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 head wear 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 possible obstacles of real time implementation were also assessed via the 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 about the mental state of a player and the effects changes in the mental state have 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.

Application Programming Interfaces (APIs) have been used to evaluated the performance of several popular facial recognition software, on the 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 software. 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.

Most facial recognition software began with the intention of security applications. There are others, such as Facebook which links images of people with their profiles. This means the recognition software currently available was not intended to be used in fast paced, elite sport environment. It was believed that the capabilities of the software could be limited by their intended security and surveillance purposes. Barr (2014) addresses the 'lack of robustness of current tools in unstructured environments' and this survey explores how detection performance is affected by situations that arise in elite tennis matches.

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 is representative of the video files to be used for future facial recognition analysis:

- 6406 images, 800×450px

To produce the set of 6406 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 files used were the broadcasts 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. and Granger, E and 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. (2014). Companies who have recently expanded their API ranges were also considered. This search produced two of the software, Microsoft API (Microsoft Cognitive Services 2016), provided by Microsoft Cognitive

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

Services, and Google Vision API (Google Cloud Platform 2016). Online demos were used to test viability, Figure 1 depicts Bernard Tomic and the detected areas found.
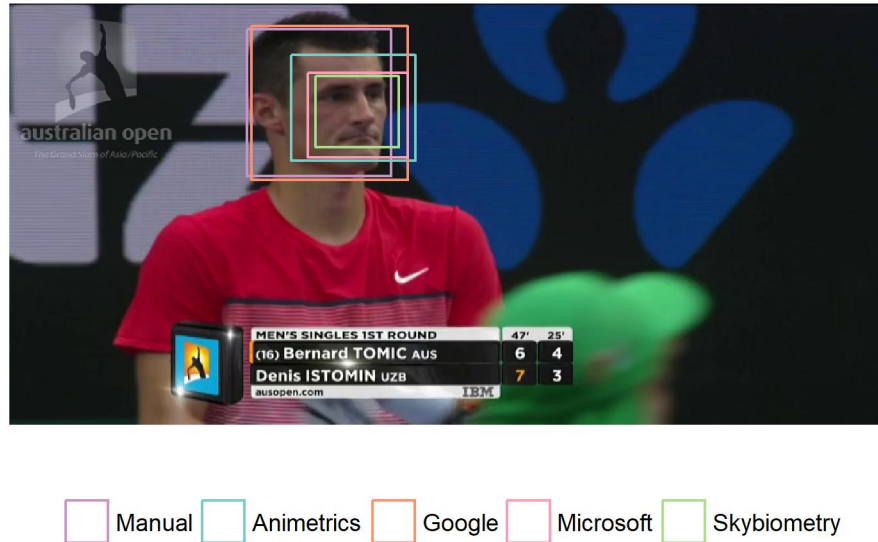

Figure 1: This image of Bernard Tomic was chosen as a trial image to be presented to each of the software before they were included in the research. Each colour represents a different detection source. It was expected that the software would be able to find this face, despite the player facing away from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

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 Sunlight, Shaded, Partially Shaded
Head Angle	Front On, Back of Head, Profile, Other
Glasses	Yes, No
Visor or Hat	Yes, No

each of these faces, information was collected regarding the attributes in Table 2.

The web app was created using the Shiny package for R (Chang et al. 2016) and has been developed into an R package called taipan, as it is a Tool for Annotating Images in Preparation for ANalysis. If there was a face in the image the annotator was able to highlight that section of the image to contain the face. This selection activated a set of attributes questions for the annotator to answer, and allowed information to be recorded for the face. The records included the location of the box in the image, and all appropriate answers. When a face was not visible in the image only the scene attributes were applicable for that image. All of 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 APIs were accessed via a R (R Core Team 2013) script, calling the APIs relied on 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 and error management of the requests was incorporated for these APIs.

