Show Me Your Face: Towards an Automated Method to Provide Timely Guidance in Visual Analytics

Davide Ceneda, Alessio Arleo, Theresia Gschwandtner, and Silvia Miksch

Abstract—Providing guidance during a Visual Analytics session can support analysts in pursuing their goals more efficiently. However, the effectiveness of guidance depends on many factors: Determining the right timing to provide it is one of them. Although in complex analysis scenarios choosing the right timing could make the difference between a dependable and a superfluous guidance, an analysis of the literature suggests that this problem did not receive enough attention. In this paper, we describe a methodology to determine moments in which guidance is needed. Our assumption is that the need of guidance would influence the user state-of-mind, as in distress situations during the analytical process, and we hypothesize that such moments could be identified by analyzing the user's facial expressions. We propose a framework composed by a facial recognition software and a machine learning model trained to detect when to provide guidance according to changes of the user facial expressions. We trained the model by interviewing eight analysts during their work and ranked multiple facial features based on their relative importance in determining the need of guidance. Finally, we show that by applying only minor modifications to its architecture, our prototype was able to detect a need of guidance on the fly and made our methodology well suited also for real-time analysis sessions. The results of our evaluations show that our methodology is indeed effective in determining when a need of guidance is present, which constitutes a prerequisite to providing timely and effective guidance in VA.

Index Terms—Guidance, visual analytics, emotions, facial analysis, machine learning

1 Introduction

Visual Analytics (VA) aims at enabling a better collaboration between humans and analytical systems by means of interactive visual interfaces [1]. Despite being straightforward to understand, the VA process hides challenges at any of its steps making it difficult to apply in practice.

For this reason, in parallel to the development of VA methods, scientists have been studying approaches to help analysts using them. The science of assisting analysts has roots in interaction science and visual interface design [2], [3]. All their nuances can be grouped together under the term *guidance* [4]. As new guidance methods are being developed, new challenges arise whose solution is vital for the instantiation of effective assistance. Among these, detecting the most appropriate moment for providing guidance is crucial, but mostly unaddressed [5]. However, as analytical methods become more and more complex, timely guidance cannot be disregarded anymore. Since different guidance is needed at different moments of the VA process, choosing the right timing is crucial to make the guidance effective. Conversely, choosing a wrong timing may mislead and sway the analyst and interfere with the analysis as a whole.

Until now, research in VA has mainly considered the analyst's interactions for detecting *what* the user may need during the analysis. For instance, if a certain incorrect behavior is detected, some countermeasures – such as providing guidance – could be taken. However, *when* such guidance should be provided is a problem that has not been sufficiently addressed in VA. To determine the right timing for providing guidance, at least two ingredients are needed: First, there should be a need for guidance. Second, the user should be ready to accept it. While the second problem entails the consideration of subjective factors and specific nuances of the single user's personality e.g., stubbornness in rejecting guidance, the first one relies on common psychological mechanisms. Within this context, we present a methodology to determine if and when a need for guidance is present, thus making important steps towards determining the correct timing for providing guidance.

Our research is grounded on the recognition of specific facial expressions that can be seen as a *doorway to the analyst's state of mind* and hence can be directly associated to a need for guidance [6]. Facial expressions are a direct consequence of changes of the analyst's state-of-mind as, for instance, in response to distressful situations during the analysis. Thus, in this work, we propose a framework to analyse a user's face and detect such changes on the fly, hence identifying moments when guidance is potentially needed. We describe and evaluate an implementation of our approach with three methodologies. The results obtained show that our method is effective in determining when the user needs guidance and shed light on the steps needed to make guidance accepted (and hence, effective) in VA. Our contributions are:

- We present a proof-of-concept solution for detecting the moment when guidance is needed exploiting facial expressions of analysts doing VA.
- We describe the training and use of a machine learning model (ML, Random Forrest) to analyze facial expressions and automatically identify the need for guidance in real time (Section 4).

The authors are with TU Wien, Austria E-mail:{name.surname}@tuwien.ac.at

- We evaluate our methodology by evaluating the trained model with both test data and in a real-time analysis scenario (Section 5) discussing its advantages, possible limitations, and illustrating how subjective traits, such as the willingness of a user to accept the guidance, are factors that play a crucial role in determination of the most appropriate timing to initiate guidance (Section 6).

2 Related Work

In this section, we describe related research in cognitive sciences, human emotion recognition and HCI.

2.1 Guidance in VA

Prior research has defined the goal of guidance as helping the analyst to overcome a *knowledge gap* [4], [7], [8]. This knowledge gap is related to the difficulties the analyst faces when solving tasks. For instance, exploring the data, or choosing appropriate analytical methods are common issues that guidance aims to alleviate. One of the first approaches providing guidance is GADGET [9]. GADGET supports analysts while creating their own visualizations. GADGET suggests possible additions to the visual design by confronting the data and a description of the tasks with a knowledge base of previously created visualizations. On the same line is VisComplete [10], which also aids analysts in creating visualizations. The system is built upon a knowledge base comprising typical visualization pipelines. Focusing on the user's interaction, the system is capable of suggesting viable visualizations. Gotz et al. [11] proposed a guidance method to suggest the most appropriate visualizations for a given task. The system automatically extracts a descriptor of the current task the analyst is pursuing, based on the interaction, and proposes them possible additions to enhance the current analysis process.

While we described just a few guidance approaches, many more exist in literature [5]. Accordingly, existing approaches do *not* really consider what the most appropriate moment to initiate guidance could be. Usually, such approaches are more interested in determining *what* type of guidance could be appropriate and provide it right away, which, as we will see in Section 3, may not always be a good strategy.

2.2 Recognizing Human Affects

Affect recognition refers to the process of identifying human emotions. Although the accuracy in discerning emotions varies from subject to subject, the ability to roughly say what a person is feeling is typically an innate ability that is tied to our evolution [12].

Human emotions are complex mental states associated with our nervous system [13]. There is not yet a commonly accepted definition of emotion. However, there is a consensus that emotions are strongly connected to experiences and events that we live. In general, many studies showed how emotions occur as a consequence of internal and external stimuli, namely, emotion elicitors, causing physiological and psychological changes in our bodies [14].

This very same sequence *event – emotions – bodily reaction* is also at the base of the way we interpret others' emotions. For instance, feeling anger can be deduced from how others articulate their speech or the tone of their voice. This theory has inspired many researchers to look for methods to automatically extract emotions from *visual* e.g., images and videos, and *non visual* stimuli, e.g., skin conductance and hearth beat [15].

In visual data analysis, a big part of literature highlights the importance of maintaining an appropriate state of mind during the analysis to foster insights [16], [17]. Data analysis can be considered an *emotion elicitor*, like many other activities we perform. Executing tasks and performing data analysis has a direct impact on the emotions and on analysts' state of mind. Sensations of being lost may arise, for instance, when facing difficulties but also feeling frustrated or sad might also be a sign of issues during the analysis process [18], [19]. In this paper, we make a step forward exploiting such emotions to our advantage. We present an automatic method that detects analysts' emotions by analyzing their facial expressions, identifying distress situations, and subsequently initiate guidance.

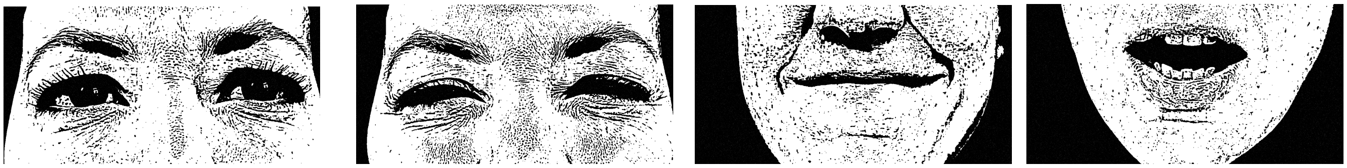
The Facial Action Coding System (FACS) [20], [21], [22] has been for years the main method for categorizing facial expressions. Emotions have a direct effect on facial traits which are expressed by the simultaneous movement of multiple facial muscles. In this respect, the FACS enumerates the so called Action Units (AUs) which represent movements of single facial muscles. Hence, AUs are a *systematic way* to study facial configurations and human affects. Emotions can be seen, in fact, as the simultaneous presence of multiple muscle movements (AUs).

