MULTIVARIATE BEHAVIORAL RESEARCH
2025, VOL. 60, NO. 3, 641–655

OPEN ACCESS

A Tutorial on the Use of Artificial Intelligence Tools for Facial Emotion Recognition in R

Austin Wyman and Zhiyong Zhang

Department of Psychology, University of Notre Dame, Notre Dame, IN, USA

ABSTRACT

Automated detection of facial emotions has been an interesting topic for multiple decades in social and behavioral research but is only possible very recently. In this tutorial, we review three popular artificial intelligence based emotion detection programs that are accessible to R programmers: Google Cloud Vision, Amazon Rekognition, and Py-Feat. We present their advantages, disadvantages, and provide sample code so that researchers can immediately begin designing, collecting, and analyzing emotion data. Furthermore, we provide an introductory level explanation of the machine learning, deep learning, and computer vision algorithms that underlie most emotion detection programs in order to improve literacy of explainable artificial intelligence in the social and behavioral science literature.

KEYWORDS

Emotion recognition; emotion detection; artificial intelligence; Google Cloud Vision; Amazon Rekognition; Py-Feat

The ubiquity of artificial intelligence (AI) in science, business, and media has led to the development of competing AI-powered tools, many of which are available for research, commercial, and personal use. One tool that has received recent attention is emotion recognition AI. Emotion recognition AI goes by many names (e.g., emotion detection AI, facial expression analysis, affective computing), but besides semantic differences, these terms equivocally refer to programs that integrate machine-learning and deep-learning algorithms to detect and label human emotions. Emotion recognition tools rely on pre-trained models, which have been taught across a large number of test subject images to understand differences in facial expressions that at times can be too subtle to be targeted by humans. Through such tools, users can input a human face image and receive immediate feedback on the emotions being presented.

Most research regarding emotion recognition AI has been conducted in the field of computer science and has seldom been applied in the social and behavioral sciences. However, emotion recognition software provides numerous opportunities for applied researchers including psychologists to explore important questions concerning emotions and related functioning, such as in emotion dysregulation, emotional intelligence, and communication (Wyman & Zhang, 2023). The lack of applications is partially due to a lack of awareness of methods developed in other fields. It is also due to controversy surrounding black box algorithms (Holm, 2019), in which the resulting model has a clear input and output but the internal calculations that allow us to arrive at the output are often not interpretable. Machine-learning and deep-learning algorithms are powerful, but they generally prioritize prediction power and ignore issues such as measurement error and bias that pervade all statistical models (Jacobucci & Grimm, 2020). The nature of such algorithms can easily lead to misuse and misinterpretation of AI tools. Therefore, Taylor and Taylor (2021) recommend that every psychologist be trained on machine learning principles in the same way that neuroscience training is required of every psychologist. In fact, Hitron et al. (2019) demonstrated that even children can understand machine learning concepts if provided with the proper building blocks. Thus, the misconception and misuse of AI tools among researchers may be avoided with a more common understanding of artificial intelligence.

Although the full breadth of opportunities for emotion recognition AI in the social and behavioral sciences has seldom been explored, there are clear advantages for researchers to use emotion recognition AI as a multimodal tool for examining emotions. Self-report measures of emotion are often unreliable due to psychometric limitations, such as state-dependence, recall bias, cognitive bias, acquiescence bias, and social desirability (Chan, 2010; Larsen & Fredrickson, 1999). Emotion recognition AI bypasses all of these limitations by providing an objective observer rating for emotions. Wyman and Zhang (2023) describes how AI has been used to develop experiment companions tools that observe emotions and report instantaneous feedback to participants, which is particularly valuable in behavioral interventions for young children. Furthermore, the estimates that AI provides are much faster than any human rater, neither self-report or observer, is able to provide, allowing for the ability to study emotions at extremely small sampling intervals (i.e., within milliseconds). This is particularly advantageous for ecological momentary assessment studies and any experiments that take place within a naturalistic setting. Even outside of large within-person study designs, emotion recognition AI is advantageous for working with large sample sizes of images, such as data from social media. As a multimodal tool, emotion recognition AI can be implemented alongside tools, such as EEG and eye-tracking (Lim et al., 2020; Zheng et al., 2014); however, these tools require implementation during study design, whereas emotion recognition AI can be implemented after data has already been collected. Emotion recognition AI provides many opportunities for both current and novel study designs in the social and behavioral sciences, and can even be used to retroactively analyze previous data sets.

However, only a good understanding of the basic idea of such techniques can open up the black box algorithm (Hitron et al., 2019) and demystify the challenges and opportunities of using AI to study emotions. Furthermore, researchers need information and guidelines on the available emotion recognition tools. Therefore, the aim of the present paper is to make emotion recognition software more accessible to researchers in the social and behavioral sciences, so individuals can make informed decisions about which program is right for their interests, and are able to immediately begin designing, collecting, and analyzing data.

In the rest of the paper, we will first provide a high-level explanation of action units, face detection, and neural networks, which are necessary for understanding the procedure behind automated emotion recognition. Then, we will provide an overview of current emotion recognition AI programs, illustrate sample R code (for both beginner and experienced programmers), and compare each program's advantages and disadvantages.

Automated emotion recognition and its evolution

Emotion recognition AI integrates decades of research in psychology, physiology, and computer science. Ekman and Friesen (1971) originally proposed the existence of six basic human emotions (happiness, sadness, anger, surprise, disgust, and fear), which are believed to be universally experienced across all human cultures. The theory of basic human emotions has been substantiated according to developmental (Izard, 2007), behavioral (Siedlecka & Denson, 2019), neurobiological (Celeghin et al., 2017; Stephens et al., 2010), and physiological evidence (Ekman, 1992; Keltner et al., 2000), which has been particularly crucial in the development of computer vision and detection algorithms.

Ekman and Friesen (1976) further developed the first physiological model of basic emotions, the Facial Action Coding System (FACS), which continues to serve as the foundation for many modern emotion recognition programs. The FACS was built by observing which muscle groups are involved in making facial expressions, and it reduced facial expressions to their simplest muscular actions, and classified each muscle action as a single action unit (AU). Ekman and Friesen (1976) proposed 42 AUs in total, although only about 20 are used in practice today (Martinez et al., 2019). AUs are often used as reference points to help classify human emotions. For example, when a person is making a happy facial expression, the AUs *Cheek Raiser* (6, referring the sixth AU) and *Lip Corner Puller* (12) would be active. Happiness is a simple example because it only involves two components, which probably explains why most emotion recognition AI programs perform disproportionately well for detecting happiness compared to other emotions that involve more AUs. In contrast, a fearful expression is the combination of seven AUs: *Inner Brow Raiser* (1), *Outer Brow Raiser* (2), *Brow Lowerer* (4), *Upper Lid Raiser* (5), *Lid Tightener* (7), *Lip Stretcher* (20), and *Jaw Drop* (26). Naturally, classifying complex emotions, such as fear, would be a difficult task for humans, especially when noise and interference from other muscle units are introduced. Before the 1990s, there was limited research involving the FACS because of the immense labor involved in training raters and processing video footage (Martinez et al., 2019). It was estimated that the FACS required 100 h of training to achieve minimal competency and typical raters would take an hour to evaluate a minute of video footage (Donato et al., 1999).

Limits in human processing led to the automation of emotion recognition research. Equipped with facial landmark models like the FACS, computer scientists applied recent advancements in machine learning and computer pattern recognition to develop the first iterations of automated emotion recognition AI (Donato et al., 1999; Lien et al., 1998; Pantic et al., 1998). The structure of early facial expression analysis relied on a sequence of tracking, dimension reduction, and classification algorithms. For example, Lien et al. (1998) manually identified the locations of facial features within an image, mapped the image onto a standardized 2D face template, and used derivative-based algorithms (e.g., dense optical flow tracking) to maintain stable action unit locations regardless of motion, as an image may shift from frame to frame. Principal component analysis was then used to examine geometric correspondence between the pixels, select the most prominent eigenvectors, and ultimately detect locations of change in the image. After transforming the displacements into vectors, data was passed through a Hidden Markov Model (HMM), a probability-based model trained with supervised learning, to classify vectors as action units. The HMM identified the most likely action units presented in the photo, which could then be interpreted according to the FACS model. These early automated emotion recognition models were innovative, yet they were not considered fully-automatic because they relied on human coders to manually label the locations of facial features prior to feed them to feature tracking algorithms.

