Detection of Deception in the Mafia Party Game

Sergey Demyanov
s.demyanov@student.unimelb.edu.au

James Bailey
baileyj@unimelb.edu.au

Kotagiri Ramamohanarao
kotagiri@unimelb.edu.au

Christopher Leckie
caleckie@unimelb.edu.au

Department of Computing and Information Systems
The University of Melbourne, Melbourne, VIC, Australia

ABSTRACT

The problem of deception detection is very challenging. Only trained people with specialist knowledge are able to demonstrate an accuracy that is sufficiently higher than random predictions. We present a multi-stage automatic system for extracting features from facial cues and evaluate it on the Mafia game database which we have collected. It is a large database of truthful and deceptive people, recorded in conditions more variable and realistic than many other databases of similar kind. We demonstrate that using the extracted features we are able to correctly classify instances with an average AUC (area under the ROC curve) equal to 0.61, significantly better than random predictions.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

Keywords

deception; classification; facial cues; action units; video processing

1. INTRODUCTION

Many organizations such as police, secret services, border security services and insurance companies depend on the recognition of deception and truthfulness of their clients. According to previous research ([22],[2]), the average person detects liars with a probability that is statistically significant, but just slightly above a random chance. However, results of other experiments have demonstrated that when people have a motivation to lie, their deception cues are present via four non-verbal channels: facial expressions, gestures and body language, verbal style and voice characteristics. Using these, trained people can achieve an accuracy of up to 73% ([6], [7]). Moreover, in [12] it is claimed that the analysis of facial expressions considered simultaneously with context could further increase this accuracy up to 90%, considerably higher than human performance. Current technologies allow us to perform such analysis automatically in real-time. Since the face is the richest source of information, detection of facial cues of deception is one of the most important parts of such a deception detection system.

In this article we present a new database of truthful and deceptive people based on the videos of the Mafia party game (also known as Werewolf), and describe a methodology for feature extraction and classification of each person's role. In each game players are assigned to be either truthful or deceptive. At the same time there are no requirements on what the players have to say and how they should behave. A more detailed explanation of the game rules is given in Section 3. This new database contains 6001 labeled episodes from 270 participants with a total duration of 5 hours.

Our goal was to identify the players' roles based on their close-up face recordings. To create features we consider the first N minutes of each game, where N varies from 5 to 35 minutes. We extract features corresponding to the particular movements of facial muscles. Some of them are caused by experienced emotions, and therefore can be the signs of verity or the cues of deception. According to these features we build an automatic classifier of truthful and deceptive people.

First we give an overview of related work (Section 2), present the Mafia database (Section 3) and discuss the methodology of our research (Section 4). In the next key sections we describe the procedure of facial movement detection (Section 5), and explain the details of feature engineering (Section 6). Later we compare the obtained accuracy with the accuracy of a random classifier and demonstrate that the predictions are statistically significant (Section 7.1). Finally we analyze the most predictive features and show that they agree with the theory (Section 7.2). The conclusion (Section 8) finalizes the article. We believe our research can boost interest in the area of deception detection from facial cues.

2. RELATED WORK

The problem of deception detection has attracted considerable interest in recent years. Work in [22] established that the leakage of cues to deception is caused by the increased cognitive load experienced by liars, and therefore cannot be avoided. In [3] the authors analyzed 158 cues including facial expressions, linguistic features, physiological features and others. They discovered that people with

ICMI 2015, November 9–13, 2015, Seattle, WA, USA.

DOI: http://dx.doi.org/10.1145/XXX.XXXXXXX.

higher cognitive loads are more likely to have less illustrators and body movements, more hesitations, longer pauses in speech, greater pupil dilation and more gaze aversion. Deceptive people can also experience a sudden increase of blood flow in the region of the eyes, which can be detected in thermal images [1]. Moreover, deception is often related with one of the 3 emotions: fear, guilt or delight [23]. The facial expressions of these emotions can also reveal these emotions.

In the experiments by Warren et al. [25], participants watched emotional and unemotional videos and were asked to lie about what they saw. The average classification accuracy for predicting lying was around 50%, however it was 64% for the group lying about emotional videos. This confirms the hypothesis that emotions can cause the non-verbal leakage of the deception cues. Their database is known as YorkDDT. Another database was collected by Frank et al. [9]. They recorded videos for an interrogation scenario with 100 participants of 2.5 minutes each. A multimodal database with 30 participants that includes video, thermal video, speech and physiological data have been collected in [**?**]. Mihalcea and Burzo [14] obtained 140 videos with truthful and deceptive people using Amazon Mechanical Turk. Zhang et al. [28] developed a system of expression classification based on facial key points and demonstrated its good performance.

A number of attempts have been made to develop an automatic system for deception detection. However, in most of them the features are not related to facial expressions. Several of them are based on linguistic features. For example, Fornaciari and Poesio used stylometric techniques to identify deception in the corpus of hearings collected in Italian courts [8]. Using only the linguistic features, Mihalcea et al. [15] could reach an accuracy $52 - 73\%$.

