50, the startup had a good chance to secure investments from an external stakeholder. Such startups were assigned a score of 1, denoting high probability of that startup to secure investments. On the contrary, if all the three judges scored a startup less than 50, it was assigned a score of 0, denoting a low probability of the venture to secure investments. The startups and their final binary-coded scores (target of research) are shown in Table 7 (Appendix A, page 35). EDA was performed on the target variable as well, as explained in Section 4.3.

The final preprocessing steps employed were feature normalisation and data reshaping. During EDA, it was found that the values of feature variables were in different ranges (described in Section 4.3). The values of facial AUs from both OpenFace and FaceReader were normalised to be in the range of [0, 1]. This was achieved using MinMaxScaler from scikit-learn (Pedregosa et al., 2011). Data reshaping was required as the RNNs used for the research accepts input in the form of a 3D tensor, with the format [batch, timesteps, feature] (Abadi et al., 2015). This was performed using the reshape function of NumPy (Harris et al., 2020), before feeding the input to the models. These steps are further mentioned in Section 4.3 and 4.5.1.

4.3 *Exploratory Data Analysis*

EDA was performed on both the feature as well as the target variables. The extracted OpenFace and FaceReader feature data (facial AUs) were further examined to check for missing values and outliers. No missing data was found for the OpenFace AUs. However, some FaceReader AUs were found to have values such as 'FIT FAILED' and 'FIND FAILED'. These were replaced with zeroes. It was also found that FaceReader did not detect AU9 and it had no values. Thus AU9 was dropped from both OpenFace and FaceReader. This meant that finally, 15 facial AUs were available as features for this research. They form the first set of features used (Set 1, Table 1). Three sets of features were further derived from this final set of features to simulate the RNN models and compare performance. The complete set of features used is shown in Table 1.

As briefly mentioned in Section 4.2, the investigation of AU values also revealed that ranges of AU intensities differed for OpenFace and FaceReader AUs. It was observed that the values of facial AU intensities in OpenFace were in a scale of 0 to 5, whereas that of FaceReader were in a scale of 0 to 1. In order to ensure all values were in the same range, all data were normalised before feeding to the models.

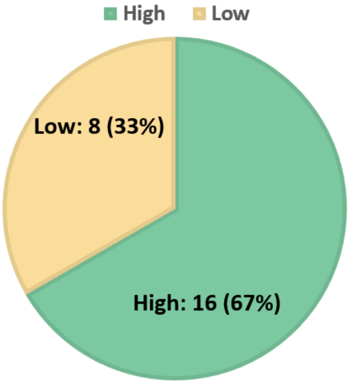


Figure 4: Class imbalance of the target variable

The target variable, which represents the likelihood of a startup to secure investments, was also explored. This inspection revealed the presence of imbalanced classes in the target variable. Figure 4 illustrates the class imbalance. It was seen that there were 16 positive (high probability of securing investments) and 8 negative (low probability of securing investments) classes respectively. The figure clearly shows that the count of the positive class is twice as higher than the count of the negative class. The imbalance in classes of the target variable was taken into account during model evaluation. This is explained in more detail in Sections 4.5.3 and 4.5.4.

4.4 *Features Used for the Research*

As explained in Section 4.3, 15 facial AUs common to both OpenFace and FaceReader were chosen as the independent variables of this research. However, this was not the only independent feature set used. Several other feature sets were developed based on the 15 AUs chosen after EDA. The complete feature sets are shown in Table 1.

Table 1: Feature sets

Feature set	AUs used
Set 1	1, 2, 4, 5, 6, 7, 10, 12, 14, 15, 17, 20, 23, 25, 26
Set 2	1, 2, 4, 5, 6, 7, 12, 15, 20, 23, 26
Set 3	6+12, 1+4+15, 4+5+7+23, 1+2+4+5+7+20+26
Set 4	6+12

As seen from Table 1, four sets of features were defined. This was done to compare the performance of the RNN models when the feature inputs are varied, which was one of the research sub-questions (Section 1). Set 1 in the table is the original 15 AUs chosen. Set 2, Set 3 and Set 4 are chosen based on the theory explained in Section 2. Set 2 represents the 11 facial AUs whose combinations correspond to the 4 basic emotions of happiness, sadness, anger and fear. Set 3 represents the actual combinations of the facial AUs in Set 2 that indicate the 4 basic emotions (Table 2). Set 4 represents the emotion of happiness. It was chosen as happiness is decisive in startups securing investments (A. Hu & Ma, 2021).

Table 2: Emotions and AUs (Ekman & Friesen, 1976)

Emotion	AU combination
Happiness	6 + 12
Sadness	1 + 4 + 15
Anger	4 + 5 + 7 + 23
Fear	1 + 2 + 4 + 5 + 7 + 20 + 26

4.5 *Experimental Procedure*

After prepossessing the raw data and conducting EDA, the extracted OpenFace and FaceReader features (Table 1) were ready to be fed into the RNN models. Two versions of RNNs were used for this research; namely the LSTM network and the GRU network. The sections below describe how the models are constructed, trained and evaluated. Tuning of the models is also discussed.

4.5.1 *Segment-Level and Video-Level Analysis*

Before diving into the particulars of the RNN models used, it is important to clarify a few points regarding the granularity of the input data. Four sets of features, as shown in Table 1, are used as inputs to the RNN models. Each row of features represents one frame of data. The LSTM and GRU models used in the research require inputs in the form of a 3D tensor, representing [batch, timesteps, feature] (Abadi et al., 2015). The feature data, thus, has to reshaped before feeding into the models. This reshaping means that the models are trained on a collection of frames, or on segments of data. This means the predictions are also made on the segment-level, rather than for a particular startup pitch video. Thus, two levels of granularity exist in the data; data at segment-level and data at video-level. Segment-level data represents the lower level of granularity whereas the video-level data (24 videos) represents higher level of granularity. The final results are reported at both levels.

