Chapter 27
Deception Detection in Videos Using Robust Facial Features with Attention Feedback

Anastasis Stathopoulos, Ligong Han, Norah Dunbar, Judee K. Burgoon, and Dimitris Metaxas

Abstract This chapter presents methods to address the problem of deception detection in videos. Current approaches to detect deception in videos are limited since they (1) are used in short videos focusing only on a small act of deception; (2) are hard to interpret; and (3) do not make use of any human model or insights that could help in the detection task. To address these limitations, a novel framework based on the Dynamic Data-Driven Applications Systems (DDDAS) paradigm is proposed that uses as input the one-dimensional Facial Action Unit (FAU) and gaze signals, and model enhancements. By using facial features as input and not the raw video, we are able to train a conceptually simple, modular, and powerful model that achieves state-of-the-art performance in video-based deception detection. The proposed DDDAS methodology allows to interpret predictions of the referenced model by computing the attention of the neural network in the time domain, identifying *key frames*. The previous can (a) enable domain scientists to perform retrospective analysis of deceptive behavior; (b) identify informative data for model re-training.

Keywords Video classification · Explainable AI · DDDAS · Deception detection

A. Stathopoulos · L. Han · D. Metaxas (✉)
Rutgers University, New Brunswick, NJ, USA
e-mail: dnm@cs.rutgers.edu

N. Dunbar
UC Santa Barbara, Santa Barbara, CA, USA

J. K. Burgoon
University of Arizona, Tucson, AZ, USA

F. Darema et al. (eds.), *Handbook of Dynamic Data Driven Applications Systems*,
https://doi.org/10.1007/978-3-031-27986-7_27

27.1 Introduction

Whenever people communicate, deception can occur whether directly or indirectly. The motivation to cause acceptance when something is unproven or false is present in our daily lives and can take many forms. Detection of misleading information is of paramount importance not just on a personal level, but also in many societal situations. Therefore, detection of deception is very important. For instance, accurate detection of deception is critical for law enforcement officials to perform their duties more effectively, or for airport security screening procedures. Therefore, the development of robust Automated Deception Detection (ADD) systems is a long sought-after goal.

Dynamic data-driven application systems (DDDAS) provide a basis for ADD, where known models, contextual knowledge, and real-time data can be used to verify and validate information. Typically, DDDAS is used to include data to update models, wherein ADD uses models to validate the integrity of the data.

Current methods for ADD are limited for the following reasons: (1) they focus on a single act of deception, typically for rather short video recordings; (2) the interpretability of the used models is narrow; and (3) these methods do not utilize any model of the human face or body (such as in [1]) to aid the detection task. Since the input to deception detection systems contains humans, modeling of faces or bodies can provide very useful cues to the video exploitation, while reducing the risk that the ADD model overfits to background noise and learns unimportant features.

This chapter presents a novel system to detect deception in video scenarios. The deception detection task can be modeled as a binary video classification task. That is, a positive label is given for a video that contains a person manifesting deceptive behavior and a negative label is given when that person is acting truthfully.

According to the Interpersonal Deception Theory [2], deception is a dynamic process, in which deceivers adjust their behavior according to how much they think they are being suspected by others. For this reason, it is posited here that datasets that contain short video clips, focusing on only a single act of deception, are not enough for modeling deceptive behavior.

To address the limitation of ADD for a single instance, we use a dataset that contains large videos (1 hour long) of people playing a version of the board game: *The Resistance Game*; it is a social role-playing game that involves deductive reasoning. Players in the Resistance Game are randomly given one of two roles, either *deceivers* or *truth-tellers*. The majority of players in a game is assumed to be truth-tellers. Deceivers know the role of everyone in the game, whereas truth-tellers do not. The game relies on deceivers trying to hide their identity and attempt to prevent the larger group (of the truth-tellers) from working together to reveal deceivers among the group. More specifics on the data are presented in Sect. 27.4.

Facial expressions can convey a lot of information about one's physical and emotional state [3]. People rely on facial expressions to "collect" both intentional and unintentional meaning during interactions. The Facial Action Coding System

(FACS) [4] was developed. FACS is a systematic way to code facial motion with respect to non-overlapping facial muscle actions called Facial Action Units (FAUs).

