

Detecting Facial Expressions in Professional Tennis Matches

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

28 July 2017

Contents

1	Abstract	1
2	Introduction	2
2.1	Sample and sampling approach	2
2.2	Software Selection	3
3	Methodology	4
3.1	Manual Annotations	4
3.2	Manual Annotation	5
3.3	Software Interaction	6
3.4	Data Processing	6
3.5	Analysis	7
4	Results	9
4.1	Amount of Faces	9
4.2	Comparing Faces Found	9
4.3	False Discoveries	11
4.4	Image Characteristics	11
4.5	Accessories	12
5	Future Work	13
6	Acknowledgements	15
6.1	APIs	15
6.2	Data Files	15
6.3	Packages	16
	References	16

1 Abstract

This report aims to analyse the application of facial detection APIs to broadcasted video of Australian Open tennis matches to find a suitable API for face detection of players throughout a match. Four APIs are contrasted based on the amount of faces returned, and whether these faces matched those found manually. Considerations of whether using a broadcast video of a sport is appropriate is based on the characteristics of the images in the sample. The characteristics of the scenes captured and the features associated with individual faces casts light on what would be most appropriate for

accurate and high volume facial detection to occur. Situations where certain APIs are less capable are highlighted, should applications require reliable results in these situations the shortcomings in time taken and performance under less than optimal conditions would present certain APIs as inapplicable.

2 Introduction

Many tennis professionals believe that tennis is a game heavily affected by the mental states of the players. The opportunity for researching this “inner game” presents itself with the hope of improving the playing and coaching of tennis players by improving their “mental game”. By statistically analyzing the faces and expressions of players during a match there is a hope that insight may be gained into the effects of the mental state on the outcome of a match. Facial expressions during competition may provide an insight into a player’s thoughts. The aim of this project is to begin to develop methods to collect accurate information about the facial expressions of elite tennis athletes during match play.

In this report, we investigate the performance of several popular facial recognition software’s through their Application Programming Interfaces (APIs), and evaluate their performance when applied to the broadcast videos of elite tennis matches. Using the broadcast videos gives an objective insight to emotions as the player progresses through a match. While it is impossible to know the thoughts and feelings of a player during a match, professionals may be able to infer this information through results produced by a recognition software. This approach opposes the previous studies that have used player’s recollections after a game to determine 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. This limited selection was based on accessible APIs believed to produce appropriate and useful facial recognition. The performance of the evaluated software was compared against manual classifications obtained by an annotation tool developed by the authors. In addition to looking at the overall performance, image factors that influences the performance of each service were also investigated.

2.1 Sample and sampling approach

The goal of the sample was to be representative of the video files that will be used for future facial recognition analysis.

- 6406 Australian Open images (2.8GB)
- 800x450px size frames from 105 match broadcast videos
- Video frames taken every 3 seconds over a 5 minute segment

The sample consisted of a set of 6404 still images. To produce these images, a still shot of the frame was taken at every three seconds, for the length of each 5 minute segment. The stills were provided

by Tennis Australia for use in this research, these segments were taken from 105 video files, which were the broadcast of the tennis Matches shown on the Seven Network during the Australian Open 2016. The sample included an equal amount of singles tennis matches played between females and males. The rounds of the competition vary 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, this included players, staff on the court and fans in the crowd. These faces were included in the manual annotations as they were likely to be found by the software selected. We felt including these additional faces would not only increase the sample by which to judge the software’s capabilities but also allow provision of information on how to differentiate between players and other people for further research. Therefore the sample was not filtered at this initial stage.

There are many matches played during the Australian Open, and they are played on the range of courts available at Melbourne Olympic Park. Therefore the sample was selected to be representative of the seven courts that have the Hawk Eye technology enabled.

2.2 Software Selection

The choice of the initial software considered for this research was informed by a report that reviewed ‘commercial off-the-shelf (COTS) solutions and related patents for face recognition in video surveillance applications.’ (D. Gorodnichy 2014)

The process of software selection to determine which we would compare was based on several criteria. The results of the report, as it considered processing speed and feature selection techniques, as well as the ability to perform both still-to-video and video-to-video recognition.

