# Detecting Facial Expressions in Video Stills from Professional Tennis Matches

*Stephanie Kobakian, Mitchell O'Hara-Wild, Dianne Cook, Stephanie Kovalchik*

*20 September 2017*

## Abstract

This paper examines the effectiveness of facial detection APIs on broadcast video stills of Australian Open tennis matches. The goal is to determine the best API to use for face detection of players throughout a match. For training purposes faces were manually tagged in 6406 images, recording the scene characteristics and features of individual faces. This included information regarding accessories, such as headwear or sunglasses being worn. This enables the performance to be assessed based on conditions, and the APIs were evaluated on their success rate at detecting these faces. The practicality of usage was also assessed via time taken to complete detection within an image.

*** (1) Check caption sentences finish with "." (2) Check spelling (3) Check no referring to tables and figures in the text like "the table below . . . " or the "figure above . . . " (4) Check all the writing is in third person, remove "We then . . . " (5) One sentence paragraphs should be very rarely used, and I suspect there are too many in the current draft. Check if the orphan sentence better belongs with the prior or after paragraph.

## 1 Introduction

Many tennis professionals believe that tennis is a game heavily influenced by the mental states of the players. The opportunity for researching this "inner game" presents itself with the hope of improving the performance, and coaching of, tennis players by improving their "mental game". By statistically analyzing the faces and expressions of players during a match insight may be gained into the effects of the mental state on the outcome of a match. The aim of this project is to develop methods to collect accurate information about the facial expressions of elite tennis athletes during match play.

The performance of several popular facial recognition software's through their Application Programming Interfaces (APIs) is evaluated based on their performance on still images derived from broadcast videos of elite tennis matches. While it is difficult to know the thoughts and feelings of a player during a match, analysts may be able to gain information through results produced by recognition softwares. This approach to understanding player's emotions during a match differs to previous standards that have used player's recollections after a game to understand their emotions.

Making use of the recognition software's currently available presents a challenge as high performance sports are not the intended use. Their capabilities are often limited to their intended security and surveillance purposes. Barr (2014) addresses the 'lack of robustness of current tools in unstructured environments' that this paper faces and applies to a sports environment.

The aims of the present study were to determine the feasibility of using currently available APIs for extracting facial information of players during broadcasts of professional matches by comparing the performance of several popular facial recognition APIs. A limited selection of accessible APIs was chosen based on their ability to produce appropriate and useful facial recognition. The performance was evaluated against manual classifications obtained in an annotation tool developed by the authors.

# 2 Methodology

In this study any reference to a 'face' is considered to be an area designated manually or by an API as an area that encloses a human face. These may or may not be actual faces.

## 2.1 Sample and sampling approach

Images from the Australian Open 2016 were provided by Tennis Australia, with goal being that the sample specified below is representative of the video files to be used for future facial recognition analysis:

- 6406 images, 800x450px

To produce the set of 6404 images, 5 minute segments were taken from 105 video files a still shot of the video was taken at every three seconds, for the length of each segment. The video file were the broadcast of the tennis Matches shown on the Seven Network during the Australian Open 2016. The sample included an equal amount of females and males singles tennis matches. The rounds of the competition vary so as to not limit the pool of players to only those who progressed, though there was a higher chance of advancing players reappearing.

The sample included images that contained the faces of many people, such as players, staff on the court and fans in the crowd. All of these faces were included in the manual annotations as they were likely to be found by the software selected. It was decided that including these additional faces would allow better evaluations of the software's capabilities, and provide information to differentiate between players and other faces captured. Therefore the sample was not filtered at this initial stage.

Matches played during the Australian Open are played on a range of courts available at Melbourne Olympic Park. The sample was selected to be representative of the seven courts that have the Hawk Eye technology enabled.

## 2.2 Software selection

The initial software to be considered was informed by a report that reviewed 'commercial off-the-shelf (COTS) solutions and related patents for face recognition in video surveillance applications' (Gorodnichy, D, Granger, E, & Radtke, P, 2014, 3).

The selection criteria included the availability, speed, feature selection and whether images and/or videos could be presented for detection. The results of Gorodnichy, Granger and Radtke's (2014) report considered processing speed, feature selection techniques, and the ability to perform both still-to-video and video-to-video recognition.

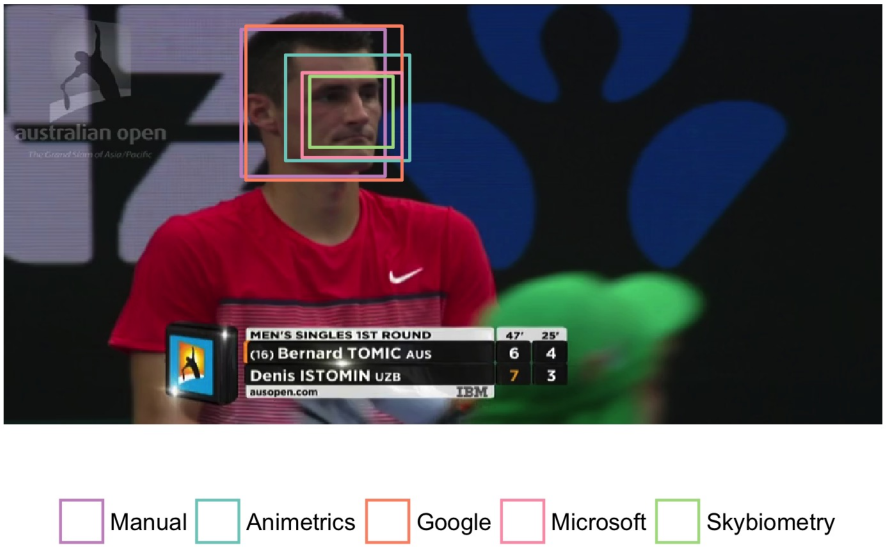


Figure 1: This image of Bernard Tomic was chosen as a trial image to be presented to each of the software before they were included in the research. Each colour represents a different API source. It was expected that the software would be able to find this face, despite the player facing away from the camera.