4.5.2 *Modelling Algorithms*

The deep learning algorithms used for this study are LSTM and GRU, which are two popular versions of RNNs. Figure 15 and Figure 16 (Appendix A, page 37) shows the model summaries of the two models. It can be seen that both the LSTM and the GRU models has a single layer of LSTM and GRU units respectively. This layer is followed by a dropout layer, which is added to prevent overfitting of the model (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). After the dropout layer, there is an dense layer, which maps the input to the output.

The LSTM and GRU models were compiled using the Adam optimiser as the optimisation algorithm, which is a robust, gradient-based optimiser that requires little memory (Kingma & Ba, 2014). The loss and activation functions used were binary cross entropy and sigmoid function respectively. These were chosen as the research problem is binary classification. The models were evaluated using (stratified) nested cross-validation, which

Table 3: Hyperparameter tuning

Hyperparameter	Values explored
Epochs	50, 75, 100
Batch size	16, 32, 64
Dropout rate	0.2, 0.3, 0.4, 0.5
Learning rate	0.01, 0.001, 0.0001
Number of hidden units	32, 64, 128, 256

is explained in detail in Section 4.5.3. The evaluation metrics chosen for studying the performance of models are discussed in Section 4.5.4.

4.5.3 *Model Evaluation*

The models were evaluated using (stratified) nested cross-validation, as explained in Section 3.3. This method was used as the dataset was relatively small, with data of only 24 participants available. Stratification was ensured within the training and test folds of both loops, in order to maintain the distribution of the target variable in the folds. Figure 3 shows an illustration of the cross-validation structure used for this research. It can be seen that the outer loop performed 4-fold cross-validation, while the inner-loop performed 2-fold cross-validation. Hyperparamter tuning was undertaken by the operations in the inner folds. The hyperparameters and their values chosen to be tuned are listed in Table 3. Other model values were left at default. Grid search was used to find the optimal parameters. The training folds of outer loop were refitted with the best hyperparameters found in inner folds, and the performance for that fold was reported on hold out test fold.

It is vital to note that the split of data into training and test folds was done in a subject-independent manner, such that the data of a participant video appears in either fold, but not both. This was important because a split made as such would introduce bias in the data. The resulting model would display an elevated performance as it will no longer be evaluated on unseen data.

4.5.4 *Evaluation Metrics*

The performance of the models was evaluated by computing several evaluation metrics. Macro averaged F1 score was the metric chosen to compare model performance, as it gives equal importance to both classes and is useful for problems with imbalanced classes (Koyejo, Natarajan, Ravikumar, & Dhillon, 2014). Other metrics, such as accuracy and ROC

AUC (Area Under the Receiver Operating Characteristic Curve) score were also noted (at segment-level). The F1 scores are computed at both segment-level and video-level. Each model was run with both OpenFace and FaceReader feature data and evaluated using nested cross-validation as described in Section 4.5.3. Below are the formulas used to calculate the F1 scores of each class, which are averaged to get the macro F1 score.

$$Precision = \frac{TP}{TP + FP} \quad \text{and} \quad Recall = \frac{TP}{TP + FN}$$

$$F1\ score = 2\ .\ \frac{Precision\ .\ Recall}{Precision + Recall}$$

where TP = True Positive, TN = True Negative, FN = False Negative, FP = False Positive.

4.6 *Software*

This section summarises all the algorithms and software packages used for this research work. This study was predominantly performed using Python (3.7.9.) (Van Rossum & Drake Jr, 1995). R (4.1.0) (Team et al., 2013) and RStudio (1.4.1717) (RStudio Team, 2020) were used for initial data preprocessing and EDA. Models were created and executed using Jupyter Notebook (6.4.3) (Kluyver et al., 2016). The libraries used in Jupyter are Tensorflow (Abadi et al., 2015), Keras (Chollet et al., 2015), Pandas (pandas development team, 2020), NumPy (Harris et al., 2020) and Scikit-learn (Pedregosa et al., 2011).

5 RESULTS

The findings of the research are presented in this section. Several models were run using the different feature sets from OpenFace and FaceReader (Table 1). This resulted in a total of 16 models, with 4 models per feature set. In the following sections, the results found for each feature set, when given as input to the LSTM and the GRU models are analysed. The ensuing sections summarises the results in alignment with the research questions of this study, such that:

- The performance of LSTM and GRU are compared
- The performance of OpenFace and FaceReader are compared
- The performance of different feature sets used are compared

Table 4: Performance metrics of all models at segment-level

Input	Models	F1 score		ROC AUC score	
		OpenFace	FaceReader	OpenFace	FaceReader
Set 1	LSTM	0.51	0.42	0.59	0.39
	GRU	0.53	0.45	0.52	0.43
Set 2	LSTM	0.52	0.40	0.60	0.43
	GRU	**0.56**	0.43	**0.61**	0.48
Set 3	LSTM	0.43	0.42	0.48	0.50
	GRU	0.37	0.41	0.38	0.45
Set 4	LSTM	0.39	0.39	0.46	0.44
	GRU	0.40	0.41	0.46	0.57

