In your face!
Analyzing facial expressions in R using free APIs

Tamas Nagy, PhD
Junior professor @ ELTE, Budapest
Former lead researcher @ Synetiq

@nthun

Vincent, we happy?

Background

- Emotional expressions on the face have been an important source of data since animals started to feel emotions
- Emotion recognition was dominated by human coding for a long time
- In the last decade, the proliferation of machine learning made automatic recognition of facial expressions scalable
- In the last years, several free web APIs (Application Programming Interface) became available
- These APIs are web based functions that process requests in the cloud and return the results to the sender
- Facial emotions can be used for scientific research, but also for market research or the measurement of customer satisfaction in stores.

How the emotion recognition algorithms work?

Find faces
Identify face position, tilt, etc.

Find facial landmarks in 3D

Find emotions based on constellation of landmarks

Aim: Use facial emotion recognition from R

- Identify free APIs for emotion recognition
- APIs can be called directly from R, this way, the data can be acquired and analyzed in the same environment
- Search for R packages to use them, or call the API from R
- Compare usability and main features
- Compare the performance of APIs
- Create convenience functions to:
  - Post API request
  - Plot the results in ggplot2

Selection of APIs

- Comparison of three APIs
- One using an R wrapper (Google Cloud Vision Face API)
- One with using a general purpose API (Microsoft Azure CS Face API)
- One with a specialized API (Kairos)
- All of them have free plans

Google Cloud Platform

Microsoft Cognitive Services

KAIROS

Methodology for the comparison of accuracy

- Standard face dataset (Cohn-Kanade Dataset; CK+)
- Selected 10 random pictures for all basic emotions (70 pictures)
  - happiness/joy, sadness, fear, anger, surprise, disgust, contempt
- The same pictures were fed to all APIs
- Plots were generated for all pictures, containing the face bounding rectangle and the emotion predictions

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (pp. 94–101). IEEE. https://doi.org/10.1109/CVPRW.2010.5543262

Outcome measures for comparison

- All 70 pictures got annotated by all 3 APIs
- **Average prediction confidence**: returned by the API, and averaged by emotion
- **Classification success** (“best guess”): get the prediction with the highest confidence, and match it to the reference
- **Confusion matrix**: which emotions were misclassified and what was the false classification

General steps for all APIs

1. Set up an account
2. Authenticate using api keys
3. (upload the images/videos to the web to get urls)
4. Call the API (using httr as a general approach or via custom made package)
5. Parse the results into a tidy dataframe
6. Plotting the results using ggplot2 and grid graphics

Check out my github repo for convenience wrappers for getting + parsing results and plotting in ggplot: nthun/face_reading_tools

Plotting function for bounding boxes and labels

```
plot_emotions <- function(emotion_df, img_path){
    if (!file.exists(img_path)) return("No such file in the path")
    # Plot the pic and the bounding recs with emotion prediction
    if (grepl(".jpg$|.jpeg$",img_path) == TRUE) {img <- readJPEG(img_path)}
    if (grepl(".png$",img_path) == TRUE) {img <- readPNG(img_path)}
    mgk_info <- tibble(height = dim(img)[1], width = dim(img)[2])
    g <- rasterGrob(img, interpolate = FALSE, width = unit(1,"npc"), height = unit(1, "npc"))

    ggplot(emotion_df) +
        aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax,
            x = (xmin + xmax)/2, y = ymin, # x and y needed for the labels
            group = face_id,
            label = label) +
        scale_x_continuous(limits = c(0, mgk_info$width)) +
        scale_y_reverse(limits = c(mgk_info$height, 0)) + # Y axis reversed in rastergrobs
        annotation_custom(g, xmin = 0, xmax = mgk_info$width, ymin = 0, ymax = -mgk_info$height) +
        geom_rect(alpha = 0, size = 2, color = "red") +
        geom_label(direction = "y", nudge_y = 1) +
        theme_void() +
        theme(legend.position = "none")
}
```

Google Cloud Vision (GCV) Face API

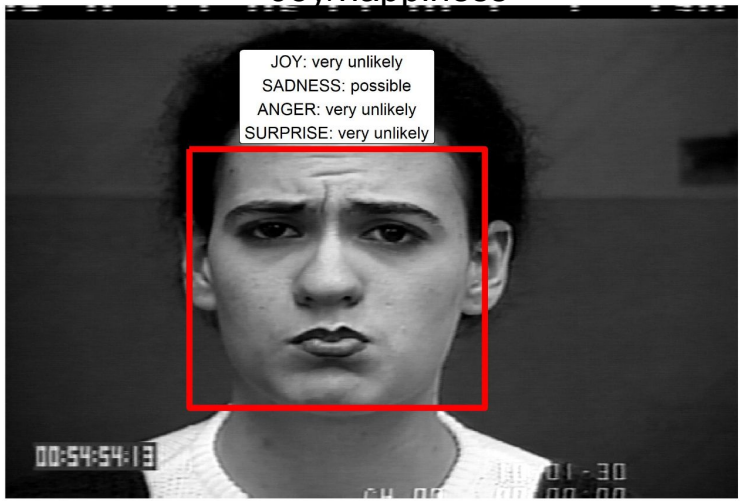
1. Has R wrapper package (RoogleVision)
2. It returns emotional classification for 4 basic emotions: joy, sadness, surprise, anger
3. Returns an ordinal classification (from very unlikely to very likely)
4. The bounding rectangles are good
5. Can only find a maximum of 5 faces per picture

Get API response using RoogleVision

```
if (!require(RoogleVision)) devtools::install_github("cloudyr/RoogleVision")
library(RoogleVision)
get_emotions_google <- function(img_path){
    if (!file.exists(img_path)) return("No such file in the path")
    gca_result <- getGoogleVisionResponse(img_path, feature = 'FACE_DETECTION') # API call via the cloudyr wrapper
    emotions <- gca_result %>% # Get emotions
                select(   happy = joyLikelihood, sadness = sorrowLikelihood, anger = angerLikelihood,
                          surprise = surpriseLikelihood) %>%
                mutate(face_id = row_number())
    coords <- # Extract bounding rectangle coordinates
               gca_result %>%
               pull(fdBoundingPoly) %>%
               mutate(face_id = row_number()) %>%
               unnest(vertices) %>%
               group_by(face_id) %>%
              summarise( xmin = nth(x,1), xmax = nth(x,2), ymin = nth(y,1), ymax = nth(y,3))
   coords %>%  # Putting together the emotions with the coordinates in a tidy data frame
            full_join(emotions, by = "face_id") %>%
            gather(emotion, value, happy:surprise) %>%
            mutate(emotion = emotion %>% str_to_upper(),
                   value = value %>% str_to_lower() %>% str_replace("_"," "))
}
```