The report outlined that Animetrics (Animetrics Inc 2016) required an 'image/face proportion should be at least 1:8 and that at least 64 pixels between eyes are required'.

SkyBiometry (Skybiometry 2016) is a 'spin-off of Neurotechnology' which was considered by Gorodnichy et al. an API that also allows for both detection and recognition.

Companies who have recently expanded their API ranges were considered. Microsoft API(Microsoft Cognitive Services 2016), provided by Microsoft Cognitive Services, and Google Vision API (Google Cloud Platform 2016).

Online demos were used to test viablity, Figure 1 depicts Bernard Tomic and the detected areas found.

### 2.3 Manual annotations

A web based annotation tool was developed to allow image annotations that would capture the location of areas selected manually, and annotation of attributes for each face, and scene.

Specific information for each face within each scene was collected. To determine which of the, sometimes many, faces in the scene it would be reasonable for software to detect a standard was created for reasonable detection.

The faces of players were recorded if it showed their face at a minimum of 20 by 20 pixels. The back of the head was not detected as a face by any software, these areas were classified manually but reclassified as other. Crowd faces were not the intended targets of the recognition however these faces contributed to our understanding of the software. The same face size standard applied to crowd members, but focus was placed on the most prominent faces. For each of these faces,

Table 1: Characteristics recorded for each image.

| Attribute | Choices |
|---|---|
| Graphic | Live Image, 2D Graphic |
| Background | Crowd, Court, Logo Wall, Not Applicable |
| Person | Yes, No |
| Shot Angle | Level, Birds Eye, Upward |
| Situation | Match play, Close up, Not player, Crowd, Off court, Transition |

Table 2: Characteristics recorded for each face. Most were easy to assess for each face, with the exception of obscured and head angle.

| Attribute | Choices |
|---|---|
| Detectable Person | Player, Staff, Fan, Not Applicable |
| Obscured Face | Yes, No |
| Lighting | Direct Sunglight, Shaded, Partially Shaded |
| Head Angle | Front On, Back of Head, Profile, Other |
| Glasses | Yes, No |
| Visor or Hat | Yes, No |

information was collected regarding the attributes in Table 2.

The web app was created using Shiny (Chang et al. 2016). This has been expanded into an R package called taipan. A tool for annotating images in preparation for analysis.

If there was a face in the image the annotator was able to highlight a section of the image to create a square 'Face Box'. This selection presented a set of Attributes questions to answer, and allowed information to be recorded for the face in the specific 'Face Box'. This recorded the x and y coordinates of the box drawn, and saved all the selections and the 'Face Box' coordinates to a CSV file.

When a face was not visible in the scene only the scene attributes applied for that image.

All the annotations for this sample were completed by one annotator to provide a consistent sample of faces annotated manually. The initial decisions of what would be reasonably detected was made by several people.

### 2.4 Software usage

The face recognition software allowed for access through a R (R Core Team 2013) script, calling the APIs using the httr(Wickham 2016b) package. The scripts looped through the images, individually posting a request to retrieve information provided and convert it into an appropriate format for analysis.

Special handling was required due to the limits on requests per minute of Skybiometry and Microsoft, time-controlling the requests was incorporated for these APIs.

Figure 2 displays the processing time required by the APIs for each image, as a letter value plot (*** add in reference to Heik'e paper). A letter value plot is like a boxplot, but more suited to skewed data like this. (Both Microsoft and Google have a few eimages each where they have taken a long time to complete and these were removed from this plot.) The times are quite variable between APIs, with Skybiometry being the most consistently fast. Microsoft and Animetrics are quite variable regardless of the number of faces found. Google is variable in time taken when there are few faces present, and then takes a little longer on average to find faces as the number of them in the image increases.

It should be considered that the time taken may be an issue for 'real time' processing. It would depend on the amount of frames sampled. This also does not account for the time taken to process the information returned.

### 2.5 Data processing

The data needed for our analysis was organised by source, separate files for each of the four APIs. They contained the information on the location of the faces found in the images and the time taken to find them. Some APIs also provided detailed information such as the estimated head angle, and locations of specific facial features.

The manual tagging app resulted in two files, one containing information about the image, and the faces.

### 2.6 Results

The manual tagging helped to provide a benchamark for faces found by APIs. The face detection rate indicated what attributes of an image or face were associated with face detection by an API. Faces that were found by multiply APIs indicated the industry expectations for face presentation. This application allowed for a broader range of angles, backgrounds and accessories that may have impacted detection. False discoveries were the instances of faces found by APIs that did not intersect with manually tagged faces. Some of these may actually have been faces or reveal sensitivities in the algorithms of the APIs. The analysis of image characteristics and accessories indicates whether the APIs were viable choices for detecting faces in sports environments.

#### 2.6.1 Modeling detection rate

A logistic regression model was used to assess the characteristics of the face and scene affecting detection, with stepwise variable selection. The binary response was whether the face was detected or not, using the manually tagged data as the benchmark, and characteristics included the setting, angle of shot and player, background, lighting and player adornments. (*** Any other variables considered but not selected for any model?) A separate model was fit for each API.