In order to become fully-automated, facial expression models needed to train more sophisticated face detection algorithms (i.e., computer vision) to recognize facial landmarks without manual specification. Deep learning, based on artificial neural networks (ANNs), helped researchers clear the hurdle of automation and the method continues to be a staple of modern emotion recognition algorithms. ANNs are roughly based on human neural networks, in which we assume there are layers of neurons and that different combinations of neurons being fired in one layer influence which neurons are fired in the next layer. ANN layers represent different attempts to scan for particular values in digital image data. The algorithm will scan for a particulate attribute and based on its presence, absence, or specific location, the algorithm will decide where to search next. ANNs repeat this process for a set number of layers (i.e., unsupervised learning), or until it can find the correct answer (i.e., supervised learning). ANNs are often viewed as black box algorithms because interpretation is a significant challenge, even more so than traditional machine learning methods. However, follow-up procedures such as back-propagation (Dayhoff & DeLeo, 2001) and image generation (Castelvecchi, 2016) can elucidate what is happening within the hidden layers, which often results in more efficient and more interpretable products. There has been further development in artificial neural network infrastructure over the last 20 years, particularly with the development of convolutional neural networks (CNNs) for image analysis; however, this is beyond the scope of the present tutorial. See Jain et al. (2021) for a more comprehensive discussion of neural networks in emotion recognition AI. With a stronger background in the theory of emotion recognition AI, one can more comfortably approach competing algorithms in the literature, make informed decisions about their empirical value, and more accurately interpret analyses.

Review and use of existing software programs

A wide range of emotion recognition AI software programs are available, each with their respective advantages and disadvantages. Some notable programs include Affectiva (Kulke et al., 2020), Amazon Rekognition (Amazon Web Services, 2016), Baidu AI Cloud (Baidu, 2019), FaceReader (Noldus, 2014), Face++ (Megvii, 2017), Google Cloud Vision (Google Cloud Platform, 2015), iMotions AFFDEX and FACET (Stöckli et al., 2018), nViso (nViso, 2016), OpenCV (Puri et al., 2020), OpenFace (Amos et al., 2016), Py-Feat (Cheong et al., 2023), and Viso AI (Viso AI, 2021). These programs also have a range of accessibility, as different consumer groups exist, each with unique data processing and analysis needs. That said, the present tutorial will only evaluate the programs we determined are most accessible to R users based on the following criteria:

1. Affordability (we prioritized programs with low per-image cost estimates);
2. Availability of technical assistance (we prioritized programs that are actively maintained and have a history of responding to user-identified issues);
3. Availability within the existing R infrastructure (we prioritized programs that can be accessed

in R environments currently without the requirement to create new packages or API wrappers); and
4. Lack of redundancy (we prioritized programs that have unique advantages relative to other programs already included in the review).

For example, iMotions is a popular model for psychologists and offers numerous advantages, including synchronization with biophysical sensors (e.g., EEG, fMRI) and experimental stimuli (Kulke et al., 2020; Stöckli et al., 2018). However, it was not included in the review because of high subscription cost, lack of compatibility with image-level data, and lack of compatibility with R (although iMotions can output results in R Markdown, iMotions is a point-and-click software that does not allow users to interact with or customize its model on the front-end of analysis). See Deshmukh and Jagtap (2017) and Pinto et al. (2023) for a more comprehensive summary of emotion recognition AI programs, their respective advantages, and sample emotion datasets to personally evaluate their efficacy.

Based on the above criteria, we selected Google Cloud Vision AI, Amazon Rekognition, and Py-Feat as the primary subjects of our evaluation. In the following, we will first discuss key features of these programs, then illustrate their usage with sample code, and finally evaluate the utility of each program for social and behavioral researchers. All code is will be available on the companion GitHub page (https://github.com/awymanquant/emotion-recognition-tutorial-R) for ease of implementation.

Google Cloud Vision

Key features

Google Cloud is a cloud computing platform that hosts dozens of pre-trained AI tools, which users can access either directly through a web browser or through their home programming environments. The platform receives input data, processes them on Google's cloud servers, which stores the AI models, and then returns the output to the user. Emotion recognition is available through the "Face Detection" feature of Google Cloud Vision. Google Cloud Vision is able to detect four emotions: joy, sorrow, anger, and surprise. The likelihood of each emotion is rated on a Likert scale from 0 to 5, with 0 representing *unknown*, 1 representing *very unlikely*, 2 representing *unlikely*, 3 representing *possible*, 4 representing *likely*, and 5 representing *very likely*. Two additional estimates of *Detection Confidence* and *Landmarking Confidence* are also available, both of which are rated on a continuous scale from 0 to 100, with 100 being the maximum estimate of the AI's detection confidence. It also includes estimates for different types of image interference, such as underexposure, which may be used to diagnose reliability limitations for a given batch of images. However, the interference estimates Google provides are rated on the same 0–5 ordinal scale, not rated on the same 0–100 continuous scale as the confidence estimates.

Although Google Cloud only has built-in API support for Python, Java, Go, C++, and Ruby, the community has developed packages that allow users to initialize Google Cloud Vision within R environments, such as the "googleCloudVisionR" package (Koncz et al., 2020). Note that the package depends on the "googleAuthR" package (Edmondson, 2023).

How to use

The R package `googleCloudVisionR` provides a way to interact with Google Cloud Vision API in R. To use it, we need to first install the package. Google Cloud service requires authentication which can be done using the R package `googleAuthR` (Edmondson, 2023). The R code below shows how to install the packages, how to authenticate the Google account, and how to obtain the Cloud Vision permissions (scopes).

More specifically, Line 1 installs the two packages `googleCloudVisionR` and `googleAuthR`, which only needs to be done once. Lines 2 and 3 load the packages into the existing R environment, which needs to be done once per new R session. To use the Google Vision service, one needs to have a Google Cloud account. Within the account, one can create an OAuth client ID. After creating it, one can then download the OAuth client as a JSON file to own computer. To authenticate R to use the service, the code on Lines 9 and 10 are used. Particularly, the function `gar_set_client` sets up the needed information including the OAuth client (`json`) and the scopes (`scopes`) of the analysis. The `json` argument should provide the location and name of the OAuth client JSON file, shown in Line 5. The `scopes` argument tells which Google Cloud service to use, as specified in Lines 6 and 7. The function `gar_auth` will open Google authentication window for a user to approve the authentication. Note that the "email" should be provided by the user.Screenshot-assisted instructions for how to access the OAuth client and scopes can be found on the companion GitHub page.

```
install.packages(c("googleAuthR","googleCloudVisionR"))
library(googleAuthR)
library(googleCloudVisionR)

json="google-client.json"
scopes=c("https://www.googleapis.com/auth/cloud-vision",
        "https://www.googleapis.com/auth/cloud-platform")

gar_set_client(json=json,scopes=scopes)
 gar_auth(email="email_associated_with_Google_Cloud_
 account")
```

Once the connection between R and Google Cloud service is build, one can start to send data and get the emotion recognized. If your data is already in an image format (e.g., jpeg or png), you may skip the follow part on how to convert videos to images and proceed to submitting the data to the Google Cloud. Here, we first provide instructions on converting a video file to individual static images because videos are commonly recorded in emotion research. The R package `av` can be used to process many different types of videos including the commonly used `mp4` and `avi` formats.