With so much communicating by the facial expressions, we opt to incorporate facial cues to create a system that detects deception in videos. Using the DDDAS paradigm, we can also find key timesteps in videos and facilitate improved detection capabilities. Our approach, Deception Detection using Robust Facial Features (DDRFF), has the following pipeline. A morphable model is superimposed to a subject's face and with the help of a feature extractor, deriving facial features. In particular, for each frame of the input video, the intensities of 17 Facial Action Units (FAUs) are computed, which are normalized with the parameters of the morphable model that is fitted to the subject's face, resulting in 17 identity-agnostic FAU intensities. Also, the gaze angles of the subject are tracked for each frame of the input video.

The 19 one-dimensional signals (17 FAU and 2 gaze signals) are concatenated channel-wise, and this signal is fed as input to a model for video classification. In particular, the model used is the Temporal Convolutional Network (or TCN). The presented approach chooses to use those FAU waveforms as a higher-level representation as opposed to raw pixels from the input video. As deception detection datasets are very small, models that operate directly on raw videos are likely to overfit to background noise. Our approach assumes that the chosen high-level representation input is more robust than raw videos, which is validated experimentally in the present work.

The chapter provides a framework for retrospective analysis of deceptive behavior. More specifically, given the predicted class of each video, an *Attention Module* is used to calculate the regions over the duration of the video that contributed substantially to the prediction of the model. If the video was classified as containing deceptive behavior, those regions could be indicative of when deception happened. Domain experts could then observe the FAU signals and how they are correlated in those time regions to gain insights about deception indicators.

The effectiveness of the DDRFF approach presented here is validated by comprehensive evaluation in three datasets. Also provided are comparisons with the state of the art, including an ablation study. The approach presented here surpasses the current state-of-the-art methods in deception detection, while at the same time is lightweight and modular. The contributions of the work presented in this chapter are summarized as follows: (a) We propose a novel DDDAS framework that achieves state-of-the-art performance on video-based deception detection as tested on three benchmarks; (b) the framework is modular, lightweight and robust; and (c) a framework is designed for retrospective analysis of deceptive behavior.

The rest of this chapter is as follows. In Sect. 27.2, related work is overviewed. Section 27.3 discusses in a more amplified context the methods developed by the authors. Section 27.4 presents experiments conducted to validate these methods, and Sect. 27.5 highlights the essence and impact of the methods. Section 27.6 offers conclusions and future work.

27.2 Related Work

Video Classification Architectures Motivated by the success of CNNs on image-related tasks and of RNNs on sequence modeling, a natural solution to video classification can be to combine CNNs and RNNs [5, 41]. Another solution for video classification can be feed-forward models that use 3D Convolutions (C3D) [6, 7] to learn spatiotemporal features. In 2014, Simonyan and Zisserman noticed that temporal features are hard to learn only by stacking images, and, therefore proposed to train a two-stream ensemble network [8] that utilizes Optical Flow as a complementary modality to RGB frames. Optical Flow captures motion features and has been shown to be effective for action recognition. Finally, in 2019 [9], a method with sparse sampling to model long-term temporal dependencies was proposed. However, for the task of video-based deception detection, one cannot use any of the previous approaches as an off-the-shelf model. Although, they are very successful in action recognition, they seem to *overfit* to the identity of the person in the video and fail to extract relevant features necessary for deception detection.

Deception Detection from Videos With the introduction of a dataset that contains video clips from real-life cases (e.g., court trials) [10, 11], several methods for detecting deceptive behavior in videos have been developed. However, the size of the dataset is typically very small (104 videos are used in practice). As a result, there are approaches reporting that handcrafted features perform much better than deep features. For instance, in [12] the authors use IDT (Improved Dense Trajectory) as low-level features to train a micro-expression detector that is used along with the IDT features for deception detection. In [13], the authors use a deep learning model that makes video-level predictions by aggregating the predictions made in short snippets, sparsely sampled from the input video. The input to the model consists of a video frame capturing appearance features and five Optical maps that model temporal features. To train this DL model, the authors of [14, 15] make use of meta-learning and adversarial learning modules.

The work in this chapter opts not to make use of such methods for training, since a major objective of our approach is for the models to be interpretable. We will employ a DDDAS approach since it offers interpretability of the model features as well as explainability of the outputs given the data. We will demonstrate that the DDDAS paradigm can benefit AI methods.

