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 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.

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).

Table 1: Image descriptions that are associated with the attributes of 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

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

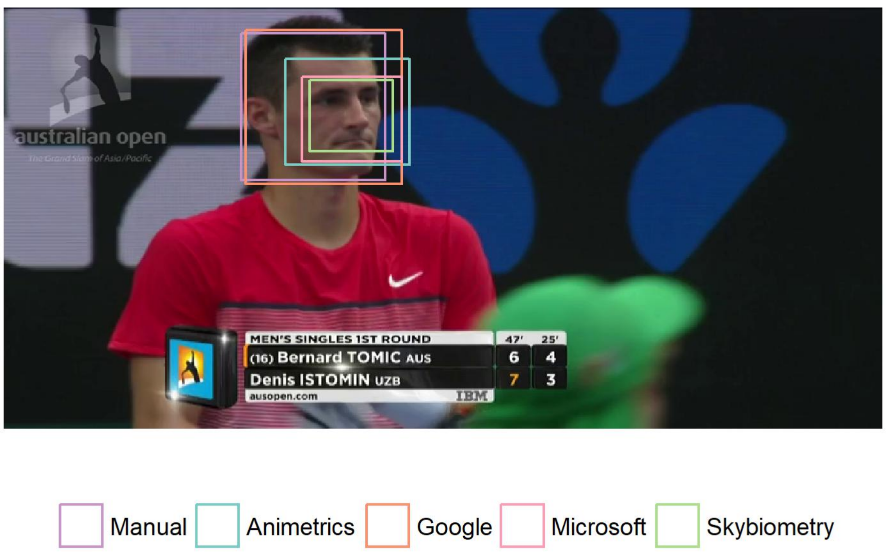

Figure 1: This image of Bernard Tomic was chosen as a trial image to be presented to each of the software before they were included in the research. Each colour represents a different API source. It was expected that the software would be able to find this face, despite the player facing away from the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

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

Table 2: Possible face descriptions for each individual face. Most of the selections were obvious for each image, 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

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, 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 author. This provided consistency across the sample of faces annotated manually. The choices of what would be reasonably detected was made by several of the authors.

2.4 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 :

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.

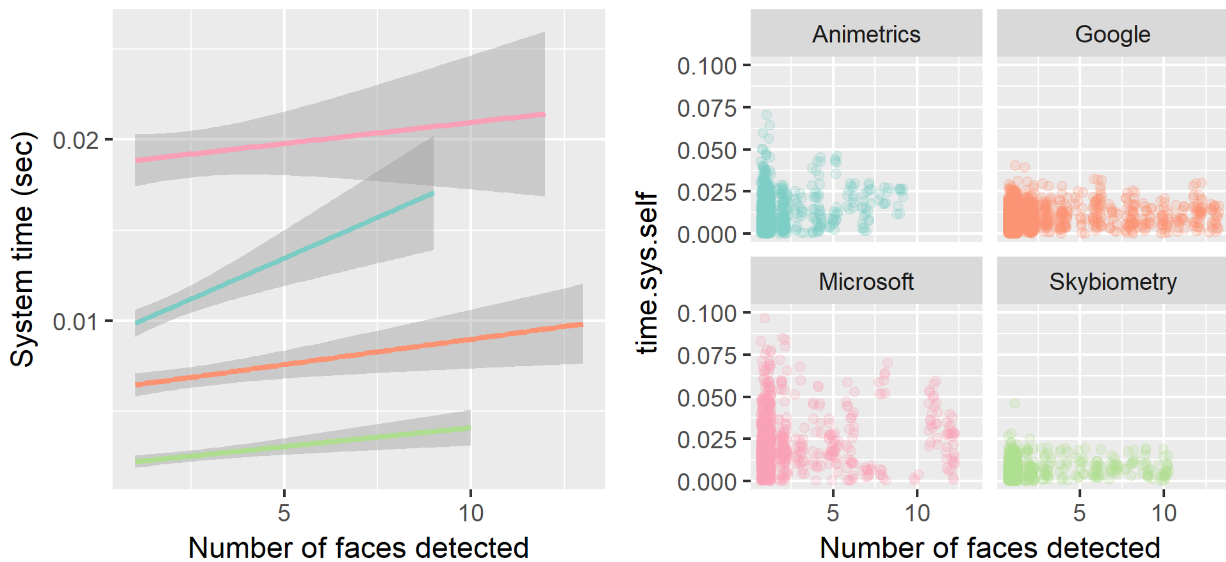

Figure 2: System time taken by the four APIs to finish searching individual images, plotted against number of faces discovered. Skybiometry is the fastest, and Microsoft is the slowest. The largest variability is when no faces are found.

2.5 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.

2.6 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.