*** The number of digits should be consistent, e.g. "-1" should be "-1.00". I don't know how to force this with the kable printing, but one way or another we have to.

For the most part, the APIs were all affected by the same factors. Those that negatively affected all APIs were glasses, visor or hat. Background only negatively affected Google's detection rate.

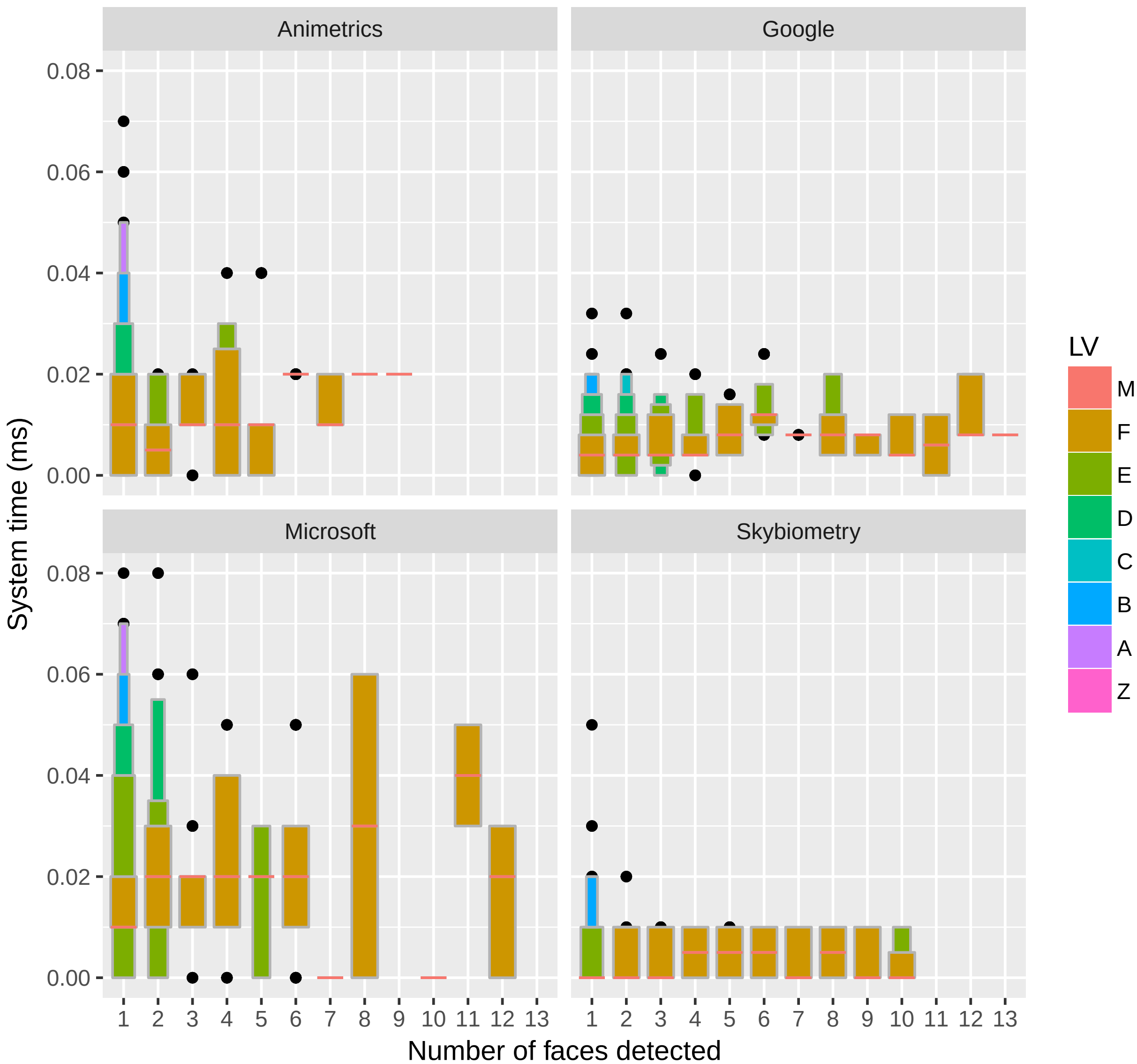


Figure 2: System time taken by the four APIs to finish searching individual images, by number of faces discovered, displayed as a letter value plot. Skybiometry is the most consistently fast.

Table 3: Evaluation of factors affecting face recognition for API, as assessed by logistic regression model. Coefficients from the four models shown (significance: ** 0.01 * 0.05 . 0.01 - NS). Most APIs wre affected by similar factors, with glasses, visor or hat, head angle, lighting, shot angle and situation all contributing significantly to face detection.