The paper by Ekman and Friesen described sixty-four action units (AU01 to AU64) [20], [21]. Nowadays, frameworks for facial analysis are able to extract most of them in real time and also determine the intensity of the detected movements [23]. For instance, AU14 (see Figure 1c) is connected to the action of the buccinator muscle and it is involved in the appearance of mouth dimples. If a person when thinking or talking uses such muscles, the corresponding presence of AU14 will be positive and its intensity related to a numerical value, e.g., 1–5. A list of further AUs can be found in Table 2 and Figure 1. As we will see, different AUs have a different importance in determining a need for guidance.

2.3 Detecting Interruptibility

Detecting a need for guidance is closely related to the research area of interruptibility, which investigates how users can be interrupted as they perform different duties [24].

In human-computer interaction (HCI), the system developed by Tsubouchi et al. [25] analyses the user's activity with a smartphone to detect appropriate moments to send notifications. They further show how providing notifications on time is the key to obtain a suitable response from users. Similarly to the previous paper, Züger et al. [26] investigated when software developers could be interrupted during their work day. Differently from the previous papers, the authors consider a larger set of input to decide when users can be interrupted. Specifically, biometric data (e.g.,



(a) AU01 Brow Raiser (b) AU07: Lid Tightner (c) AU14: Dimpler (d) AU26: Jaw Drop

Fig. 1: Facial expressions associated with a need for guidance. Events and experiences, like issues occurring during VA, have a direct effect on our mental state and emotions. When emotions occur our body reacts to them and our facial traits change. We can analyse facial traits (AUs) and use them as hints to initiate guidance. (a) AU01 is associated with the brow raiser muscle; (b) AU07 is related to the lid tightener; (c) AU14 regulates the appearance of dimples; (d) AU26 regulates the movement of the jaw.

heart beat) and interaction data are employed to obtain accurate predictions. Interruptions can be counterproductive for learners, i.e., students. Qu et al. [27] investigated when students can be interrupted as they learn new concepts. The authors utilize a combination of different input sources derived from the students' interaction with the learning environment and eye-gaze activity to detect their focus and attention span. Finally, also Barral et al. [28] exploited eye-gaze data to enhance and support the exploration of narrative visualizations in users with different levels of visualization literacy.

Whereas these approaches show how "negative" emotions and the disruption of the workflow are the result of choosing to interrupt the user at the wrong moment, our aim is to investigate if and when those emotions can be instead a signal for guidance.

Interruptibility in the Visualization Field. Moving to visualization, most of the literature is not concerned with deciding if a user can be interrupted. Conversely, visualization approaches assume that users can be indeed interrupted because they are already stuck and need assistance. In other words, the visualization literature is more focused on detecting *what* guidance may be needed instead of *when* this should be provided. For instance, Cook et al., [29] exploited task descriptors to assist the completion of mixed-initiative VA tasks. In a study that is more closely related to ours, Fan et al. [30] used ML techniques to analyze the speech of users using a visual interface. Using speech analysis, the system provides feedback to interface designers. Conversely to our purposes, this method works only with speak aloud protocols and during the design phase, for usability and testing purposes. Thus, it requires an active effort of the end-user. In our work, we do not focus on the design process but rather in analysing users' visual appearance *during the analysis*, so that no active effort is required from them to express their need for guidance.

3 Towards Timely Guidance in VA

While the research in HCI showed us how choosing an inappropriate moment can have dire consequences for the user activity [31], the guidance approaches we described in Section 2.1 are mostly concerned with deciding *what* guidance the user might need to continue the analysis.

We argue that typical analysis scenarios are generally more complex than those described in the aforementioned studies on guidance. Typically, many events take place between the start of the analysis and moment when guidance is needed. Hence, at a certain moment, guidance may or may not be needed at all depending on how the analysis developed until that point. In such complex situations, the aforementioned guidance approaches would probably fail in determining if the guidance is needed with the consequence that their efficacy would be compromised.

An early work tackling the problem of timely guidance has been authored by Mark Silver [2]. Silver states that guidance should be provided when there is an *opportunity*. Such an opportunity is related to the existence of a decisional moment in which the analyst is required to make a "discretionary judgement", like deciding about the next step to make. For instance, in VisComplete [10], the analyst has to decide how to complete the visual design and only then the system provides guidance. In absence of such moments, Silver argues the benefits of providing guidance are minimal.

However, detecting decisional moments is not always possible. Battle et al. [32] state that performing exploratory analysis is a clear example of such situations. Other examples of problematic situations are, for instance, stalled analyses. In stalled analyses, it is usually not immediately clear if analysts are simply doing something else or if they just do not want any assistance. Finally, analyzing the interaction history alone, which represents the solution adopted by many literature approaches could also not achieve a sufficient level of accuracy.

In summary, looking for interaction patterns and analysing task descriptors, as described in Section 2.1, may be useful to detect what guidance the user may need but show their limitations when applied to decide *when* guidance has to be provided. Our argumentation is that providing guidance on time is a complex decision, which partly involves the presence of a real need for guidance and the user's willingness to accept it. While user's propensity to receive help is more strictly related to the problem of interruptibility and tied to specific traits of the single user's personality, e.g., stubbornness in rejecting help, tackling the first challenge instead has more to do with how humans deal with data analysis and how they face problematic situations. In this paper, we describe a novel solution to tackle the latter problem.

Our hypothesis is that changes of facial expressions could be exploited to determine if guidance is needed at a given moment.

Therefore, we describe how we built and evaluated a framework to detect a need for guidance by detecting specific facial features, thus shading light on a possible time span to safely provide it to the user.

4 A ML Approach to Timely Guidance

Our method can provide an answer to the question – *"Does the analyst require guidance?"* – by analyzing a video stream of the analyst's face. In the following, we describe the procedures we set up to collect and label data for training and testing purposes.

4.1 Overview of the Training Procedure

We provide a short overview of the training process, which is portrayed in Figure 2. To collect training data, we set up a set of tasks to be performed on a large dataset using Tableau Desktop[1]. Using a webcam, we collected a video of the analysts' face while performing tasks. We employed OpenFace v2.0, a state-of-the-art tool for facial expression analysis [23] on the recorded footage to extract the analysts' facial features. The participants could ask for guidance when needed using a button. When this happened, the software stored the corresponding time which was used to label the data. An evaluator assisted the process and could add additional labels. The evaluator was positioned in a deferred position and s/he did not interact with the participants except for moments when they explicitly requested guidance. Finally, the labeled data was fed to a ML algorithm for discerning moments when guidance was needed. A detailed description of the training process follows.

4.2 Task and Dataset

Task Requirements. When designing the task and the evaluation procedure, we defined three requirements we wanted to meet: (1) The task should include open exploration and there should be multiple alternative ways for solving it, so that not all users would face the same problems at the same time. (2) The task should be general enough to demonstrate the applicability of our methodology to multiple VA scenarios. (3) The task should not be too simple not to require guidance.

Given the requirements, we settled for an open-ended exploratory data analysis scenario [33]. This allowed us to focus on the definition of a generic analysis goal rather than on defining a precise list of tasks. This also allowed us to test our methodology in a situation in which other strategies usually fail – see [32]. In summary, we let analysts free of choosing the strategy they wanted.