Table 5: Performance metrics of all models at video-level

Input	Models	F1 score	
		OpenFace	FaceReader
Set 1	LSTM	0.49	0.36
	GRU	0.51	0.37
Set 2	LSTM	0.56	0.38
	GRU	**0.63**	0.36
Set 3	LSTM	0.40	0.40
	GRU	0.30	0.40
Set 4	LSTM	0.38	0.40
	GRU	0.40	0.40

The results found are compared with the baseline model, which is chosen as the LSTM model, with its input as the first feature set (Set 1, Table 1) corresponding to OpenFace. This choice of baseline was made as the LSTM model is more frequently used than GRU. OpenFace AUs was chosen for the same reason, and feature Set 1 was chosen as all other feature sets are derived from it. All models are evaluated using (stratified) nested cross-validation as explained in Section 4.5.3 and their performance are evaluated based on (macro) F1 scores. The F1 scores and ROC AUC scores of all models at the segment-level are shown in Table 4. Model performance at the video-level are shown in Table 5. The best scores are highlighted in both tables. Accuracy scores of the models at segment and video levels are shown in the Table 8 (Appendix A, page 36). Since nested cross-validation was used for model evaluation, all metrics reported in the tables are the average values across all folds. In the following sections, the F1 scores discussed are all at the segment-level, unless stated otherwise.

5.1 *Feature Set 1*

The first set of features used as input to the LSTM and GRU models was all AUs common to OpenFace and FaceReader. As explained earlier in Section 4.3, although 16 AUs are common to the two software, AU9 was not included, thus this feature set consisted of 15 facial AUs.

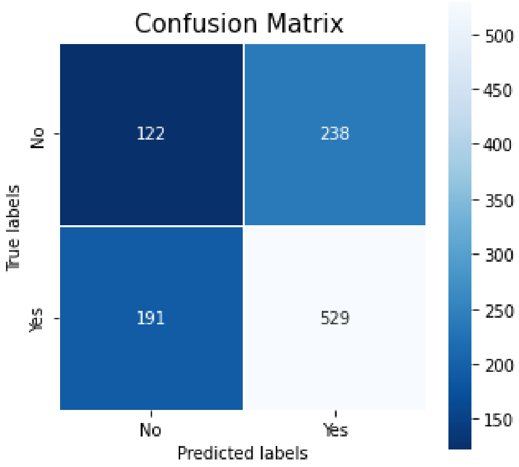


Figure 5: Confusion matrix at segment-level of GRU model with OpenFace input

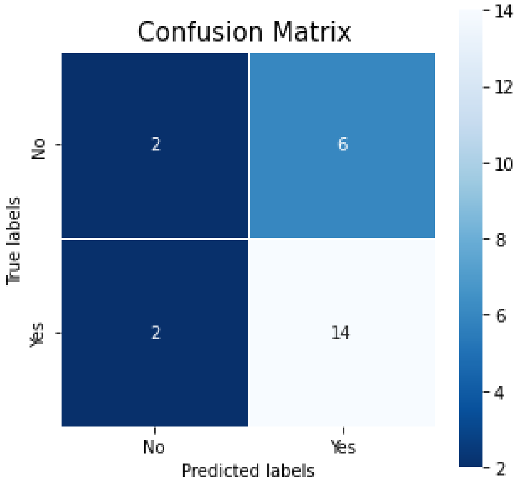


Figure 6: Confusion matrix at video-level of GRU model with OpenFace input

It was observed that out of the four models ran using this feature set as input, GRU with OpenFace AUs showed the best overall performance, with a F1 score of 0.53. This was also reflected at the video-level (F1 score = 0.51). The performance of the GRU model was higher than that of LSTM with OpenFace input, which was chosen as the baseline, the F1 score of which was 0.51. The performance of LSTM and GRU models with FaceReader input were comparable, with their F1 scores being 0.42 and 0.45 respectively. The lowest performance for this feature set was observed for LSTM with FaceReader input. GRU was seen to perform better than LSTM for both OpenFace and FaceReader inputs. When comparing OpenFace and FaceReader, the F1 scores were higher when OpenFace input was used. The confusion matrices [1] for the best performing model at both segment and video-levels are shown in Figure 5 and Figure 6.

5.2 *Feature Set 2*

The second set of features used as input to the LSTM and GRU models was all the facial AUs whose combinations made up the four basic emotions, namely, happiness, anger, fear and sadness. This corresponded to 11 AUs common to OpenFace and FaceReader (Table 1).

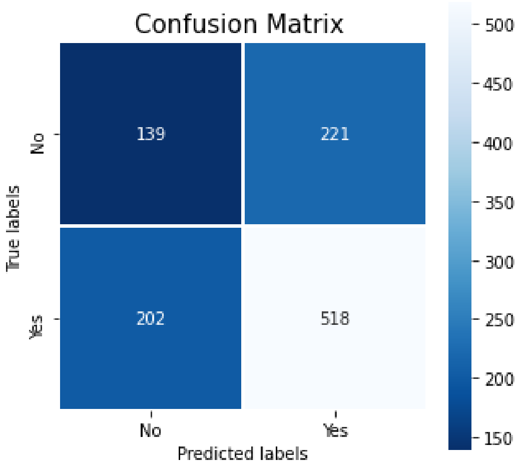


Figure 7: Confusion matrix at segment-level of GRU model with OpenFace input

It was found that for this selection of features, GRU performed better than LSTM for OpenFace data, with F1 scores of 0.56 and 0.52 respectively. A similar trend was seen for FaceReader input, with F1 scores of GRU and