From the software analysed we considered availability for use within the time frame of the report. This led us to choose Animetrics FaceR. The report outlines that for Animetrics, ‘one requirement is that image/face proportion should be at least 1:8 and that at least 64 pixels between eyes are required’. We realize this could present challenges given the pixel sizes of faces in the image set. It will also allow for an extension from detecting to recognizing people in the data set.

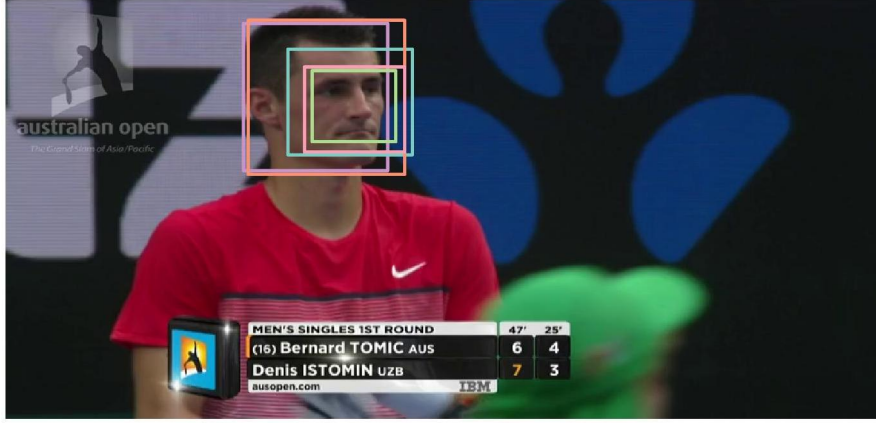
After considering several other off-the-shelf products, we did not choose any other software’s from those analysed as they were not as readily available as other products on the market.

This led to SkyBiometry, an API that also allows for both detection and recognition. The cloud-based software as a service, is a ‘spin-off of Neurotechnology’, a software considered by the report.

We then chose to consider companies who are expanding their API ranges. This resulted in the choice of Microsoft API, provided by Microsoft Cognitive Services. This detects faces and return a square area where the face was located, and predicts facial features. It also allows the possibility of video stream detection.

The final software we chose to analyse was Google Vision API. Due to Google’s expansion in many web based solutions we searched for a facial recognition software.

We were able to try the online demos to see whether these software were viable, we used the following Image displayed in Figure 1:



Manual
 Animetrics
 Google
 Microsoft
 Skybiometry

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. It was expected that the software would be able to find this face, despite the player facing away from the camera.

Table 1: This table lists the possible image descriptions that are associated with the attributes of each image. The most appropriate description from each list was selected. There are more options for more complex attributes, there were five expected situations the image may depict.

Attribute	Choices
Graphic	Live Image, 2D Graphic
Background	Crowd, Court, Logo Wall, Not Applicable
Person	Yes, No
Shot Angle	Level With Players, Birds Eye, Upward Angle
Situation	Court in Play, Court Player Close-Up, Court Close-Up Not Player, Crowd, Off Court Close Up of

3 Methodology

A ‘face’ is considered to be any area designated manually or by an API as a location that encloses a human face. These may or may not be actual faces.

3.1 Manual Annotations

An annotation tool was constructed to allow image annotations that would capture the location of areas selected manually. We refer to the faces in these areas as Manual Face Classifications. This tool also allowed selections of attributes of each face, and attributes describing the features of the Scenes.

The aim was to collect specific information on each different face within the scene. 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.

Table 2: This table lists the possible face descriptions that were appropriate for each individual face. The most appropriate description for each face was built by selecting one option from each list describing a particular attribute of the face that was selected. Most of these categories selections were made obvious in the image, Obscured and Head Angle were more difficult to choose for some faces.

Attribute	Choices
Detectable Person	Player, Other Staff Member (on court), Fan, Not Applicable
Obscured Face	Yes, No
Lighting	Direct Sunlight, Shaded, Partially Shaded
Head Angle	Front On, Back of Head, Profile, Other
Glasses	Yes, No
Visor or Hat	Yes, No

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 faces were classified manually but reclassified as other. Crowd shots provided difficulty in determining which faces were reasonable to classify. These 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, we collected information on the following attributes:

3.2 Manual Annotation

To record the details of attributes for each face and scene a Shiny (Chang et al. 2016) App was created. We called this Application our ManualClassificationProgram. This helped to provide information for all attributes quickly and consistently.