Joy/happiness

Anger

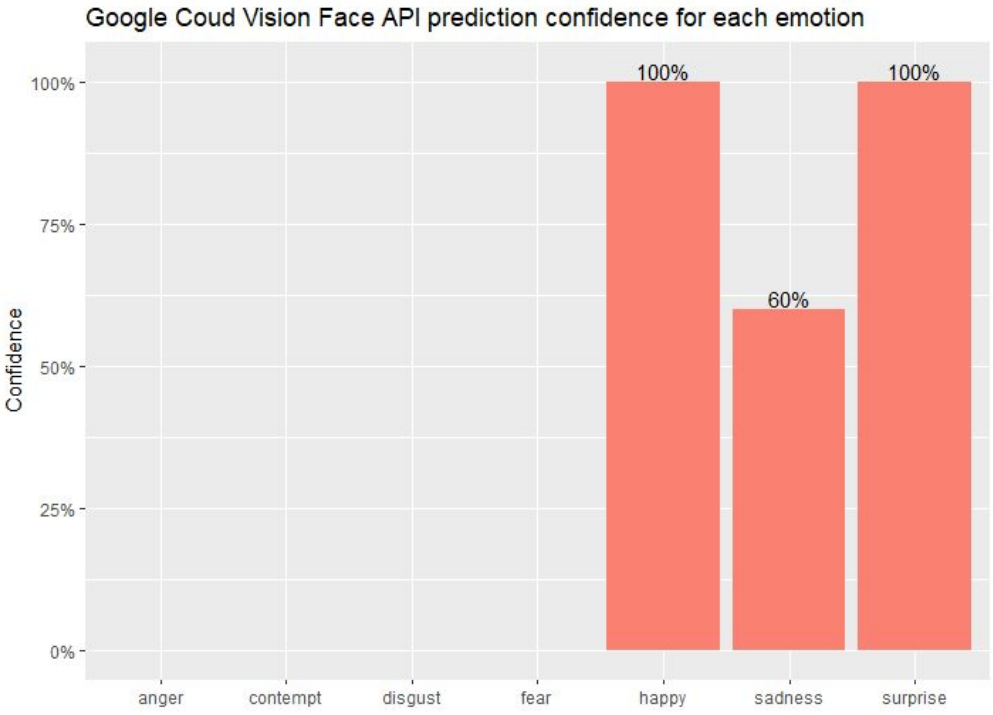


Sadness

Surprise

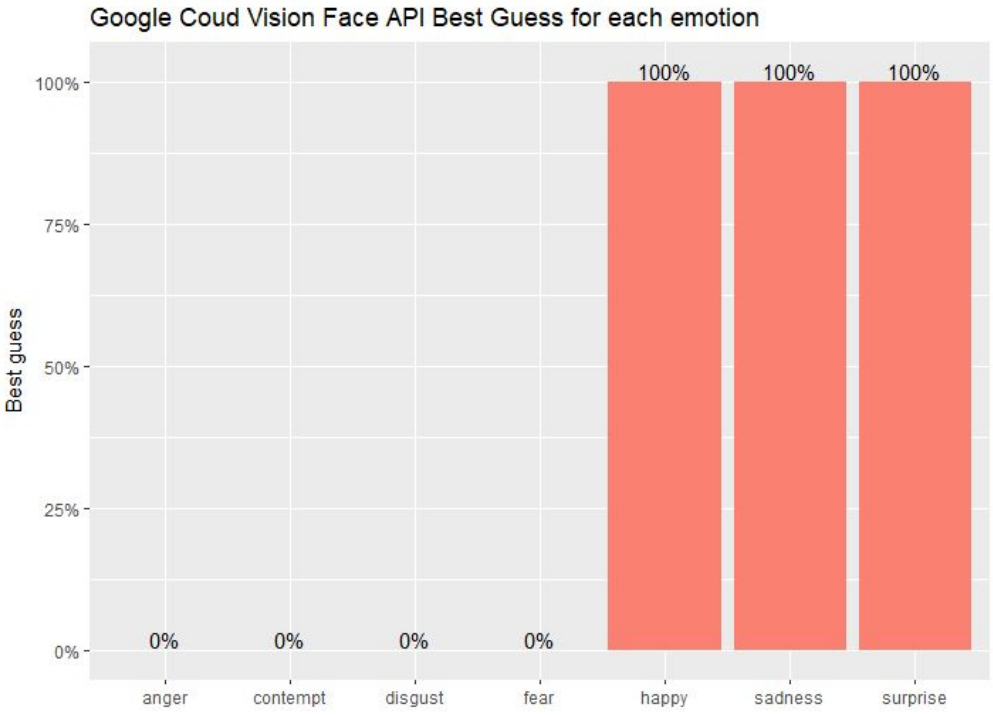
Average prediction confidence

- Prediction confidence was high for happiness and surprise
- It was medium for sadness
- Virtually non-existent for anger