Figure 2 displays the distribution of processing time required by the APIs for each image, as a violin plot (Hintze 1998). Both Microsoft and Google had few images where they had taken a significant amount of time to complete, and these were removed from this plot due to concerns regarding the network. The times taken to search are quite variable between APIs, with Skybiometry consistently spending less time than other APIs. Microsoft and Google's times vary regardless of the number of faces 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 process the information returned.
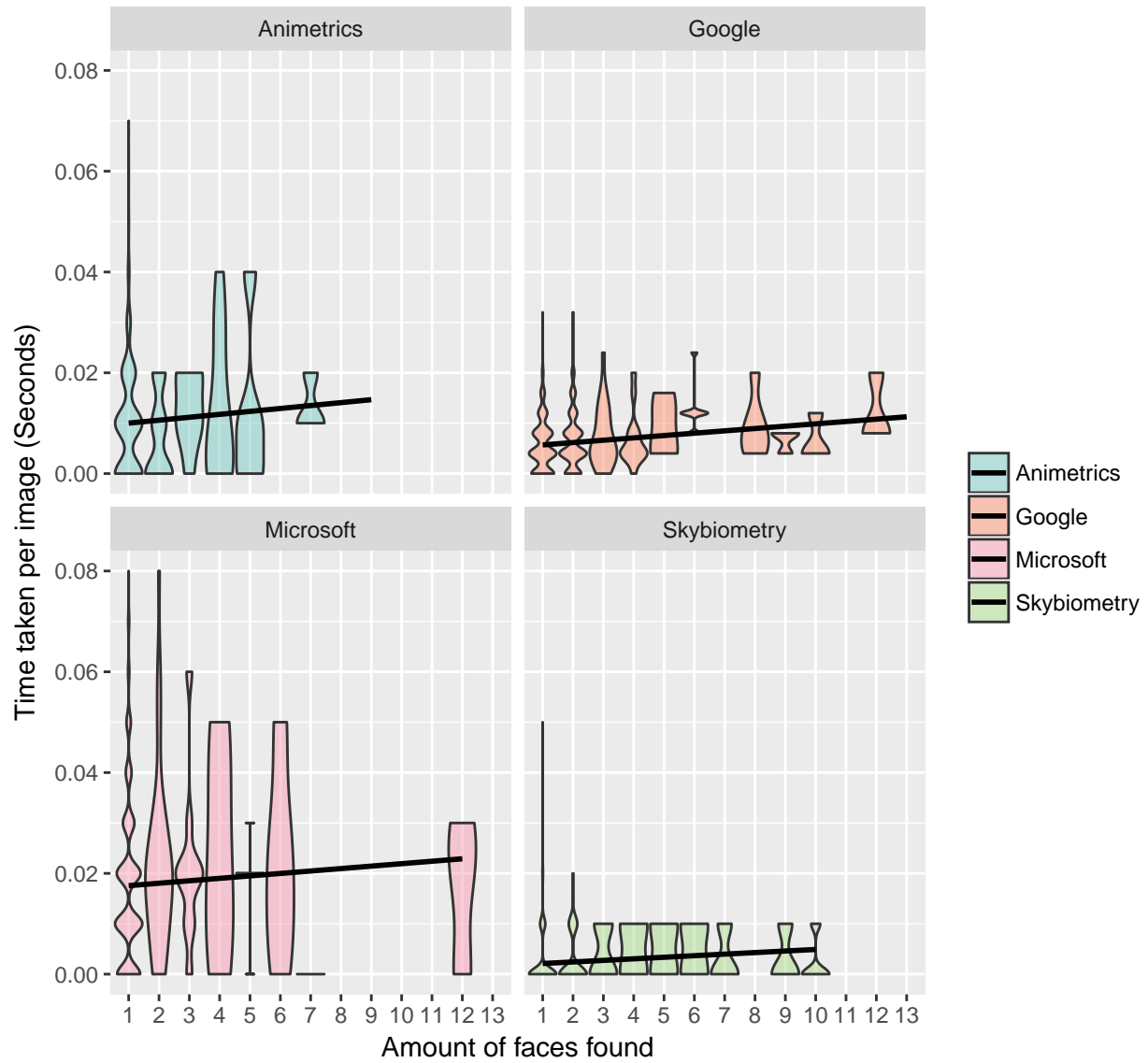
Figure 2: System time taken by the four APIs to finish searching individual images, by number of faces discovered, displayed as violins. The width of the violin shows the most common times for the group of images.