The code below provides an example. After we install and load the package `av`, the function `av_video_images` can be used to convert a video to images. In the function, the argument `video` specifies the video file on the computer. The complete path to the file can be used if the file is not in the R working directory. The argument `destdir` specifies where to save the extracted image. Here, we use "myimagefolder" to create a folder under the folder where the video file is located. The format of the extracted image is specified by the argument `format`. The package supports either `jpeg` or `png` format, defaulting to `jpeg`. The `png` format can be used if higher quality of images are necessary. Additionally, the function allows you to specify the frames per second (`fps`), or how many image to extract per second. The default is `NULL` to get all the images in the video. As there tends to be little change in emotion estimates within a short period of time, e.g., one second, it is often helpful to set `fps` to a small value to increase the efficiency of the analysis. Here, we set `fps=2` to extract 2 images for every second of the video. One can also set a fraction for this parameter. For example, `fps=0.2` will extract 1 image every 5 s. We are extracting frames from a sample image in the RAVDESS data set (Livingstone & Russo, 2018). The specific video used (`02-01-03-02-02-01-01.mp4`) is available on the companion GitHub for readers to follow along.

```
# convert mp4 to jpeg
install.packages("av")# install the package av
library(av)# load the package av

av_video_images(video="02-01-03-02-02-01-01.mp4",
                destdir="myimagefolder",
                format="jpeg",
                fps = 2)
```

Once the code is run, it will create a group of images with the names `image_000001`, `image_000002`, `image_000003`, and so on in the folder `myimagefolder`. For the sample analysis, we select the second frame (`image_000002`) as it has the best visualization of the target emotion, joy (Figure 1).

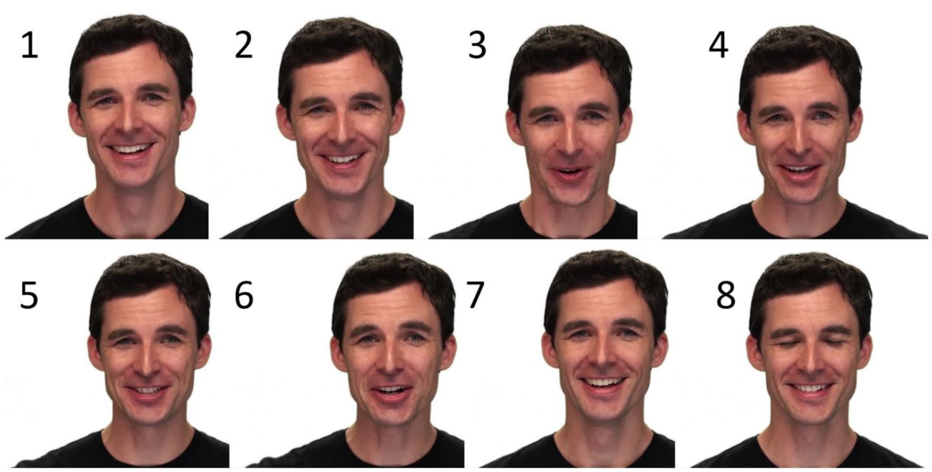

Figure 1. Frame-level output of "av_video_images" function for sample RAVDESS video.

To actually conduct emotion recognition using Google Cloud Vision API, we can use the function `gcv_get_image_annotations()`. We call the annotation function `gcv_get_image_annotations()` for emotion recognition and stores the output as an object `API.call`. We isolate the columns of interest, (1, 4 to 9), from the output. This code can also be repurposed in a for loop to process multiple images sequentially. An example batch of 1,1415 images took 12.39 min to complete, or an average of 0.0088 min (0.5 s) per image.

In the function `gcv_get_image_annotations()`, we can specify the image to use and the feature of Google Cloud Vision that we are interested in. For emotion recognition, we set the argument `feature` to `FACE_DETECTION`. The output of the function includes more information than emotion data. Here, we only keep the image_path, detection_confidence, landmarking_confidence, the four image likelihoods. Interference estimates were excluded for the reasons explained above, but may be reintroduced by including the columns 10 through 12.

```
# send image to GCP and retrieve face detection estimates
API.call< -gcv_get_image_annotations
  (imagePaths="myimagefolder/image_000002.jpeg  ",feature
  ="FACE_DETECTION")
do.call('rbind',API.call[,c(1,4:9)])
```

The code above results in the following output. Joy has the highest likelihood, which matches the labeled emotion of the image from the RAVDESS data set.

```
image_path                "image_000002.jpeg"
detection_confidence      "0.98828125"
landmarking_confidence    "0.6696827"
joy_likelihood            "VERY_LIKELY"
sorrow_likelihood         "VERY_UNLIKELY"
anger_likelihood          "VERY_UNLIKELY"
surprise_likelihood       "VERY_UNLIKELY"
```

Discussion

Because Google Cloud Platform is primarily intended for commercial use, it only offers limited free usage, which is enough for trying out the service but paid service is needed for large research. There is a per-unit (individual image) cost associated with emotion recognition. As of January 2024, Google charges $1.50 per 1,000 units for 1,001 to 5,000,000 units and $0.60 per 1,000 units for 5,000,001 and higher. Additionally, the first 1,000 units are free each month. Free trials may also be available to offset the cost of usage. The subscription service provides access not only to Google's substantial library of pre-trained models, but also Google's cloud servers, which speed up analyses for large batches of image data. Although Google's pricing is a significantly cheaper option than most APIs in the market, requiring a paid subscription creates an additional challenge for budgeting experiments, which may be prohibitive for some users.

Compared to some other emotion models, Google Cloud Vision also has limited utility. Longstanding psychological research has proposed and confirmed the existence of six basic emotions (Celeghin et al., 2017; Ekman, 1992; Ekman & Friesen, 1971; Izard, 2007; Siedlecka & Denson, 2019), yet Google only provides information for four of these, neglecting fear and disgust. Although not an emotion per-say, it also lacks a measure of neutrality to communicate uncertainty in the algorithm or that the emotion being presented in the image is not one of the four available categories. Each emotion being rated on a 0–5 Likert scale also limits its ability to detect emotion variability, often resulting in emotion estimates plateauing for long periods of time. It is advantageous that the overall likelihood (or Detection Confidence) is rated on a continuous scale, but extending this scale to the individual emotion estimates would grant much greater ability to examine individual differences in emotion functioning.

The accuracy performance of Google's emotion recognition AI has been well-documented in the computer science literature; however, given the frequency of updates for cloud-based AI, results even within the last year may be inaccurate in describing how the tool performs currently. Nevertheless, there are certain trends in Google's performance that appear to be stable features across iterations. Google Cloud Vision detects certain emotions better than others. Its true positive rate (TPR) for joy is consistently high, ranging from 99.47% (Bryant & Howard, 2019) to 100% (Khanal et al., 2018, 2023). Khanal et al. (2023) demonstrated, however, that anger (TPR = 0.26), surprise (TPR = 0.26), and sorrow (TPR = 0.10) perform significantly worse. Google Vision carries a high risk of missing anger and sorrow measurement, which, as negative affect states, may be of particular interest to emotion researchers. However, for questions related to positive affect states (e.g., joy), its near perfect true positive rate (1.00), positive prediction value (0.98), and negative prediction value (1.00) make Google a sufficient tool for analyses (Khanal et al., 2023).

It is important to note that these accuracy estimates correspond to full-forward-facing face detection, such as when an individual is staring straight at the camera. Accuracy from half-side and full-side angles is much lower for the API, yet this is a common limitation for all face detection algorithms. In fact, Google Cloud Vision's ability to handle rotated-pose images is a particular strength of the algorithm. Whereas most algorithms achieve greatly reduced accuracy from half-side and 0% accuracy for full-side angles, Google Cloud Vision is able to maintain similar accuracy at half-side profiles and non-zero accuracy at full-side profiles (Khanal et al., 2018). In experiment designs where participants may move frequently or are unable to focus on the camera for the entire study, Google Cloud Vision may be particularly advantageous because it has a higher probability of detecting non-forward profiles and potentially reducing missing data.