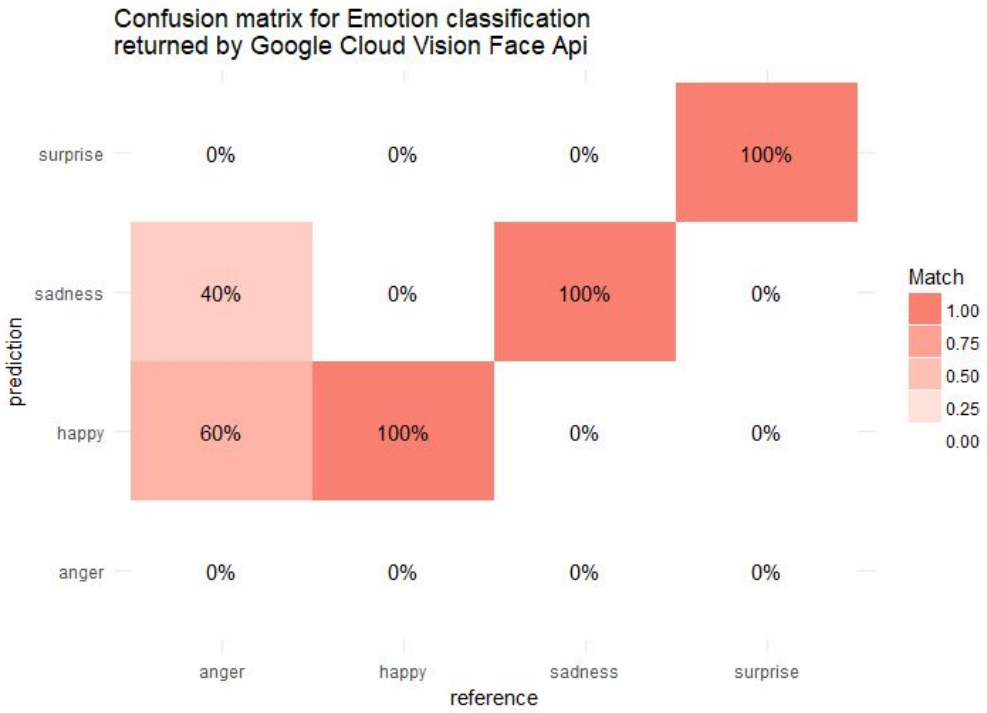
Classification accuracy

- Happiness, sadness, and surprise were identified in all pictures
- Anger was never identified
- Contempt, disgust, and fear are not measured by GCV



Confusion matrix

- Classification of happiness, sadness, and surprise are perfect
- Classification of anger is totally off; anger is usually either as happiness or sadness



Google Cloud Vision Face API results

- The easiest to use (R package, direct file upload)
- However setting up the account and the authentication requires some effort
- Limited number of emotions
- Performance for three emotions was good
- Anger was not recognized at all (possible bug?)
- Videos can only be analyzed if taken apart frame-by-frame, and sent to the API

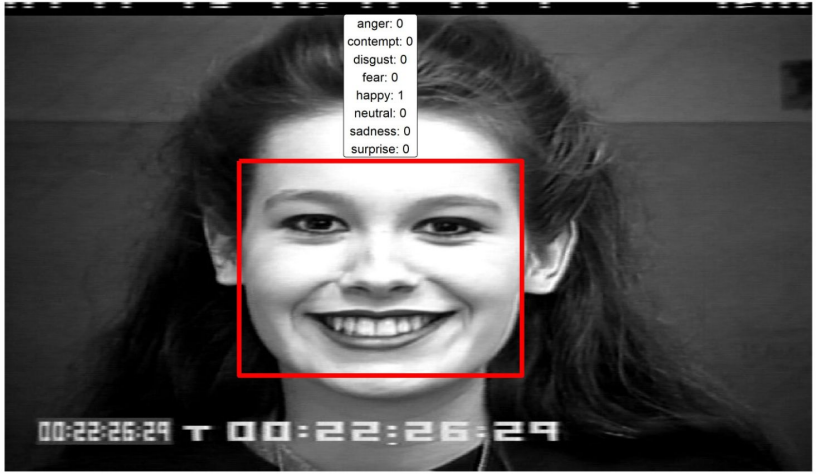
Microsoft Azure Cognitive Services Emotion API

- Maximum number of faces on an image is 64
- Recognizes all seven basic emotions plus neutral face
- Returns the exact prediction confidence for all emotions
- Predictions are not independent (they add up to 100%)