[1] 'No' indicates low probability of investment (class 0), 'Yes' indicates high probability of investment (class 1)

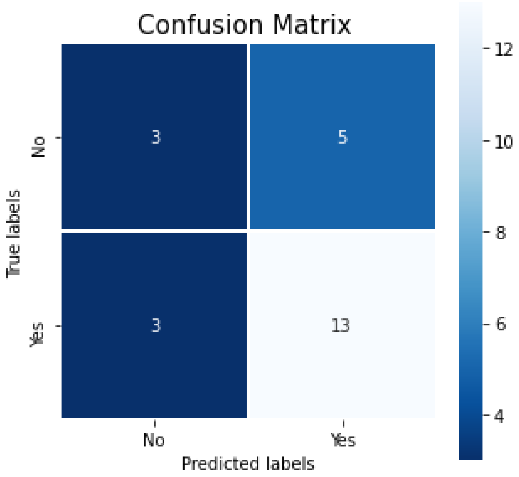


Figure 8: Confusion matrix at video-level of GRU model with OpenFace input

LSTM models being 0.43 and 0.40 respectively. Again, the best overall performance for the feature set was displayed by GRU with OpenFace input, which is higher than the performance of the baseline model (F1 score = 0.51). This was also the highest performance seen amongst all the 16 models which were run. The highest video-level performance was also found corresponding to this model (F1 score = 0.63). The confusion matrices for the best performing model at the segment and video-levels are shown in Figure 7 and Figure 8 respectively. As for feature Set 1, the performance of GRU classifier is better than LSTM for this feature set as well. Also, the performance were better with OpenFace AUs as inputs than FaceReader AUs. This is noted at both segment-level and video-level.

5.3 *Feature Set 3*

This set of features consisted of combinations of facial AUs indicating the basic emotions, instead of the individual AUs as chosen for Set 2. The four AU combinations correspond to the emotions of happiness, anger, fear and sadness respectively (Table 1).

For this feature set as input, the performance across all models were generally poor. However, now it was seen that out of the four models run for this set, LSTM with OpenFace input performed the best, with a F1 score of 0.43. This performance was comparable to the performance of LSTM and GRU with FaceReader inputs, whose F1 scores were seen to be 0.42 and 0.41 respectively. The lowest performance was seen for GRU with OpenFace input (F1 score = 0.37). It was observed that LSTM model performed better than GRU and that the performance of OpenFace

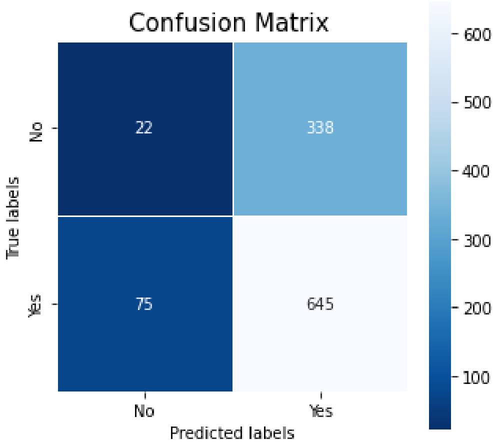


Figure 9: Confusion matrix at segment-level of LSTM model with OpenFace input

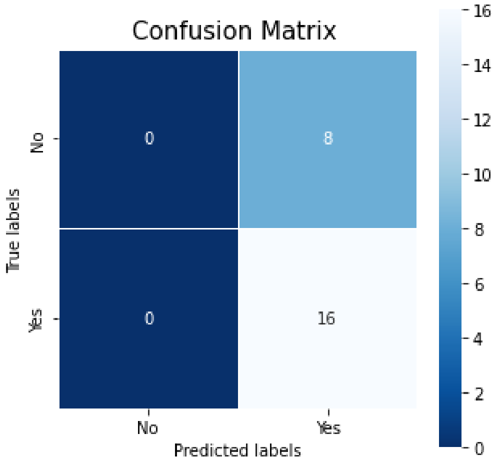


Figure 10: Confusion matrix at video-level of LSTM model with OpenFace input

was better than FaceReader for LSTM and vice versa for GRU. Confusion matrices of the best model is shown in Figure 9 and Figure 10.

5.4 *Feature Set 4*

This feature set consisted of only one feature variable, which is the combination of AUs that denote the emotion of happiness (Table 1).
The performance of all four models for this feature set were also found to be poor, and they were comparable. For the first time, the best performing model was seen to be GRU with FaceReader input, with a F1 score of 0.41. GRU models on OpenFace and FaceReader inputs performed better than the LSTM models, both of which had F1 scores of 0.39. Performance of