Thermal imaging approaches are also popular. Warmelink et al. [24] applied it to identify deception in airports. Their system demonstrated an accuracy higher than 60%. Rajoub et al. performed similar experiments with thermal videos on a set of 492 samples from 25 participants. While they were able to reach a very high level of accuracy of 87% for within-person predictions, the results of inter-person predictions (around 60%) were similar to other experiments. Another attempt to use thermal imaging has been performed by Jain et al. in [10]. They also obtained a classification accuracy for predicting lying of around 62%. In [1] the authors presented a system based on a multimodal approach, combining physiological features such as temperature, heart rate and pulse with linguistic features. The obtained accuracy was shown to be better than random, achieving 70% in some scenarios.

Some attention has also been paid to micro-expressions. The authors of [17] developed the system of micro-expression detection using LBP-TOP features. They evaluated it on the corpus of video clips from [25] and their own database, and obtained quite promising results. Another database of 195 micro-expressions was collected in [27, 26]. The authors also provide the labels of the appearing action units with their timestamps. However, there were no attempts to use these databases and algorithms for deception detection.

3. THE MAFIA GAME DATABASE

We present a new database collected from the Mafia TV show, which contains 5 hours of videos from 270 participants. Unlike others, it was recorded in much more natural conditions than the experiments in other publications. Comparing with other deception detection databases in Table 1, it makes the Mafia database one of the largest. The source videos [1], the episode timestamps and player labels [2] are available online. We hope that it will be interesting for the research community and will boost progress in the area of deception detection.

Table 1: The main parameters of deception detection video databases

database	Participants	Duration, min	Link
Mafia DB	270	300	
Amazon Turk DB	140	315	[14]
RU-FACS-1	100	250	[9]
Multimodal DB	30	75	[1]
YorkDDT	20	23	[25]

The Mafia party game (also known as Werewolf) was invented in 1986 by the students of Moscow State University studying in the Department of Psychology. This is how the game is described in Wikipedia: "[It is] modeling a conflict between an informed minority (the Mafia) and an uninformed majority (the innocents). At the start of the game each player is secretly assigned a role affiliated with one of these teams. The game has two alternating phases: 'night', during which the Mafia may covertly 'murder' an innocent, and 'day', in which surviving players debate the identities of the mafiosi and vote to eliminate a suspect. After elimination the player reveals his role. Play continues until all of the Mafia has been eliminated, or until the Mafia outnumbers the innocents". While the 'day' stage contains long discussions and voting, during the 'night' stage the 'Mafia' members silently show who they want to eliminate, and usually agree on a candidature within about 10-20 seconds.

The Mafia TV show series was shown on the Russian TV channel MuzTV in 2009 and 2010. The game participants were Russian celebrities and TV channel spectators. In total there were 30 series in this period [3]. The length of each of them is about 45 minutes. In each game there were 9 players. Two of them were from the Mafia team, the others were innocent. The role depends on the card the player gets at the beginning of the game. A black card corresponds to Mafia and a red card corresponds to innocents. Depending on the situation, each game had from 2 to 4 rounds of 'day'-'night' pairs.

Since during the 'day' stage the players from the Mafia team pretend to be innocent, we labeled them as deceptive. Others were labeled as truthful. Thus, our database contains 60 deceptive and 210 truthful players, 270 players in total. Note that the label of each player depends only on his role and is not changing within a game regardless of the current player's actions and words. Moreover, note that 'night' stage recordings are very short and do not contain discussions, so we excluded all 'night' appearances from our database.

4. METHODOLOGY

[1] `https://www.youtube.com/user/muzTV/search?query=mafia`

[2] `https://sites.google.com/site/mafiadatabase`

[3] 60 more series were shown later. Thus, the database can be increased by 3 times.

All facial expressions are caused by particular combinations of facial muscles. The list of these muscles is known from physiology. Based on this list, the Swedish anatomist Carl-Herman Hjortsjö developed the Facial Action Coding System (FACS), that was later published by Paul Ekman et al. in 1978 and revised in 2002 [5]. This system defines 27 possible **Action Units** (AU) related with particular muscles (9 of them are in the upper part of the face and 17 in the lower part) and 6 basic emotions (fear, sadness, happiness, anger, disgust and surprise) that consist of different action units. The visual appearance of these action units can be found online [4].

Facial expressions can also be classified as *posed*, *spontaneous* or *concealed*. Posed expressions appear deliberately in order to cause a certain impression. In contrast, spontaneous expressions appear unconsciously as a reaction on the ongoing events. Concealed expressions are also spontaneous, but their appearance is suppressed and therefore they have much smaller amplitudes.