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 changed the display and presented a set of Attributes with radio buttons, this allowed information to be recorded for the face in the specific ‘Face Box’. This recorded the x and y coordinates of the corner points of a box drawn by the mouse, and when the save button was hit it saved all the radio button selections and the ‘Face Box’ coordinates to a CSV file.

When a face was not selected, the radio buttons showed the Scene attributes and the radio buttons with the possible selections the annotated was able to choose from. When in this display, selecting the save button would then save the Scene selections to a specific CSV file.

If there were issues, the CSV files were able to be edited, this was reserved for extreme circumstances. As a lot of care was taken to ensure the first selections were correctly submitted and applied to the correct Faces and Scenes.

All the annotations for this sample were completed by one author. This was chosen to provide consistency across the sample of faces annotated manually. However the initial choices of what would be reasonably detected were made by several of the authors.

3.3 Software Interaction

The software choices allowed for POST requests to be sent via the internet. To access the APIs through R we enlisted the `httr` package, using functions from this package a script was written for Google, Animetrics, Microsoft and Skybiometry. These scripts contained loops that would move through the images, individually posting a request for each image to be analysed. These scripts included retrieving the information provided and converting it into a usable format for our analysis. One interesting anomaly was found when using the Skybiometry software as it limited the amount of requests per minute. We accounted for this by stalling the posts for the amount of waiting time the software notified, and checking until the time lapsed and the script could continue looping.

The amount of time taken for each software to process facial detection on the individual images is presented in the histogram :

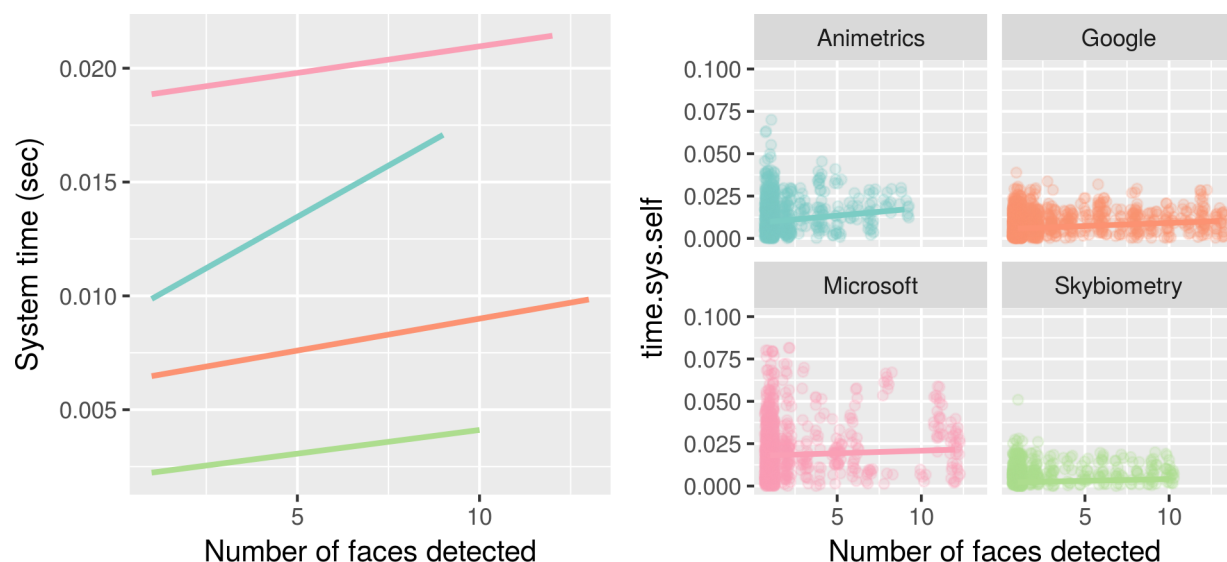


Figure 2: System time taken by the four APIs to finish searching individual images, plotted against number of faces discovered. Skybiometry is the fastest, and Microsoft is the slowest. The largest variability is when no faces are found.