In summary, Google Cloud Vision is a simple and efficient algorithm to detect emotions in images that are under less-than-ideal circumstances. This advantage, however, is paired with substantial measurement error and a lack of certain emotional information, which may limit the range of questions that can be answered. Taking advantage of its free trials and low starting prices, Google's API is a friendly program to start learning and practicing emotion recognition AI before progressing to more valid and reliable tools.

Amazon Rekognition

Key features

Amazon Web Services (AWS) is another cloud computing platform where users can access pre-trained AI models for data processing and analysis. In particular, Amazon Rekognition offers a wide range of detection, labeling, and identification features for image and video formatted data, but this review will only discuss its image processing feature.

After processing an image, users receive a list of descriptive statistics from the output. Although the initial output is complex and lengthy, it is easy to isolate the information that we are interested in analyzing. For example, we can get X and Y coordinates that specify the location of key facial features, such as the jawline, mouth, or nose. Position data can also be used to explore specific locations of the eyes (each eye is reduced to four quadrants), which may have implications for eye-tracking research. In fact, eye-tracking data are also a part of the output, which provides yaw and pitch estimates corresponding to each eye. It also includes information on whether certain traits are present (e.g., facial hair, sunglasses, smiles) or demographics (e.g., gender and age range).

Related to emotion recognition, Amazon Rekognition provides information for eight types of emotions: calm, sad, confused, happy, surprised, disgusted, fear, and angry. Rekognition also provides the confidence of detecting an emotion, rated on a continuous scale of 0–100. It is important to note that the confidence values do not necessarily indicate the intensity of emotion that is present in the image; it simply describes how confident the algorithm is that a given emotion is present in the image. It is important to note that the algorithm organizes the detected emotion in the order of decreasing confidence estimates, meaning special attention should be paid to the column values when combining results from multiple processed images.

Similar to Google, AWS provides API for Python, Java, Go, and C++, as well as toolkits for Kotlin, .NET, PHP, Rust, Azure DevOps, JetBrains, PowerShell, Visual Studio, and its own program, AWS CDK. Nonetheless, Amazon Rekognition is accessible in R environments *via* the packages "paws" (Kretch & Banker, 2023a) and "paws.machine.learning" (Kretch & Banker, 2023b).

How to use

To use Amazon Rekognition in R, we first need to install and load the packages `paws` and `paws.machine.learning`. As with the Google API, we need account credentials to access the service. After creating an account with Amazon, one can get an access key id and associated secret access key (password). The information can be provided to R as environment variables. The sample R code is given below. Sample screenshots for how to access the corresponding IDs and access keys for Amazon may be found on the companion GitHub page.

```
library(paws)
library(paws.machine.learning)
Sys.setenv(
  AWS_ACCESS_KEY_ID="your_access_key_id",
  AWS_SECRET_ACCESS_KEY="your_secret_
  access_key",
  AWS_REGION="your_aws_region"
)
```

Next, we can send an image to Amazon AWS for emotion recognition. The code below does the job where only the image file name needs to be changed for a different image. Note that we save the results in the `API.call` object, which includes information as described earlier. To extract the emotion information, the last line of the R code can be used. To analyze a set of images, a loop approach may also be used here. Our example batch of 1,415 images took 7.58 min to complete, which is an average of 0.005 min (0.3 s) per image.

```
## initialize the connection
client< -paws::rekognition()

## the file name to use
imagefile< -"myimagefolder/image_000002.jpg"

## read the image data into R for use
image< -list(Bytes=readBin(imagefile,"raw",file.size(imagefile)))

## Request the analysis
API.call< -client$detect_faces(Image=image,"EMOTIONS")

## Organize the results
do.call('rbind',API.call$FaceDetails[[1]]$Emotions)
```

The code above provides the output below for our sample image. `"HAPPY"` has the highest confidence value, which matches the label of the image in the data set. Note that Amazon always presents its estimates in descending confidence order, so when processing multiple images in a loop, the order of estimates may vary from image to image.

	Type	Confidence
[1,]	"HAPPY"	100
[2,]	"CONFUSED"	0.001733502
[3,]	"SURPRISED"	0.001400709
[4,]	"CALM"	0.000679493
[5,]	"SAD"	0.0001966953
[6,]	"DISGUSTED"	0.0001490116
[7,]	"FEAR"	4.470348e-05
[8,]	"ANGRY"	3.576279e-05

Discussion

Similar to Google Cloud Vision, Amazon AWS Rekognition is primarily a commercial resource and is only accessible by paid subscription, although free trials are available. As of January 2024, Amazon charges $0.001 per image for the first million images, with cheaper rates available for 4 million, 30 million, and over 35 million images, respectively. Their free trial covers the first 5,000 images each month for a 12 month period. Additional costs are incurred if users wish to analyze video data or store metadata from image and video analyses on the AWS cloud, which can be beneficial given the exhaustive attributes produced by their detection models. Amazon Rekognition is more affordable than most commercial AI products, but financial accessibility should be assessed on an individual budget basis.

The eight emotions that Amazon Rekognition provides are a significant improvement over Google's four. Rekognition accounts for the six basic emotions (sadness, happiness, fear, disgust, surprise, and anger) and goes beyond it by including confused and calm as well. The inclusion of measurement for confusion introduces a diversity of new research questions that can be explored using emotion recognition AI, such as studying the student attention, comprehension, and engagement in educational settings (Anbusegaran, 2021; Ramakrishnan et al., 2019; Tabassum et al., 2020). The availability of calm estimates, as an indicator of neutrality, also greatly improves upon the validity issues in Google's model. Furthermore, Rekognition's continuous scale of 0–100 better approximates the variability of a given data set, which is important for examining intra- and inter-individual differences.

Rekognition has also significantly improved in performance over the past five years due to frequent maintenance and updates by AWS. For example, Al-Omair and Huang (2018) reported that the average accuracy across all Rekognition emotion estimates was 64%, whereas Rafael et al. (2020) reported an average accuracy of 76%. Other studies have reported an average true positive rate of 50.7% in early years (Bryant & Howard, 2019) and 86.8% in later years (Yang et al., 2021), and an average positive predictive value of 55.3% (Bryant & Howard, 2019) compared to 85.7% later (Kim et al., 2021). Kim et al. (2021) also highlights Rekognition's efforts to reduce sources of algorithmic bias, particularly age and gender bias, over time. Arguably, the greatest argument for Rekognition's improvement in recent years is its inclusion of new detection functions, most notably confusion and fear. Early studies involving Rekognition were not able to measure fear, but recent estimates show that it is one of the most accurate fear detection models available (TPR = 68.3%), outperforming Face++ and Affectiva (Yang et al., 2021).

As with Google, not all emotions have comparable accuracy estimates. Naturally, happiness is the most accurate measure with a true positive rate of 100% (Yang et al., 2021), comparable to Google's estimate. The other emotions, however, are massive improvements over Google's model, anger (TPR = 0.77), surprise (TPR = 0.89), and sadness (TPR = 0.90). Similar trends are observed for positive predictive value as well (Kim et al., 2021). Limited research has investigated how Amazon Rekognition functions under rotated-pose conditions, with most literature reporting accuracy estimates from a full-frontal angle. However, as mentioned in the sample code section, image interference, such as when a participant is poorly angled for the camera, interrupts the emotion detection program and forces it to provide null values for all its parameters. Therefore, although complete accuracy statistics are not known, it is evident that Rekognition can be less robust to image interference than Google Cloud Vision.

Given limitations with pose-invariance, it is important that researchers who are designing studies using Rekognition develop plans to avoid image interference during data collection. Options include limiting distractions in the testing environments (which may cause participants to break full-frontal attention with the camera), deliberately instructing participants to remain still or confining their range of motion, or setting up multiple cameras in the environments (which can capture angles that the full-frontal camera is not able to).

In summary, Amazon Rekognition is an efficient option for collecting emotion data. One analysis provides the six basic emotions, two additional emotions, landmark estimates, eye tracking estimates, validity estimates, and much more, each rated on continuous scales. However, interpreting, cleaning, and sorting through dozens of null parameters and missing data is not a friendly task for beginners. The best way to reduce missing data and measurement error is to create distraction-free and interference-free environments for experiments, so that cameras can capture participants' data reliably. Proactive users may find that Rekognition is a tool to apply existing knowledge of emotion recognition AI with more validity and reliability.