In the book "Telling lies" [4], Paul Ekman provides a comprehensive analysis of the nature of deception. Chapter 5 describes the facial cues of deception. Ekman states that posed and spontaneous expressions are caused by different parts of the brain, and therefore have some subtle differences. One of them is that some facial muscles are involved only in spontaneous expressions and they cannot be readily inhibited. These muscles are called *reliable*. For example, only 10 percent of people can pull the corners of their lips down keeping their chin muscle fixed. The characteristic of being hard to suppress was called the inhibition hypothesis. This hypothesis was later confirmed in the experiments of [18].

Since facial expressions are supposed to be related with action units, we used this information to extract features. In order to do this we collected examples of action units and searched for their appearance in the Mafia database. The similarity scores of each player were used as features. We explain the full procedure in detail below. First we describe the algorithm of frame processing, and then provide the details of feature engineering.

5. FRAME PROCESSING

Since the camera shows players from all angles and distances, players wear glasses and gesticulate in front of the face, as well as other difficulties, frame processing is not an easy task. For this purpose we developed a multi-stage automatic procedure which is briefly described in Algorithm 1.

5.1 Face detection

As the first element we employed the Viola-Jones algorithm [21] from the OpenCV library to detect faces. This library provides a detector that is already learned, so we could apply it straight away. In our experiments we considered only one face on each episode, so when we met more than one, we chose the one that is closest to the previously detected face. Once the face was detected, we applied the same algorithm for eye detection. The library provides different classifiers for the left and right eyes, so we approximately identified the regions for eyes and applied the classifiers in these regions. If no eyes were detected, we considered it as a false registration. In the case when only one eye was not detected on the current frame, but it was detected on the previous frame, we measured the displacement of the other eye and applied it to the current one. This situation happened quite often for people with glasses, and it allowed to treat such frames with the standard procedure. We used the position of detected eyes for initial in-plane rotation, so that the eyes become horizontally aligned.

Algorithm 1 Frame processing

1. Face and eyes detection using OpenCV library
2. Initial in-plane rotation using eyes coordinates
3. Facial keypoint feature detection using Luxand FaceSDK
4. Width, height and angle normalization using keypoint features
5. Linear and non-linear image registration
6. Grid displacement computation for a sequence of frames.

5.2 Facial feature detection and normalization

After initial processing we applied the proprietary Luxand FaceSDK[5] software for facial feature detection. For a given face it returns the location of 66 facial points: 11 for each eye, 14 for mouth, etc. If the toolbox could not detect the features, we omitted such a frame. First we employed the obtained features to compute more precise coordinates of the left eye and right eyes (average of all left and right eye features, Fig. **??**). These coordinates were used for additional in-plane rotation, so that the eyes are located on the horizontal line.

Second, we used these features to perform width and height normalization. To normalize width we computed the distance between eyes and scaled the image to make this distance equal to 80 px. Similarly we computed the visible nose height (the mean of the difference between y-coordinates of the left and right nose corners and its top), and scaled the image to make it equal to 50 px.

Third, we performed normalization with respect to other two types of rotation. While the in-plane rotation is a linear transformation and it can be easily suppressed using the eye coordinates, the other two out-of-plane types of rotation appear as a non-rigid transformation. In our problem we have only 2 regions of interest: eyes and mouth, so instead of registration of the whole face we cropped these two regions and treated them separately. These regions were considered as vertical cylinders with predefined radius values, so an up-down region rotation appears as a squeeze/stretch of an image in the vertical dimension, and therefore is a linear transformation. The same cylinder model makes it possible to handle a left-right rotation. We estimated the rotation angle using the eyes and nose coordinates obtained earlier, and computed how these cylinders would appear without rotation. In other words, we considered the visible face region as a projection of a cylinder on a rotated axis and computed its projection on the original axis. It leads to a non-linear

[4] `http://www.cs.cmu.edu/~face/facs.htm`

[5] `https://www.luxand.com/facesdk/`

stretching of the part that is closer to the observer and otherwise.

5.3 Image registration

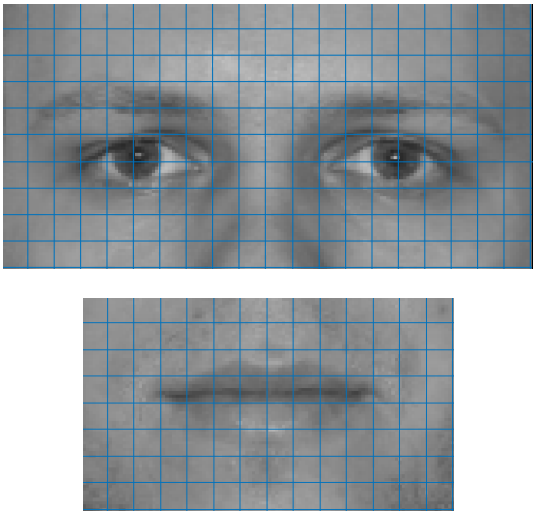


Figure 1: An example of normalized eyes and mouth with the 8×8 uniform grids

After face normalization we perform image registration. It allows allows us to obtain a description of non-linear movements within a face, that are related to action units. This procedure is based on the method of nonrigid registration using free form deformations, described in [19], and used in [11].