Temporal Convolutional Networks Recently, feed-forward architectures are increasingly used for sequence modeling over RNNs. Those architectures are called Temporal Convolutional Networks (TCNs). The TCN's main component is one-dimensional *causal* convolutions, meaning that there is no information leakage from future to past timesteps. By using dilated convolutions, TCNs can have exponential receptive fields relative to their depth; and thus, they are able to model long-term dependencies compared to RNNs. Recent works on speech and language modeling [16–18] replace recurrent architectures and make only use of TCNs. The authors of [19] show that TCNs can outperform baseline recurrent architectures across

a variety of sequence modeling tasks. TCNs are also being used by the signal processing community in a variety of tasks, such as blind source separation [20]. The work presented here, also chooses to use a TCN architecture for the methods developed.

27.3 Method

The work here presents a novel framework for non-verbal deception detection in videos that consists of two main modules: (1) a feature extractor, and (2) a video classification model (predictor), as shown in Fig. 27.1.

27.3.1 Facial Action Unit (FAU) Signals

The face can reveal a plethora of signals related to deception. To create a video-based ADD system, it is important to utilize the FAU signals effectively. Thus, a methodology is needed to extract salient information from video in order to be used in the system developed here. Instead of using as input the raw videos, which may contain a lot of noisy information for the analysis task, a higher-level representation is chosen to make the learning procedure easier and ensure that the model actually learns features relevant to deceptive behavior.

In particular, let $\mathrm{D} = \left\{\mathbf{v}^{(i)}, y^{(i)}\right\}_{i=1}^{M}$ denote the dataset, and which contains M video-label pairs. Each video $\mathbf{v}^{(i)}$ in the dataset is actually a tensor of size $T \times H \times W \times 3$. Instead, of using the video tensor as input to the model the facial features are extracted frame-wise, using the OpenFace toolkit [21]. More

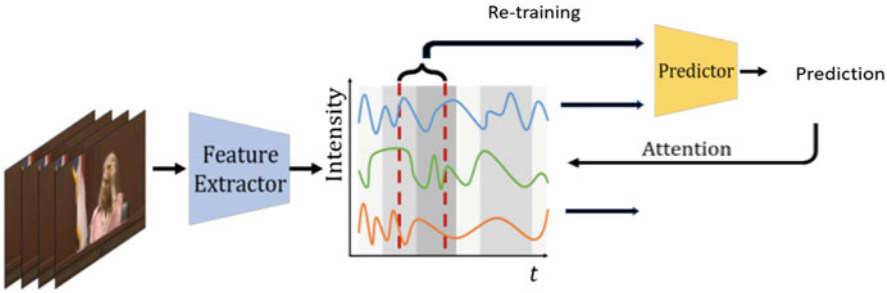

Fig. 27.1 Illustration of the proposed (DDRFF) framework. FAU intensities and gaze angles are extracted from video sequences which are considered as 1D normalized channel-wise concatenated signals to train a predictor model. Model attention is computed to enable retrospective analysis of deceptive behavior. For large videos, we can use the attention to identify key timesteps, which we use for model re-training (DDDAS)

specifically, for each video $\mathbf{v}^{(i)}$, the normalized intensities are computed of N FAUs $\left\{x_j^{(i)}\right\}_{j=1}^{N} \in [0, T]$; in the experiments conducted here, $N = 17$.

The N FAU signals are concatenated as

$$\mathbf{x}^{(i)} = \Big\|_{\text{ch}=1}^{N} \left(x_{\text{ch}}^{(i)}\right) \tag{27.1}$$

where || represents channel-wise concatenation.

The FACS coding system [4] is one of the most comprehensive and objective systems for describing facial expressions. If one thinks of Facial Action Units (FAUs) as a basis, any facial muscle movement can be decomposed to a combination of FAUs. Therefore, the claim is made in the present work, that by replacing $\mathbf{v}^{(i)}$ with $\mathbf{x}^{(i)}$ the system keeps a less noisy representation of the facial behavior for each subject.

The framework presented here is quite general, since it can incorporate a variety of different input features by simply stacking them as extra channels in the input signal. The gaze angles can also be concatenated and used as input to the classification model.

27.3.2 Video Classification Model

The core module of the ADD method is the video classification model. Although this model aims to predict the class of a video, the input to the model is a 1-D signal with 19 channels carrying the appropriate video information and not the raw video itself.