Dataset. For the analysis, we chose a dataset of reported wildlife incidents with aircraft[2]. The dataset consists of approximately 180k rows describing wild animals, e.g., birds, striking aircraft. The dataset is regularly maintained by the Federal Aviation Administration and contains many dimensions, like the weather conditions, the type of aircraft and animals involved, etc..

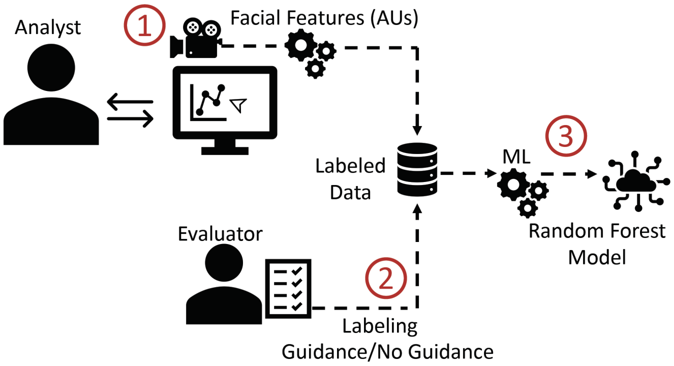


Fig. 2: The procedure we set up to train a ML model: (1) we collected facial features of analysts performing data analysis; (2) we labeled this data according to the reported moments in which they needed guidance; and (3) using this labeled data, we trained a ML model to automatically detect a need for guidance.

Open-ended Task. We asked the analysts to solve the following task:

You are put in charge by an institution (e.g., the government, or an national aviation agency) to analyse the data using a VA tool, understand the phenomenon and find a possible solution to the problem of animal striking aircraft.

We asked them to analyze the data and produce a set of guidelines or suggestions to possibly reduce the number of events. For instance, participants tried to analyse if accidents were caused by specific animals or on the time of the day. On the base of such findings, they imagined possible solutions, like installing birds dissuaders or make aircraft parts undergo deeper maintenance routines.

The task we chose is open to many solutions and multiple resolution strategies. The dataset allows users to explore multiple data dimensions. Also, the topic does not require specific domain knowledge and can be easily grasped by non-experts. Still, to be sure everyone understood the task, we prepared supplementary material comprising a description of all the data fields and three newspaper articles describing aircraft accidents due to wildlife strikes, each one focusing on different aspects of the problem to get them involved in the task. We handed them this material before the study began.

4.3 Participants and Software Environment

We asked eight visualization experts to take part in the data collection procedure (i.e., recording their facial expressions when trying to solve the given task). They are all part of our research group and experts in using visualization tools.

After signing a consent form concerning the video recording, we asked them to perform the open-ended task using Tableau. We chose this tool because it offers a direct way to explore data, create visualizations and it fits our requirement for open-ended exploration. Whereas none of the involved participants used Tableau with assiduity, many of them were already familiar with it. However, to be sure all participants had a similar knowledge of the environment before starting the study, we provided all of them an introductory tutorial to the tool, using a different dataset –

[1] https://www.tableau.com, accessed:05/2021
[2] https://wildlife.faa.gov, accessed:05/2021

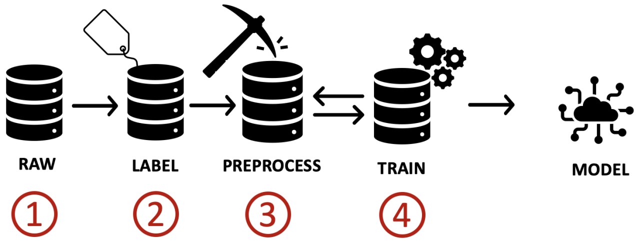


Fig. 3: Following a standard training procedure: (1) raw data was produced by OpenFace. (2) we labeled the data, either manually and automatically. (3) we pre-processed the data keeping only features relevant to our study. (4) we trained the ML model. Eventually, we repeated (3) and (4) to refine the model and obtain higher accuracy.

the Super Store dataset integrated in Tableau. Through the tutorial, we guided them through the main characteristics of the tool. For instance, the tutorial asked them to create a bar chart by selecting two dimensions, the sales of a shop and the sales' date, and later asked to filter out some data (i.e., display all but the sales related to chairs and office supplies).

4.4 Model Training

During the study, we captured facial features of the analysts which we used for training. The training pipeline is represented in Figure 2 and 3.

4.4.1 Data Labeling

The whole open-ended analysis lasted between 30 to 40 minutes per participant, introduction and tutorial excluded. At the end of the process, the video of each participant was processed with OpenFace for extracting the facial features (see (1) in Figure 2). The video was captured at a resolution of 1920x1080 pixels and 30 frames per second. Each frame was analysed to extract a feature vector, which corresponds to a row of the training dataset.

Afterwards, we labeled the data (see (2) in Figure 2), i.e., each row was identified with one of two labels. The labeling was performed, in first instance, by using labels created by the user during the study. Additional labels were inserted manually, in a subsequent moment, as detailed in the following paragraphs.

Automatic Labeling. During the analysis, we encouraged participants to ask for guidance when needed by pressing a button we positioned on the top-right corner of the Tableau interface. When pressed, the button stored the time. Thanks to the timestamp recorded, all the data falling in the instants preceding the timestamp were automatically labeled as instants in which guidance was needed. The guidance could be requested at any time and any number of requests could be submitted during the whole procedure. The actual guidance support was then provided by the evaluator. Typically, the participants asked the evaluator how certain functions of Tableau could be accessed or used. The evaluator pointed them to the requested functions, or to the menus to access them. Other times, the participants could not understand how to interpret certain graphs. In those rarer occasions, the evaluator suggested them how to change visualization, or read the graph. The provision of guidance affected participants' *feelings* as a stark decrease of the need for guidance.

Manual Labeling. We complemented the initial labeling with additional labels. During the procedure, we encouraged participants to think aloud. In particular, we asked them to talk about their current analysis goal and if they were facing difficulties while pursuing it. Hence, the evaluator was always aware of the analysis status and could possibly help and provide guidance if necessary. The participants were already familiar with talk-aloud protocols and it was perceived as a natural behavior, especially when users experienced issues. In case we noticed they were silent for protracted periods, the evaluator encouraged them to talk by asking simple questions, as "what are you doing right now?" or "what are you trying to achieve?".

The think-aloud protocol was not used directly as an input to the ML model. However, it helped us detecting additional moments in which guidance was potentially needed, i.e., when the button was not used. To detect such moments, we payed attention to cues and subtle requests for guidance, like analysts exclaiming *"I am experiencing problems doing this..."* or *"I do not know how to do that..."*. When this happened the evaluator marked the corresponding timestamp which was later considered as an additional moment when users needed guidance. This was done because we noticed that the guidance button was pressed just in extreme situations, like for instance, when the analysts were really unnerved or when they could not continue the analysis. This procedure allowed us to consider also those moment in which guidance was not directly requested, but some lower level of guidance could have been advised. In average the button for requesting guidance was pushed five times per session. In addition, the evaluator noted on average three additional moments per session.

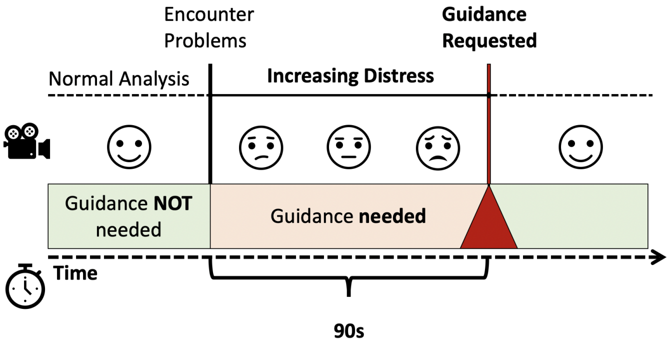


Fig. 4: The labeling scheme. We label the data as "guidance is needed" in correspondence of moments of increased distress. Remaining data frames were labeled "guidance not needed".