Py-Feat

Key features

Although the present tutorial is intended for R, it is helpful to note that there are many emotion recognition packages that have been developed for Python. In general, Python has more functionality for AI methods simply due to a coding preference of developers. Nevertheless, R programmers can still gain access to these emotion recognition packages, thanks to the R package `reticulate` (Ushey et al., 2023), which allows users to code using Python in R environments and make objects and functions from the different languages to communicate with each other.

The Python Facial Expression Analysis Toolbox (Py-Feat) provides a large library of pre-trained models that allow users to conduct a wide range of facial expression analyses, from more standard processes like face detection and landmarking to more complex ones like action units and identity detection (Cheong et al., 2023). The toolbox is highly customizable, allowing users to interchange individual algorithms and substitute or develop their own. For example, a typical emotion recognition operation consists of five algorithmic elements: face detection (Deng et al., 2019), landmark detection (Chen et al., 2018), action unit detection (Chen & Guestrin, 2016), facial pose estimation (Albiero et al., 2020), and an emotion model (Pham et al., 2020). A default algorithm is provided for each of these components, but users can opt to swap any element with another algorithm, whether by preference or necessity. Although functionality is available for both image and video analyses, we will focus on its image capabilities. Compared to Google Cloud Vision and Amazon Rekognition, a key advantage of Py-Feat is that all the algorithms available in its toolbox are free of charge.

Py-Feat provides estimates for seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. Each emotion estimate represents a percentage of the whole image, rated as a 0-1 decimal. Naturally, the sum of all emotion estimates is equal to 1. Py-Feat also allows users to generate visualizations of the action units and facial poses that are responsible for its presented emotion estimates. Users can opt to either highlight facial features on the original image or project the features onto a standardized template face. Regardless of the visualization option chosen, the opportunity to directly observe the attributes of your data that result in an algorithmic decision opens the black box of emotion recognition AI and greatly advances the goal of explainable AI.

How to use

To access Python and its packages, one must first install and load the `reticulate` package in R. Then, one needs to install the necessary Python environment for use with `Py-Feat`. For example, the function `virtualenv_install` creates a Python environment called `r-py-feat` (can be any chosen name by a user) and installs the Python package `py-feat` within it. It will also install all the necessary dependent packages to use `py-feat`. The package `reticulate` allows R to install many different copies of Python environments. Many of Py-Feat's dependencies update and change over time, so for the sake of consistency, we set our virtual environment to use Python 3.8.10, which has been validated to work with Py-Feat. Users may encounter issues replicating our sample code for more recent versions of Python. To use the one we set up in the future, simply load it using R code `use_virtualenv("r-py-feat")`.

```
library (reticulate)
install_python ('3.8.10')
use_python_version ('3.8.10')
virtualenv_create ('r-py-feat', python='3.8.10')
virtualenv_install ("r-py-feat", "py-feat")
```

To use Py-Feat for emotion recognition in R, the following code can be used. Line 1 simply initializes the python environment for Py-Feat. Line 3 loads the Py-Feat module into R for use. Py-Feat provides many different detection methods (Cheong et al., 2023) and here the default choices are supplied with the Line 4. An image can then be supplied for emotion detection as shown in Line 6. Again, one can analyze a batch of images using a `for` loop. Our example batch of 1,415 images took 44.43 min to complete, which is an average of 0.03 min (1.8 s) per image.

```
1 use_virtualenv ("r-py-feat")
2
3 feat< -import ("feat")
4 detector< -feat$Detector ()
5
6 res< -detector$detect_image ("myimagefolder/image_
  000002.jpeg")
7 res[,c("anger","disgust","fear","happiness","sadness",
  "surprise","neutral")]
```

The output includes the estimates of each of the seven emotions discussed earlier, in which happiness has the highest probability of being observed in the sample image.

```
             [,1]
anger        7.815199e-05
disgust      9.324357e-05
fear         5.976111e-04
happiness    9.748173e-01
sadness      9.838692e-05
surprise     2.417541e-02
neutral      1.397787e-04
```

Specific values for AUs can be obtained with the following code. Happiness consists of AUs 6 and 12, both of which have high probabilities of being active in the current image, so happiness is the most probable emotion here. Although other AUs are also high (e.g., 7, 11, 20, 25), they do not align with the classification model for a specific emotion, whereas AUs 6 and 12 are reliable indicators of happiness.

```
>do.call('rbind', res[, names(res)
  [substr(names(res),1,2)=='AU']])
        [,1]
AU01    0.52407897    AU14    0.81276035
AU02    0.24766958    AU15    0.15845187
AU04    0.26715773    AU17    0.12945455
AU05    0.23616673    AU20    1.00000000
AU06    0.95762956    AU23    0.09344121
AU07    1.00000000    AU24    0.01337716
AU09    0.67707616    AU25    0.99998069
AU10    0.99551094    AU26    0.68287283
AU11    1.00000000    AU28    0.01822461
AU12    0.99029797    AU43    0.11547881
```

In addition, Py-Feat is able to visualize AUs, either on the face of the original image or on a computer-generated template; however, visualization requires a special class of dataframe objects called `Fex` objects, which are currently not available in R. This data class is only available in Python, but Fex objects can still be accessed through RStudio. Users can create a separate Python chunk in a R markdown file and run the detector function, which automatically converts image estimates as `Fex` objects in Python.

```
1 !pipinstallscikit-learn==1.3.0
2 !pipinstallpy-feat
3 !pipinstallmatplotlib
4
5 importfeat
6 detector=feat.Detector()
7 res=detector.detect_image("myimagefolder/image_
  000002.jpeg")
8 figs=res.plot_detections(faces="aus",muscles=True,
  add_titles=False)
9
10 importmatplotlib.pyplotasplt
11 plt.show(figs)
```

Note, that it is a known issue that the `plot_detections()` function has been limited by a recent update in the Scikit-Learn package (one of Py-Feat's dependencies), which interrupts the plotting of AU locations. To circumvent this issue in the current Py-Feat version, we install a previous version of Scikit-Learn ($\leq$1.3.0) in line 1. Visualization corresponding to the output above is represented below (Figure 2).

Discussion

In contrast to Google Cloud Platform and Amazon Web Services, which were primarily designed for

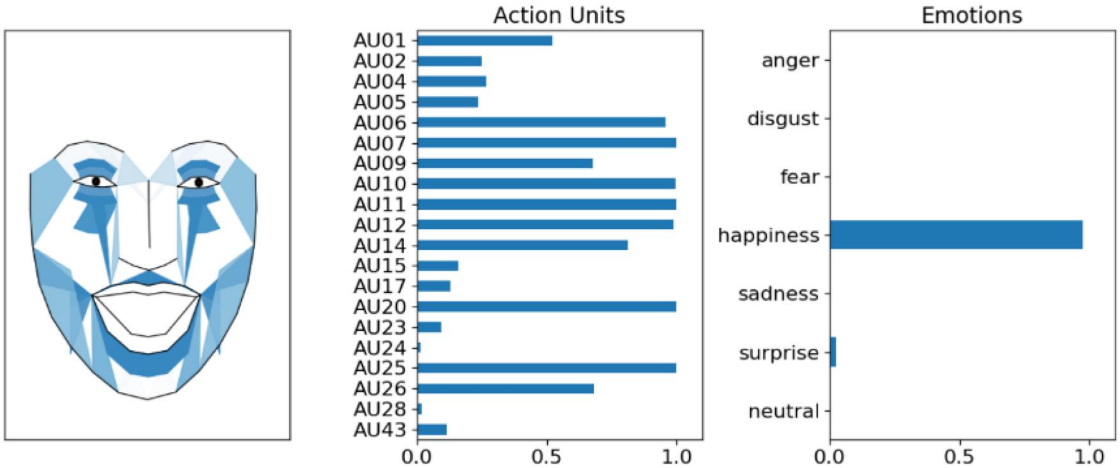

Figure 2. Py-Feat Visualization of action units, emotions, and active muscle regions projected onto a sample template face.

commercial users, Py-Feat was designed more aligned with social and behavioral research. Moreover, Py-Feat was created to provide user-friendly, open-source facial expression analysis software for researchers (Cheong et al., 2023), which makes it the most accessible program included in the tutorial. The toolkit is completely free to use, including all of its emotion recognition features, customizable models, and visualization capabilities. Py-Feat does not utilize cloud-based infrastructure, which places the burdens of securing the necessary hardware to store models and analyze data on the users. Although the difference in per-image efficiency between Py-Feat and cloud-computing programs seems minuscule, the time spent analyzing images becomes more stark over time. For example, the per-image difference between Py-Feat and AWS was 1.5 s, yet to analyze the same batch of 1,415 images, Py-Feat took 36.85 min longer than AWS. Users with particularly large samples of image data and less powerful computers may have to rely on supercomputers or cloud-computing servers to process data efficiently.