| variable | A | | G | | M | | S | |
|---|---|---|---|---|---|---|---|---|
| bgCrowd | - | | -1.22 | * | - | | - | |
| bgLogo wall | - | | - | | - | | - | |
| glassesYes | -1.05 | ** | -0.72 | ** | -0.97 | ** | -0.88 | ** |
| headangleOther | -1 | ** | -0.49 | * | -1 | ** | -1.09 | ** |
| headangleProfile | -4.19 | ** | -0.85 | ** | -4.25 | ** | -4.09 | ** |
| lightingShaded | -0.55 | ** | -0.78 | ** | -0.65 | ** | -0.88 | ** |
| obscuredYes | -0.76 | ** | -0.78 | ** | -0.75 | ** | -0.63 | ** |
| shotangleBirds Eye | 0.7 | . | 1.79 | ** | - | | - | |
| shotangleLevel | - | | 1.33 | ** | 0.87 | . | 1.01 | . |
| shotangleLevel:bgCrowd | - | | 1.92 | ** | - | | - | |
| shotangleLevel:bgLogo wall | - | | 1.24 | . | - | | - | |
| shotangleUpward | - | | 2.7 | ** | - | | - | |
| shotangleUpward:bgCrowd | - | | - | | - | | - | |
| situationCrowd | 0.62 | . | 1.74 | ** | 0.7 | . | 0.67 | . |
| situationMatch play | -1.54 | ** | -2.53 | ** | -2.68 | ** | -1.24 | ** |
| situationNot player | 1.21 | . | 2.25 | * | - | | - | |
| situationOff court | 0.61 | * | - | | - | | - | |
| visorhatYes | -1.2 | ** | -0.94 | ** | -0.75 | ** | -0.76 | ** |

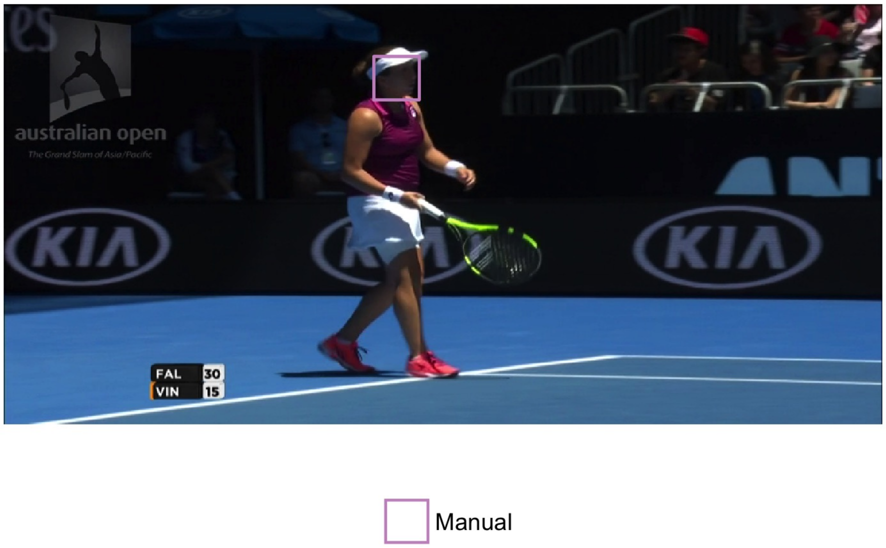


Figure 3: The face captured manually in this image was not captured by any API. The head angle, shot angle, lighting, wearing the visor and being captured across the court during the match all contribute to the struggle of the APIs.

Situation had different effects on each API. Results for Google were most interestng in the shotangle had differing effects on the detection.

When forcing no intercept, all variables are significant at the 0 level, and obscured is only significant at the 0.001 level.

### 2.6.2 Detection rates

Each API returned areas in the image that indicated the face location. These are defined using the four points, marking the four corners of a rectangle. Figure 4 shows a barchart of the number of faces detected by each API. The Google Vision API detected largest number of faces, much more than the other three APIs, and most closely matched the number recorded by manual tagging.

### 2.6.3 API comparison

Table 4 cross-tabulates the faces found by each API relative to the manually tagged faces. For a perfect match with the manually tagged faces, the off-diagonal elements should be zero. Google has the closest match to the manual tagging, and Skybiometry, which was the fastest to run, performed very badly. It is interesting to see the high proportion of faces detected by the APIs that were not considered to be faces by manual tagging.

The mismatches were manually examined, and showed that the 638 potential faces found by Google but not tagged manually were actually faces, but were crowd members. They would need to be used, for the long term purpose of the study, to detect and tag players' emotions during a match.

Comparatively, visual inspection of the 527 potential faces that did not match manually annotated

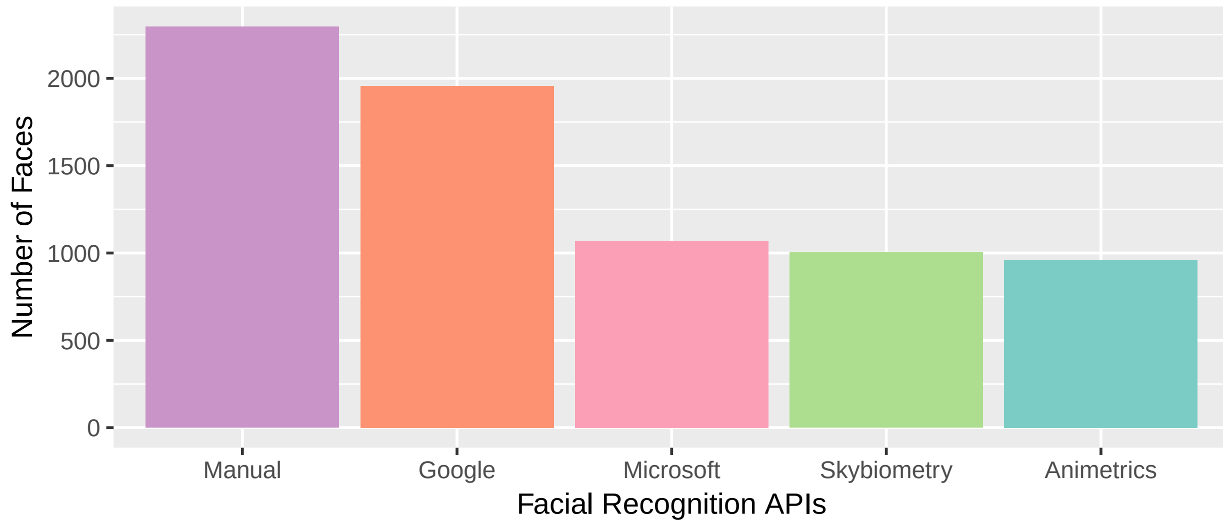


Figure 4: The bar chart of the number of faces detected by each API. Google's Facial Recognition API recognized almost 1000 more faces than the next best API, Microsoft.

faces showed the Animetrics results contained many potential faces that were actually very unusual results, which are discussed later.