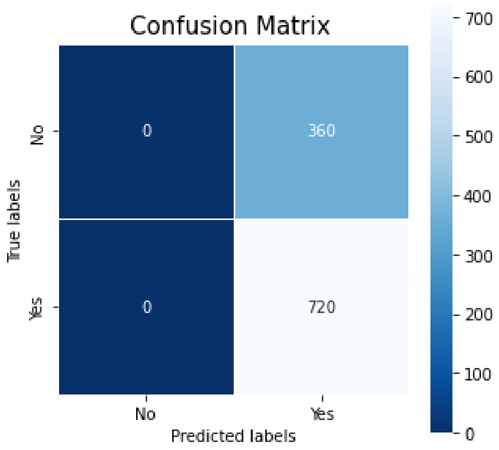


Figure 11: Confusion matrix at segment-level of GRU model with FaceReader input

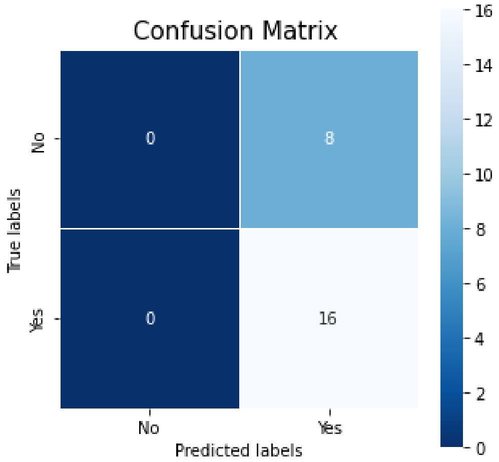


Figure 12: Confusion matrix at video-level of GRU model with FaceReader input

OpenFace and FaceReader data was close. The confusion matrices of the best performing model is shown in Figure 11 and Figure 12.

5.5 *Comparisons with Baseline Model*

As mentioned previously, the baseline model was chosen to be the LSTM model with OpenFace input of feature Set 1. It was seen that across all the four feature set inputs used, the best performing model was found when feature Set 2 input of OpenFace was used in combination with the GRU model. It had an F1 score of 0.56 at the segment level and 0.63 at the video-level. Mean ROC curve across all cross-validation folds for the best model is plotted in Figure 14 (Appendix A, page 37). None of the best models of other feature sets, except for Set 2, outperformed the baseline model. Figure 13 illustrates the performance of the best models found for each feature set, along with the baseline model. It can be clearly seen that out the four feature sets tested, feature Set 2 gave the best results.

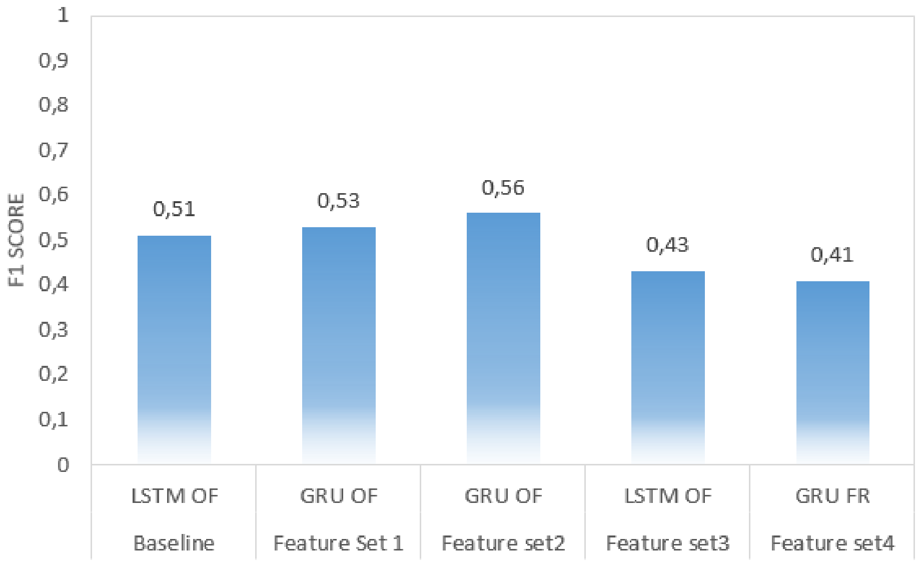


Figure 13: Comparison of baseline model with the best models of all feature sets

6 DISCUSSION

This section discusses the findings of the research. The goals of the study is revisited and the findings of research are presented according to the research questions that were defined. This is followed by a discussion of the limitations of the research and scope for future work. The contributions of the research are also highlighted.

6.1 *Goals and Findings of the Research*

The primary goal of the research was to predict the likelihood of a startup securing investments from external stakeholders, by analysing the facial expressions of entrepreneurs during pitching sessions, using RNNs. The features used for this purpose were facial features extracted from startup pitching videos in terms of AUs, coded according to FACS. These AUs were extracted using OpenFace and FaceReader. The target variable was a binary-coded variable indicating the likelihood of investment, coded

1 and 0, denoting high and low probability of securing investments respectively. Two popular variants of RNNs were chosen to solve this binary classification problem, namely, the LSTM and the GRU models. Several combinations of AUs were used as inputs to the models to compare performance. In order to answer the main research question of this study, three research sub-questions were formulated. The following paragraphs discusses the findings of the research based on each sub-question. All discussed performance metric are the macro F1 scores at segment-level, unless stated otherwise.

The first research question explored which RNN model performed best for the task at hand. It was seen that generally, GRU outperformed LSTM. The best performing model across all four feature sets was found to be the GRU model with OpenFace feature Set 2 as input, which had an F1 score of 0.56 at the segment-level and F1 score of 0.63 at the video-level. The highest performance displayed by LSTM model was an F1 score of 0.52. There was no literature found that compared the performance of LSTM and GRU models in the context of entrepreneurial decision making. The finding of GRU outperforming LSTM in this research may be compared to the findings of the work Huang et al. (2019), which compared LSTM and GRU models trained on facial AUs for a different task and for a small dataset. Their results showed GRU outperforming LSTM for a dataset of 16 subjects, which is comparable to the findings of this research with a dataset of 24 subjects. The finding is also in line with the work of Yang et al. (2020), which found that GRU performed better on smaller datasets, whereas LSTM performed better on larger datasets. The performance of GRU was better than LSTM in terms of time taken for execution as well. For the same input features, GRU was seen to be perform 31.2% faster than the corresponding LSTM models, which is in line with the findings of Yang et al. (2020), who found GRU was 29.29% faster than LSTM for the same input. This is likely due to the structural differences of LSTM and GRU, with GRU having one gate lesser than LSTM, and thus, lesser internal computations are performed (Rana, 2016).