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 manipulate and save the information returned.

3.4 Data Processing

The data needed for our analysis was spread across six files. For each software we had the information on the location of the Facial Bounding Boxes, as well as the time taken for the software to find the information. Some of the software also provided a more detailed level of information.

The collation of the results from the Manual Recognition Program created two CSVs.

A single data set was created to combine all necessary information in the previously mentioned files for our analysis. The information in the data set, was carefully considered. It considers the identify of each face, and all relative face attributes, as well as the image file the face was found in, from

this information each face was able to be uniquely identified. Also included was information on the software that found it, and the time it took the software to identify the face. It also has a record of how many faces had been identified in the image by counting each additional recognized face. To do so, we gathered the name of the file the face was found in and the API that found it. The automatically determined time values were also included. The minimum and maximum x and y values were drawn from different values in each software’s CSV files. This required some processing to align the differing values to be comparable.

To find whether the software were recognizing the same faces a function was created. As the location and size of the boxes around the faces were recorded, these values were used to see if a particular identified face box matched a manually identified face, or a region found by another software. This function uses the information of each face and compares the intersecting regions of the polygons created by the x,y coordinates of Manual Faces and other software’s faces, to determine if the same face was recognized. We determined the ratio of intersecting area to total area must be greater than 0.1 to be considered the same face. This allowed us to compare the identification areas, as well as contrast the identified faces of each software. This contributed another variable, boxID, to the data set.

3.5 Analysis

Using the data set of the combined API and manual results, we were able to compare the performance of the software. Firstly, we considered how many individual faces the software were able to detect in Figure 4.

However, individual faces are not beneficial if they do not correspond to the faces manually annotated. Figure 5 was created by defining groups depending on the API that recognized each particular face. The UpSetR (2017) package helps visualise set intersections. Where in this circumstance faces boxes may overlap on the same face, each bar shows the number of individual faces that have boxes resulting from each of the APIs highlighted below the bar.

3.5.1 Modeling Face Detection Probability

This shows how influential certain attributes are in determining whether a face will be detected, a hit, or not, a miss.

A step wise method was used to determine the best regression model for predicting a hit or miss. The regression model that provided the best AIC, included the scene attributes: Shot Angle, Background, the interaction between these two; whether the scene was a graphic, and the situation on the court when the image was taken. It also included the lighting on the particular face, and whether the specific face was accessorised with glasses and headwear, a visor or hat.

Regression analysis showed that the situation variable had significant differences between intercept level of “Court Close-Up Not Player” and the categories of “Court in Play” and “Court Player Close Up”. This would mean that the probability of all the APIs finding the face that had been found manually was significantly less if the situation depicted was “Court in Play” and “Court Player Close Up” rather than a “Court Close-Up Not Player”.

The bar charts of the situations show that the situation is important in influencing the detection of a face. When the face is accessorised with them, all APIs have a significantly increased chance of

finding the face in comparison to the base rate category of the “Court in Play”. While the image being a Graphic had a large impact it was not significant, this is likely because of the small number of images with this attribute. It also shows that the shot angle is only significant for one or two APIs at each level.

Table 3: The table of regression coefficients denotes the significant variables, and levels, for the four API types. The dashes represent coefficient that were not significant. The symbols to the right of the regression coefficients denote the significance level associated with each coefficient. The background variable, and levels of the interaction between shot angle and background were only significant for Google.

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	1.92	*	4.04	**	1.6	.	1.31	.
shotanglePlayer Shoulder Height	2.04	*	3.58	**	1.89	*	1.87	*
shotanglePlayer Shoulder Height: bgCrowd	-		1.92	**	-		-	
shotanglePlayer Shoulder Height: bgLogo wall	-		1.24	.	-		-	
shotangleUpward Angle	-		4.95	**	-		-	
shotangleUpward Angle: bgCrowd	-		-		-		-	
situationCourt in play	-2.75	**	-4.78	**	-3.7	**	-2.1	**
situationCourt player close-up	-1.21	.	-2.25	*	-		-	
situationCrowd	-		-		-		-	
situationOff court close up of player	-		-1.91	.	-		-	
visorhatYes	-1.2	**	-0.94	**	-0.75	**	-0.76	**

The variables were kept if they were significant for any level of the variable for any type. Given the significant coefficients above it can be seen that the largest decrease in probability of a face being found by the APIs was the face profile being captured.