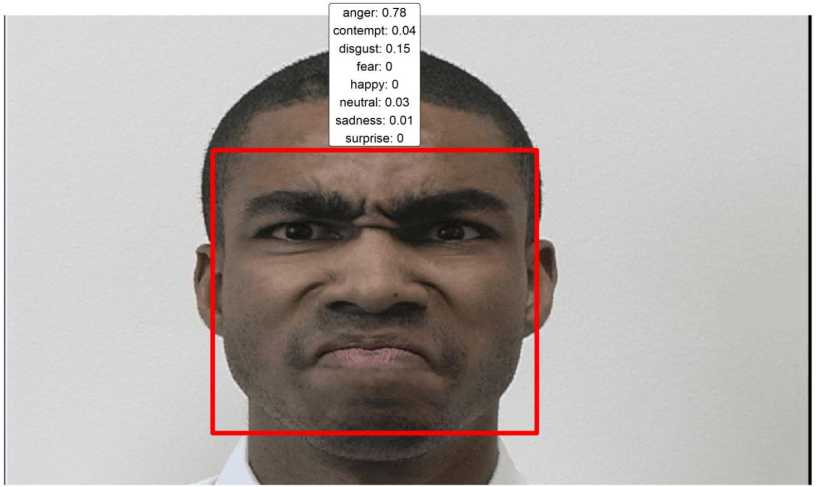
Microsoft
Cognitive Services

Get API response using MACS Face API

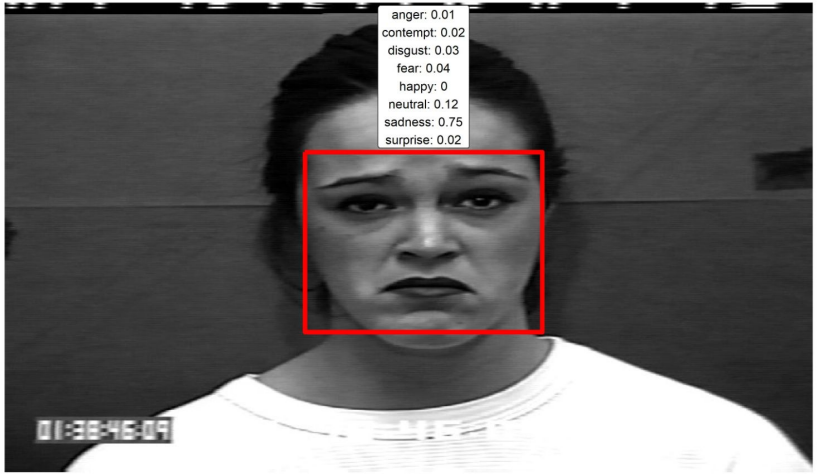
```
api_url <- "https://westus.api.cognitive.microsoft.com/emotion/v1.0/recognize?visualFeatures=Faces"
get_emotions_microsoft <- function(image_url, api_key){
    API_response <- POST(url = api_url,
                                content_type("application/json"),
                                add_headers(.headers = c('Ocp-Apim-Subscription-Key' = api_key)),
                                body = list(url = image_url),
                                encode = "json")
    df <- content(API_response)
    map_dfr(df, "faceRectangle") %>% # Put results in a tidy data frame
        mutate(face_id = row_number()) %>%
        full_join(map_dfr(df, "scores") %>%
                          mutate(face_id = row_number()),
                  by = "face_id") %>%
        mutate(xmin = left,
               xmax = left + width,
               ymin = top, # It is calculated from the top left
               ymax = top + height) %>%
        select(face_id, xmin:ymax, anger:surprise) %>%
        gather(emotion, value, -(face_id:ymax)) %>%
        mutate(value = value %>% round(2)) # Round values for 2 decimals
}
```