The second sub-research question focused on the data sources (software) used for the research. This question explored which of the two software performed the best with the chosen models. It was seen that OpenFace generally performed better than FaceReader for all the experiments. This could be due to difference in the values of AU intensities detected by the two software. The AUs selected from OpenFace generally had higher values than the ones detected by FaceReader. This finding could be further explained by the work of Lewinski et al. (2014), who found that certain FaceReader AUs (7, 10, 20, 23, 24) gave poor performance on two image datasets they considered for analysis. Out of those AUs, three AUs (7, 20, 23) are used in three out of the four feature sets used for this research.

The third sub-research question explored which AUs chosen as feature variables for the models could yield better performance. This comparison was made as no literature was found reviewing a comparison of this nature. Four feature sets, two of which were individual AUs and two were combinations of AUs indicating different emotions were defined as features. It was seen that the feature set of individual AUs corresponding to basic emotions (Set 2) gave the best performance. It was interesting to note that the two feature sets of individual AUs (Set 1 and Set 2) performed better than combination of AUs (Set 3 and Set 4). This insight could be useful for future studies.

6.2 *Limitations and Recommendations for Future Work*

The main limitation of the research is a lack of ample data. It was observed that both GRU and LSTM models classified the positive classes more effectively than negative classes, which can be seen from the confusion matrices in Section 5. This is reflected in the overall low F1 scores. One reason the performance of the classifiers could not be improved even after tuning for hyperparameters could be the lack of training data. Both LSTM and GRU models were trained and tested on the limited number of samples (24 videos). It was seen that the models were overfitting in the cross-validation folds. Getting more data could solve the overfitting problem. Additionally, deep learning algorithms generally require large amounts of training data to learn patterns from them. This also explains the large variance seen in the accuracy values of the models (Table 8). With more training data available, the accuracy and F1 scores of the models could be improved. Despite this limitation, the results of the research are indeed insightful and lay groundwork for using RNNs for prediction of entrepreneur funding decisions.

With that said, some recommendations for future work have to be discussed, as not all possibilities of solving the classification task were explored in this research. First would be to apply the methods explored in this research on a larger dataset, which could improve the performance of the classifiers. If a sufficiently large dataset is available, a separate hold out test set can be kept aside at the beginning, before entering cross-validation, so as to understand the true performance of the model on unseen data. The resampling technique of cross-validation gives an estimate of model performance on unseen data by taking a portion of training data as test data. If a separate hold out test is available, the final performance could be evaluated on that set. This will corroborate the skill of the model that was estimated by resampling.

Second, intuitions other than what was suggested in this research to binary-code the target variable could be investigated. Third, the stochastic nature of the algorithms results in different results every time for the same data as the models are randomly initialised every time. This can also contribute to the variance seen across the folds and in the value of the mean performance metric. Due to time constraints, this research was conducted for only 2 repeats of nested cross-validation. Ideally, it would be better to explore increasing the number of repeats, as it could yield better estimate of mean performance of the model. Finally, the future research could explore altering the model configurations, such as adding more LSTM or GRU layers if more data is available, or even try combination of algorithms such as CNN with LSTM or GRU and compare findings.

6.3 *Contribution of the Research*

This research explored how entrepreneurial pitching videos could be examined to predict investments. The research bridges the gap in the performance comparison of LSTM and GRU networks for video analysis. It also compares two popular feature extraction software and gives an indication of their performance. It provides an insight into which AUs could be used for predictive tasks as such.

7 CONCLUSION

The objective of this study was to predict the likelihood of whether a startup would secure investments or not, predicted by analysing the facial expressions of entrepreneurs from pitching video data. Data was extracted from videos in terms of facial AUs using two software; OpenFace and FaceReader. Different combinations of AUs were defined. Two popular variants of RNN, the LSTM and GRU models were used for the analysis. The model performance for each set of input features was investigated on the basis of (macro) F1 scores, to account for the class imbalance of the target variable. It was found that the overall best performance was achieved when using the GRU model in combination with OpenFace data. The best F1 score that was achieved was 0.56 at video segment-level.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/` (Software available from tensorflow.org)

Abedi, W. M. S., Sadiq, A. T., & Nadher, I. (2020). Modified cnn-lstm for pain facial expressions recognition.

Baltrusaitis, T., Zadeh, A., Lim, Y. C., & Morency, L.-P. (2018a). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th ieee international conference on automatic face & gesture recognition (fg 2018)* (pp. 59–66).

Baltrusaitis, T., Zadeh, A., Lim, Y. C., & Morency, L.-P. (2018b). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th ieee international conference on automatic face & gesture recognition (fg 2018)* (pp. 59–66).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F., et al. (2015). *Keras.* GitHub. Retrieved from `https://github.com/fchollet/keras`

Deng, D., Chen, Z., Zhou, Y., & Shi, B. (2020). Mimamo net: Integrating micro-and macro-motion for video emotion recognition. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 2621–2628).

Dhall, A., Ramana Murthy, O., Goecke, R., Joshi, J., & Gedeon, T. (2015). Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 acm on international conference on multimodal interaction* (pp. 423–426).

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2625–2634).

Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., & Pal, C. (2015). Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 acm on international conference on multimodal interaction* (pp. 467–474).