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



□ Manual

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.

4 Results

4.1 Amount of Faces

Each API returned areas that indicated the potential location of a face. These areas are defined using the four points returned by the APIs. The Google Vision API detected a large amount of faces, much more than Microsoft, Skybiometry or Animetrics. This can be seen in Figure 4.

To evaluate the performance in terms of the overall accuracy of each algorithm we considered the amount of faces they classified that matched faces that were selected manually. This sample contains all the manually annotated faces and all the faces recognized by the four APIs.

Figure 4, the bar chart of faces per API shows Google produces the most detections. Having a high volume of detected faces returned is valuable for the future use where faces of the player must be captured over time during sports matches.

4.2 Comparing Faces Found

The individual faces, the specified area of the image, found by each API are compared to those found manually and by the other APIs.

Manual	Face	No Face
Face	434	1861
NoFace	527	787

Table 4: Animetrics

Manual	Face	No Face
Face	1319	976
NoFace	638	676

Table 5: Google

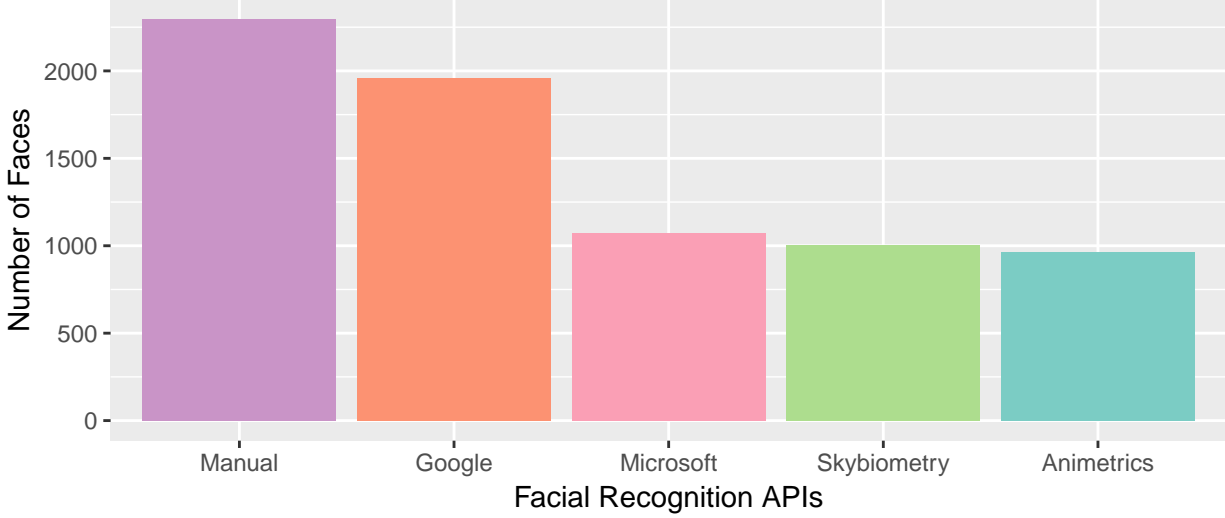


Figure 4: Face Boxes Per API: The bar chart shows the number of Face Boxes produced by each API, comparing the height of the bars indicates that Google’s Facial Recognition API recognized almost 1000 more faces than the next best API, Microsoft.

Manual	Face	No Face
Face	565	1730
NoFace	505	809

Table 6: Microsoft

Manual	Face	No Face
Face	493	1802
NoFace	512	802

Table 7: Skybiometry

Each table above considers the individual faces found in the set. The upper left quadrant of tables shows the faces found manually and by each API. The ideal set would have majority of faces found belong to this category. Table 4, the comparison of Animetrics and Manually annotated faces, gives the lowest amount in this regard. Table 5, Google, found the largest amount of faces that matched those manually annotated.