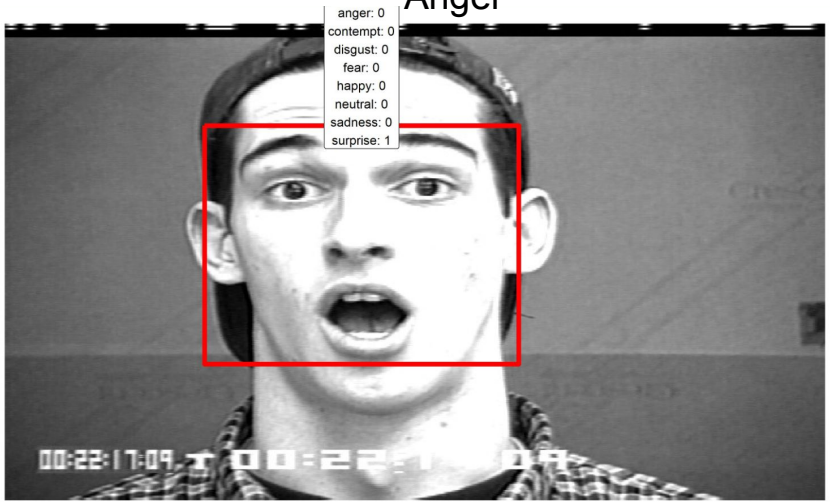
Joy/happiness



Anger



Sadness



Surprise

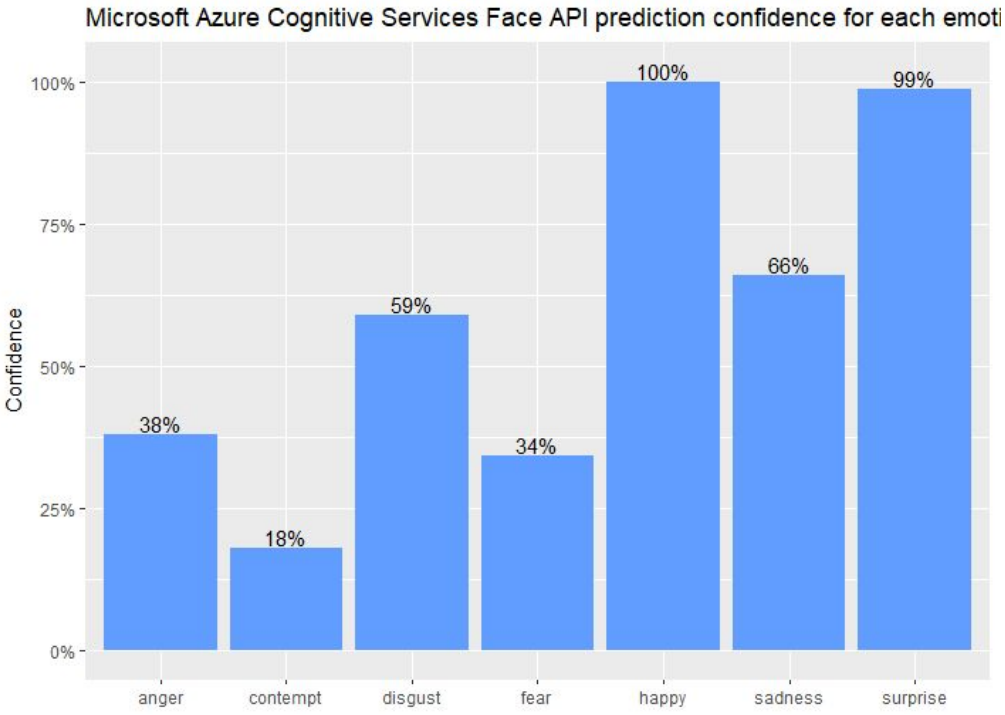
Disgust

Contempt

Fear

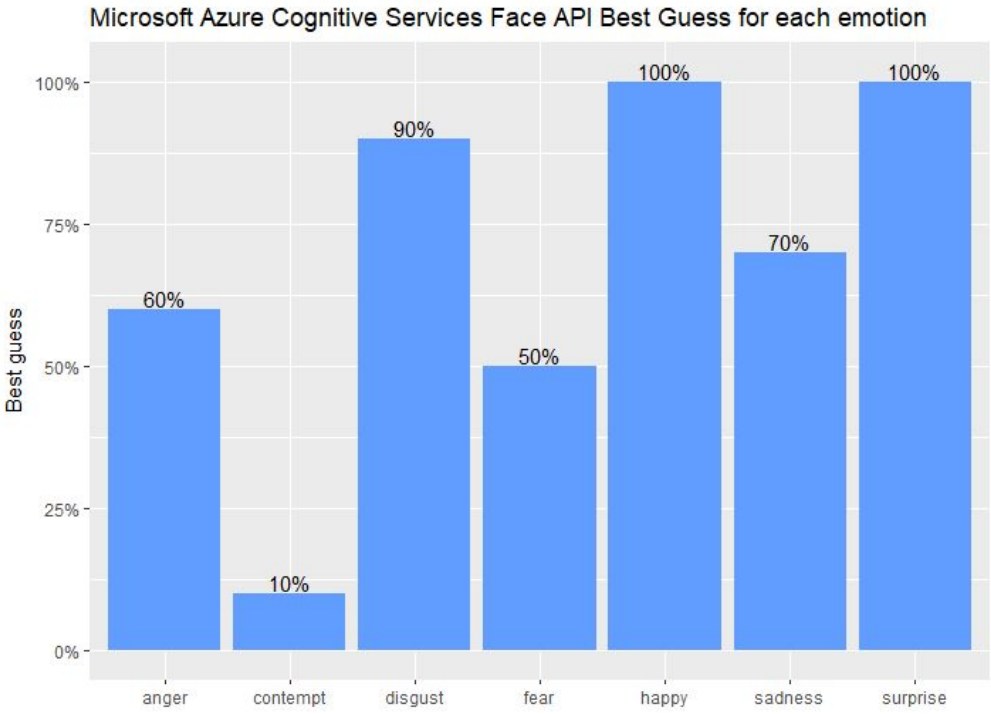
Average prediction confidence

- Prediction confidence was high for happiness and surprise
- It was medium for disgust and sadness
- Low for anger, fear, and especially contempt



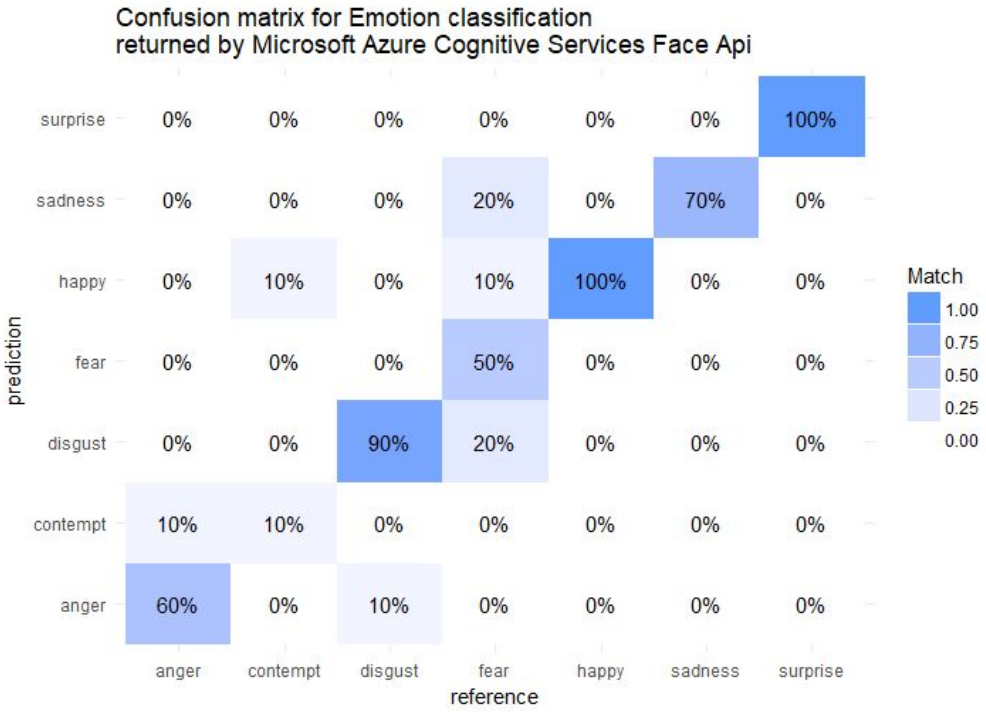
Classification accuracy

- Solid performance for happiness, surprise, disgust
- Fear, anger, and sadness are identified in the majority of cases
- Contempt was recognized poorly



Confusion matrix

- Happiness, sadness, surprise, and disgust are usually classified well
- Fear is misclassified as other emotions in 50% of the cases
- Contempt and anger are often recognized as neutral emotional states



Microsoft Azure Cognitive Services results

- Setting up an Azure account can be confusing at the first time
- Can recognize several faces on the same picture
- Solid accuracy for most basic emotions
- Contempt is recognized poorly
- Can analyze videos

Microsoft
Cognitive Services

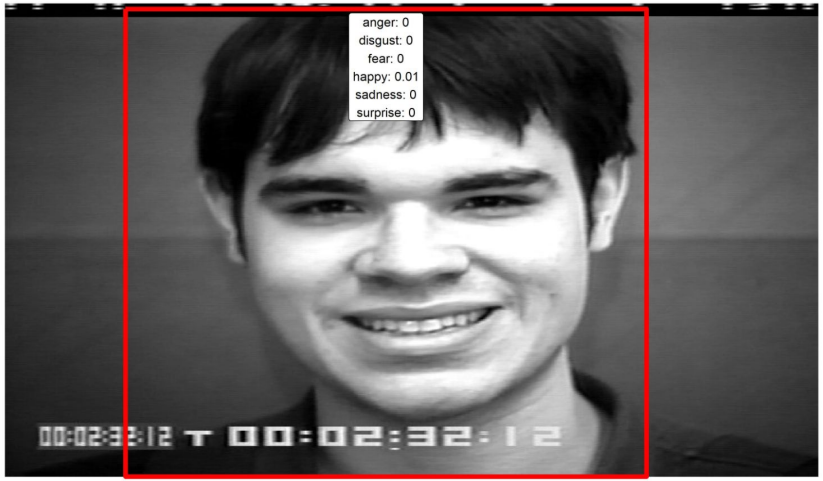
Kairos API

- Smaller company
- The API can only do face recognition and analysis
- Returns six basic emotions
- Predictions are continuous (0-100) and independent for each emotion
- Files can be fed from the web, or uploaded to Kairos's server independently
- Quotas: 25 requests/minute, 1 500 requests a day