Table 4: Contingency tables of each API capture against manual tagging. Interestingly all APIs reported faces that would not be considered faces by manual inspection.

| Animetrics | | |
|---|---|---|
| Manual | Face | No Face |
| Face | 434 | 1861 |
| NoFace | 527 | 787 |

| Google | | |
|---|---|---|
| Manual | Face | No Face |
| Face | 1319 | 976 |
| NoFace | 638 | 676 |

| Microsoft | | |
|---|---|---|
| Manual | Face | No Face |
| Face | 565 | 1730 |
| No Face | 505 | 809 |

| Skybiometry | | |
|---|---|---|
| Manual | Face | No Face |
| Face | 493 | 1802 |
| No Face | 512 | 802 |

*** I've decided that these should be proportions.

Figure 5 is a set visualisation to examine the intersection of the APIs, and the manual annotations. This is produced with the UpSetR (Gehlenborg 2017) package. The black bubbles below the bars indicate the API combination that corresponds to the bar count. The largest count comes from manual tagging, followed by Google and manual tagging, and then Google. Faces detected by all four API and manual annotations were the fourth largest count. This says that the APIs agreed on about a quarter of the faces from the manually tagged collection. The fifth group was the most surprising – these were faces captured by Animetrics alone, that were false discoveries.

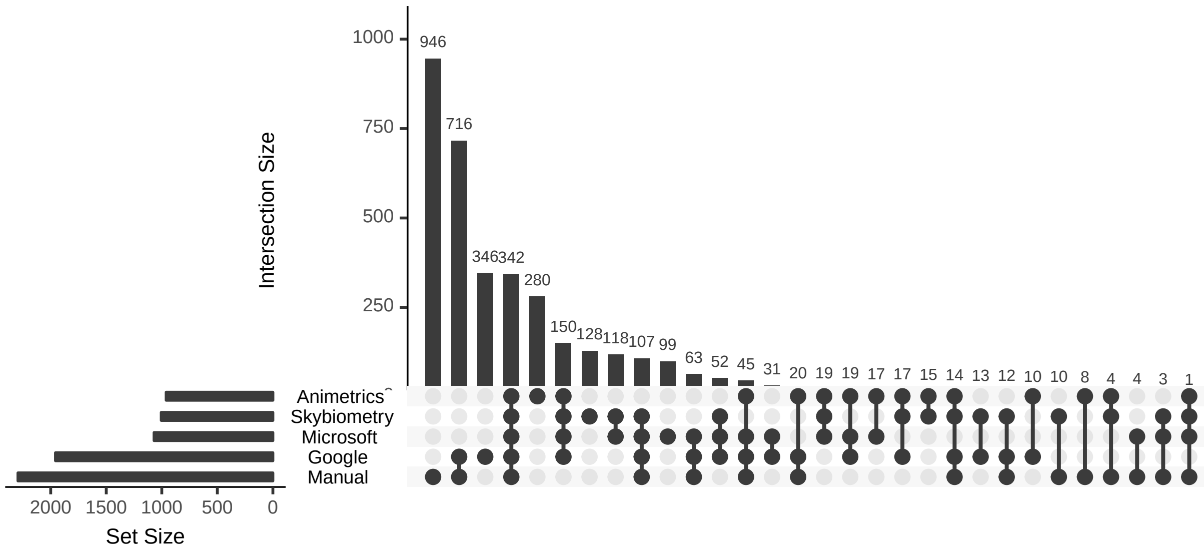


Figure 5: Faces Per API Combination. The Bar Chart shows the faces that were recognised by multiple API or found manually. The largest group, with 809 faces, is faces only found by Manual annotations. The following group were the 716 faces recognized both Manually and by the Google API. These combinations may give some indication as to the circumstances when some APIs perform better than others.

Table 5: Combinations of image attibutes that are most common in the image set. The combination of the Logo Wall background and the Level angle are shared by three of the five image combinations.

| situation | bg | shotangle | detect | count |
|---|---|---|---|---|
| Close-up | Logo wall | Level | Player | 720 |
| Match play | Logo wall | Level | Player | 238 |
| Crowd | Crowd | Upward | Fan | 149 |
| Close-up | Court | Birds Eye | Player | 133 |
| Close-up | Logo wall | Level | Staff | 117 |

### 2.7 False discoveries

Animetrics provided many surprises. Figure 6 shows four images and the captures where non-faces were detected as faces. In image (a) the player's back is detected as a face. In image (b) the KIA logo was reported as a face. In image (c), where logos on the player's and ballboy's shirts were returned as faces. In image (d) where there are a lot of crowd faces, if you look closely one of the detections is actully a fist (to the right on the center).

The other APIs also returned some non-faces, but the most egregious mistakes came from Animetrics.

### 2.8 Image characteristics

We then considered the characteristics of the images that the API found Potential Faces in.

