Performance Comparison of Common Face Detection APIs using Video Stills from Professional Tennis Matches

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, using a specially constructed web application, recording the scene characteristics and features of individual faces, such as head wear or sunglasses. API performance was evaluated on their success rate at detecting the tagged faces, and also relative face and scene attributes.

Keywords: data science, sports analytics, shiny, R

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 et al. (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 $\times$ 450px

To collect the images, 5 minute segments were taken from 105 video files, with a still shot stored every three seconds. 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, pp.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 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.

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, and to minimize human labor, some rules were applied to the manual tagging:

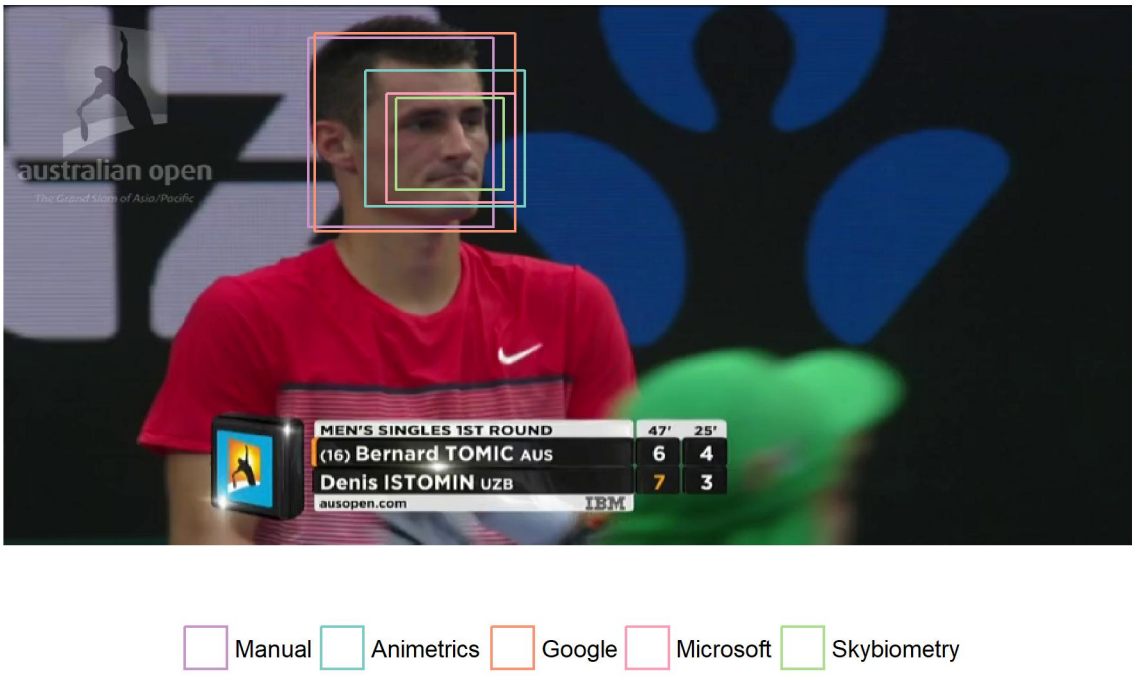

Figure 1: This image of Bernard Tomic was chosen as a trial image to be presented to each of the software before they were included in the research. Each colour represents a different detection source. It was expected that the software would be able to find this face, despite the player facing away from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

Table 1: Characteristics of the scene occuring when the still shot was taken, one option for each category was selected for each image.

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

Table 2: Characteristics of the faces captured, one option for each category was selected for each face manually tagged.

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

- The faces 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.

Table 2 summarises attribute information recorded for each face.

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. The annotator was able to highlight the section of the image containing a 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 answers to the questions providing the attributes of the face. When a face was not visible in the image only the scene attributes were recorded for the 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 2017) script, calling the APIs relied on the httr (Wickham 2016*b*) 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 shows the distribution of processing time required by the APIs for each image. The violin plot display created by Hintze & Nelson (1998) 'combines the box plot and the density'. A categorical variable splits the data along the x axis, as would be seen in a box plot display. A density plot is then incorporated on the y axis. This is produced by manipulating a density display; switching the axes allows the continuous variable to increase along on the y axis, then the violin shape is created by mirroring the density. The width of the violin will now be twice the length that would be have been measured vertically in a density plot.