2.6.1 Modeling face detection probabiity

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 "Not player" and the categories of "Match 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 "Match play" and "Court Player Close Up" rather than a "Not player".

Table 3 shows that for all of the APIs are significantly less likely to find a face accessorised with glasses or hats, or obscured by an object. All APIs are less likely to find a face if the face was not captured front on to the camera. The faces captured when the situation was match play significantly decreased the likelihood of a hit for all APIs.

Google was the only API that benefited from the interaction that was significant in increasing the likelihood of a hit when a face was captured at an angle level to the face, and had a background of the logo wall.

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.

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	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	**

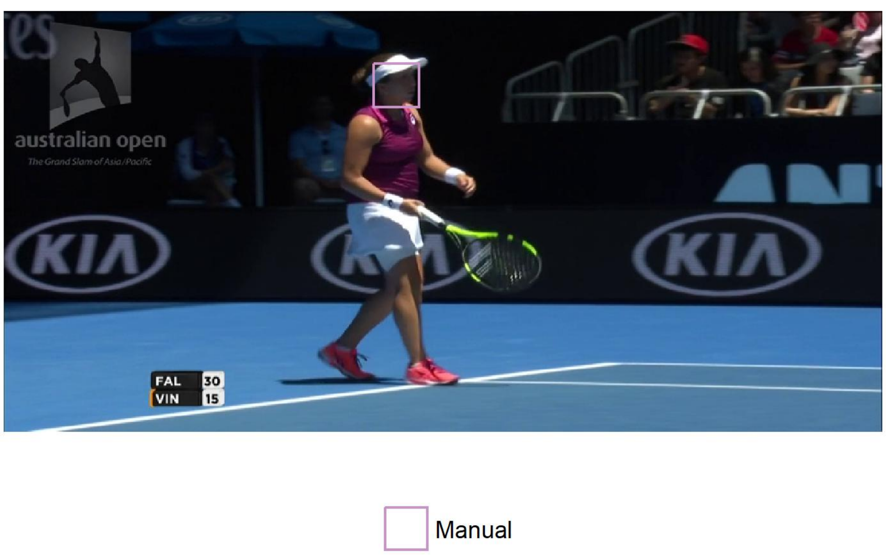

Figure 3: The face captured manually in this image was not captured by any API. The head angle, shot angle, lighting, wearing the visor and being captured across the court during the match all contribute to the struggle of the APIs.

3 Results

3.1 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.

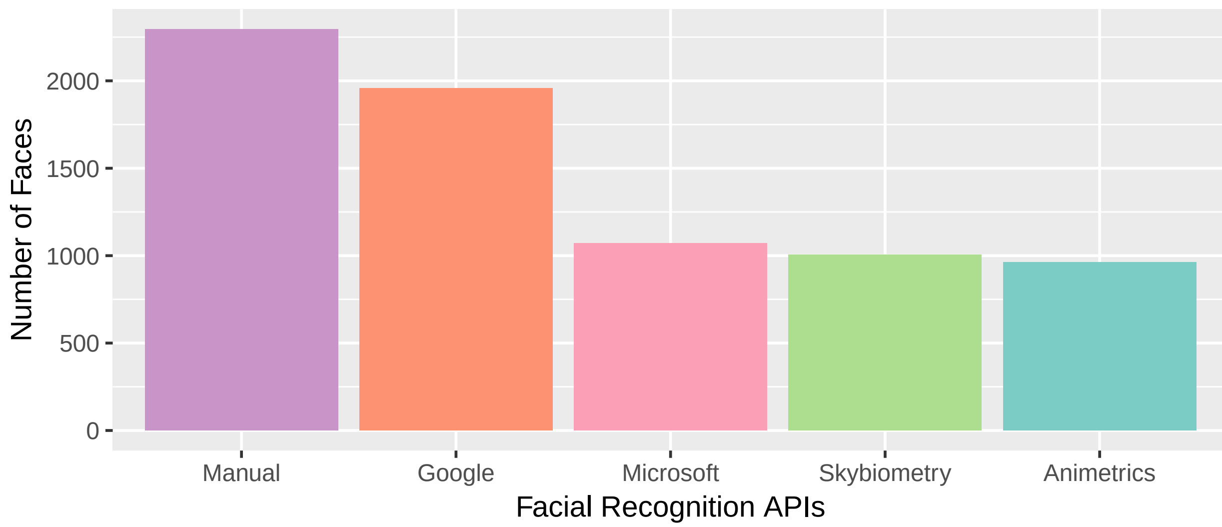

Figure 4: Face Boxes Per API: The bar chart shows the number of Face Boxes produced by each API, comparing the height of the bars indicates that Google's Facial Recognition API recognized almost 1000 more faces than the next best API, Microsoft.