Inspired by the success of TCNs in the domains of signal processing [20] and sequence modeling [16–19], the TCN supports the video classification task. The input to the model is a waveform that carries high-level facial features. This is in contrast to other deception systems [13] which need to extract the necessary features from the raw video for the classification task.

Base Network The input to the model $\mathbf{x}^{(i)}$ is convolved with 128 1-D kernels of size L to produce 128 1-D features maps. Those features maps are then passed through a ReLU (Rectified Linear Unit) and a Batch Normalization [22] layer. The output is then processed by a dense layer implemented as 1-D convolution with kernel size 1.

Residual Blocks and Classification Layer To be able to capture long-term dependencies in the input, our approach uses residual blocks [23] with dilated convolutions [24]. Such a block can be seen in Fig. 27.2. Finally, two other feature maps are average pooled and a Fully Connected layer is applied to get the final prediction.

It should be noted here that the present method can model inputs of arbitrary length, is extremely memory and computation efficient, and is scalable. Further-

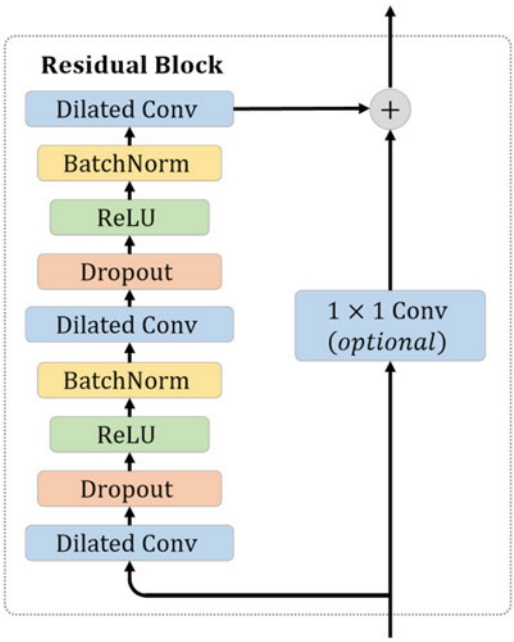

Fig. 27.2 Residual blocks used in the proposed video classification model

more, using a DDDAS approach, we can control the extent to how dependencies are captured in the time domain by changing the model receptive field of the model, which can be accomplished by stacking more dilated convolutional layers, increasing filter sizes, or using larger dilation factors. More details on the implementation of the video-based deception detection model are presented in the next section.

Bayesian Ensemble To boost the performance of our approach, a Bayesian neural network (BNN) variant is introduced to the base model and implemented. Briefly, a BNN tries to estimate the posterior over network weights during training. For a classification model, the predictive distribution over labels given input x is $P(y|x) = \mathrm{E}P_{(w|\mathrm{D})}P(y|x,w)$, where $P(y|x,w) = \mathrm{softmax}(f(x|w))$ are the predicted probabilities given a specific feed-forward network f parameterized by weight w. Notice that since estimating the posterior $P(w|\mathrm{D})$ is often intractable [25, 26], variational inference is commonly adopted. The posterior is approximated by $P(w|\mathrm{D}) \approx Q_\theta(w)$ and trained by maximizing the evidence lower bound (ELBO) [27, 28]. Finally, at testing time, the BNN prediction is approximated by Monte Carlo sampling (with K samples),

$$P\left(y|x;\theta\right) = \frac{1}{K}\sum_{k=1}^{K}\mathrm{softmax}\left(\hat{f}^{(k)}(x)\right). \tag{27.2}$$

27.3.3 Attention Module

Given a trained model, it is beneficial and informative to visualize the most prevalent features for deception detection. To this end, an attention mechanism is used. As opposed to self-attention [29], the method presented here uses Gradient-weighted Class Activation Mapping (Grad-CAM) [30]-like method, which is flexible and does not require any change in the architecture of the model. As a recap, the Grad-CAM assigns importance to each pixel as the gradient of network output w.r.t. a certain feature layer. For 1-D time series indexed by subscript t, denoting the class score as Y^c and feature map in the k-th channel as F^k, where the importance weight can be computed as $\frac{\partial Y^c}{\partial F_t^k}$. Aggregating the importance weights for all pixels, we can obtain the neuron importance weight of the k-th channel for class c, $\alpha_k^c = \frac{1}{z} \sum_t \frac{\partial Y^c}{\partial F_t^k}$ where Z is the normalization constant (global average pooling).