The problem is the following: for two similar images we want to find a non-linear transformation that maps one image into the other one. As in [11], we also use a sum of squared distances between pixels as a similarity measure. We model the transformation by the displacement of nodes of the uniform grid in Fig. 1 with the size of a cell 8×8 px. For the displacements p_k of the nodes of the uniform grid x_k, the value for the pixel x is taken from the pixel $T(x)$ of the original image:

$$T(x) = x + \sum_{x_k \in N_x} p_k \beta\left(\frac{x - x_k}{\sigma}\right),$$

Here N_x is the set if 16 nodes around the point x, p_k's are the displacements of these grid nodes, β is the cubic multidimensional b-spline polynomial function and σ is the regular grid spacing (8). To avoid problems with the corner pixels, the regular grid has 2 rows and columns outside the image from each size. Thus, the parameters of the model are the coordinates of the extended uniform grid nodes ($16 \times 26 \times 2 = 832$ variables for the eye region and $14 \times 20 \times \times 2 = 560$ variables for the mouth region). The minimization problem is solved using the L-BFGS algorithm.

Before solving non-rigid registration problem, we first equalize the intensity histograms in order to avoid variations in brightness. Second, we perform affine registration, which also minimizes the SSD measure. It gives us the 3×3 affine transformation matrix, which aligns frames linearly. The non-linear transformation is performed in two steps with different grid spacing. First the SSD measure is minimized only for a half of nodes (i.e., using double spacing), and after it the solution is updated using all nodes.

Because non-rigid registration works well only when the difference between frames is small enough, we applied it only to nearby frames. To obtain the transformation function between two frames on an arbitrary distance, we sequentially applied the grids for frames between them. In other words, if we know all functions T^i_{i-1} for $i = 1 \dots N$, then the function T^N_0 is just their composition, i.e.

$$T^N_0(x) = T^N_{N-1}(\dots T^2_1(T^1_0(x)) \dots).$$

Here T^i_{i-1} is the function that transforms the points from the frame $i-1$ to the frame i. However, this process is not precise and good results can be obtained only for a sequence of not more than 30 frames. Note that the functions $T(x)$ are defined only for pixels in the image, while the output $T(x)$ may be outside the image. In this case we do not have 16 nodes around this output (that is an input for the next function) and we have to approximate the displacements for these non-existing nodes. This leads to inaccurate results of the points near the border.

6. FEATURE ENGINEERING

Algorithm 1 allows us to compute the displacement of grid nodes in a sequence of consecutive frames. These node displacements give a compact representation of the movements between the first and the last frames. Here we describe the procedure of feature extraction using this information. Its scheme is given in Algorithm 2.

Algorithm 2 Feature extraction

1. Extract all episodes from the Mafia videos using Algorithm 1
2. Choose a collection of AU examples from the MMI database
3. Compute the displacement grids for the onsets and offsets of chosen examples using Algorithm 1
4. Find the subsequences of the episodes that are most similar to the AU examples
5. Compute their similarity scores
6. Aggregate the similarity scores over all episodes of each player (choose the minimum)

6.1 Episode extraction

We processed all source Mafia videos to extract the sequences of neighboring frames that are registered to each other using Algorithm 1. We refer to them as *episodes*. However, within a single episode some frames might be omitted for different reasons. For example, it might happen because the face is occluded by a hand and therefore it is not detectable. Their number should be small enough to have the registered frames be close to each other and large enough to keep the episodes continuous, for example in the situations of people with glasses. In our case the maximum allowed number of consecutive omitted frames in a single episode was 10. It the number was larger, we ended the episode

and started a new one. Given that each pair of frames required about 3 seconds for registration, it was the most time consuming part of the total pipeline.

We extracted episodes in the interval from 7 to 42 minutes, when all the game actions were taking place. In total we obtained 6733 episodes in 30 games with an average of 224 episodes per game, 29 per player. The minimum and maximum number of episodes in one game were 175 and 284. The average length of each episode was 77 frames, i.e., about 3 seconds. Each episode was manually viewed and labeled according to the player appearing on it. If the episode was corrupted, contained non-players or players after the game, it was labeled as 0 and was omitted in further calculations. The total number of episodes after elimination is 6001.

6.2 MMI database

The MMI database ([16]) contains a wide range of videos and images representing different emotions. Some of the videos are also labeled according to the Action Units they contain. Additionally they contain the information about AU stage (onset (or start), peak, offset (or finish), neutral) for each frame. We employed this information for feature selection.