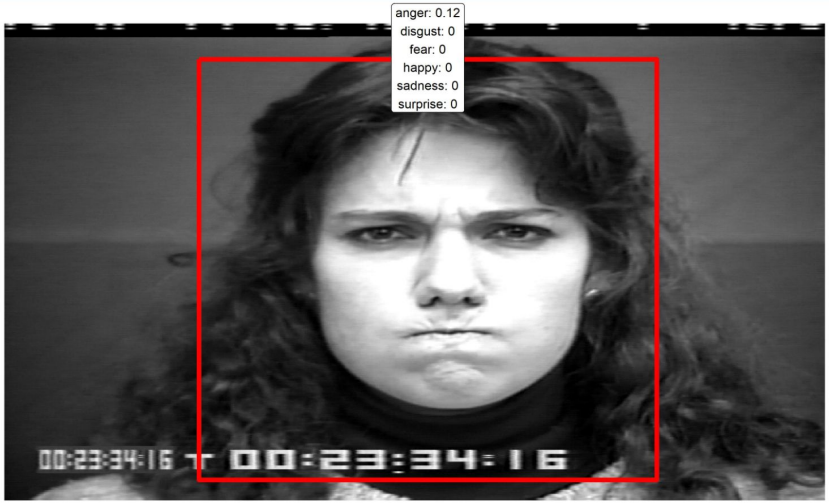
KAIROS

Get API response using the Kairos API

```
get_emotions_kairos <- function(img_url, api_id, api_key){
    req_url <- paste0(api_endpoint_url,
                      img_url %>% URLencode(reserved = TRUE),
                      uri_parameters)
    api_response <- POST(url = req_url,
                         content_type('application/json'),
                         add_headers(.headers = c( "app_id" = api_id,
                                                   "app_key" = api_key)))
    response_list <- content(api_response)
    if (response_list$frames[[1]]$people %>% length > 0){
        df <- tibble(face_id = map_int(response_list$frames[[1]]$people, "person_id")+1) %>% # one indexing
              bind_cols(map_df(response_list$frames[[1]]$people, "face"),
                        map_df(response_list$frames[[1]]$people, "emotions")) %>%
              mutate(xmin = x,
                     xmax = x + width,
                     ymin = y, # It is calculated from the top left
                     ymax = y + height) %>%
              select(face_id, xmin:ymax, anger:surprise) %>%
              gather(emotion, value, -(face_id:ymax)) %>%
              mutate(value = value %>% divide(100) %>% round(2))
              return(df)
       } else {tibble()} # Sometimes the API does not recognize faces, in that case, return empty tibble
}
```