Then the attention map can be obtained from $A^c_{\text{Grand}-\text{CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c F^k\right)$.

Motivated by [31], a positive gradient at a specific location implies increasing the pixel intensity in F^k results in a positive impact on the prediction score. As such, a new channel-weighted attention mechanism A_{ch}, is adopted as:

$$A^c_{\text{ch}} = \frac{1}{K}\text{ReLU}\left(\sum_k \sum_t \text{ReLU}\left(\frac{\partial Y^c}{\partial F_t^k}\right) F^k\right). \tag{27.3}$$

With the help of the Attention Module, we can identify key timesteps in the video. This is useful especially for larger videos, where the subject exhibits deceptive behavior sparsely in time. Following the DDDAS paradigm, the model is re-trained using only the *key frames* discovered by the attention mechanism. Model re-training improves performance for large videos, as shown by the experiments in Sect. 27.4.4.

27.4 Experiments on Video-Based Deception Detection

In this section, we first introduce additional implementation details. Then, we present the evaluation setting of the approach. Evaluation is done on three datasets with comparisons with the state-of-the-art methods. The datasets used are the *Real-Life Trial* dataset [10] and *Bag-of-Lies* [32], which are available to the public. We also used a dataset that contains very long videos of people playing a version of the board game *The Resistance*. In the remainder of this chapter, that dataset is referred to as *The Resistance* dataset. An ablation study is performed on the input components of the model method. Additionally, the Attention Module is used to find the *key frames* for every video, and namely, the frames which influence most the decision of the model. By inspecting the value of the input in those frames,

valuable insight is derived as to what the model "thinks" is deception as a function of facial movements (FAUs intensities).

27.4.1 Implementation Details

Since the videos in the *Real-Life Trial* dataset are captured with different frame rates, the time scale of the extracted features is not similar for every video. To handle this inconsistency, the values are interpolated in the videos captured with frame rates less than 30 fps.

Training The weights of the model are initialized as described in [33]. Training of the model involves sampling k values of the input signal, corresponding to k consecutive frames in the input video. For the *Real-Life Trial* [10] dataset, $k = 180$, while for *Bag-of-Lies,* $k = 110$. The value of k is chosen to be the minimum number of frames that a video contains in the corresponding dataset. In the present work, the models are trained for 100 epochs in total. Starting with a learning rate of 0.001 and using the method described in [34], the model's learning rate is dynamically updated using the gradient with respect to the learning rate of the update rule itself.

In the preliminary experiments, it is observed that inserting residual blocks to the model, as evaluated in for *Real-Life Trial* [10] dataset, did not increase the performance of the model. Thus, it was determined that the base network along with the classifier layer is necessary for this dataset and we omitted any residual blocks. However, for the other two datasets, we observe a small performance boost when using residual blocks, and thus they were included in the evaluation study.

Inference During inference, the videos are split into segments, each of which contains k frames and the model performs a forward pass for each one of them, computing softmax scores for them individually. The final prediction of the model is the averaged softmax scores for all segments.

27.4.2 Experiments on Real-Life Trial Dataset

The *Real-Life Trial* dataset [10] is a publicly available database for the evaluation of deception detection models. It consists of 121 videos from real-life court room trials. As the utility of some videos is questionable, the recent approaches of [12] and [13] have opted to include only a subset of the dataset in their experiments. In particular, both methods use only 104 out of the 121 available videos in the dataset, including 54 deceptive and 50 truthful videos.

Evaluation Protocol For the purpose of evaluation, a tenfold cross-validation was performed, as suggested in [12] and [13]. However, both works claim that the dataset consists of 58 identities and decide to split the dataset into folds based on identities

instead of video samples. Nonetheless, the subset used contains only 42 identities and some subjects appear only in 1 video, while others can appear in more than 10% of the total videos. The imbalanced data can be problematic for the cross-validation folds and, in some cases, a fold can contain only four videos.

The cross-validation evaluation procedure can result in distorted results based on the videos in each validation fold. For this reason, the approach presented here uses a more objective procedure, and namely, the dataset is split based on video samples, ensuring that there are enough videos for validating the model in every fold. Unlike Face-focused cross-stream network (FFCSN) [13], the method presented here does not model a subject's appearance and thus it is robust in that setting.