Visual inspection showed that the 638 potential faces noted in Table 5, that were found by Google but not found manually were actually faces, however they were not all of players. Some of these crowd members were beyond what would be expected of an emotion recognition application for detecting faces of tennis players during a match. These were deemed unlikely to be necessary when attempting to find faces of players during tennis matches, and neglected during manual annotation.

Comparatively, visual inspection of the 527 potential faces that did not match manually annotated faces showed the Animetrics results contained many potential faces that were actually very unusual results.

Figure 5 helps to understand faces that were found by a combination of APIs, and the manual annotations. The black bubbles the bars show the API combination that detected all the faces counted in the bar above. Faces common to all four API and manual annotations were the fourth largest group, it was obvious that relevant faces were captured more commonly by Google and manually. The fifth group was the most suprising, this captured a lot of faces that were false discoveries.

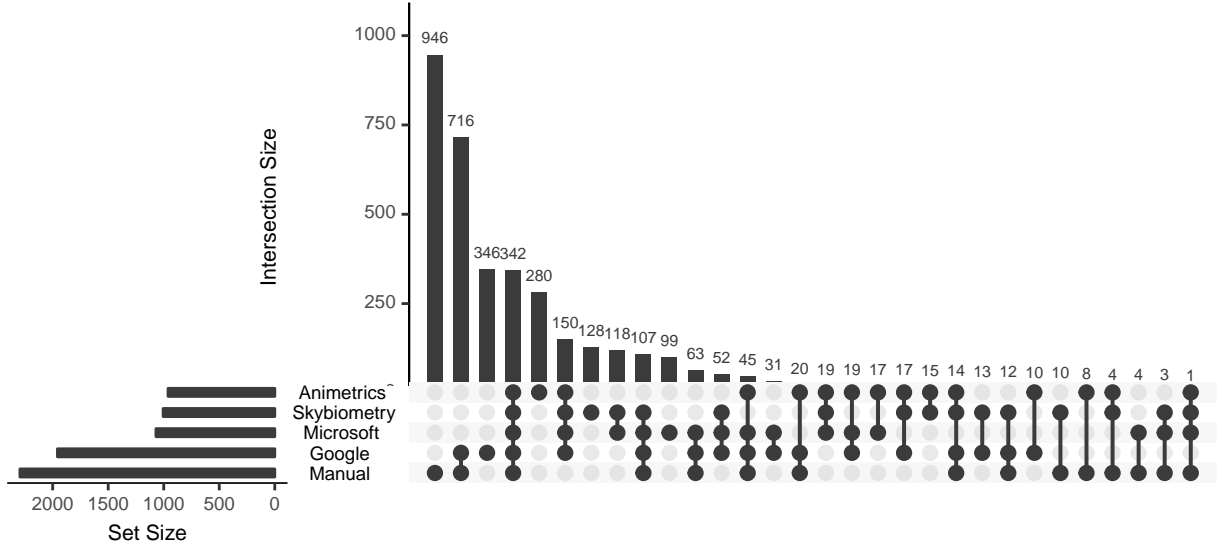


Figure 5: Faces Per API Combination. The Bar Chart shows the faces that were recognised by multiple API or found manually. The largest group, with 809 faces, is faces only found by Manual annotations. The following group were the 716 faces recognized both Manually and by the Google API. These combinations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

4.3 False Discoveries

Examples are included of faces only found by one of the APIs, as other than Google, the APIs produced several unusual false discoveries. These highlight the potential differences in the detection methods used by each of the APIs. This would need to be taken under consideration to prevent time wasted on sorting actual faces from false discoveries in future applications.

Figure 6 depicts four images that contained unusual results of faces. Figure a highlights a very large area, most interesting about this is the face of the ball boy captured within this area. Figure b also shows unusual faces, the KIA logo was returned by Animetrics many times as a face. This discredits the ability of the API to distinguish actual faces from tricks of lights and colours in the complex setting of colourful tennis matches.

Examples of tricks of colours are shown in Figure c, where logos on the shirts were returned as faces. These discoveries discredit the use of Animetrics for facial detection applications during sports matches.