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 scene, and the other contained information regarding the faces.

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. Every face found in an image manually, or by any API was compared. From their locations the intersecting area between them was found. When the ratio of this intersecting area to total area was greater than 0.1 the IDs were altered to reflect the two faces would be considered the same face. This allowed comparison of the identification areas on the same face, as well as contrast the identified faces of each software.

2.6 Results

The manual tagging helped to provide a benchmark 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 multiple APIs indicated either obvious facial features were visible or the APIs may be using similar criteria to detect faces. 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 that may have affected the detection likelihood of the face. A step-wise variable selection method was used to find variables of significance for any of the APIs. 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. Variables that were not included denoted whether there was a graphic or live shot visible, the role of the person whose face was captured and whether or not there was a person in the image. detect A separate model was fit using the faces for each API.

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. Situation had different effects on each API. Results for Google were most interesting in the shot angle 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.

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	-1.00	**	-0.49	*	-1.00	**	-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	0.70	.	1.79	**	-		-	
shotangleLevel	-		1.33	**	0.87	.	1.01	.
shotangleLevel:bgCrowd	-		1.92	**	-		-	
shotangleLevel:bgLogo wall	-		1.24	.	-		-	
shotangleUpward	-		2.70	**	-		-	
shotangleUpward:bgCrowd	-		-		-		-	
situationCrowd	0.62	.	1.74	**	0.70	.	0.67	.
situationMatch play	-1.54	**	-2.53	**	-2.68	**	-1.24	**
situationNot player	1.21	.	2.25	*	-		-	
situationOff court	0.61	*	-		-		-	
visorhatYes	-1.20	**	-0.94	**	-0.75	**	-0.76	**
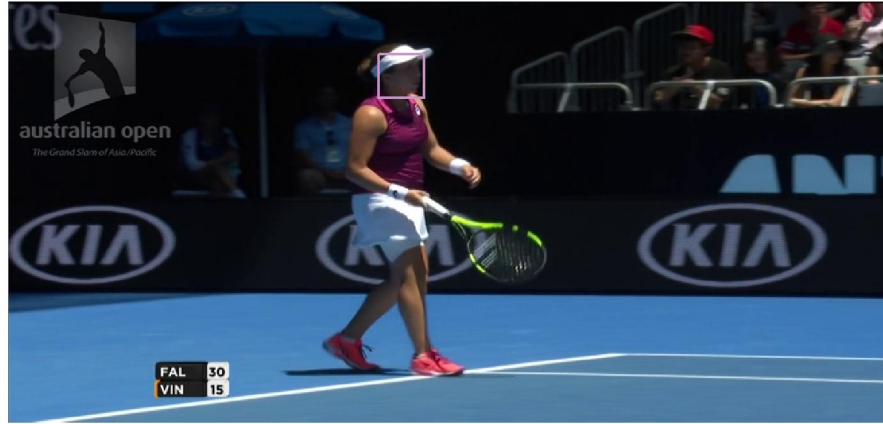
Manual

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.

2.6.2 Detection rates and API comparison

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.

Table 4 displays the proportion of the manually tagged faces that the APIs detected. Google had by far the best detection rate. In addition, the APIs also detected non-faces, in images that produced no manually tagged faces. Google had the highest of these, and the detections were for the most part helpful, primarily of tiny faces in the crowd, that were ignored by manual tagging. They would not be needed for the long term purpose of the study, to detect and tag players' emotions during a match.

In comparison, examining the 527 images in which Animetrics detected faces, not reported manually, revealed surprising results, which are discussed in more detail later.

Table 4: Proportion of API matches with the 2295 manually tagged faces, and faces detected in the 1314 images with no manually tagged faces.