Ekman, P., & Friesen, W. V. (1976). Measuring facial movement. *Environmental psychology and nonverbal behavior*, *1*(1), 56–75.

Ekman, P., & Friesen, W. V. (1978). *Facial action coding system: Investigator's guide.* Consulting Psychologists Press.

Fortin-Côté, A., Beaudin-Gagnon, N., Campeau-Lecours, A., Tremblay, S.,

& Jackson, P. L. (2019). Affective computing out-of-the-lab: The cost of low cost. In *2019 ieee international conference on systems, man and cybernetics (smc)* (pp. 4137–4142).

Gao, Y., & Glowacka, D. (2016). Deep gate recurrent neural network. In *Asian conference on machine learning* (pp. 350–365).

Graves, A., Mayer, C., Wimmer, M., Schmidhuber, J., & Radig, B. (2008). Facial expression recognition with recurrent neural networks. In *Proceedings of the international workshop on cognition for technical systems.*

Hans, A. S. A., & Rao, S. (2021). A cnn-lstm based deep neural networks for facial emotion detection in videos. *INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES*, *7*(1), 11–20.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

Hellmann, T. (2005). *Entrepreneurship in the theory of the firm: the process of obtaining resources* (Tech. Rep.). Citeseer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hu, A., & Ma, S. (2021). Persuading investors: A video-based study. *Available at SSRN 3583898*.

Hu, B., Guo, W., Yang, H., Liu, Z., & Xu, Y. (2021). Deep neural networks for depression recognition based on 2d and 3d facial expressions under emotional stimulus tasks. *Frontiers in Neuroscience*, *15*, 342.

Huang, H.-H., Fukuda, M., & Nishida, T. (2019). Toward rnn based micro non-verbal behavior generation for virtual listener agents. In *International conference on human-computer interaction* (pp. 53–63).

Jack, R. E., Garrod, O. G., & Schyns, P. G. (2014). Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. *Current biology*, *24*(2), 187–192.

Keltner, D., Sauter, D., Tracy, J., & Cowen, A. (2019). Emotional expression: Advances in basic emotion theory. *Journal of nonverbal behavior*, 1–28.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., ... Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (p. 87 - 90).

Koyejo, O., Natarajan, N., Ravikumar, P., & Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. In *Nips* (Vol. 27, pp. 2744–2752).

Lee, D., & Anderson, A. (2016). Form and function in facial expressive behavior. *Handbook of emotions*, 495–509.

Lewinski, P., den Uyl, T. M., & Butler, C. (2014). Automated facial coding: validation of basic emotions and facs aus in facereader. *Journal of Neuroscience, Psychology, and Economics*, *7*(4), 227.

Liebregts, W., Darnihamedani, P., Postma, E., & Atzmueller, M. (2020). The promise of social signal processing for research on decision-making in entrepreneurial contexts. *Small Business Economics*, *55*(3), 589–605.

Liebregts, W. J., Urbig, D., & Jung, M. M. (2018-2021). *Survey and video data regarding entrepreneurial pitches and investment decisions.* (Unpublished raw data)

Mehrabian, A. (2017). Communication without words. In *Communication theory* (pp. 193–200). Routledge.

Nagy, B. G., Pollack, J. M., Rutherford, M. W., & Lohrke, F. T. (2012). The influence of entrepreneurs' credentials and impression management behaviors on perceptions of new venture legitimacy. *Entrepreneurship Theory and Practice*, *36*(5), 941–965.

Namba, S., Sato, W., & Yoshikawa, S. (2021). Viewpoint robustness of automated facial action unit detection systems. *Applied Sciences*, *11*(23), 11171.

Noldus, F. (2014). Tool for automatic analysis of facial expression: Version 6.0. *Wageningen, the Netherlands: Noldus Information Technology BV*.

pandas development team, T. (2020, February). *pandas-dev/pandas: Pandas.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3509134 doi: 10.5281/zenodo.3509134

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peleckis, K., Peleckienė, V., & Polajeva, T. (2016). Towards sustainable entrepreneurship: role of nonverbal communication in business negotiations. *Entrepreneurship and sustainability issues*, *4*(2), 228.

Prince, E. B., Martin, K. B., Messinger, D. S., & Allen, M. (2015). *Facial action coding system.* The Sage Encyclopedia of Communication Research Methods.

Rana, R. (2016). Gated recurrent unit (gru) for emotion classification from noisy speech. *arXiv preprint arXiv:1612.07778*.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

Rodriguez, P., Cucurull, G., Gonzàlez, J., Gonfaus, J. M., Nasrollahi, K., Moeslund, T. B., & Roca, F. X. (2017). Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE transactions on cybernetics*.

RStudio Team. (2020). Rstudio: Integrated development environment for r [Computer software manual]. Boston, MA. Retrieved from http://www.rstudio.com/

Savran, A., Sankur, B., & Bilge, M. T. (2012). Regression-based intensity estimation of facial action units. *Image and Vision Computing*, *30*(10), 774–784.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Team, R. C., et al. (2013). R: A language and environment for statistical computing.

Van Rossum, G., & Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Warnick, B. J., Davis, B. C., Allison, T. H., & Anglin, A. H. (2021). Express yourself: Facial expression of happiness, anger, fear, and sadness in funding pitches. *Journal of Business Venturing*, *36*(4), 106109.

Yang, S., Yu, X., & Zhou, Y. (2020). Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 international workshop on electronic communication and artificial intelligence (iwecai)* (pp. 98–101).