Labeling Scheme. The timestamps detected with the aforementioned procedures were used to label the data. The labeling was performed using R[3]. The labeling scheme is portrayed in Figure 4. Specifically, we chose a binary scheme: *"Guidance is needed"* and *"Guidance is not needed"*.

We assumed that in the moments preceding the press of the button, the symptoms of distress would have resulted

[3] https://www.r-project.org accessed: 05/2021

in evident facial expressions. Hence, we labeled the data collected in such intervals as “Guidance is needed”. The precise time interval was set to 90 seconds, i.e., all the data collected in the *90 seconds* interval preceding the actual push of the button was marked with this label. The negative label was assigned to all the other frames. In this way we put in correlation facial expressions and guidance requests.

The 90 seconds interval was chosen because we noticed that in that time analysts started to show signs of distress. For instance, many started to frown. For the sake of being exhaustive, we also tried shorter and longer intervals. In the end, the 90 second interval proved to be the most appropriate.

4.4.2 Preprocessing

In the third step of our pipeline (see Figure 3), we applied a dimensional reduction to check for unnecessary dimensions, while testing the accuracy of the resulting model. The raw data coming from OpenFace includes more than five-hundred dimensions and fifty-five AUs. Besides the AUs, a myriad of other low-level data corresponding to facial characteristics and position of facial landmarks are captured, including the x,y, and z coordinates of the single points composing them. For our purpose, most of such dimensions were redundant, as for instance, the position of single landmarks and their coordinates. Therefore, although we wanted to take initially into consideration a higher number of features, at the end of this phase, we kept only the fifty-five AUs captured by OpenFace.

The dataset was generally very tidy. However, we had to take care of the rows corresponding to video frames that OpenFace was not able to interpret (less than 1% of the total rows). This happened, for example, when the analyst’s face moved outside the video frame. Usually, this situation lasted no more than a couple of seconds. A negative value in the ”success” dimension was sufficient to identify such cases.

The AUs describe the presence (or absence) of facial features as categorical values (AU present/not present). For a correct training – most of the ML algorithms work better with numbers – these dimensions were transformed into numerical values using a one-hot encoding. Also the values representing the intensity of the AUs were normalized to values between zero and one for better processing.

Finally, we performed a feature selection. Theoretically, it exists a subset of dimensions that contributes the most to determine if guidance is needed or not. Appropriate feature selection also helps to avoid the risk of over-fitting the model, since it avoids unnecessary dimensions. Feature selection can be done either automatically i.e., an algorithm chooses what features to keep based on a ranking of their importance, or manually, where the control over which feature to keep is kept by the analyst. We decided for mix of automatic and manual selection. In other words, we ran the automatic feature selection algorithm, we inspected the results and decided which dimension to keep.

At the end of the process, we kept all the AUs and the AUs intensities but we removed all the dimensions related to the head pose and eyes position. The eye position, for instance was removed because we thought it could lead to a biased model. In our setup, the button for asking for guidance was positioned in the top-right corner of the analysis environment. This forced analysts to move the head and eyes towards that corner before asking for guidance. If we would have included this dimension, the system could have learned that when analysts looked at the top-right angle of the screen they needed help. Hence, we removed them. The head pose dimension captures the location of the head with respect to camera, in millimeters, and it was also a potential source of bias. The values are calculated considering the camera as the origin of the coordinates system. Since the position of the camera may vary, we decided to remove also this feature. Finally, the system also suggested to remove (i.e., it had a very low ranking) the head pose and position. Removing such dimensions resulted in increased accuracy.

4.4.3 Training

In a last step, we trained our ML model with the preprocessed data (see (3) in Figure 2): We tested multiple algorithms and compared their prediction accuracy. A random forest classifier (RF) performed best and was kept for further evaluation.

Figure 5 shows the results of this comparison, performed automatically using R, which took care of the whole process. With the aim of having a fair comparison R fine-tunes all of them with a same strategy. We applied a ten-fold cross validation repeated five times to fine tune the hyperparameters of all the ML models. To understand what this strategy means in practice, we can think of how the RF algorithm works. During the training, while the model builds multiple decision tree, the tuning process optimizes the number of trees and the number of variables that are used to split and branch them [34]. Usually this parameter is referred to as *mtry*. In our case, after the tuning, fourteen variables were chosen ($mtry = 14$) for the best performing model. The comparison among the models was based on the following metrics: (1) The ROC-AUC, which measures the area under the receiver operating characteristic (ROC) curve. The curve plots the true positive rate of a model against its false positive rate. The area under the resulting curve (AUC) shows how much a model is able to discern between classes. (2) The sensitivity, sometimes called *recall*, measures the true positive rate, which is the proportion of actual positives cases that are correctly identified as such. Consequently, high sensitivity of a model means that the number of false negatives is low. Finally, (3) the specificity measures a model’s ability to correctly identify negative cases – in our case, moments when the analyst did not need guidance. A high specificity is also an indication of few false positives.

After preprocessing, the dataset still consisted of about 200k+ rows. Training times mostly depended on the algorithm used: Some of the models, like SVMs, are generally not fitted to elaborate this huge amount of data because its kernel requires memory space that scales quadratically with the number of rows [35]. Others, like the Random Forest algorithm, required less resources and less time.

4.4.4 Class Balance

The data we collected was very skewed. Especially when shorter intervals were chosen for labeling, e.g., thirty seconds, the classes were mostly unbalanced, with a clear dominance of the negative class. Thus, we adopted two strategies

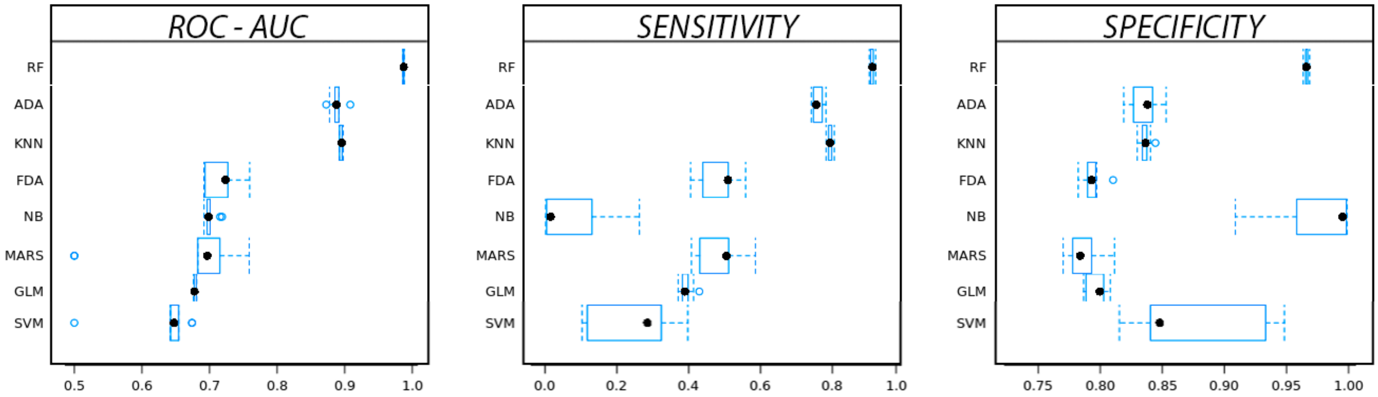


Fig. 5: Comparison of multiple ML models in our training scenario. We report the ROC-AUC curve, the specificity, and the sensitivity of the models we trained. It is possible to notice that the Random Forest model performs the best in terms of ROC-AUC which can be translated to a high accuracy. List of abbreviations: **RF**: Random Forest. **KNN**: K-Nearest Neighbors. **ADA**: Adaboost. **FDA**: Flexible Discriminant Analysis. **NB**: Naive Bayes. **MARS**: Multivariate Adaptive Regression Splines. **GLM**: Generalized Linear Model. **SVM**: Support Vector Machines.

to balance the classes. First, we tried to under-sample the negative class. Second, we tried the Synthetic Minority Over-sampling Technique (SMOTE) [36]. The SMOTE works by generating reasonable values for the minority class while it under-samples the other class until a balance is reached. In both cases, we aimed for a class balance around 60-40%. The initial results we obtained showed that the RF model trained with the data obtained with the two balancing strategies had similar performance to the one obtained using the whole dataset (i.e., similar accuracy, sensitivity, and specificity). Hence, for the final tests we kept the RF model trained with data obtained with the simple undersampling strategy, as it was easier to manipulate (less data means a smaller model) and had faster prediction times.