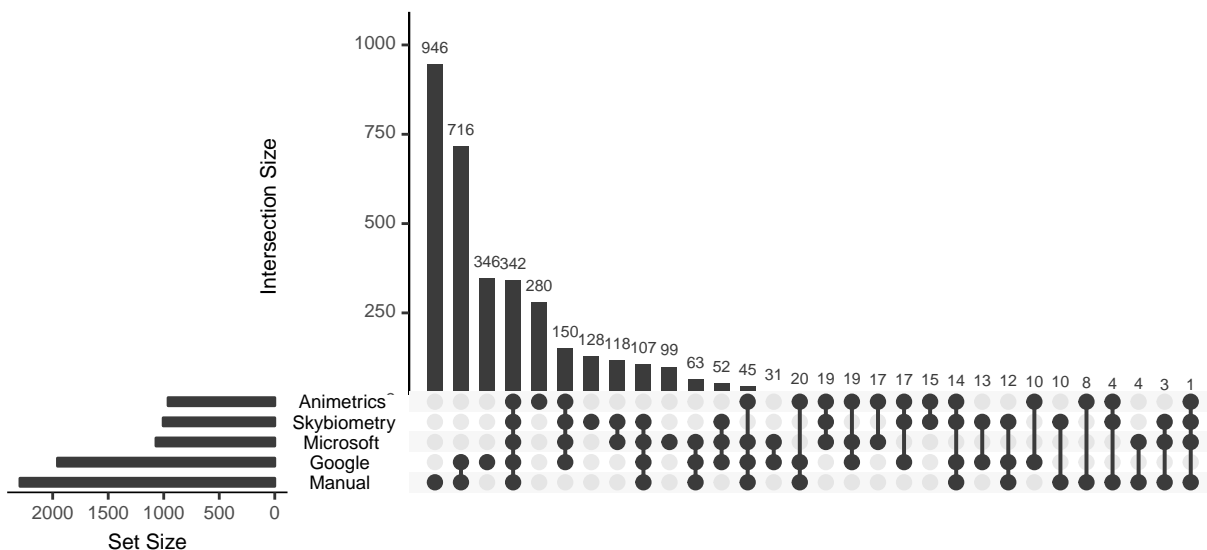|	Faces found		Non-faces found	
API	Count	Prop	Count	Prop
Google	1319	0.57	638	0.49
Microsoft	565	0.25	505	0.38
Skybiometry	493	0.21	512	0.39
Animetrics	434	0.19	527	0.40

Figure 4: Set visualization, to examine intersections between API and manual face detections: bar charts show counts, by API (lower left) and for combinations (main part of plot), and dot display (bottom) indicates the group intersections. All five way intersections are examined, but only the first few have large counts. The largest group is the manual tagging with 946 faces uniquely identified. The intersection between manual tagging and Google is 716 faces. The intersection of all four APIs and manual tagging is fourth highest, at 342.

Table 5: Top five most frequent image attribute 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

Figure 4 is a set visualization 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.

2.7 False discoveries

Animetrics provided many surprises. Figure 5 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 ball boy'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 actually a fist (to the right on the center).

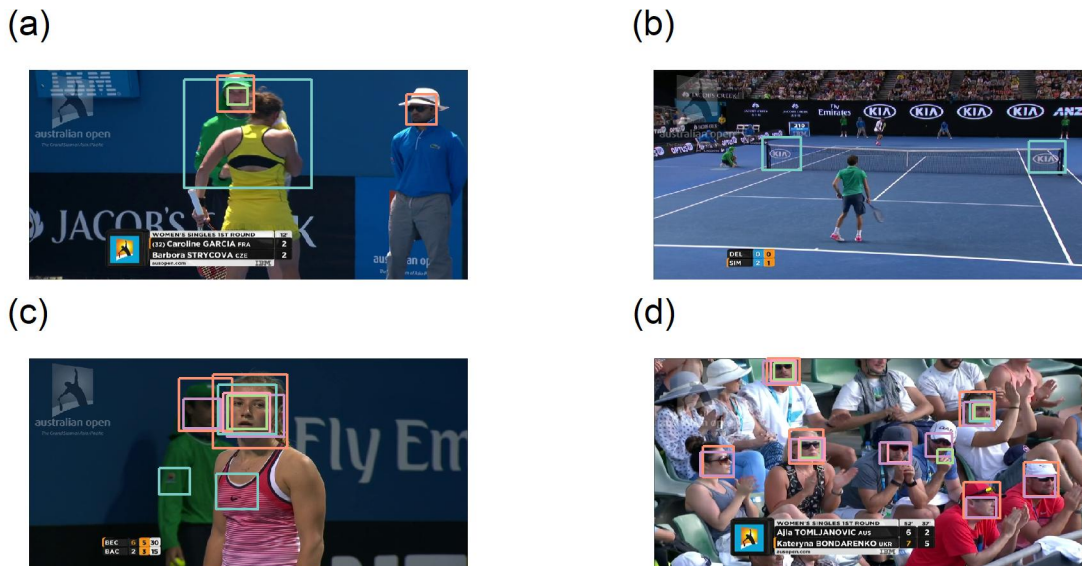The other APIs also returned some non-faces, but the most egregious mistakes came from Animetrics.