Theory suggests [4, 18], that the action units caused by reliable muscles (like AU1) correspond to felt emotions. Depending on the emotion it can be a sign of either truthfulness or deception. Therefore, it was plausible to find the appearance of such action units. For this purpose we employed the same image registration procedure for MMI examples as for the Mafia database. We selected a maximum of 3 examples of non-empty onsets and offsets for each of the 14 most common action units (namely 1, 2, 4, 5, 6, 7, 9, 12, 15, 16, 17, 20, 23 and 24), 79 in total. The indices of video clips with AU examples from the MMI database including the duration of onset and offset stages are presented in the Table 2. The choice of 3 examples provides a balance between the variety of AU representation and the generalization ability of the trained model. AU templates are similar, and the corresponding features are highly correlated. If the number of features was too large, this could result in overfitting behavior by standard classifiers such as the employed logistic regression. The examples were chosen randomly (in fact, in lexicographical order of their numbers) in order to avoid inflation of reported performance due to over tuning of feature selection.

For each example we performed image registration of the frames with the 'onset' and 'offset' labels. Then we used the computed transformation functions to compute the new coordinates of the uniform grid nodes on the last frame of the resulting sequences. For each of these sequences we also computed the cumulative linear transformation by multiplying the transformation matrices for each frame in the sequence. The new coordinates were multiplied on the inverse cumulative transformation matrix in order to suppress head movements. Since different action units appear in different areas of the eyes and mouth, we used only the subset of nodes located in these areas. The list of these nodes for each AU is also provided on the database website. The displacements of these nodes uniquely identify the appeared action unit.

There might be some concerns about the validity of usage of posed AU examples from the MMI database in order to find the spontaneous appearance of AU in the Mafia database. However, our algorithm takes the difference between posed and spontaneous expressions into account. More specifically, that most of the difference lies in the temporal dimension, which is effectively handled in our algorithm by considering all possible pairs of first and last frames to detect action units. Second, our algorithm detects the presence of AUs, rather than their type. AU type is another discriminative factor, which might further boost accuracy, but is more difficult to correctly detect. The employed algorithm is more preferable than approaches like CERT, where the AU appearance scores are based on each frame independent of others. On the other hand, our displacement grids are based on changes in time, thus containing richer information, which is exploitable in our framework.

Table 2: The indices of Action Units from the MMI database, which are used as examples. The numbers of onset and offset frames are provided in the brackets

AU	Examples (onset and offset duration)		
1	1931 (3, 0)	24 (18, 18)	582 (4, 6)
2	144 (3, 9)	145 (10, 13)	1649 (6, 8)
4	1047 (4, 7)	1384 (5, 3)	1823 (6, 13)
5	1 (5, 9)	1275 (6, 11)	144 (3, 9)
6	1074 (5, 5)	1088 (7, 8)	123 (11, 16)
7	1316 (0, 2)	1874 (3, 7)	1973 (5, 6)
9	1384 (5, 5)	1964 (11, 10)	199 (3, 5)
12	123 (17, 37)	124 (28, 39)	125 (12, 19)
15	1077 (7, 3)	1152 (7, 7)	1153 (5, 5)
16	134 (5, 6)	135 (7, 17)	14 (11, 18)
17	1152 (8, 6)	1153 (4, 3)	12 (10, 14)
20	1088 (7, 8)	1812 (13, 4)	1813 (10, 7)
23	382 (8, 4)	611 (12, 12)	
24	1874 (11, 8)	1931 (3, 0)	1973 (4, 6)

6.3 Feature extraction

The next step was to find the appearance of action units in the extracted episodes. Since they might have appeared at any moment, we considered all subsequences of consecutive frames of total length not more than 30. For each of them we computed the displacement of the subset of uniform grid nodes specific for this AU, the same way as we did for MMI examples. Then we computed the Euclidean distance between the displacement vectors of each episode subsequence and each onset and offset of examples of the action units from the MMI database obtained in the previous step.

We assume that within a single episode a particular action unit can appear only once. Given that, we computed the minimum distance among all subsequences and recorded it as a feature for this episode. Thus, for each episode we obtained 79 features (40 onset and 39 offset features). For convenience we recorded them with a negative sign. We will refer to them as the similarity scores.

The next goal was to create features for each player in each game. As before, we used maximum to aggregate the scores over each player episodes. We thus selected the maximum similarity scores over all episodes for a particular player and consider them as features for this player.

While a higher similarity score corresponds to a higher probability of AU appearance, the similarity scores are still quite noisy. In fact, after the visual examination we deter-

mined that only the first quartile of their values correspond to the episodes that are similar to the related action unit. Other 3 quartiles seemed to contain random episodes, regardless of the score value. To incorporate this knowledge in the dataset we subtracted the 78% - percentiles (7/9, the percentage of truthful players) and set all the negative values to 0. Therefore, we produced a sparse dataset with no difference in the feature values between non-top values.