4.4 Image Characteristics

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

Table 8: This table outlines the combinations of image attributes that are most common in the image set. The combination of the Logo Wall background and the Player Shoulder Height angle are shared by three of the five image combinations.

situation	bg	shotangle	detect	count
Court player close-up	Logo wall	Player Shoulder Height	Player	720
Court in play	Logo wall	Player Shoulder Height	Player	238
Crowd	Crowd	Upward Angle	Fan	149
Court player close-up	Court	Birds Eye	Player	133
Court player close-up	Logo wall	Player Shoulder Height	Other staff on court	117

We then considered that there would be an uneven amount of faces with certain image attributes. 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 in Figure 7 Plot a. show the amount of faces that matched those found manually given the angle the faces were captured from. There is a greater amount 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 less faces that matched those found manually. The most common faces in our set were those captured at Player Shoulder Height and found manually and by an APIs.

The mosaic in 7, Plot b shows the number of Faces captured during each possible situation. It contrasts how many of the images captured in situations either did or did not match the faces annotated manually using the colour pink for potential faces that did not match, and blue for those that did. It can be seen that the largest portion of the faces were captured in a close up situation while the players were on court.

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 Player Shoulder Height is common to the highest amount of faces. There are also no images that were taken at an upward 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.

4.5 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.

Image a.



Image b.



Image c.

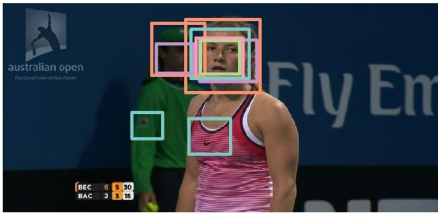


Image d.

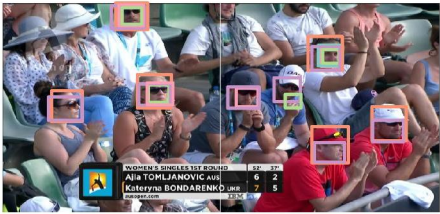


Figure 6: Animetrics provided many results that were quite unusual. The top left image presents a potential faces that actually defines an area containing a player’s back and a ball boy’s torso. The top right shows two potential faces that are actually the KIA logo on the net. The bottom left image was very interesting as the player’s face was found, but the shirts of both the player and ball boy were deemed potential faces. The bottom right image had a lot more faces that usually recognised. Animetrics found a fist to the right on the centre that it considered a potential face.

Table 9: This cross table compares the amount of faces that were captured wearing accessories.

Glasses	Headwear		Total
	No	Yes	
No			
N	981	962	1943
Total(%)	42.7451%	41.9172%	
Yes			
N	92	260	352
Total(%)	4.0087%	11.3290%	
Total	1073	1222	2295

Table 10: This cross table compares the amounts and percentages of faces captured and whether the person was wearing glasses. The first row is the faces that were tagged as not wearing glasses, the second row is faces that were. For the API to not have been affected by the prescence of glasses, the proportion should be similar to that found manually. Animetrics had the highest proportio difference, 93.31% of the faces it found did not have glasses. Google is the closest proportion, 88.62% of the faces it found were wearing glasses.

Glasses	Headwear					Total
	Manual	Animetrics	Google	Microsoft	Skybiometry	
No						
N	1943	405	1169	514	450	4481
Column(%)	84.6622%	93.3100%	88.6277%	99.9725%	91.2770%	