(a)　　　　　　　　　　　　　　(b)

(c)　　　　　　　　　　　　　　(d)

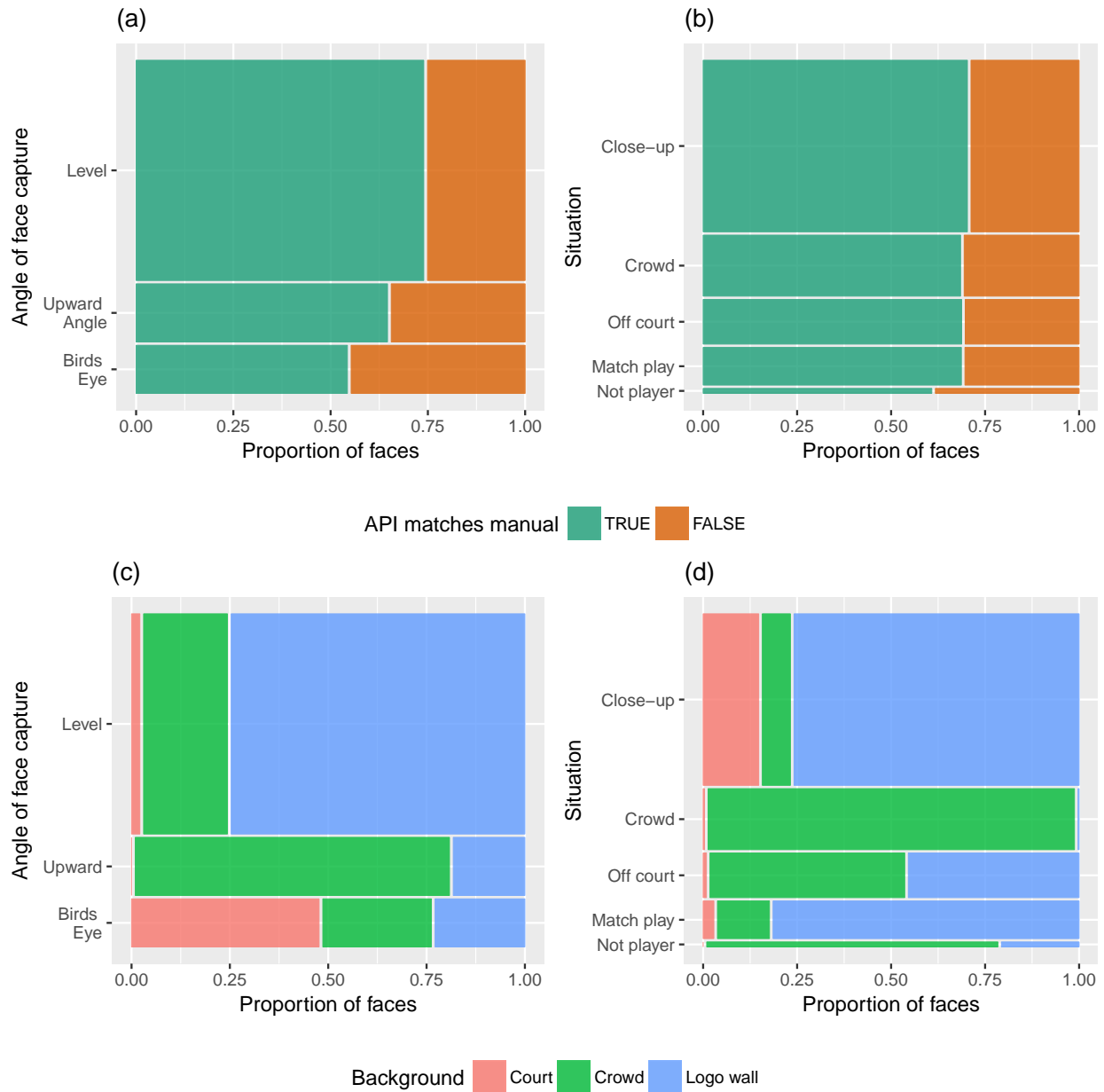Figure 5: Four images in which Animetrics located unusual faces.