7. EXPERIMENTAL RESULTS

During the game some players are eliminated and do not appear in the later stages. This causes an imbalance in the total observation time. Moreover, the eliminated players are more likely to be truthful, because only truthful players are eliminated during the 'night' stage. This might also introduce a bias in the features. In order to validate the obtained results we created a number of datasets, corresponding to the different game duration. We split all 35 minutes on 10 second intervals and considered game durations in the range from 5 to 35 minutes, totally 181 intervals. For each of them we selected only those episodes that completely fall into this interval. Thus, each set of these episodes gives an independent dataset.

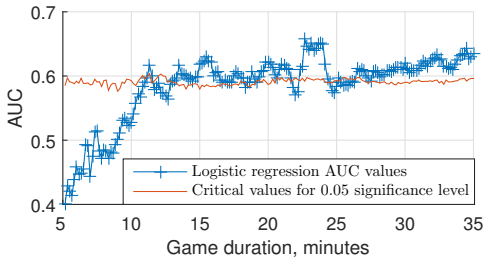
7.1 Classification

Since the dataset is highly unbalanced, we used the area under the ROC curve (AUC) as a performance metric instead of classification accuracy. We employed the simple multinomial logistic regression classifier. Before classification we eliminated the players without observed episodes. There are 22 of them for the 5 minute game duration, and only one (player 3 in the game 16) for the 35 minutes. We performed 30-fold cross-validation, every time leaving out the players from a single game. Thus we kept the same proportion of the players in the training and test sets as in the full dataset.

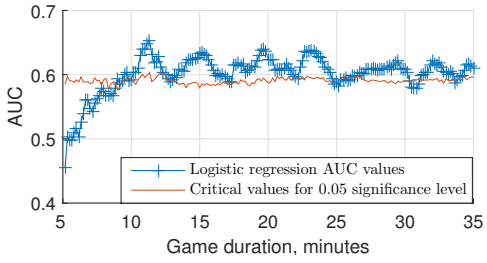
Fig. 2 presents the plots of overall AUC depending on the game duration. It demonstrates a clear increase of performance in the range from 5 to 12 minutes, that corresponds to the increase of the number of considered episodes. This is a natural behavior: once we take more information into account, the predictions become more and more accurate. After that the AUC remains on the same level around 0.61. The maximum duration of 35 minutes gives the $AUC = 0.6391$. It corresponds to the accuracy 70.26%, while the random predictions give the average accuracy 65.35%. Choosing a classification threshold such that the number of predicted positive examples is equal to the one in the training set, we get the same corresponding precision and recall: 0.8082, and so is the F1-score. From Fig. 2 (b,c) we can see that eyes features independently perform better than mouth features, which reach the significance level only after 25 minutes. The reason might be a larger variety of mouth movements, caused by speech. Since it acts as noise, more time is required to collect meaningful statistics. We also evaluated the performance of onset and offset features independently. As expected, the results are worse than when they are combined together. However, onset features appear to be more predictive. Similar results were obtained using a linear SVM classifier with the regularization parameter $C = 1000$.

In order to assess the statistical significance of the results, we computed the critical AUC values. We tested the hypothesis that such AUC values could be obtained for a random distribution of labels. For each dataset, corresponding to a particular game duration, we performed the classification with the randomly permuted labels, 400 permutations. Permutations allowed to preserve the ratio of positive and negative instances. The critical values for the 0.05 significance level are presented by the orange curve in Fig. 2. As we can see from it, the obtained AUC is higher than the critical curve for almost every dataset based on more than 13 minutes. As it follows from the definition of critical values, the probability of this to be random is less than 0.05. It confirms that computed features are indeed the label predictors, containing relevant information.

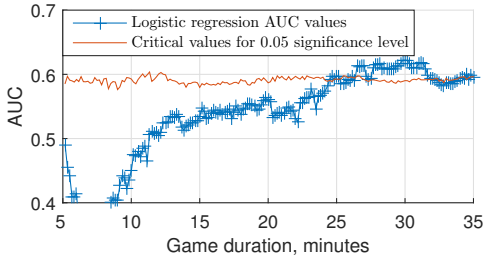
Figure 2: Plots show the AUC (area under ROC curve) values as a function of game duration for eyes, mouth and all features. The orange line represents the critical values for the 0.05 significance level.



(a) All features



(b) Eyes features



(c) Mouth features

We also computed the mean of AUC for all time intervals $5-35$ (0.5849) and time intervals $15-35$ (0.6106). Using the same randomization, we computed their p-values: 0.0104 and 0.0026 accordingly. Both of them are much lower than 0.05, indicating a result due to random chance is unlikely.