We then considered that there would be an uneven amount of faces with certain image attributes.

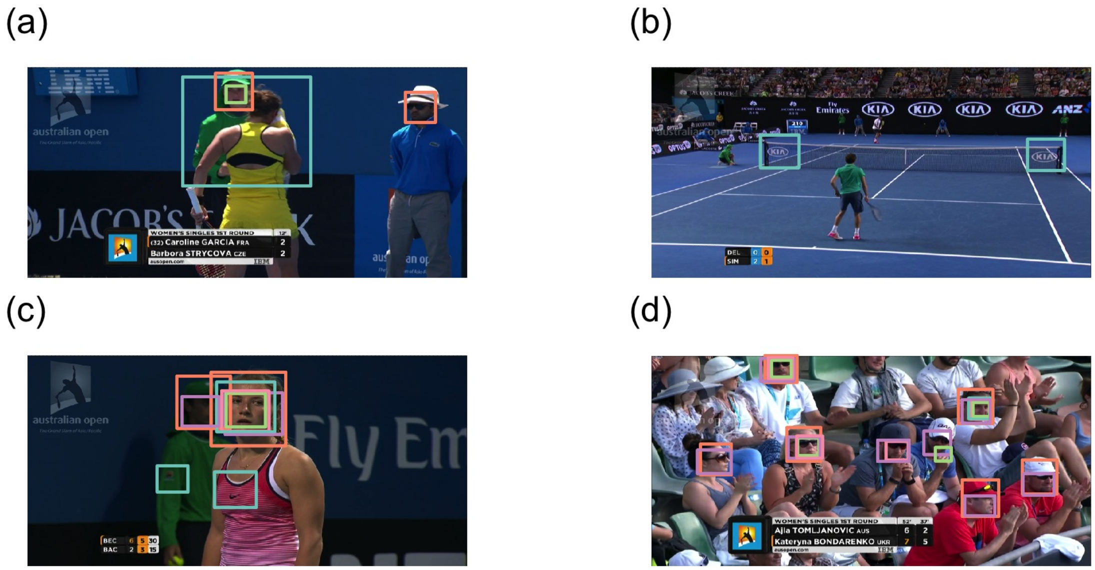


Figure 6: Animetrics provided many results that were quite unusual: (a) one of the detected face is a player's back, (b) the KIA logo on the net is detected as a face, (c) is very interesting as the player's face was found, but the shirts of both the player and ball boy were reported as faces, (d) had a lot more faces than many images but Animetrics also found a fist (to the right on the center) that it considered a face.

These relationships can be explored in mosaics for the images that were all considered manually. The scene information was recorded and the combinations were shown to find how many potential faces were found with the combination of scene attributes.

This showed that crowd members faces were often recognized, this is helpful as it shows a strong ability of Google's algorithm to recognize faces, even when these faces are not the goal of the research. It also allowed for an increase in understanding how attributes of a face or image impact on the API detection.

The colours in the mosaic (a) in Figure 7 show the proportion of faces that matched those found manually, given the angle the faces were captured from. There is a greater proportion of API faces that matched the faces found manually than those that did not match. However given the face was captured from a birds eye angle there were a lot less faces proportionally that matched those found manually. The most common faces in the set were those captured Level to player's faces, these faces were often recognised by the APIs as well.

The mosaic (b) in 7, shows the amount of Faces captured during each possible situation. It contrasts the proportion of the images captured in each situation that either did or did not match the faces annotated manually. It can be seen that the largest portion of the faces were captured in a close up situation and there were more of these faces that were found by the APIs that did match manually derived faces. This proportion is steady across most situations, from this we learn the situation may not be influencing the APIs rate of detection.

This stacked bar chart, Figure 7 Plot c, allows consideration of the interaction between two image attributes. We are able to see that the background being the Logo Wall and the angle of Level is common to the highest amount of faces. There are also no images that were taken at an upward