9

Figure 6: Relationships between court and face characteristics, and API detection, using mosaic plots: (a) Angle of face capture, (b) court situation, (c) angle of capture by background, and (d) situation of capture by background. Most face captures occurred at a level angle, and this produces slightly better detection. Most situation face captures are close up, but the detection rate is identical regardless of situation. Most level angles had the logo wall in the background, while most birds-eye had the court as background, as we would expect. Close ups were mostly against the logo wall.

2.8 Image characteristics

Image characteristics (Table 1) were constructed during manual tagging. The scene information was tallied and the most common combinations of these features shown in Table 5.

Figure 6 examines the relationships between several of the image characteristics, and face detection by the Google API, using mosaic plots. Plot (a) compares the angle of capture by detection rate. Colour is mapped to detection. Most images were taken at a level angle. The highest capture rate occurred when the angle was level, and least with a bird's eye, as might be expected. Plot (b) compares the situation with face detection. Close-up was the most common situation. The detection rate was the same across situations.

The mosaic 6 (b) in Figure 6, 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 amount 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, it can be inferred the situation may not be influencing the APIs rate of detection.

Plots (c) and (d) show the three way relationship between capture angle, situation, background and face detection. In the level capture angle most faces were detected against the logo wall, but with bird's eye most were captured against the court. In close-ups the most faces were detected against the logo wall.

2.9 Player accessories

The use of accessories like glasses and head wear, visors or hats, is common during the Australian Open as play is mostly outdoors during daytime. Table 6 shows player use of head wear and glasses, as tagged manually.

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

Table 6: Usage of head wear and glasses by players. Most players did not wear glasses, and roughly half had hats or visors.

	No head wear	Head wear	Total
Glasses	92	260	352
No glasses	981	962	1943
Total	1073	1222	2295

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. A long term aim would be to create a program that automated the collection of player emotion data from throughout a match. This information would be presented in a timeline

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

	Headwear		Glasses	
API	Yes	No	Yes	No
Google	0.48	0.69	0.43	0.60
Microsoft	0.19	0.32	0.14	0.26
Skybiometry	0.16	0.28	0.12	0.23
Animetrics	0.11	0.27	0.08	0.21

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.

Google found many faces that did not match manually annotated faces, this showed manual errors were a possibility. To search for these Taipan could be extended to allow the face identified to be displayed and confirmed this would allow the annotator to confirm manually whether or not these are faces.

Given that certain scene attribute combinations produced more successful facial recognition than other combinations, a limit should be implemented for the sample of images sent to Google Vision API. This would not help to reduce cost. 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 was most successful for Google to return faces.

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

Acknowledgements

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: Animetrics Face Recognition API, FaceR API by Animetrics Inc (2016).

- Google: Google Cloud Vision API (2016)

- Microsoft: Microsoft Azure Cognitive Services Face API, published by Microsoft Cognitive Services (2016).

- Skybiometry: Skybiometry (2016) 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. and Granger, E and 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.

Hintze, Nelson, J. L. 1998. "Violin Plots: A Box Plot-Density Trace Synergism." *The American Statistician* 52: 181–84.

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/.