To evaluate the method in the present work, the average classification accuracy (ACC) is computed, and also the average area under the precision-recall curve (AUC) across the cross-validation folds, as suggested in prior works. Earlier works [10, 11, 35, 36] use the ACC metric, while most recent works [12, 13] use the AUC to account for the imbalance of the positive and negative classes.

Baselines We use two baseline methods to compare our method. The first baseline method is the one proposed in Bag-of-Lies [32] along with the relevant dataset. To do the comparison, a video input is split into 20 chunks and a single representative frame is selected from each chunk. A vector is constructed by extracting Local Binary Pattern (LBP) [37] features for each frame and we concatenate them in the order they appear in the video, as proposed in [32]. The combined feature vector is then used further for classification using Support Vector Machine (SVM) [38], Random Forest [39], and Multi-layer Perceptron (MLP) [40].

The second baseline method used to compare our method is the Temporal Segment Network (TSN) [9]. It is a two-stream neural network that utilizes a sparse temporal sampling strategy and video-level supervision to enable learning using the whole video. For each segment sampled from the video, TSN inputs an RGB image to the spatial stream and five Optical Flow maps to the temporal stream. Then, the outputs from each segment are combined using a consensus function H (such as softmax) to get the final video-level prediction. To compare TSN with our method, the publicly available TSN code is utilized.[1]

Comparative Results The presented method is compared with the state-of-the-art alternative [32] as well as with prior approaches [10–12, 35, 36]. Most of these methods are multi-modal. Thus, comparison with them is on equal terms, which requires reporting their results by using only visual cues. The comparison results are given in Table 27.1.

Table 27.1 shows that the method, introduced here by the authors, outperforms all the other methods on the Real-Life Trial dataset [10]. This validates the hypothesis that the performance of a model which is trained for video-based deception detection

[1] https://github.com/yjxiong/tsn-pytorch

Table 27.1 Comparative results (%) on the Real-Life Trial dataset [10]

Method	ACC	AUC
LBP [37]	75.00	76.15
TSN [9]	77.55	81.78
[36][a]	67.20	–
[10][a]	68.59	–
[11][a]	75.42	–
[35][a]	78.58	–
[12][a]	–	83.47
FFCSN [13][a]	89.16	91.89
FFCSN [13][a, b]	*93.16*	*96.71*
Ours	**92.36**	**97.27**

Note that for all methods, the results are reported only with visual cues, even if multi-modal results are given as well
For methods indicated with an "[a]", we report the results directly from their papers
For "[b]" the authors use meta-learning and adversarial learning. The results do not include any data augmentation

Table 27.2 Ablation study results (%)

Modality	ACC	AUC
Gaze	79.73	83.05
FAUs	91.15	95.43
FAUs + Gaze	**92.36**	**97.27**

will benefit by using as input higher-level features, instead of raw videos. The results in Table 27.1 show that by using 1-D features, one can create a model that is simple and easy to train, yet performs better than previous approaches.

Ablation Study Results The ablative study is conducted here on the input features of the new method presented by the authors in this chapter. In particular, the performance of the model is measured by using (1) only Gaze signals, (2) only FAUs signals, and (3) FAUs + Gaze signals. The results are shown on Table 27.2. As expected, the method performs better using FAU signals than using just Gaze signals. However, the results show that the best performance is obtained when the gaze and FAU features are combined, meaning that FAU and Gaze signals are complementary.

Table 27.3 Comparative results (%) on the Bag-of-Lies dataset [32]

Method	ACC	AUC
LBP [37]	55.12	55.32
TSN [9]	56.94	57.62
Ours	**64.47**	**67.08**

27.4.3 Experiments on Bag-of-Lies

In another experiment, the *Bag-of-Lies* [32] dataset is used for evaluating our method. This study consists of 35 subjects, each of whom is shown 6–10 images and then is being asked to describe them. Each participant is free to describe the image honestly or deceptively and the answer is recorded in a video. The video recordings do have the same length, they are ranging from 3.5 seconds to 42 seconds.

The total number of samples in the dataset is 325 with an even distribution of truth (163) and of lie (162) samples. Although this dataset offers information on other modalities as well (audio and electroencephalogram (EEG) signals), here only the visual modality is used in our experiments.