Yüksel, K., & Skarbek, W. (2019). Convolutional and recurrent neural networks for face image analysis. *Foundations of Computing and Decision Sciences*, *44*(3), 331–347.

Zott, C., & Huy, Q. N. (2007). How entrepreneurs use symbolic management to acquire resources. *Administrative science quarterly*, *52*(1), 70–105.

APPENDIX A

Table 6: Facial action units and action descriptors (Ekman & Friesen, 1978)

AU number	FACS name
0	Neutral face
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowerer
5	Upper lid raiser
6	Cheek raiser
7	Lid tightener
8	Lip toward each other
9	Nose wrinkler
10	Upper lip raiser
11	Nasolabial deepener
12	Lip corner puller
13	Sharp lip puller
14	Dimpler
15	Lip corner depressor
16	Lower lip depressor
17	Chin raiser
18	Lip pucker
19	Tongue show
20	Lip stretcher
21	Neck tightener
22	Lip funneler
23	Lip tightener
24	Lip pressor
25	Lips part
26	Jaw drop
27	Mouth stretch
28	Lip suck

Table 7: Startups and their investment probability scores

Index	Startup	Score
1	Little Sister	1
2	FLIPR	1
3	Bubble Pop	0
4	RecognEyes	1
5	HOTIDY	1
6	FitPoint	1
7	SOLON	1
8	tAIste	0
9	Choos3 Wisely	1
10	SmArt	0
11	wAIste	1
12	Chattern	1
13	FindIT	0
14	Ar-T-ficial	1
15	Recipe-Me	0
16	Salix	1
17	Peech	1
18	HoodFood	0
19	LockUp	0
20	Ziggurat	1
21	Young Boosters	1
22	PREA	0
23	Whitebox	1
24	Soccer Academy	1

Table 8: Accuracy of models

Input	Models	Segment-level		Video-level	
		OpenFace	FaceReader	OpenFace	FaceReader
Set 1	LSTM	58.80 (8.77)	57.50 (4.87)	62.50 (7.22)	58.33(14.43)
	GRU	60.28 (5.59)	56.39 (0.71)	66.67 (0.00)	58.33 (8.33)
Set 2	LSTM	59.35 (3.53)	57.04 (8.06)	66.67 (20.41)	62.50 (7.22)
	GRU	60.83 (3.41)	57.41 (10.11)	66.67 (11.79)	58.33 (14.43)
Set 3	LSTM	61.76 (5.09)	66.67 (2.27)	66.67 (0.00)	66.67 (0.00)
	GRU	46.11 (12.44)	64.26 (2.63)	41.67 (18.63)	66.67 (0.00)
Set 4	LSTM	62.87 (7.00)	65.65 (1.76)	62.50 (7.22)	66.67 (0.00)
	GRU	66.48 (0.32)	66.67 (0.00)	66.67 (0.00)	66.67 (0.00)

Note: Standard deviations are given in brackets.

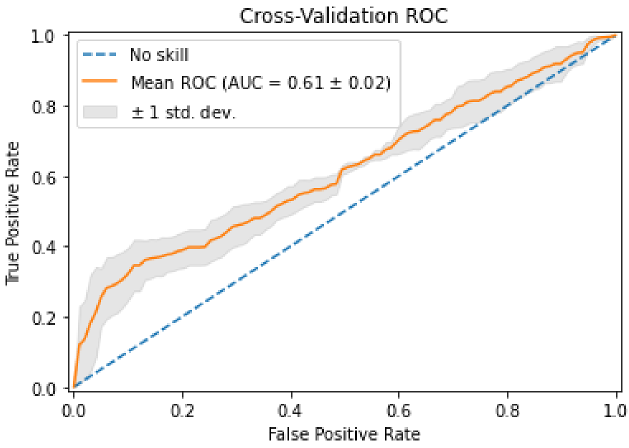
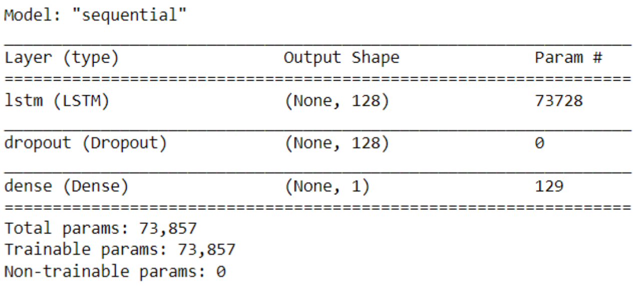
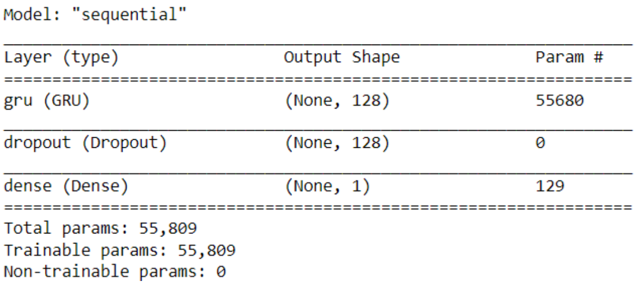


Figure 14: ROC curve for overall best performing model (GRU with OpenFace AUs of Feature Set 2)



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 128)               73728
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 73,857
Trainable params: 73,857
Non-trainable params: 0
```

Figure 15: LSTM model summary



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru (GRU)                    (None, 128)               55680
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 55,809
Trainable params: 55,809
Non-trainable params: 0
```

Figure 16: GRU model summary