However, one advantage of open-source software is that it is transparent and can be adjusted by the user. For example, the pre-trained models for facial expression analysis can be selected by individual users. In addition, users can load pre-trained models from other existing literature, or develop their own models, and integrate them into Py-Feat. Even though open-source software may not necessarily function on par with leading commercial programs (i.e., Google and AWS), they offer a higher ceiling than licensed products so that dedicated users can transform the software to suit their research needs. However, modifying the toolkit will require a strong background in Python, which makes this option less accessible for R programmers. Therefore, our discussion just focuses on the models available in Py-Feat.

Py-Feat still has a distinct advantage over Google Cloud Vision regarding its measurement of emotions. Py-Feat provides information for the basic six emotions and a measure of neutrality, in the case that the algorithm is not able to decipher the emotion being presented in an image. Furthermore, these emotions are measured on a continuous scale of 0 to 1, which greatly outperforms Google's six-point Likert scale. Although Amazon Rekognition's emotion scale allows for more variability (0–100), Py-Feat's continuous scale has a more appropriate interpretation for the kinds of analyses that are valuable to social and behavioral researchers. Rekognition rates emotions by the degree to which the algorithm is confident that the emotion is being presented in the image, whereas Py-Feat rates emotions by their percentage of the whole emotional expression in the image. The latter interpretation is more aligned with contemporary psychological research (Cheong et al., 2023), providing a more valid interpretation of phenomenon like emotional intensity and latency (Haines et al., 2019).

Since Py-Feat is a more recent toolkit, there has not been much literature examining its performance in comparison with other established tools. Cheong et al. (2023) discussed the performance of Py-Feat in terms of F1 scores, an accuracy metric used for binary classification. Py-Feat achieved an average F1 score of 0.55, which outperformed iMotions Cheong et al., 2023), but previous benchmarking data for Amazon Rekognition yielded an average F1 score of 0.87 (Yang et al., 2021). Naturally, Py-Feat's estimation of happy faces yielded high accuracy ($F1 = 0.77$), whereas all other emotions yielded moderate accuracy, ranging from 0.48 to 0.55 (Cheong et al., 2023). Other studies comparing open-source emotion recognition AI led to similar results (Hsu & Sato, 2023; Namba et al., 2021), suggesting that Py-Feat performs on par with other open-source programs, but struggles to compete against larger commercial AI models. Although Py-Feat's forward-facing accuracy may be underwhelming, Namba et al. (2021) suggests that the program is incredibly robust against rotated-pose images, detecting no significant differences in accuracy across four angles 15 degrees apart (0, 15, 30, 45). Py-Feat is similarly robust to high and low luminance (Cheong et al., 2023), making the tool a great option for handling real-world images (Santana et al., 2023) and reducing missing data.

Py-Feat would also be advantageous when there is a research question related to an underlying mechanism behind an emotion detection model. As mentioned previously in our discussion of ANNs, researchers can open the black box of deep-learning models by forcing the network to generate an image with the same parameters it used to make its classification decision (Castelvecchi, 2016). Py-feat provides specific values for AUs, whereas most other emotion recognition models do not, giving Py-Feat a distinct advantage in interpretability. In other programs, users have to take the confidence and emotion estimates of the model at face value, but the availability of AUs allows Py-Feat users to work backwards from classification decisions and make more holistic determinations of whether the outputted emotion estimates are accurate. Furthermore, it allows Py-Feat users to draw more generalized about the strength of the ANN model as a whole. If a model is able to closely reproduce the original input image based on its parameters, it is believed that the model sufficiently understands how to reach the correct classification. Likewise, users can look at the model-identified AUs, compare them to actual AUs, and interpret whether the model is able to identify the location and change in facial landmarks in a way that would be intuitive to human observation. Py-Feat's visualization function can help with both of these diagnostic and interpretation goals, which is another unique feature that is not available in commercial emotion recognition programs.

In summary, Py-Feat is a tool that anyone can download for free to learn about emotion recognition AI and visualize the theory behind the algorithms. Minimal knowledge of Python is needed to access the tool with R and reticulate. It provides full coverage of basic emotions, neutrality, and the variability to effectively measure individual differences; however, analyzing data will take a lot longer than cloud-based programs on not very powerful computers. Py-Feat's forward-facing accuracy tends to underperform compared to commercial emotion recognition programs, yet its overall robustness and visualization make the toolkit an effective companion to researchers.

Table 1. Summary of pros and cons for Google Cloud Vision, Amazon Rekognition, and Py-feat.

		Google	Amazon	Py-Feat
Pros	Fast processing speed	X	X	
	Generous free trial quotas	X	X	
	Free and open source			X
	Large range of emotions		X	X
	Large range of variability		X	X
	Above-average pose-invariant accuracy	X		X
	Visualization options			X
	High TPR for all emotions		X	
Cons	Slow processing speed			X
	Paid subscription required	X	X	
	Average pose-invariant accuracy		X	
	Below average forward-facing accuracy	X		X
	Limited emotion range	X		
	Requires limited knowledge of Python			X

Conclusion

Social and behavioral perspectives have contributed to the development of sophisticated AI models, such as computer vision and deep-learning neural networks, by providing the theoretical infrastructure to help computers mimic internal human processes. As a result, AI is a powerful tool that can assist social and behavioral research in numerous dimensions, which has been demonstrated by improving the handling of noisy neuroimaging data (Hosseini et al., 2021), the construct validity of assessments from natural language processing (Kjell et al., 2019), and the prediction accuracy from data mining methods (Yarkoni & Westfall, 2017). Thus, the increased implementation of AI modeling in social and behavioral research has the opportunity to mutually benefit both disciplines.

Likewise, emotion recognition AI was built upon decades of emotion research and has the potential to contribute to future emotion research (Wyman and Zhang (2023) for a discussion of current and potential applications). Although significant validity and reliability concerns are still present, especially regarding the interpretability of AI models, emotion recognition AI offers distinct advantages over traditional measurement approaches, including the visualization of emotions, the precision of emotion variation at adjacent time points, and the ability to observe passive emotion functioning. Each program offers distinct advantages and disadvantages that may appeal, which may appeal differently to different researchers depending on the level of specificity in their research interests, the suitability of research environments, and other study design constraints. See Table 1 for a comprehensive summary of advantages and disadvantages of each program discussed in the tutorial.