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.

3.2 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.

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 4, Google, found the largest amount of faces that matched those manually annotated.

Table 4: Counts of captures of different APIs against manually tagged faces.

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

Visual inspection showed that the 638 potential faces noted in Table 4, 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.

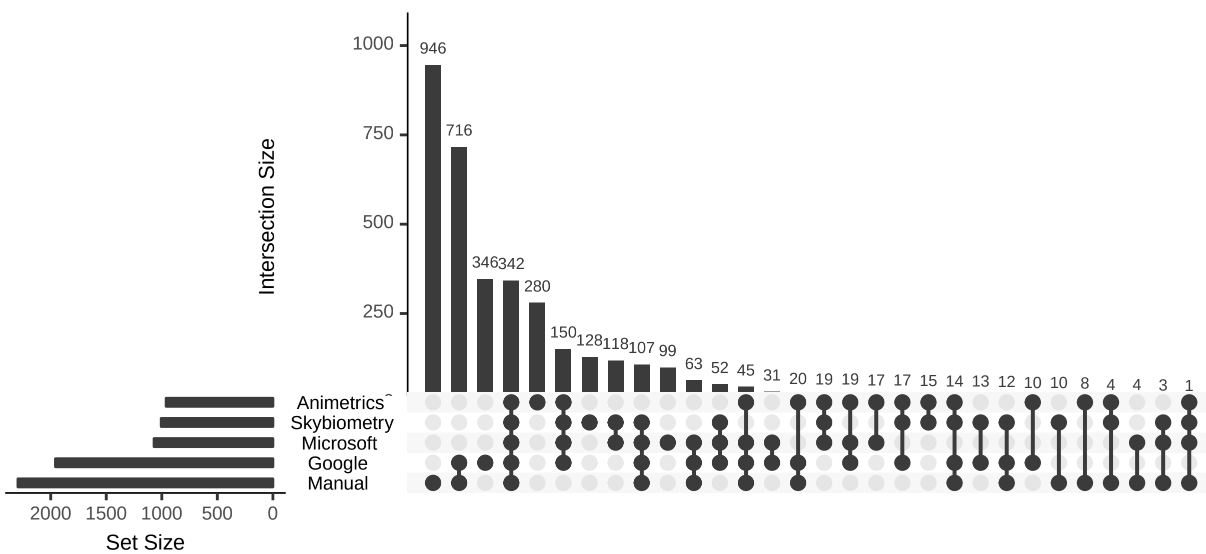

Figure 5: Faces Per API Combination. The Bar Chart shows the faces that were recognised by multiple API or found manually. The largest group, with 809 faces, is faces only found by Manual annotations. The following group were the 716 faces recognized both Manually and by the Google API. These combinations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

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 surprising, this captured a lot of faces that were false discoveries.

3.3 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.

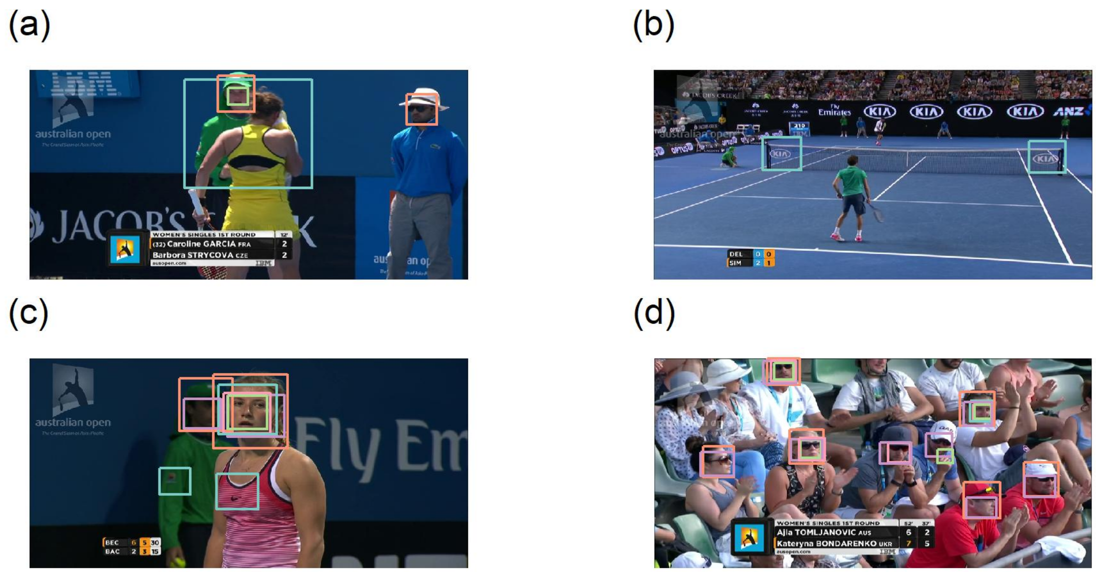

Figure 6: Animetrics provided many results that were quite unusual. The top left image presents a potential faces that actually defines an area containing a player's back and a ball boy's torso. The top right shows two potential faces that are actually the KIA logo on the net. The bottom left image was 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.