It is interesting to compare the obtained accuracy with a human baseline. We can compute a rough estimation the following way. Recall that a game consists of repeating 'day'-

'night' phases. While in the 'day' phase players try to eliminate a deceptive player, in the 'night' phase they always lose one truthful player. For a predefined number of players (7 vs 2) there is a limited number of possible outcomes (10). Assuming a random choice of eliminated players in the 'day' phase, we can compute the probability of each of these outcomes. The Mafia team wins in 4 of them, with the total probability equal to $221/315 \approx 70.16\%$. In fact, the Mafia team won in 21 among 30 games, i.e., exactly in 70% of cases. It confirms that people detect deception very close to random, even when they get information from all channels including video, audio and context. The same result was established earlier in psychological research [2].

7.2 Feature analysis

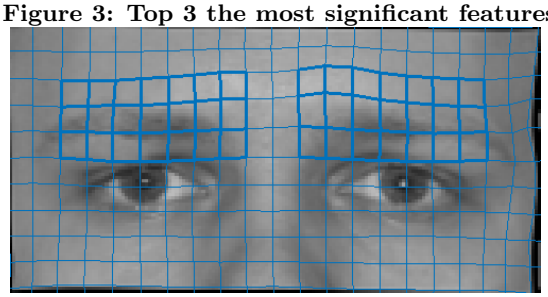
The logistic regression classifier also provides p-values for the obtained coefficients. We used these coefficients to identify the most statistically significant features. The top 3 of them are presented on Fig. 3. Note that all of them are caused by the reliable muscles [13], and therefore are difficult to be simulated. We also computed the corresponding feature coefficients. While their amplitude depends on the similarity scores and does not provide any information, their signs show how the feature influences the final prediction. Positive coefficients increase the probability of truthful people, while negative ones decrease it. Here we use the average of coefficients over all 30 folds and game durations from 15 to 35 minutes, when the datasets are based on a sufficient number of episodes.

The first of the top 3 features is an example of the onset of Action unit 1 (inner brow raiser). It has a positive coefficient, with the p-value 0.030. The second feature represents the offset of the AU20 (lip stretcher). It has a negative coefficient with the p-value 0.037. The third feature is connected with the AU16 (lower lip depressor). It also has a positive coefficient. Its p-value is 0.039.

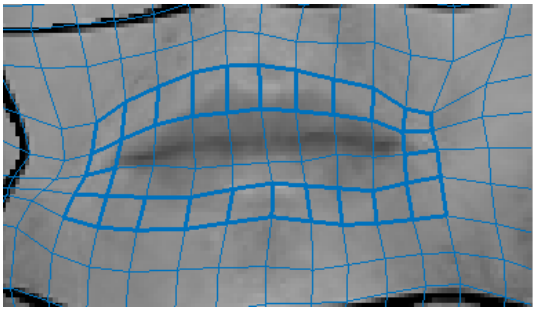
We can provide a possible explanation of these results. Theory says that deceptive people tend to experience fear, guilt or delight [23]. Opposite to that, sadness has been shown to be a sign of verity [20]. It is known [5] that AU1 and AU16 might be caused by sadness, while AU20 can appear as a result of fear. Therefore, their coefficient signs do not contradict the theory.

We also tested the performance of these 3 most significant features independently. None of them achieved statistically significant classification accuracy, which confirms that only a combination of cues can give a reliable result. This was earlier stated in [23].

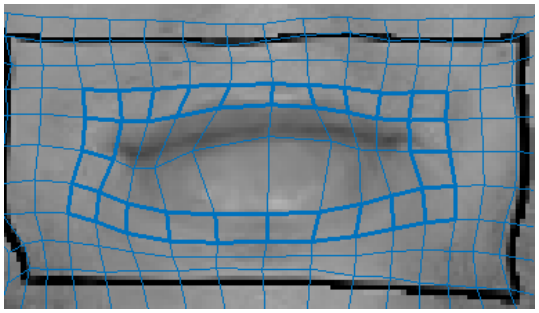
Figure 3: Top 3 the most significant features



(a) 582(AU1)



(b) 1088(AU20)



(c) 135(AU16)

8. CONCLUSION

In this article we presented an automatic system of deception detection based on the features extracted from movements of eyebrows, eyes and mouth on videos. We demonstrated that these features contain sufficient information to achieve a classification accuracy significantly higher than random predictions. The list of the most predictive features can be explained according to the results previously established in the psychological literature. We also introduced the new Mafia database of truthful and deceptive people, which was recorded in more natural conditions compared to previous studies. This database contains in total 5 hours of video, which make it one of the largest available. We hope that it will become a benchmark for assessing algorithms for deception detection. The demonstrated performance is not good enough to be used in practice, but it establishes a lower bound for the future.