Our tutorial provides the first step for researchers in designing studies related to emotion recognition AI, processing, and analyzing data. The three identified programs (Google Cloud Vision, Amazon Rekognition, and Py-Feat) are sufficient for most research questions and image data sets, but future work should also provide directions for other programs, such as iMotions (Stöckli et al., 2018) and OpenFace (Amos et al., 2016). There is large room for improvement regarding the performance of these methods, yet, as discussed within the Amazon Rekognition section, these tools are updating and evolving over time, with each time increasing their utility for social and behavioral researchers. As emotion recognition AI continues to evolve, we can expect that detecting more complex constructs (e.g., confusion and attention) will become more accurate and more available across different models. Just as Amazon Rekognition increased the number of emotions it can detect over years of software updates, we can expect many other emotion recognition models will try to improve their range of capabilities, which will simultaneously expand the number of hypotheses that can be explored with this technology. We can also expect that relics of training bias in performance, such as racial, gender, and age bias (Kim et al., 2021) will be greatly reduced in future versions. Although the status quo of artificial intelligence is uncertain, the joint exploration and development of computational methods and social and behavioral research can help researchers arrive at a better understanding of both processes. We also hope the increasing use of such tools will further enhance their performance.

Article information

Conflict of Interest Disclosures: Each author signed a form for disclosure of potential conflicts of interest. No authors reported any financial or other conflicts of interest in relation to the work described.

Ethical Principles: The authors affirm having followed professional ethical guidelines in preparing this work. These guidelines include obtaining informed consent from human participants, maintaining ethical treatment and respect for the rights of human or animal participants, and ensuring the privacy of participants and their data, such as ensuring that individual participants cannot be identified in reported results or from publicly available original or archival data.

Funding: This work was supported by Grant R305D210023 from the Institute of Education Sciences and by Grant 2236418 from the National Science Foundation. However, the contents of the paper do not necessarily represent the policy of the Department of Education or the National Science Foundation, and you should not assume endorsement by the Federal Government.

Role of the Funders/Sponsors: None of the funders or sponsors of this research had any role in the design and conduct of the study; collection, management, analysis, and interpretation of data; preparation, review, or approval of the manuscript; or decision to submit the manuscript for publication.

References

Albiero, V., Chen, X., Yin, X., Pang, G., & Hassner, T. (2020). *img2pose: Face alignment and detection via 6DoF, face pose estimation*. CoRR, https://arxiv.org/abs/2012.07791.

Al-Omair, O. M., & Huang, S. (2018). A comparative study on detection accuracy of cloud-based emotion recognition services [Paper presentation]. Proceedings Of the 2018 International Conference on Signal Processing and Machine Learning, SPML 2018 (pp. 142–148), Shanghai, China, November 28–30, 2018.

Amazon Web Services. (2016). *Amazon Rekognition – video and images*. https://docs.aws.amazon.com/rekognition.

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, *6*(2), 20.

Anbusegaran, S. (2021). Unobtrusive assessment of student engagement levels in online classroom environment using emotion analysis (Publication No. 2265) [Master's thesis, Georgia Southern University]. Electronic Theses and Dissertations. https://digitalcommons.georgiasouthern.edu/etd/2265/.

Baidu, A. I. C. (2019). *Face recognition*. https://intl.cloud.baidu.com/product/face.html.

Bryant, D. G., & Howard, A. M. (2019). A comparative analysis of emotion-detecting AI systems with respect to algorithm performance and dataset diversity In V. Conitzer, G. K. Hadfield, & S. Vallor (Eds.), *Proceedings of the 2019 AAAI/ACM conference on AI, Ethics, and Society* (pp. 377–382). ACM. https://doi.org/10.1145/3306618.3314284.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature*, *538*(7623), 20–23. https://doi.org/10.1038/538020a

Celeghin, A., Diano, M., Bagnis, A., Viola, M., & Tamietto, M. (2017). Basic emotions in human neuroscience: Neuroimaging and beyond. *Frontiers in Psychology*, *8*, 1432. https://doi.org/10.3389/fpsyg.2017.01432

Chan, D. (2010). So why ask me? Are self-report data really that bad? In C. E. Lance & R. J. Vandenberg (Eds.), *Statistical and methodological myths and urban legends* (pp. 329–356). Routledge.

Chen, S., Liu, Y., Gao, X., & Han, Z. (2018). *MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices*. CoRR, http://arxiv.org/abs/1804.07573.

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. CoRR, http://arxiv.org/abs/1603.02754.

Cheong, J. H., Jolly, E., Xie, T., Byrne, S., Kenney, M., & Chang, L. J. (2023). Py-Feat: Python facial expression analysis toolbox. *Affective Science*, *4*(4), 781–796. https://doi.org/10.1007/s42761-023-00191-4

Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial neural networks: Opening the black box. *Cancer*, *91*(S8), 1615–1635. https://doi.org/10.1002/1097-0142(20010415)91:8+<1615::AID-CNCR1175>3.0.CO;2-L

Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., & Zafeiriou, S. (2019). *RetinaFace: Single-stage dense face localisation in the wild.* CoRR, abs/1905.00641. http://arxiv.org/abs/1905.00641

Deshmukh, R. S., & Jagtap, V. (2017). A survey: Software API and database for emotion recognition [Paper presentation]. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 284–289). Madurai, India.

Donato, G., Bartlett, M. S., Hager, J. C., Ekman, P., & Sejnowski, T. J. (1999). Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(10), 974–989. https://doi.org/10.1109/34.799905

Edmondson, M. (2023). *googleAuthR: Authenticate and create Google apis* [R package version 2.0.1]. https://CRAN.R-project.org/package=googleAuthR.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, *6*(3–4), 169–200. https://doi.org/10.1080/02699939208411068

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, *17*(2), 124–129. https://doi.org/10.1037/h0030377

Ekman, P., & Friesen, W. V. (1976). Measuring facial movement. *Environmental Psychology and Nonverbal Behavior*, *1*(1), 56–75. https://doi.org/10.1007/BF01115465

Cloud Platform, G. (2015). *Vision API – image content analysis*. https://cloud.google.com/vision/docs/detecting-faces.

Haines, N., Southward, M. W., Cheavens, J. S., Beauchaine, T., & Ahn, W.-Y. (2019). Using computer-vision and machine learning to automate facial coding of positive and negative affect intensity. *PLOS One*, *14*(2), e0211735. https://doi.org/10.1371/journal.pone.0211735

Hitron, T., Orlev, Y., Wald, I., Shamir, A., Erel, H., & Zuckerman, O. (2019). Can children understand machine learning concepts? The effect of uncovering black boxes [Paper presentation]. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland* (pp. 1–11). ACM.

Holm, E. A. (2019). In defense of the black box. *Science*, *364*(6435), 26–27. https://doi.org/10.1126/science.aax0162

Hosseini, M.-P., Hosseini, A., & Ahi, K. (2021). A review on machine learning for EEG signal processing in bioengineering. *IEEE Reviews in Biomedical Engineering*, *14*, 204–218. https://doi.org/10.1109/RBME.2020.2969915

Hsu, C.-T., & Sato, W. (2023). Electromyographic validation of spontaneous facial mimicry detection using automated facial action coding. *Sensors*, *23*(22), 9076. https://doi.org/10.3390/s23229076

Izard, C. E. (2007). Basic emotions, natural kinds, emotion schemas, and a new paradigm. *Perspectives on Psychological Science*, *2*(3), 260–280. https://doi.org/10.1111/j.1745-6916.2007.00044.x

Jacobucci, R., & Grimm, K. J. (2020). Machine learning and psychological research: The unexplored effect of measurement. *Perspectives on Psychological Science*, *15*(3), 809–816. https://doi.org/10.1177/1745691620902467

Jain, P. R., Quadri, S. M. K., & Lalit, M. (2021). Recent trends in artificial intelligence for emotion detection using facial image analysis [Paper presentation]. IC3 2021: Thirteenth International Conference on Contemporary Computing, Noida, India, August 5–7, 2021 (pp. 18–36).

Keltner, D., Ekman, P., Gonzaga, G., & Beer, J. (2000). *Facial expression of emotion* (pp. 236–248). Guilford Publications.

Khanal, S. R., Barroso, J., Lopes, N., Sampaio, J., & Filipe, V. (2018, June 20–22). Performance analysis of Microsoft's and Google's emotion recognition API using pose-invariant faces [Paper presentation]. Proceedings of The 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion, DSAI 2019, *Thessaloniki, Greece*, 2018 (pp. 172–178).