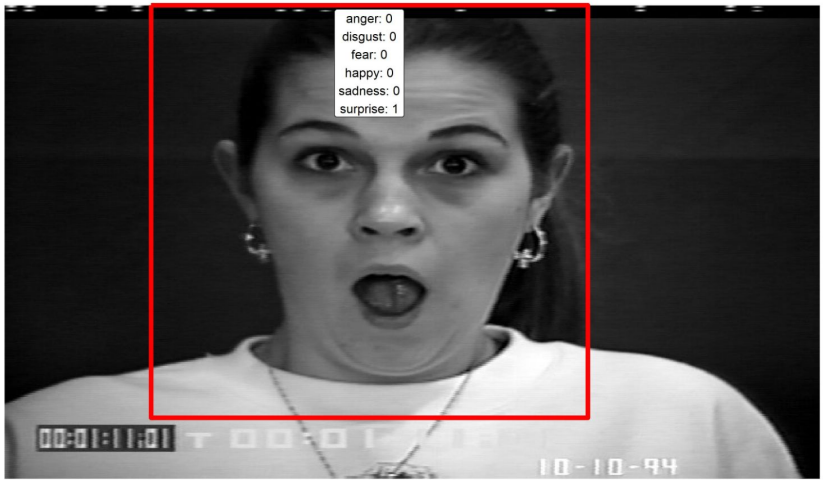


Joy/happiness

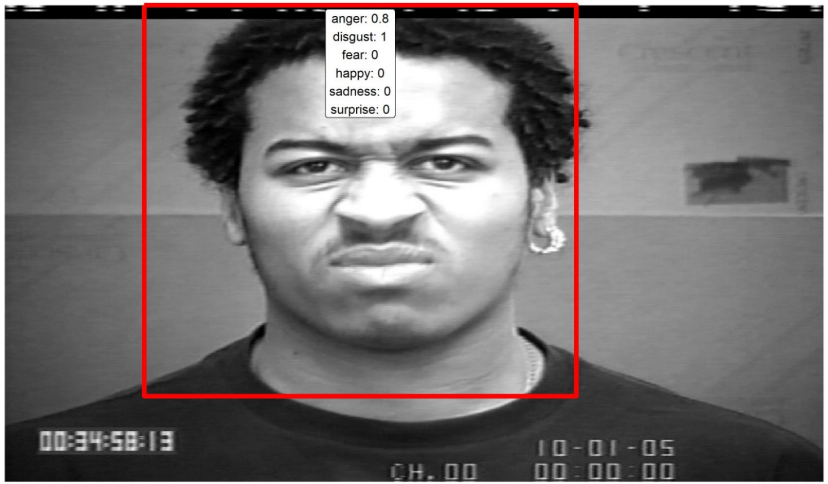


Anger

Sadness



Surprise

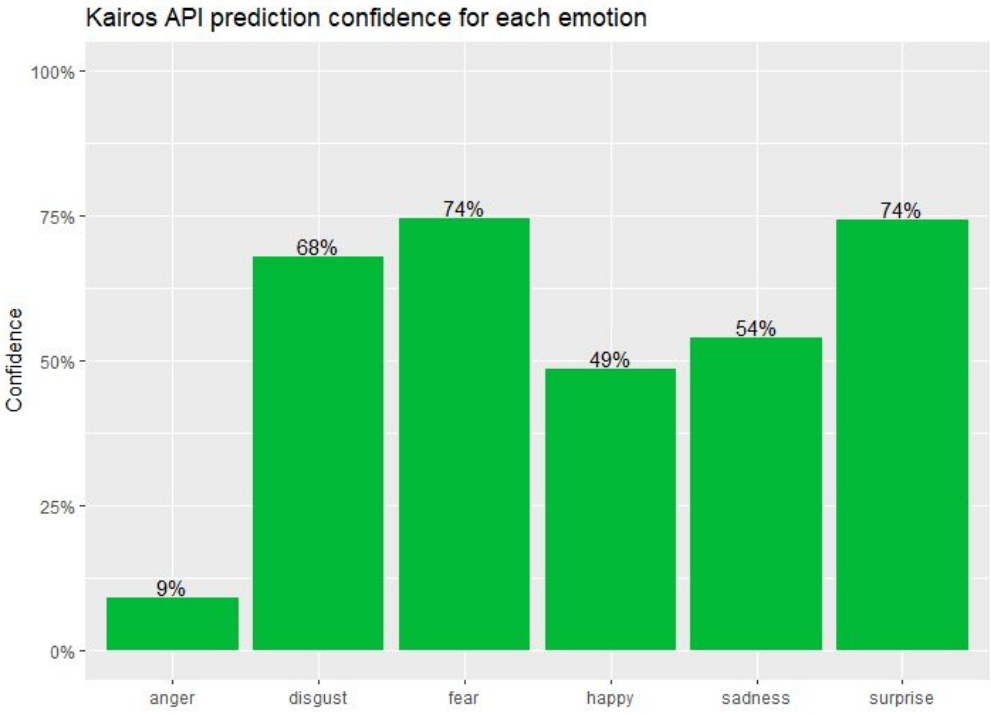


Disgust

Fear

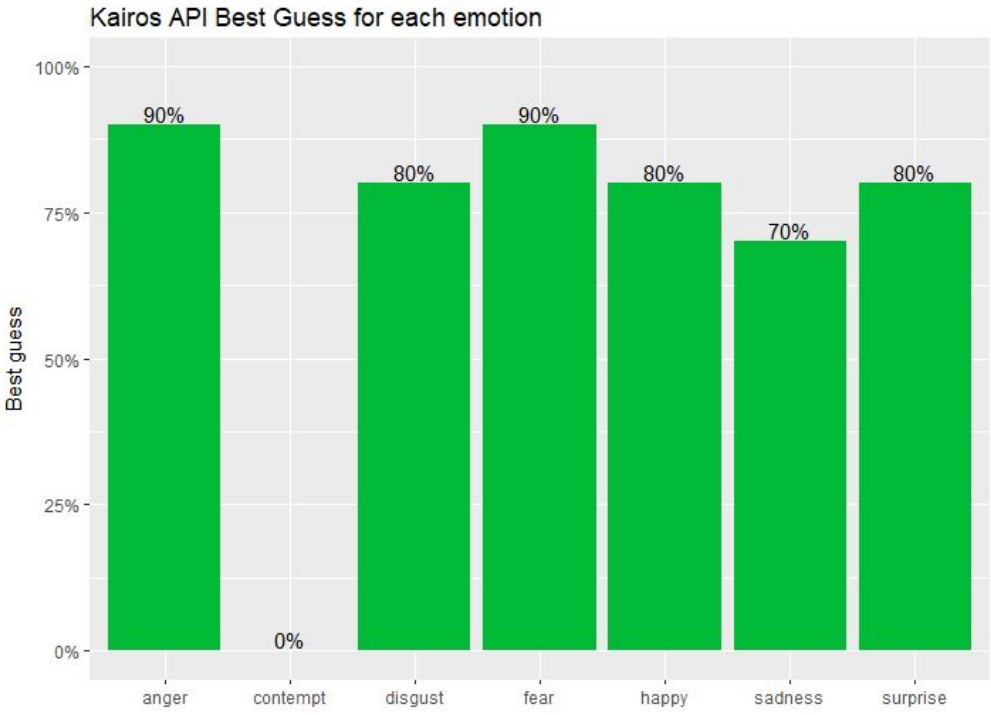
Average prediction confidence

- The confidence of the model is not very high for any emotions
- Confidence is only very low for anger. However it was still the highest confidence for angry faces, therefore it was appropriately classified



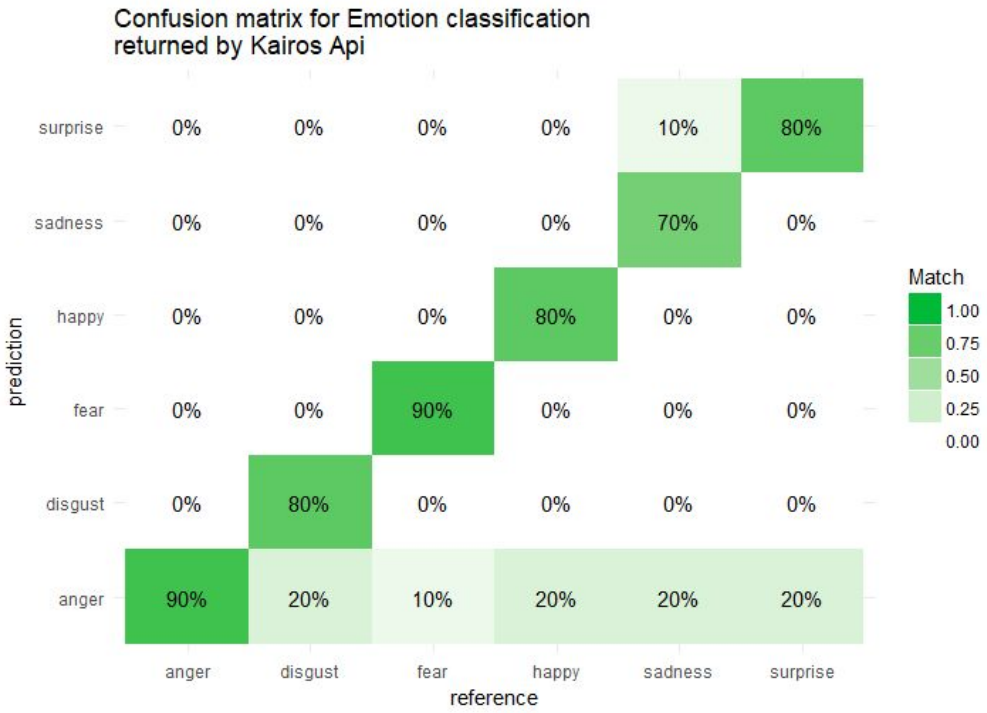
Classification accuracy

- Most emotions were recognized well
- None of the emotions were recognized in all pictures
- Contempt is not measured by Kairos