5 Multi Stage Model Evaluation

In this section, we elaborate on the strategies we used to evaluate our model.

5.1 Overview and Evaluation Strategies

After training, we ran some tests to evaluate if the model could detect a need for guidance in different scenarios. Also, we investigated if the need for guidance could be related to specific facial features, as hypothesised.

We adopted two standardized evaluation procedures [37] – see Section 5.2. In first instance, part of the labeled data was used to test the model (80-20 method). In a second trial, we applied a k-fold cross evaluation. Finally, we tested our methodology in a real-time analysis scenario – see Section 5.3. Results are shown in Table 1. For the first two tests, we report the sensitivity, the specificity and the F1 measure, which is the harmonic mean of specificity and sensitivity. F1 values above 50% are generally considered positive. For the real-time evaluation, we only discuss accuracy, since we lacked the ground truth to calculate the other metrics.

A Base Line for Evaluation. Before performing the evaluation, we determined a baseline accuracy. Often, this is referred to as the *no-information rate* or the probability of making a correct prediction with a simple guess. If the accuracy of the model is worse than guessing, it is clear that the trained model is not working well.

A naive choice would be setting this value to 50%, as we have two classes. However, since we are working with a skewed dataset this values can be raised to 56.4%, which represents the percentage of negative class instances in the dataset.

5.2 Evaluating Performance: 2 Scenarios

5.2.1 *Basic Evaluation*

Typically, models are evaluated with a so-called 80-20 strategy: around 80% of the labeled data is used for training while the remaining 20% is reserved for testing [37].

This is a typical (and easy) evaluation methodology. The drawback is that it does not tell us how the model performs with unseen data, since test and training data are taken from a same dataset. Although the results should be taken with a grain of salt, they were encouraging. The tests highlight high accuracy ($\approx$ 98%), as well as high sensibility and the specificity. We interpreted this result as a sign that our model can – most of the times – correctly identify a need for guidance.

5.2.2 *K-fold Cross Validation*

To get a more representative view of model performance, we performed a k-fold cross validation [38]. The idea of a k-fold evaluation strategy is simple: repeat multiple times a given test, each time with different settings and compare the results. Earlier, we showed how this idea can be applied to fine-tune *the model parameters*. This time, instead, we apply it for evaluation purposes. The evaluation consists of a few steps: Split the dataset into k groups and pick one for testing. Use the remaining data for training. Check the accuracy of the predictions and repeat selecting a different group as test data.

We created the groups manually, each containing the data of a different analyst. Then, we trained the model $k-1$ times and applied it to predict the values of the remaining

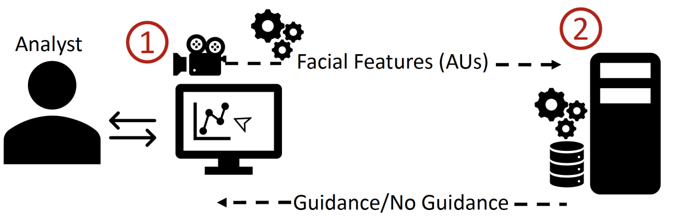


Fig. 6: A schematic representation of the procedure we set up to test the trained model in a real-time analysis scenario. At first, (1) each video frame is used to extract facial features of the analyst. Afterwards, the facial features are sent to a server (2) that instantly produces an answer – "Guidance" or "No Guidance" required – and sends it back to the client.

fold. We report the average accuracy and other performance metrics of these iterations in Table 1. The results show an accuracy of 72.6%, which is much lower than the results of the 80-20 evaluation. However, it gives us a much better estimation of how the model would work in practice, since the model is tested on completely unseen data – including individual facial features and expressions.

5.3 Real-Time Evaluation

In a last scenario, we assessed how the predictive model performed in a real-time analysis and if it could detect a need for guidance "on-the-fly" during the analysis. For this evaluation, we had to create a new pipeline in order to process the data in real time and decide on the fly if guidance was needed. We adopted this procedure to simulate what a functioning guidance system would have done during a normal VA session.

The scheme of the real-time evaluation setting is portrayed in Figure 6. The evaluation setup was very similar to the one used for training purposes: Using the same dataset we employed for collecting data, we asked *five* additional participants to perform the same open-ended task, using Tableau. Similarly to the training scenario, also these participants had previous experience with Tableau and data exploration.

In comparison to the training setting, though, we set up a new client/server architecture to process the data as it was created. In these real-time sessions the participants were filmed through a virtual conference tool while performing the task and the data was sent to a local server for real-time analysis. This different setup was imposed by the regulations due to the ongoing COVID-19 pandemic and the need to avoid close social contacts.

Evaluation	Accuracy	Sensitivity	Specificity	F1
No-Info Rate	*56.4%*	-	-	-
20% Test Data	*97.62%*	*98.62%*	*98.09%*	*98.4%*
K-fold Cross Validation	*72.6%*	*59%*	*55%*	*56%*
Real Time Analysis	*≈70%*	-	-	-

TABLE 1: Performance metrics of the trained ML model. The first row describes the case of guessing the labels (guidance/no guidance), i.e., the *No Information Rate*. We evaluated the model in multiple ways: We applied a 80-20 test and a k-fold cross validation strategy (the reported data is an average of k tests). Finally, we describe how the model performed in a real time analysis scenario. We report the accuracy, the sensitivity, specificity, and the F1 combined metric.

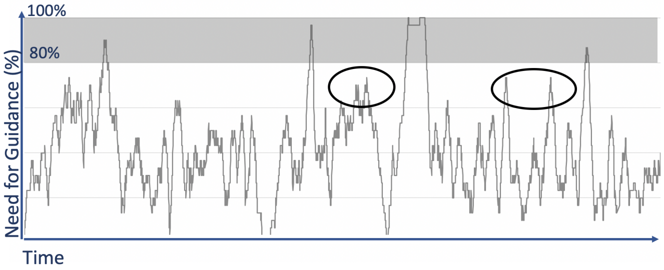


Fig. 7: How the need for guidance developed in a real-time evaluation. X-axis: the time of the analysis, ≈15 mins. Y-axis: the avg. need for guidance. When the detected need was higher than 80% we interrupted the analysis and asked the participants if they needed help. The peaks in the gray zone were correctly identified as the analyst needing guidance. The peaks highlighted with the circles were not detected, although the participant reported that some guidance could have been necessary at that moment.

When our prototype identified a need for guidance, we interrupted the participants and subsequently asked them two **Quick Questions**: 1) *How would you rate the grade of interruptibility, on a scale from 1 (not interruptible) to 5 (highly interruptible)?* And 2) *How would you rate your mental workload, on a scale from 1 (very low) to 5 (very high)?*. If the participant acknowledged the need for guidance, the prediction was considered correct; false otherwise. Afterwards, the participants were provided with the requested guidance, when the detection was correct, and could continue the analysis.