Khanal, S. R., Sharma, P., Fernandes, H., Barroso, J., & de Jesus Filipe, V. M. (2023). *Performance analysis and evaluation of cloud vision emotion Apis*. arXiv preprint arXiv: 2303.12974. https://doi.org/10.48550/ARXIV.2303.12974

Kim, E., Bryant, D., Srikanth, D., & Howard, A. M. (2021). Age bias in emotion detection: An analysis of facial emotion recognition performance on young, middle-aged, and older adults. In M. Fourcade, B. Kuipers, S. Lazar, & D. K. Mulligan (Eds.), *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event,* USA, *May 19*–21, 2021 (pp. 638–644). ACM.

Kjell, O. N., Kjell, K., Garcia, D., & Sikström, S. (2019). Semantic measures: Using natural language processing to measure, differentiate, and describe psychological constructs. *Psychological Methods*, *24*(1), 92–115. https://doi.org/10.1037/met0000191

Koncz, T., Varkoly, B., Lukacs, P., & Kocsis, E. (2020). *GoogleCloudVisionR: Access to the 'Google cloud vision' API for image recognition, ocr and labeling* [R package version 0.2.0]. https://CRAN.R-project.org/package=googleCloudVisionR.

Kretch, D., & Banker, A. (2023a). *Paws: Amazon web services software development kit* [R package version 0.4.0]. https://CRAN.R-project.org/package=paws.

Kretch, D., & Banker, A. (2023b). *Paws.machine.learning: 'amazon web services' machine learning services* [R package version 0.4.0]. https://CRAN.R-project.org/package=paws.machine.learning.

Kulke, L., Feyerabend, D., & Schacht, A. (2020). A comparison of the Affectiva iMotions Facial Expression Analysis Software with EMG for identifying facial expressions of emotion. *Frontiers in Psychology*, *11*, 329. https://doi.org/10.3389/fpsyg.2020.00329

Larsen, R. J., & Fredrickson, B. L. (1999). Measurement issues in emotion research. *Well-Being: The Foundations of Hedonic Psychology*, *40*, 60.

Lien, J. J., Kanade, T., Cohn, J. F., & Li, C. (1998). Automated facial expression recognition based on FACS action units. *3rd International Conference on Face & Gesture Recognition* (FG '98), April 14–16, 1998, Nara, Japan (pp. 390–395).

Lim, J. Z., Mountstephens, J., & Teo, J. (2020). Emotion recognition using eye-tracking: Taxonomy, review and current challenges. *Sensors*, *20*(8), 2384. https://doi.org/10.3390/s20082384

Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS One*, *13*(5), e0196391. https://doi.org/10.1371/journal.pone.0196391

Martinez, B., Valstar, M. F., Jiang, B., & Pantic, M. (2019). Automatic analysis of facial actions: A survey. *IEEE Transactions on Affective Computing*, *10*(3), 325–347. https://doi.org/10.1109/TAFFC.2017.2731763

Megvii. (2017). *Face++ cognitive services*. https://www.faceplusplus.com.

Namba, S., Sato, W., & Yoshikawa, S. (2021). Viewpoint robustness of automated facial action unit detection systems. *Applied Sciences*, *11*(23), 11171. https://doi.org/10.3390/app112311171

Noldus. (2014). *FaceReader: Tool for automatic analysis of facial expression*. https://www.noldus.com/facereader.

nViso. (2016). *Artificial intelligence emotion recognition software*. https://nviso.ai/facial-action-unit-detection-and-recognition.

Pantic, M., Rothkrantz, L. J. M., & Koppelaar, H. (1998). Automation of non-verbal communication of facial expressions. In A. Verbraeck & M. Al-Akaidi (Eds.), *EUROMEDIA 1998 featuring Webtec-Mediatec-Comtec-Aptec*, January 5–6, 1998 (pp. 86–93). De Montfort University SCS, The Society for Computer Simulation International.

Pham, L., Vu, T. H., & Tran, T. A. (2020, January 10–15). Facial expression recognition using residual masking network [Paper presentation]. 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event, /Milan, Italy, 2021 (pp. 4513–4519).

Pinto, L. V. L., Alves, A. V. N., Medeiros, A. M., Costa, S. W. D. S., Pires, Y. P., Costa, F. A. R., & da Rocha Seruffo, M. C. (2023). A systematic review of facial expression detection methods. *IEEE Access*, *11*, 61881–61891. https://doi.org/10.1109/ACCESS.2023.3287090

Puri, R., Gupta, A., Sikri, M., Tiwari, M., Pathak, N., & Goel, S. (2020). *Emotion detection using image processing in python*. CoRR, https://arxiv.org/abs/2012.00659.

Rafael, G., Kusuma., & H., Tasripan. (2020). The utilization of cloud computing for facial expression recognition using amazon web services [Paper presentation]. 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) (pp. 366–370). Surabaya, Indonesia.

Ramakrishnan, A., Ottmar, E., LoCasale-Crouch, J., & Whitehill, J. (2019, May 14–18). Toward automated classroom observation: Predicting positive and negative climate [Paper presentation]. 14th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2019, Lille, France, 2019 (pp. 1–8). Lille, France.

Santana, O. J., Freire-Obregón, D., Hernández-Sosa, D., Lorenzo-Navarro, J., Sánchez-Nielsen, E., & Castrillón-Santana, M. (2023). Facial expression analysis in a wild sporting environment. *Multimedia Tools and Applications*, *82*(8), 11395–11415. https://doi.org/10.1007/s11042-022-13654-w

Siedlecka, E., & Denson, T. F. (2019). Experimental methods for inducing basic emotions: A qualitative review. *Emotion Review*, *11*(1), 87–97. https://doi.org/10.1177/1754073917749016

Stephens, C. L., Christie, I. C., & Friedman, B. H. (2010). Autonomic specificity of basic emotions: Evidence from pattern classification and cluster analysis. *Biological Psychology*, *84*(3), 463–473. https://doi.org/10.1016/j.biopsycho.2010.03.014

Stöckli, S., Schulte-Mecklenbeck, M., Borer, S., & Samson, A. C. (2018). Facial expression analysis with affdex and facet: A validation study. *Behavior Research Methods*, *50*(4), 1446–1460. https://doi.org/10.3758/s13428-017-0996-1

Tabassum, T., Allen, A. A., & De, P. (2020). Non-intrusive identification of student attentiveness and finding their correlation with detectable facial emotions [Paper presentation]. In *Proceedings of The 2020 ACM Southeast Conference, ACMSE 2020, Tampa, Florida* (pp. 127–134). ACM.

Taylor, J. E. T., & Taylor, G. W. (2021). Artificial cognition: How experimental psychology can help generate explainable artificial intelligence. *Psychonomic Bulletin & Review*, *28*(2), 454–475. https://doi.org/10.3758/s13423-020-01825-5

Ushey, K., Allaire, J., & Tang, Y. (2023). *Reticulate: Interface to 'python'* [R package version 1.34.0]. https://CRAN.R-project.org/package=reticulate.

Viso AI. (2021). *Facial emotion analysis*. https://viso.ai/application/emotion-analysis.

Wyman, A., & Zhang, Z. (2023). API face value: Evaluating the current status and potential of emotion detection software in emotional deficit interventions. *Journal of Behavioral Data Science*, *3*(1), 1–11. https://doi.org/10.35566/jbds/v3n1/wyman

Yang, K., Wang, C., Sarsenbayeva, Z., Tag, B., Dingler, T., Wadley, G., & Goncalves, J. (2021). Benchmarking commercial emotion detection systems using realistic distortions of facial image datasets. *The Visual Computer*, *37*(6), 1447–1466. https://doi.org/10.1007/s00371-020-01881-x

Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/1745691617693393

Zheng, W.-L., Dong, B.-N., & Lu, B.-L. (2014). Multimodal emotion recognition using EEG and Eye Tracking Data. *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5040–5043). IEEE.