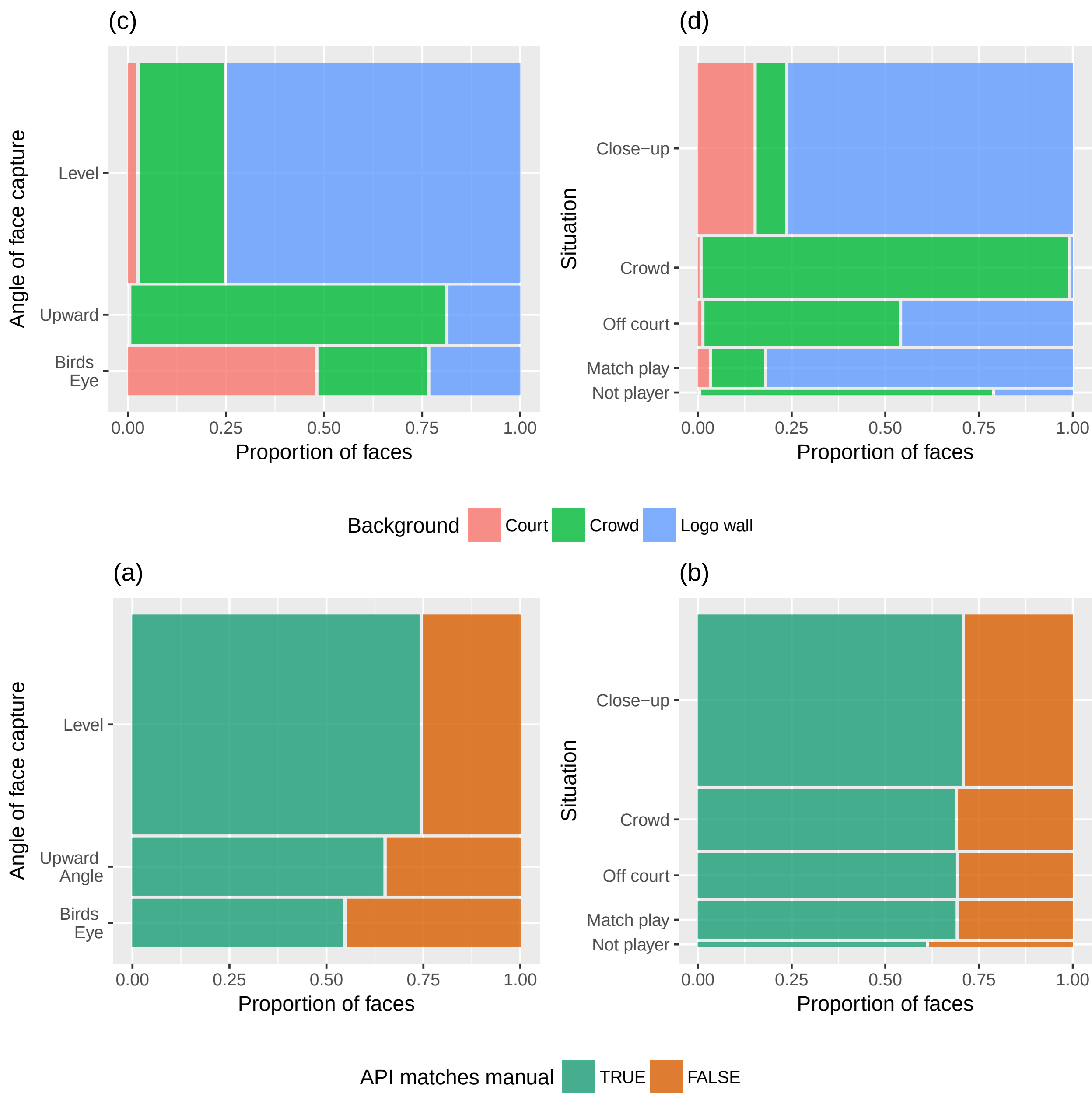


Figure 7: The mosaic plots indicate the proportion of faces given the angle, or situation, that matched faces found manually. The stacked bar charts show the amount of faces captured at certain angles, or situation.

angle with the background of the court.

This stacked bar chart, Figure 7 Plot d, allows consideration of the interaction between the background of the image captured and the situation that could possibly be occurring. As seen previously, the logo wall is the most common background, but it is never the background of an image of the crowd. The court is the background of an image only when players are captured, this occurs during close ups and while the court is in play.

### 2.9 Accessories

The use of accessories like glasses and headwear, visors or hats, was considered as the Australian Open takes place on both indoor and outdoor courts. It was assumed that outdoor courts would lead to the use of these accessories and these accessories may contribute to the performance of a recognition software.

```
##
##              No       Yes
##   No  0.5048893 0.4951107
##   Yes 0.2613636 0.7386364
```

Table 6: Proportions of faces captured manually tagged as wearing headwear given they wore glasses, or did not.

| | Head wear | No Head wear |
|---|---|---|
| Glasses | 0.50 | 0.50 |
| No Glasses | 0.26 | 0.74 |

Table 6 of the proportions tells that of the faces wearing glasses, there were a similar amount of faces captured with and without headwear. Of those without glasses only 26% wore headwear. This may be influenced by weather and sunlight on the court during Australian Ope n matches.

Table 7: Conditional proportions of faces of players wearing glasses or not, captured by the different software, in comparison to the manually tagged faces. API's unaffected glasses will have similar proportions for both categories. Animetrics had the biggest proportion difference, and thus, most affected. Google is the least affected.

| | Animetrics | Google | Microsoft | Skybiometry |
|---|---|---|---|---|
| No | 0.21 | 0.6 | 0.26 | 0.23 |
| Yes | 0.08 | 0.43 | 0.14 | 0.12 |

## 3 Future Work

The long term goal is to better understand how the emotion's felt by a player during a match affect player performance. Ultimately we would aim to create a program that automated the collection of player emotion data from throughout a match. This information would be presented in a timeline that allowed match performance, in the form of points won, to be aligned with the emotions felt at certain times throughout.

Considering the images used during our study were stills derived from Broadcast video files, it would be useful to extend further research to deal with the video files directly. The Google Vision API which produced the best recognition in images does not yet have the potential to detect faces and emotions in a video.

It should also be considered that these are software focused on providing recognition in certain controlled scenarios. If the study was controlled to focus on certain camera angles that align with the facial angles these security programs are intended to recognize faces in.

Given that Google found many faces that did not match manually annotated face, we considered that we should check for manual errors. There is the possibility that we could create another app that shows the Facial Bounding Boxes identified by each program, this would allow the annotator to confirm manually whether or not these are faces.

Given that certain scene attribute combinations produced more facial recognition than other combinations we should consider limiting the sample of images sent to Google Vision API. This would not only reduce cost but also provide a greater level of detail of the emotions felt by a player during a match. To provide a greater level of information at all points in a match it would be beneficial to derive images from a single camera feed. This feed should match the scene attributes that provided the most Google faces.

To undertake sentiment analysis, we would take the boxes of faces found in this set of images. Allowing each face a border of pixels, we would crop the images and produce an individual face image that would form the data set for emotion recognition. We also feel that incorporating audio information from the microphones worn by players may assist in sentiment analysis. By including this information we would be able to define differences between certain emotions that may not be able to be found by facial features only.