Architecture of the prototype. As mentioned, the analysis of facial features was done on the fly. To account for this change, we modified the facial analysis software, so that the detected AUs would be sent immediately for analysis. We set up a simple local server accepting GET requests. Once received, the requests were passed to a R script running the predictive model and decide if guidance was needed. Client and server were connected through the same physical network, to ensure a fast and reliable communication. However, due to some delays and to prediction times, in contrast to the offline setting, we were not able to decode the same amount of frames per second. The compromise for a stable facial analysis and feature detection was analysing 15 frames per second, which did not prejudice the fluidity of the study. The whole preprocessing of the data was performed on the fly by the client.

For each frame sent, the server replied with a textual answer guidance/no guidance. When a certain amount of positive answers were detected (see paragraph "Aggregation Strategy" for the details), an evaluator received a notification on the client side and asked the analyst if this was really the case, if they really needed guidance, and asked the aforementioned quick questions. At the end of the analysis, the participants were asked a further set of **After-study Questions:** We asked whether 1) *the participants felt the* *guidance was administered at the right time* and 2) *if they found it frustrating to be interrupted*. Furthermore, 3) *if they found invasive that the guidance was provided them with the voice*, or 4) *if they would have preferred a different type of notification e.g., popups, sounds, etc.*. Finally, 5) *if they were concerned for their privacy and if they would have used our method in their daily workflow.*

Aggregation Strategy. With RF models each prediction is independent from the others. Hence, it could happen that, according to how the analyst behaves, the prediction could vary multiple times in a short period of time. To account for this variability, which would cause confusion if we would have to notify the user for each change, we employed the following aggregation strategy.

Predictions received in a fixed temporal interval were grouped together and the final decision, whether guidance was needed or not, was based on the average number of "guidance is needed" labels, as in a majority-voting strategy. Each group was populated with 15 predictions, as we saw it did not result in sudden changes of predictions. In practice, we implemented this strategy using a sliding-window. Predictions were stored in a queue as they arrived to the client and when such queue was full, newly arrived predictions would cause the older ones to be removed. Positive and negative labels were mapped to 0 and 1 values and their average calculated. If the resulting value was greater than ≥ 0.8 the participant was notified..

Results. The result obtained thanks to this evaluation method are very encouraging and in line with the ones obtained with the k-fold cross evaluation. The test reported an average accuracy of 70% (see Table 1). The accuracy was calculated by matching how many times in a session the system correctly detected a need for guidance and the participant confirmed the detection. Beyond the accuracy, the evaluation shows how and why our methodology could effectively detect a need for guidance but also how, sometimes, failed. This mostly occurred due to how we set the notification threshold, and to a specific will of some participants in rejecting the guidance. We report on these specific cases, as well as the results of the questionnaires in the following paragraphs.

Accuracy of Predictions – A representation of how a typical real-time evaluation worked is shown in Figure 7. The figure shows on thee y-axis the average predicted need for guidance of a participant, calculated according to the aggregation strategy discussed in the previous paragraph. What we can see is that the need for guidance was typically detected correctly in the majority of the cases: we can see four peaks, in this specific case, falling in the gray detection zone. Thanks to the talk-aloud protocol, we related those peaks to moments in which the analyst was not sure how to pursue a goal, or was actively looking for some command in the interface.

However, as it can be seen, in a couple of other moments the analyst might have needed additional support from the system, but the need didn't reach the threshold, and thus, guidance was not provided. This is represented by the two circled peaks in Figure 7. A first moment, corresponding to the left-most circle, corresponds to the analyst feeling frustrated due to some mistakes s/he made when creating a visualization. In a second case (see the right-most circle) the analyst was trying to interpret the meaning of a visualization with no success. In both moments, the average aggregated need for guidance was ≈ 0.75, and hence lower than the notification threshold.

This happened with a similar pattern also in the other evaluation sessions and it shows a possible flaw of our aggregation strategy. We discuss how to solve this problem with a more sophisticated strategy in Section 6.

Willingness to reject guidance – In two occasions the participants rejected the provided guidance. By talking with them, we related the rejection to their specific will to try and proceed by themselves, rather than a flaw of our methodology. In other words, participants needed guidance, but were not yet ready to accept it. Reported answers that helped us clarifying the case were "*I need help, it is true, but not now. I would like to try by myself at first, ask me back in a minute*" and "*I really do not know what I should do now, but I am bit stubborn. I usually try many times before giving up*".

Looking at the precise numbers, when a need for guidance was correctly identified, the participants reported an average interruptibility value of 1.2 (out of 5), i.e. they stated to be highly interruptible. At the same time, though, they reported a high mental workload at the moment of the notification – average answer 3.5. This means the participants were indeed thinking hard how to overcome the encountered issue, but at the same time they reported to be open to receive help. A reported answer that clearly summarizes such situations is "*Yes, I was thinking how I could interpret this visualization, but still, I enjoyed receiving guidance*".

General Comments – The final set of questions provided us with additional means to evaluate the overall soundness of our methodology. The participants reported they felt the guidance was given them at the right moment, in most occasions – see Figure 7 for a discussion of possible missed detections. Two participants acknowledged that the correct timing could have been fine-tuned, but recognized that a need for guidance was indeed present whenever asked by the system. For what it concerns the way the guidance was provided, the participants reportedly appreciated that the notification was given them by voice, and noted that they were not frustrated with the notification. However, three of them also recognized that the actual instructions on how to proceed would have been better if provided visually, with labels and popups. Finally, we discussed if the participants had concerns about the privacy. They all noted they care about their privacy and how their data is treated. However, they also stated that the intents of the study and how their data would be utilized were clear from the beginning. According to them this was sufficient to ease their minds. Asking whether they would have used our methodology as part of their workflow, they answered that as far as the data is kept locally and the user is correctly informed about the purposes of video capture, they would have accepted to have their expressions analysed to receive in change system guidance. One participant additionally mentioned that s/he would have appreciated an open-source implementation of the software.

5.4 Need for Guidance and Facial Expressions

The results reported in the past sections shed light on the accuracy of our model in detecting a need for guidance. In addition to these results, we also checked that what we obtained supported also our initial thought that specific facial features can be associated to a need for guidance.

With this is aim in mind, we inspected what AUs had the largest impact in determining a correct prediction, determined if these results made sense and if they were in line with the literature on this topic. Table 2 shows the impact of the different AUs.

In summary, what we see is that the need for guidance can be correctly inferred from facial displays that previous research associated to difficulties with completing a task [6], [15]. Our research confirms these findings, extends them to a VA context and additionally, relates them to a need for guidance.

Looking at the specific facial expressions, we see that the presence and frequency of mouth dimples (AU14) had a great impact in determining a need for guidance. In previous research, AU14 has been related to learning gains [15], [39]. Other studies showed that AU14 might also indicate contemplative states, which could imply that the analyst is reasoning or deciding what to do next [6]. AU17 (Chin Raiser) and AU04 (Brow Lowerer) are also associated with a guidance request (see Table 2). Their simultaneous presence has been also correlated to thoughtful states [6].

Considering complex facial expressions, the contemporary presence of AU01 (Brow Raiser), AU02 (Outer brow raiser), AU04 (Brow lowerer), and AU25 (Lips Part/Jaw Drop), which are all present in our study, have been related to difficulties during the analysis, in previous literature [18]. AU07 (Lid Tightner), which also plays an important role in determining a need for guidance, has been instead connected to states of confusion [40], [41]. Finally, also the frequency of eye-blinks (AU45) was positively associated with guidance but it was never highlighted in previous research.