Evaluation Protocol The evaluation of the method presented here uses the same protocol as in [32]. A threefold cross-validation is performed across participants (with 12, 12, and 11 participants in each fold). The same metrics are used for the evaluation of deception detection methods as in the *Real-Life Trial* dataset. Similarly, the results reported are average of cross-validation over folds.

Baselines To evaluate the method on the *Bag-of-Lies* dataset [32], the same baselines are used as those for the Real-Life Trial dataset. The implementation of the method using LBP features matched the results reported in [32].

Results Table 27.3 shows the results of the proposed method and the two baseline methods. One can see that the proposed method clearly outperforms both. Since the *Bag-of-Lies* dataset was introduced recently, there are no other methods that report results based on using this dataset. One thing to note is that this dataset is more challenging compared to the *Real-Life Trial* dataset.

27.4.4 Experiments on the Resistance Game

The last experiment contains a set of videos that capture a group of 5–8 people while playing a version of the *Resistance Game*, a social role-playing game.

Players in the Resistance Game are randomly given one of two roles, deceivers or truth-tellers. Deceivers know who are the individuals who have the same role as them in the game, whereas truth-tellers have no clue on who are deceivers or the other truth-tellers. The majority of players in a game is truth-tellers and it's assumed that there are 2–3 deceivers. The game proceeds in rounds or missions as they are called in the game. There are three to seven missions in a game.

The players should nominate and elect a leader, who in turn nominates team members to go on a mission. All the players vote if that particular team should go on a mission or if a different team should be chosen. When on a mission, the team members vote in secret for the success or failure of the mission. The deceivers want the mission to fail, while the truth-tellers want it to succeed. When a mission succeeds, all truth-tellers get one point, whereas when it fails the deceivers get a point. The team with the highest score at the end of the game wins. The game relies on deceivers trying to hide their identity and attempt to prevent the larger group from working together to reveal the deceivers among the group. Furthermore, it provides players with a lot of opportunities to exhibit deceptive behaviors.

The dataset contains a set of videos involving 285 players collected from five sites in three different countries to account for possible heterogeneity in deceptive behavior among different cultures. The videos used are very long and their average duration is 46 minutes. In the experiments conducted, a balanced subset of the dataset was used, containing 230 videos.

The videos in this dataset are very long, and therefore deception only occurs in a few and short duration parts of the video. Thus, by using the Attention Module to identify key timesteps for deception detection and re-training the detection model, performance is improved, as can be seen in Table 27.4.

Evaluation Protocol All videos contain different persons. A fivefold cross-validation across videos is also performed. To evaluate this approach, we use the same metrics and baselines as in the previous datasets.

Results In Table 27.4, one can see the results of the presented method and the baseline. *The Resistance* dataset is very difficult, which can be noticed by the fact that both baselines perform no better than a random classifier in that dataset. It can be speculated that this happens because not enough data are available. The input videos are very long and the supervisory signal is weak (only 1 label for the whole video). As a result, both baselines overfit to background noise and fail miserably.

Table 27.4 demonstrates that the present approach achieves substantially better performance than both baselines. This can be attributed to the fact that the new method uses FAU and gaze signals as input, which can be beneficial for the learning procedure. The model is asked to learn parameters that model the correlations of high-level facial information, and it is robust to background changes and other noisy information present in the videos.

Furthermore, re-training the model with the key frames identified by the Attention Module increases the performance even further. This can be attributed to the

Table 27.4 Results (%) of our method on the Resistance Game dataset. Since a balanced subset of the dataset is used, only the classification accuracy is reported

Method	ACC
LBP [37]	49.56
TSN [9]	51.15
Ours	**71.08**
Ours + re-training	**74.52**

Fig. 27.3 The Real-Life Trial [10] dataset: (*left*) Screenshot of frames from original videos; (*middle*) facial landmark and head-pose bounding box visualizations by OpenFace [21]; (*right*) FAU waveforms and attention visualizations of the predictor model

fact that our method has a mechanism to adaptively include more informative data for training as explained previously (DDDAS).

27.5 Attention Visualization

The Attention Module detects the *key frames* of the input video for deception detection. Using this method, one can analyze the frames that the associated model considers important for classifying a video as one that contains deceptive behavior. One can treat the attention aspect as an extra 1-D signal and visualize it along with the other waveforms to search for consistent patterns in deceptive behaviors. An example of such visualization was shown in Fig. 27.3.