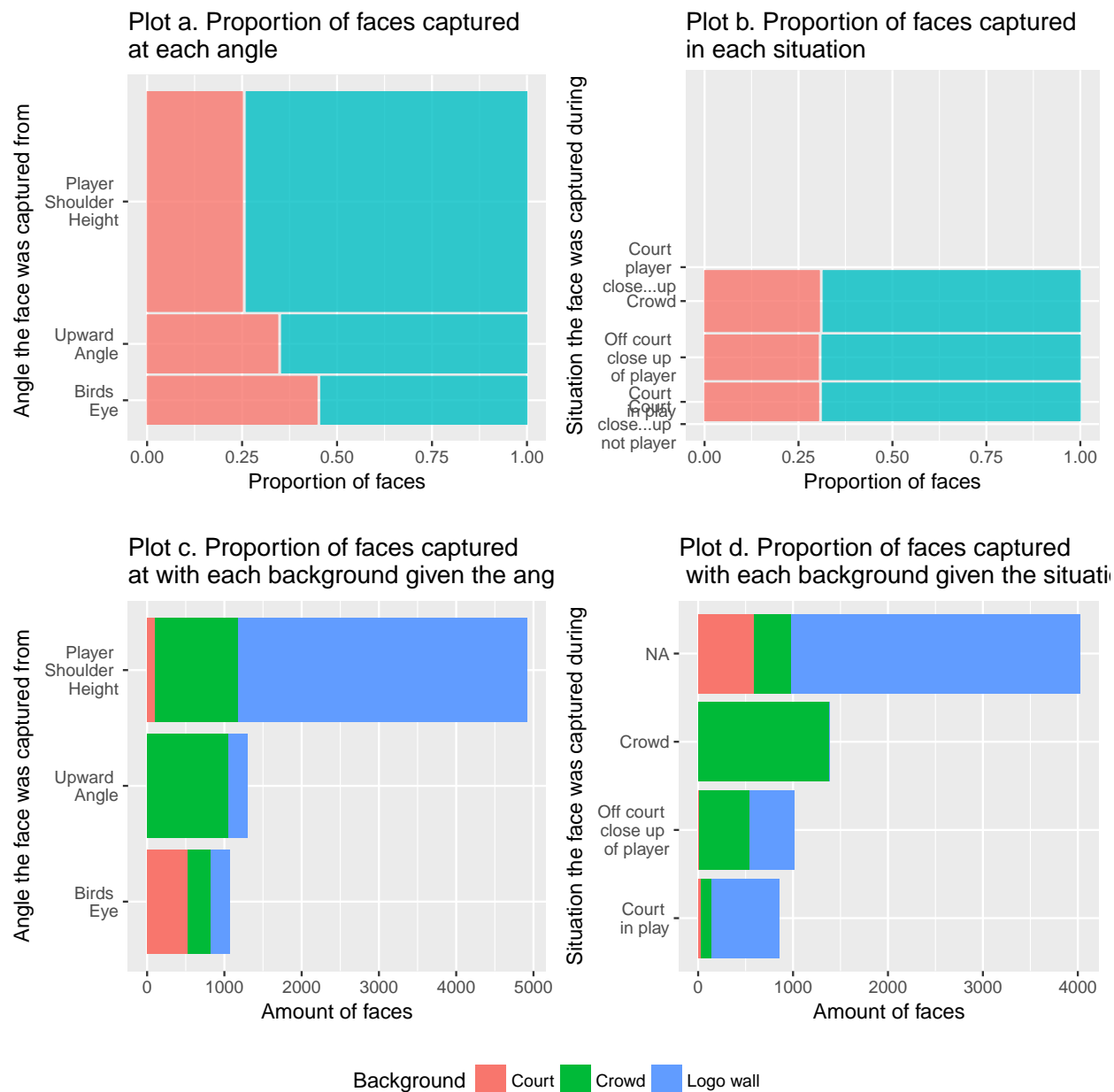


Figure 7: The mosaic plots indicate the proportion of faces given the angle, or situation, that matched faces found manually. The stacked bar charts show the amount of faces captured at certain angles, or situation, .

used in this research 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.

6 Acknowledgements

The APIs, the scripts used to access them and the resulting data set are contained in the supplementary materials.

6.1 APIs

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.

6.2 Data Files

All data files used are available on the github page.

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.

Each script that was automated for POST requests created a classified faces csv.

6.3 Packages

The R packages were used to create the following:

- bookdown (Xie 2017a)
- descr (DirkENZmann 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 2016)
- kfigr (Koochafkan 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)

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>.
- D. Gorodnichy, P. Radtke, E. Granger. 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.

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/>.

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. 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. 2016. *Gtable: Arrange 'Grobs' in Tables*. <https://CRAN.R-project.org/package=gtable>.

———. 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/>.