Action Units	Associated Facial Muscle	AU Importance
AU14	*Dimpler*	*100%*
AU17	*Chin Raiser*	*92%*
AU25-26	*Lips Part and Jaw Drop*	*87%*
AU10	*Upper Lip Raiser*	*85%*
AU04	*Brow Lowerer*	*75%*
AU01	*Inner Brow Raiser*	*68%*
AU07	*Lid Tightner*	*58%*
AU45	*Blink*	*50%*

TABLE 2: Table collecting common AUs, the associate facial muscle, and their relative importance in determining the need of guidance. The AU importance is related either to percentage of moments in which guidance was needed and at the same time the given AU was also present, as well as to the weight of the feature in the model.

6 Discussion

The results of our research show how the selected model performs in different contexts but it also provides indications how we could improve it.

Accepting Guidance. While the results we obtained are promising, additional steps are needed to tackle effectively the problem of providing timely guidance. As mentioned, although detecting a guidance need allows us to isolate a possible time window to initiate guidance, it does not necessarily mean that afterwards such guidance will be accepted by the user. As shown by the real-time evaluation, in a couple of occasions the participants in fact rejected the help, although highlighting that guidance was needed. In this regard, more research is certainly needed to shed light on mechanisms to cope with such scenarios, but also to understand if our methodology has a higher accuracy in respect to existing literature approaches.

Participants and Performance Analysis. The results obtained show variability in the accuracy depending on the applied evaluation strategy. As shown by the metrics of the basic 80-20 evaluation, the accuracy of predictions is very high ($\approx$ 98%). The accuracy, however, is much lower – 70-72% – in the k-fold cross validation and in the real-time analysis. This is due to how parts of the same dataset were used for the first evaluation. In this, the k-fold cross validation and the real-time evaluation offer a better view of the performance of our model since employ the data of unseen analysts. We want to point out that our work represents only a preliminary study. One of the most obvious limitations is the small number of participants involved - eight for training and five for testing. These results, which are also sustained by high specificity values, show that in order to make the methodology really effective in production scenarios, a wider set of user data should be collected i.e., involving more analysts to encompass for variable facial expressions, behaviors, and different nuances. Another consideration we make is that the processing times for the real-time scenario are not yet suitable to support a reliable application of our methodology in a production environment. While we tuned our implementation for fast and reliable communication, we still had to decrease the number of processed frames in order to obtain a smooth processing. In this regard, better hardware, parallelizing the computation, but also resorting to ensembles of smaller (but faster) models could all be added to make this approach fit for production contexts.

Apriori Assumptions. We hypothesized that facial expressions were indicators for the analyst's need of guidance. This lead us to not choose a precise set of tasks, but rather let the analysts perform an open-ended task. The results, especially the one associated with the real-time evaluation, seem to back up our initial assumptions. The predictive model we built succeeded in detecting guidance in multiple instances, no matter what the task was. For example, the system correctly detected the need of guidance when the analysts did not know how to use certain functions but also when they had problems interpreting the visualization. In this, we can see how the predictive model can account for the lack of both, operational and domain knowledge, which represent two different types of knowledge gaps [42].

False Negatives. In the real-time evaluation, we chose a simple strategy to detect when guidance was needed. We grouped and averaged the predictions the system made in a certain time interval and if the resulting value was higher than a predefined threshold, we provided the analyst with guidance. However, this strategy as we have seen, can fail: Two moments when guidance was needed were not correctly identified (see Figure 7). The strategy of using a static threshold is error-prone as the actual value can be subjective. Some people display emotions in a very subtle way, while others display them very obviously. To mitigate this problem, a tuning phase could be imagined to fine-tune the threshold before the analysis starts. More advanced solutions are also possible. For instance, instead of choosing a fixed threshold, it is possible to predict the need for guidance by looking at the steepness of the curve (see Figure 7) or for how long the average value remains in a range of values. These approaches have the potential to further improve our results and are good starting points for future research.

Privacy Issues. The way our tool detects a need for guidance may directly open to possible privacy intrusions. According to the answers we collected, privacy issues are indeed a problem that should be carefully considered. The questionnaires point directly to possible solutions: 1) have an informed consent 2) keep the data local and 3) keep isolated the elaboration of facial features. These seems to be sufficient preconditions to ease the mind of most of the participants, who also stated that under such conditions they would trade a small amount of their privacy for having guidance. These answers also highlight how our method could be improved: In our implementation we demand the elaboration of some features to a local server, which should be avoided. In this behavior, however, our approach resembles many other literature approaches which also exploit user data (e.g., interaction, heart-beat etc.) for detecting intents, as shown in the related work section.

Binary Predictions. In the current implementation, our model can predict a binary need of guidance (yes or no). In future developments, it could be interesting to consider also the intensity and strength of such a need. Our model is not able to understand the actual knowledge gap – “what does the user need?” – that leads to the need of guidance. Thus, we can identify when an analyst needs assistance, but we cannot say why, or how to solve it. In our environment, this was solved by external observers who could just ask the analyst about the problems they were facing. An automated solution would probably need to employ task-related strategies. Combining such strategies with our solution poses an important challenge for future research.

Emotions and Need for Guidance. The intensity of AU04 (Brow Lowerer) alone has been related to self-reported feelings of frustration and thoughtful states [6]. Moreover, AU04 is correlated with contemplative states and confusion when occurring during the first phases of the analysis, which may later evolve into frustration if the issues that caused the confusion remain unsolved. However, in our approach, the temporal sequence of events was not a discriminant factor for deciding whether guidance was needed, and therefore it is not possible to determine if AU04 should be interpreted as a contemplative state or a sign of unresolved analytical issues. To tackle this problem, the use of specific Neural Network for video analysis, considering the actual temporal sequence of the emotions, might help. In this, we see opportunities for further research.

7 Conclusion

In this paper, we investigated whether facial features can be used to detect a need for guidance in realistic scenarios. We explore the use of a ML model in our proof-of-concept implementation of an automatic system for guidance need detection. The results of the evaluation show its potential in interpreting the user’s expressions in offline and real-time scenarios. Finally, we propose a solution to one of many important challenges in the context of providing effective guidance during VA tasks. Our solution for automatically identifying moments when guidance is needed is an important step towards providing timely guidance in VA.

Acknowledgments

This work was funded by The Vienna Science and Technology Fund (WWTF) with the grant ICT19-047 (GuidedVA), by the Research Cluster ”Smart Communities and Technologies (Smart CT)” at TU Wien, by the Austrian Research Promotion Agency (FFG), grant #880883 (DoRIAH) and by the Austrian Science Fund (FWF), grant P31419-N31 (KnoVA).

References

[1] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, “Challenges in visual data analysis,” in *Proc. of IEEE Symposium on Information Visualization*. IEEE, 2006, pp. 9–16.

[2] M. S. Silver, “Decisional guidance for computer-based decision support,” *MIS quarterly*, pp. 105–122, 1991.

[3] E. Horvitz, “Principles of mixed-initiative user interfaces,” in *Proc. of SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999, pp. 159–166.

[4] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, “Characterizing guidance in visual analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 111–120, 2017.

[5] D. Ceneda, T. Gschwandtner, and S. Miksch, “A review of guidance approaches in visual data analysis: A multifocal perspective,” *Computer Graphics Forum*, vol. 38, no. 3, pp. 861–879, 2019.

[6] G. C. Littlewort, M. S. Bartlett, L. P. Salamanca, and J. Reilly, “Automated measurement of children’s facial expressions during problem solving tasks,” in *Face and Gesture 2011*. IEEE, 2011, pp. 30–35.