From the experiments conducted, one can conclude that the facial signals constitute important cues for detecting deception in videos. The authors believe that the Attention Module can be used as a tool to quantitatively study the role of facial signals as deception indicators.

27.6 Conclusion

This chapter presents a novel framework for video-based deception detection and analysis of deceptive behavior. By using one-dimensional FAU and gaze signals, the authors show that one is able to train a conceptually simple, modular, and powerful model that performs really well in practice. The comprehensive evaluation results illustrate that the model achieves state-of-the-art performance, even though it is far less intricate than previous approaches. This highlights the usefulness of facial information for the task of non-verbal deception detection. Finally, this chapter presents a novel approach to interpret the model's predictions, by computing the attention of the video classification model used. The framework supports DDDAS methods as it can dynamically choose the "right" data to be used for model re-training, improving performance when large videos are used. Finally, the Attention Module has proven to be a useful tool for retrospective analysis of deceptive behavior by domain experts.

References

1. L. Tran and X. Liu, "Nonlinear 3d face morphable model," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7346–7355, 2018.
2. D. B. Buller and J. K. Burgoon, "Interpersonal Deception Theory," *Communication Theory*, vol. 6, no. 3, pp. 203–242, 03 1996. [Online]. Available: https://doi.org/10.1111/j.1468-2885.1996.tb00127.x
3. J. Burgoon, L. Guerrero, and K. Floyd, *Nonverbal communication*, 1st ed. Allyn Bacon, 2010.
4. P. Ekman and E. L. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, 1997.
5. J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, April 2017.
6. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4489–4497.
7. S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan 2013.
8. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 1, 06 2014.
9. L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2740–2755, Nov 2019.
10. V. P'erez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: ACM, 2015, pp. 59–66. [Online]. Available: http://doi.acm.org/10.1145/2818346.2820758
11. V. P'erez-Rosas, M. Abouelenien, R. Mihalcea, Y. Xiao, C. Linton, and M. Burzo, "Verbal and nonverbal clues for real-life deception detection," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2336–2346. [Online]. Available: https://www.aclweb.org/anthology/D15-1281
12. Z. Wu, B. Singh, L. Davis, and V. S. Subrahmanian, "Deception detection in videos," *AAAI*, pp. 1695–1702, 2018.
13. M. Ding, A. Zhao, Z. Lu, T. Xiang, and J.-R. Wen, "Face-focused cross-stream network for deception detection in videos," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
14. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
15. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.
16. J. Gehring, M. Auli, D. Grangier, and Y. Dauphin, "A convolutional encoder model for neural machine translation," 01 2017, pp. 123–135.
17. N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," 2016. [Online]. Available: https://arxiv.org/abs/1610.10099
18. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

19. S. Bai, J. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 03 2018.
20. Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019. [Online]. Available: https://doi.org/10.1109/TASLP.2019.2915167
21. T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, May 2018, pp. 59–66.
22. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167
23. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
24. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.
25. C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.
26. A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.
27. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
28. Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L . Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
30. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
31. L. Wang, Z. Wu, S. Karanam, K.-C. Peng, R. V. Singh, B. Liu, and D. N. Metaxas, "Sharpen focus: Learning with attention separability and consistency," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 512–521.
32. V. Gupta, M. Agarwal, M. Arora, T. Chakraborty, R. Singh, and M. Vatsa, "Bagof-lies: A multimodal dataset for deception detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
33. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 1502, 02 2015.
34. A. Baydin, R. Cornish, D. Rubio, M. Schmidt, and F. Wood, "Online learning rate adaptation with hypergradient descent," 03 2017.
35. M. Gogate, A. Adeel, and A. Hussain, "Deep learning driven multimodal fusion for automated deception detection," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–6.
36. M. Jaiswal, S. Tabibu, and R. Bajpai, "The truth and nothing but the truth: Multimodal analysis for deception detection," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 938–943.
37. T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.
38. C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, p. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411

39. Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, Aug 1995, pp. 278–282 vol.1.
40. G. E. Hinton, *Connectionist Learning Procedures*. IEEE Press, 1990, p. 11–47.
41. Haiqiang Zuo, Heng Fan, Erik Blasch, and Haibin Ling, "Combining Convolutional and Recurrent Neural Networks for Human Skin Detection, "*IEEE Signal Processing Letters*, 24(3):289–293, Mar 2017