Confusion matrix

- Solid results for all emotions, if we consider the “best guess”



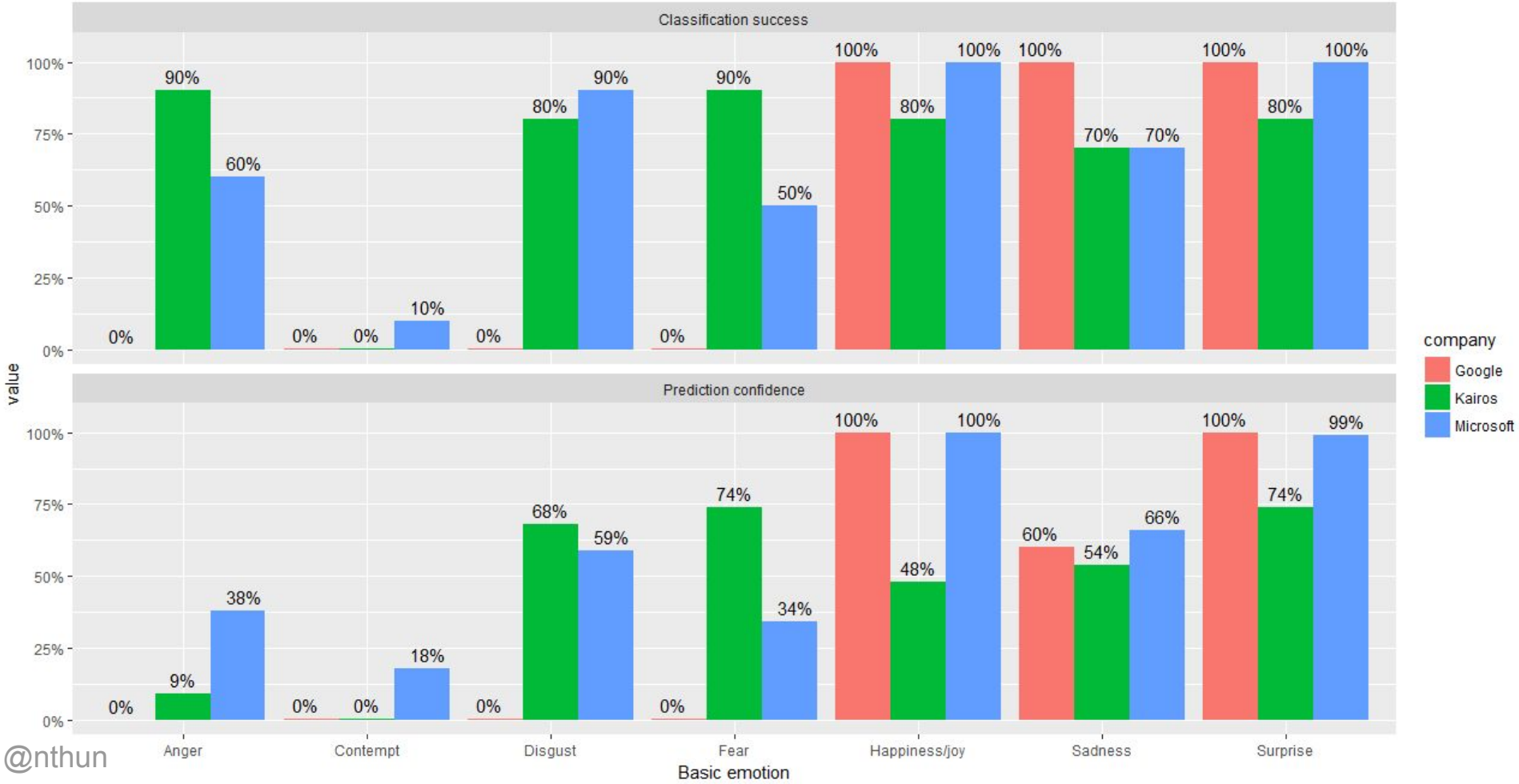
Kairos API

- The easiest to set up an account
- Not always recognizes faces, especially those that are not frontal
- Faces per picture is very limited (2?)
- Best results for emotion classification
- Can analyze videos
- Great API documentation

KAIROS

Comparison of APIs

Based on Classification success ("Best guess") and
Confidence (average confidence of predictions)



@nthun

Comparison of features and limits

	Google Cloud Platform	Microsoft Cognitive Services	KAIROS
Maximum recognized number of faces on a single picture	5	64	2
Can analyze videos by default	No	Yes	Yes
Number of recognized emotions	4	7	6
Continuous predictions	No	Yes	Yes
Independent predictions	Yes	No	Yes
Upload files directly	Yes	Yes (No)	Yes
R wrapper available	Yes	No	No
Limits: requests per minute (free plan)	1000	20	25
Limits: requests per month (free plan)	20M	30K	~45K

Summary

- There are several great APIs that provide free facial emotion analysis
- Data can be retrieved directly from R
- Emotional classification is just one factor to consider

Possibilities, future directions

- Analyze videos frame by frame, even when video feed is is not supported by the API (use ffmpeg to cut the video frame-by-frame)
- By getting data about facial expressions into R can help you to make analysis about any video, e.g. films, youtube videos, public announcements, political speeches, debates, etc.
- Combine the best capabilities of all APIs (segmenting images using MACS and ffmpeg, and recognizing emotions using Kairos)
- Analyze live video feed (Microsoft)
- Build applications that can read facial emotions

Yes, we happy Jules!