[7] D. Ceneda, T. Gschwandtner, S. Miksch, and C. Tominski, “Guided visual exploration of cyclical patterns in time-series,” 2018, visualization in Data Science (VDS at IEEE VIS 2018).

[8] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, M. Streit, and C. Tominski, “Guidance or no guidance? a decision tree can help,” in *Proc. of the Eurographics Workshop on Visual Analytics*, 2018.

[9] I. Fujishiro, Y. Takeshima, Y. Ichikawa, and K. Nakamura, “Gadget: Goal-oriented application design guidance for modular visualization environments,” in *Proc. of IEEE Symposium on Information Visualization*. IEEE, 1997, pp. 245–252.

[10] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva, “Viscomplete: Automating suggestions for visualization pipelines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1691–1698, 2008.

[11] D. Gotz and Z. Wen, “Behavior-driven visualization recommendation,” in *Proc. of Conference on Intelligent User Interfaces*. ACM, 2009, pp. 315–324.

[12] M. S. Bartlett, P. A. Viola, T. J. Sejnowski, B. A. Golomb, J. Larsen, J. C. Hager, and P. Ekman, “Classifying facial action,” in *Advances in Neural Information Processing systems*, 1996, pp. 823–829.

[13] P. E. Ekman and R. J. Davidson, *The nature of emotion: Fundamental questions.* Oxford University Press, 1994.

[14] M. Lewis, “The emergence of human emotions,” *Handbook of Emotions*, vol. 2, pp. 265–280, 2000.

[15] J. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. Lester, “Automatically recognizing facial expression: Predicting engagement and frustration,” in *Educational Data Mining*, 2013.

[16] D. Archambault and H. C. Purchase, "Mental map preservation helps user orientation in dynamic graphs," in *GD*. Springer, 2013, pp. 475–486.

[17] H. C. Purchase, E. Hoggan, and C. Görg, "How important is the "mental map"?–an empirical investigation of a dynamic graph layout algorithm," in *GD*. Springer, 2006, pp. 184–195.

[18] A. Kapoor, W. Burleson, and R. W. Picard, "Automatic prediction of frustration," *International Journal of Human-computer Studies*, vol. 65, no. 8, pp. 724–736, 2007.

[19] D. Ceneda, T. Gschwandtner, and S. Miksch, "You get by with a little help: The effects of variable guidance degrees on performance and mental state," *Visual Informatics*, vol. 3, no. 4, pp. 177–191, 2019.

[20] P. Ekman and W. V. Friesen, *Facial action coding system: Investigator's guide*. Consulting Psychologists Press, 1978.

[21] P. Ekman, W. V. Friesen, and J. C. Hager, "Facs investigator's guide," *A Human Face*, p. 96, 2002.

[22] P. Ekman, W. Friesen, and J. C. Hager, "Facial action coding system, a human face," *What is the ETC*, 2002.

[23] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 2018, pp. 59–66.

[24] M. Czerwinski, E. Horvitz, and S. Wilhite, "A diary study of task switching and interruptions," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, pp. 175–182.

[25] K. Tsubouchi and T. Okoshi, "People's interruptibility in-the-wild: Analysis of breakpoint detection model in a large-scale study," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. UbiComp '17. Association for Computing Machinery, 2017, p. 922–927. [Online]. Available: https://doi.org/10.1145/3123024.3124556

[26] M. Züger, S. C. Müller, A. N. Meyer, and T. Fritz, "Sensing interruptibility in the office: A field study on the use of biometric and computer interaction sensors," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–14.

[27] L. Qu, N. Wang, and W. L. Johnson, "Choosing when to interact with learners," in *Proceedings of the International conference on Intelligent User Interfaces*, 2004, pp. 307–309.

[28] O. Barral, S. Lallé, and C. Conati, "Understanding the effectiveness of adaptive guidance for narrative visualization: a gaze-based analysis," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 1–9.

[29] K. Cook, N. Cramer, D. Israel, M. Wolverton, J. Bruce, R. Burtner, and A. Endert, "Mixed-initiative visual analytics using task-driven recommendations," in *Proc. of IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 2015, pp. 9–16.

[30] M. Fan, K. Wu, J. Zhao, Y. Li, W. Wei, and K. N. Truong, "Vista: Integrating machine intelligence with visualization to support the investigation of think-aloud sessions," *TVCG*, vol. 26, no. 1, pp. 343–352, 2019.

[31] M. Czerwinski, E. Cutrell, and E. Horvitz, "Instant messaging: Effects of relevance and timing," in *People and computers XIV: Proceedings of HCI*, vol. 2, 2000, pp. 71–76.

[32] L. Battle and J. Heer, "Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau," in *Computer Graphics Forum*, vol. 38, no. 3. Wiley Online Library, 2019, pp. 145–159.

[33] C. North, "Toward measuring visualization insight," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 6–9, 2006.

[34] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[35] C. K. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing systems*, 2001, pp. 682–688.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence research*, vol. 16, pp. 321–357, 2002.

[37] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, 2018.

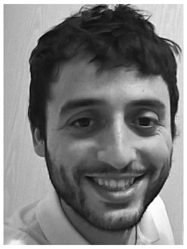
[38] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[39] S. K. D'Mello, S. D. Craig, and A. C. Graesser, "Multimethod assessment of affective experience and expression during deep learning," *International Journal of Learning Technology*, vol. 4, no. 3-4, pp. 165–187, 2009.

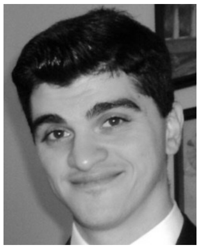
[40] S. D. Sims and C. Conati, "A neural architecture for detecting user confusion in eye-tracking data," in *Proceedings of the 2020 International Conference on Multimodal Interaction*, 2020, pp. 15–23.

[41] S. Lallé, C. Conati, and G. Carenini, "Predicting confusion in information visualization from eye tracking and interaction data." in *IJCAI*, 2016, pp. 2529–2535.

[42] M. Chen, D. Ebert, H. Hagen, R. S. Laramee, R. Van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver, "Data, information, and knowledge in visualization," *IEEE computer graphics and applications*, vol. 29, no. 1, pp. 12–19, 2008.



Davide Ceneda is a Post-Doc at the CVAST group at the Institute of Visual Computing and Human-Centered Technology, TU Wien. He earned his PhD in 2020 at TU Wien defending a thesis called "Guidance-Enriched Visual Analytics". His research interests include visualization, human perception, visual analytics and guidance.



Alessio Arleo is a Post-Doc at CVAST at the Institute of Visual Computing and Human-Centered Technology, TU Wien. He earned is PhD in 2018 at the University of Perugia (Italy) defending a thesis called "Distributed Large Graph Visualization: Algorithms and Experiments". He holds a BSc degree in Computer and Electronic engineering and a MSc in Computer and TLC engineering. His research interests include graph drawing and visualization, distributed algorithms, software engineering.

Theresia Gschwandtner is a scientific researcher at CVAST at the Institute of Visual Computing and Human-Centered Technology, TU Wien. Her research interests are visual analytics & information visualization, with a specific focus on data quality, uncertainty, time-oriented data and guidance in VA.

Silvia Miksch is University Professor and head of the Research Division "Visual Analytics" (CVAST), Institute of Visual Computing and Human-Centered Technology, TU Wien. She served as paper co-chair of several conferences including IEEE VAST 2010, 2011 and 2020 and VIS Overall Papers Chair (IEEE VIS 2021) as well as Euro- Vis 2012 and on the editorial board of several journals including IEEE TVCG and CGF. She acts in various strategic committees, such as the VAST steering committee and the VIS Executive Committee. In 2020 she was inducted into The IEEE Visualization Academy. Her main research interests are Visualization/Visual Analytics (particularly Focus+Context and Interaction) and Time.