This research is in its beginning. Future research might improve all parts of the presented system. For example, one might explore developing more accurate methods of action unit detection, extraction of more meaningful features or usage of other modalities such as voice tone and language features. The multimodal approach that combines all types of features seems to be the most promising. Another significant drawback of the algorithm is its speed. The process of image registration takes about 3 seconds per frame, which does not it to be used for real-time predictions. This procedure might be accelerated by using less expensive features or more powerful hardware like a GPU. It would be a big step forward to make the algorithm work in real-time without loss of accuracy. Moreover, one can increase the database by processing all other 60 videos and labeling extracted episodes. That would make the database the largest one. We believe that this research will stimulate interest in the challenging problem of automatic deception detection.

9. REFERENCES

[1] Mohamed Abouelenien, Veronica Pérez-Rosas, Rada Mihalcea, and Mihai Burzo. Deception detection using a multimodal approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 58–65. ACM, 2014.

[2] Charles F Bond and Bella M DePaulo. Accuracy of deception judgments. *Personality and social psychology Review*, 10(3):214–234, 2006.

[3] Bella M DePaulo, James J Lindsay, Brian E Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. Cues to deception. *Psychological bulletin*, 129(1):74, 2003.

[4] Paul Ekman. *Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage (Revised Edition)*. WW Norton & Company, 2009.

[5] Paul Ekman and Wallace V Friesen. *Manual for the facial action coding system*. Consulting Psychologists Press, 1978.

[6] Paul Ekman and Maureen O'Sullivan. Who can catch a liar? *American psychologist*, 46(9):913, 1991.

[7] Paul Ekman, Maureen O'Sullivan, and Mark G Frank. A few can catch a liar. *Psychological Science*, 10(3):263–266, 1999.

[8] Tommaso Fornaciari and Massimo Poesio. Automatic deception detection in italian court cases. *Artificial intelligence and law*, 21(3):303–340, 2013.

[9] M Frank, J Movellan, M Bartlett, and G Littleworth. Ru-facs-1 database. *Machine Perception Laboratory, UC San Diego*, 1, 2012.

[10] Uday Jain, Bozhao Tan, and Qi Li. Concealed knowledge identification using facial thermal imaging. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1677–1680. IEEE, 2012.

[11] Sander Koelstra, Maja Pantic, and Ioannis Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(11):1940–1954, 2010.

[12] David Matsumoto, Hyi Sung Hwang, Lisa Skinner, and MG Frank. Evaluating truthfulness and detecting deception. *FBI Law Enforcement Bulletin*, 80:1–25, 2011.

[13] Marc Mehu, Marcello Mortillaro, Tanja Bänziger, and Klaus R Scherer. Reliable facial muscle activation enhances recognizability and credibility of emotional expression. *Emotion*, 12(4):701, 2012.

[14] Rada Mihalcea and Mihai Burzo. Towards multimodal deception detection–step 1: building a collection of deceptive videos. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 189–192. ACM, 2012.

[15] Rada Mihalcea, Verónica Pérez-Rosas, and Mihai Burzo. Automatic detection of deceit in verbal communication. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 131–134. ACM, 2013.

[16] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 5–pp. IEEE, 2005.

[17] Tomas Pfister, Xiaobai Li, Guoying Zhao, and Matti Pietikainen. Recognising spontaneous facial micro-expressions. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1449–1456. IEEE, 2011.

[18] Stephen Porter, Leanne ten Brinke, and Brendan Wallace. Secrets and lies: Involuntary leakage in deceptive facial expressions as a function of emotional intensity. *Journal of Nonverbal Behavior*, 36(1):23–37, 2012.

[19] Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *Medical Imaging, IEEE Transactions on*, 18(8):712–721, 1999.

[20] Thomas E Slowe and Venu Govindaraju. Automatic deceit indication through reliable facial expressions. In *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, pages 87–92. IEEE, 2007.

[21] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[22] Aldert Vrij. *Detecting lies and deceit: The psychology of lying and implications for professional practice*. Wiley, 2000.

[23] Aldert Vrij. *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley & Sons, 2008.

[24] Lara Warmelink, Aldert Vrij, Samantha Mann, Sharon Leal, Dave Forrester, and Ronald P Fisher. Thermal imaging as a lie detection tool at airports. *Law and human behavior*, 35(1):40, 2011.

[25] Gemma Warren, Elizabeth Schertler, and Peter Bull. Detecting deception from emotional and unemotional cues. *Journal of Nonverbal Behavior*, 33(1):59–69, 2009.

[26] Wen-Jing Yan, Su-Jing Wang, Yong-Jin Liu, Qi Wu, and Xiaolan Fu. For micro-expression recognition: Database and suggestions. *Neurocomputing*, 136:82–87, 2014.

[27] Wen-Jing Yan, Qi Wu, Yong-Jin Liu, Su-Jing Wang, and Xiaolan Fu. Casme database: a dataset of spontaneous micro-expressions collected from neutralized faces. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–7. IEEE, 2013.

[28] Zhi Zhang, Vartika Singh, Thomas E Slowe, Sergey Tulyakov, and Venugopal Govindaraju. Real-time automatic deceit detection from involuntary facial expressions. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.