## Acknowledgements

The authors wish to thank Tennis Australia for access to the data. The analysis was conducted using R, and the following packages: bookdown(Xie 2017a), descr(Dirk Enzmann et al. 2016), EBImage (Pau et al. 2010), ggthemes (Arnold 2017), ggmosaic (Jeppson, Hofmann, and Cook 2017), grid (R Core Team 2017), gridExtra (Auguie 2016), gtable (Wickham 2016a), kfigr (Koohafkan 2015), knitcitations (Boettiger 2015), knitr (Xie 2017b) , pander (Daróczi and Tsegelskyi 2015), readr (Wickham, Hester, and Francois 2017), RefManager (McLean 2014), tidyverse (Wickham 2017), UpSetR (Gehlenborg 2017), xtable (Dahl 2016).


## Supplementary material

The APIs, the scripts used to access them, and the resulting data set are contained in the supplementary materials, available at https://github.com/mvparrot/face-recognition/paper. The APIs evaluated were:

- Animetrics: This name used throughout the paper refers to the Animetrics Face Recognition API, FaceR API by Animetrics Inc (2016).
- Google: Google Cloud Platform (2016) Refers to the Google Cloud Vision API
- Microsoft: This has been used to refer to the Microsoft Azure Cognitive Services Face API, published by Microsoft Cognitive Services (2016).
- Skybiometry: Skybiometry (2016) References the spin off detection and recognition software of Neurotechnology.

Data files provided are:

- ManualClassifiedFaces.csv resulted from the use of the Manual Annotation Application, it contains information about each face identified manually.
- ManualClassifiedScenes.csv also resulted from the use of the app, it contains information about each image.

# 4 References


Animetrics Inc. 2016. "Animetrics Face R Api." July. http://api.animetrics.com/documentation.

Arnold, Jeffrey B. 2017. *Ggthemes: Extra Themes, Scales and Geoms for 'Ggplot2'.* https://CRAN.R-project.org/package=ggthemes.

Auguie, Baptiste. 2016. *GridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Barr, Bowyer, J. 2014. "The Effectiveness of Face Detection Algorithms in Unconstrained Crowd Scenes." *IEEE Winter Conference on Applications of Computer Vision (WACV)*. doi:10.1109/WACV.2014.6835992.

Boettiger, Carl. 2015. *Knitcitations: Citations for 'Knitr' Markdown Files*. https://CRAN.R-project.org/package=knitcitations.

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. 2016. *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Dahl, David B. 2016. *Xtable: Export Tables to Latex or Html*. https://CRAN.R-project.org/package=xtable.

Daróczi, Gergely, and Roman Tsegelskyi. 2015. *Pander: An R Pandoc Writer*. https://CRAN.R-project.org/package=pander.

Dirk Enzmann, Jakson Aquino. Includes R source code and/or documentation written by, Marc Schwartz, Nitin Jain, and Stefan Kraft. 2016. *Descr: Descriptive Statistics*. https://CRAN.R-project.org/package=descr.

Gehlenborg, Nils. 2017. *UpSetR: A More Scalable Alternative to Venn and Euler Diagrams for Visualizing Intersecting Sets*. https://CRAN.R-project.org/package=UpSetR.

Google Cloud Platform. 2016. "Google Cloud Vision Api." July. https://cloud.google.com/vision/

docs/.

Gorodnichy, D, Granger, E, & Radtke, P, 2014. "Survey of Commercial Technologies for Face Recognition in Video." Division Report 22. DRDC Centre for Security Science, Ottawa ON Canada K1A 0L8: Canada Border Services Agency.

Jeppson, Haley, Heike Hofmann, and Di Cook. 2017. *Ggmosaic: Mosaic Plots in the 'Ggplot2' Framework*. https://CRAN.R-project.org/package=ggmosaic.

Koohafkan, Michael C. 2015. *Kfigr: Integrated Code Chunk Anchoring and Referencing for R Markdown Documents*. https://CRAN.R-project.org/package=kfigr.

McLean, Mathew William. 2014. *Straightforward Bibliography Management in R Using the Refmanager Package*. http://arxiv.org/abs/1403.2036.

Microsoft Cognitive Services. 2016. "Microsoft Face Api." July. https://docs.microsoft.com/en-us/azure/cognitive-services/face/.

Pau, Gregoire, Florian Fuchs, Oleg Sklyar, Michael Boutros, and Wolfgang Huber. 2010. "EBImage—an R Package for Image Processing with Applications to Cellular Phenotypes." *Bioinformatics* 26 (7): 979–81. doi:10.1093/bioinformatics/btq046.

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

———. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Skybiometry. 2016. "Skybiometry Api." July. https://skybiometry.com/documentation/.

Wickham, Hadley. 2016a. *Gtable: Arrange 'Grobs' in Tables*. https://CRAN.R-project.org/package=gtable.

———. 2016b. *Httr: Tools for Working with Urls and Http*. https://CRAN.R-project.org/package=httr.

———. 2017. *Tidyverse: Easily Install and Load 'Tidyverse' Packages*. https://CRAN.R-project.org/package=tidyverse.

Wickham, Hadley, Jim Hester, and Romain Francois. 2017. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Xie, Yihui. 2017a. *Bookdown: Authoring Books and Technical Documents with R Markdown*. https://github.com/rstudio/bookdown.

———. 2017b. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.name/knitr/.