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.

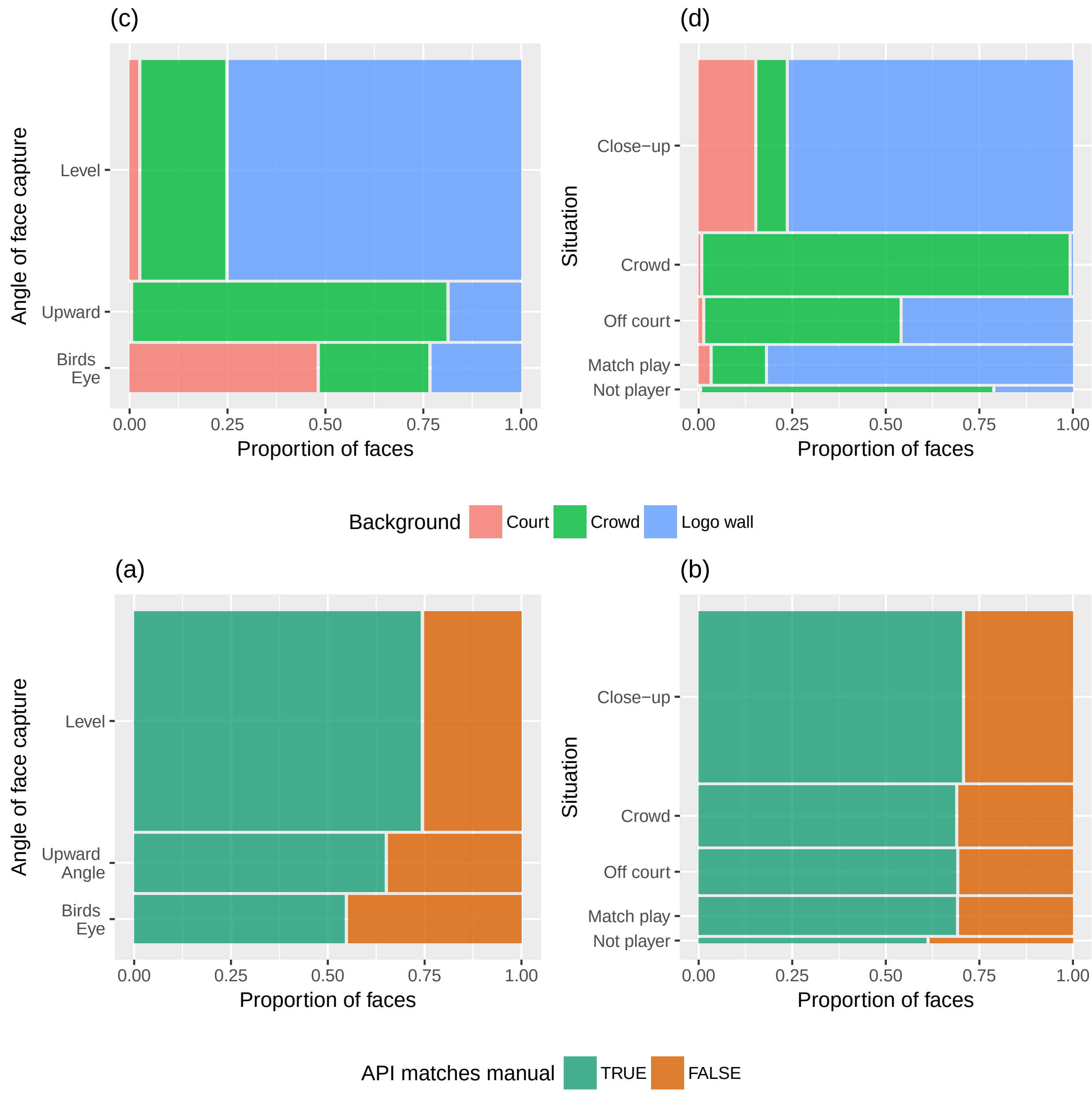

Figure 7: The mosaic plots indicate the proportion of faces given the angle, or situation, that matched faces found manually. The stacked bar charts show the amount of faces captured at certain angles, or situation.

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

3.4 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. 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 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.

3.5 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.

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

4 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 2016), 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.

5 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. 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/.