As seen in the facets of Figure 2, both Microsoft and Google had few images which took 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. The lines overlaid in each plot represent the smoothed linear model of the amount of faces, it can be seen that all APIs take longer to complete detection of faces in an image when there are more faces. As all APIs had more images where only one or two faces were found it resulted in more variability in the amount of time taken.

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.

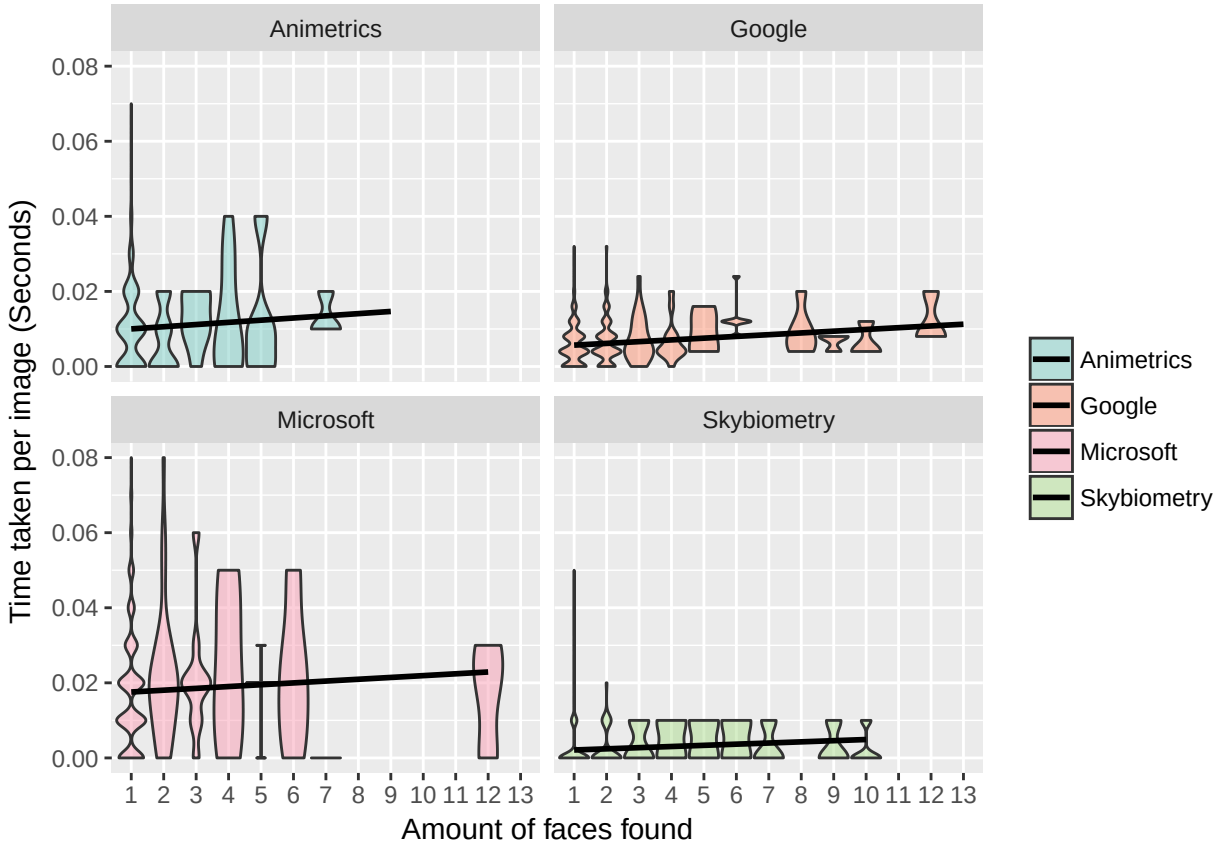

Figure 2: System time taken by the four APIs to finish searching individual images, by number of faces discovered, displayed as violins. The width of the violin shows the most common times for the group of images.

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.

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.

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.

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	**

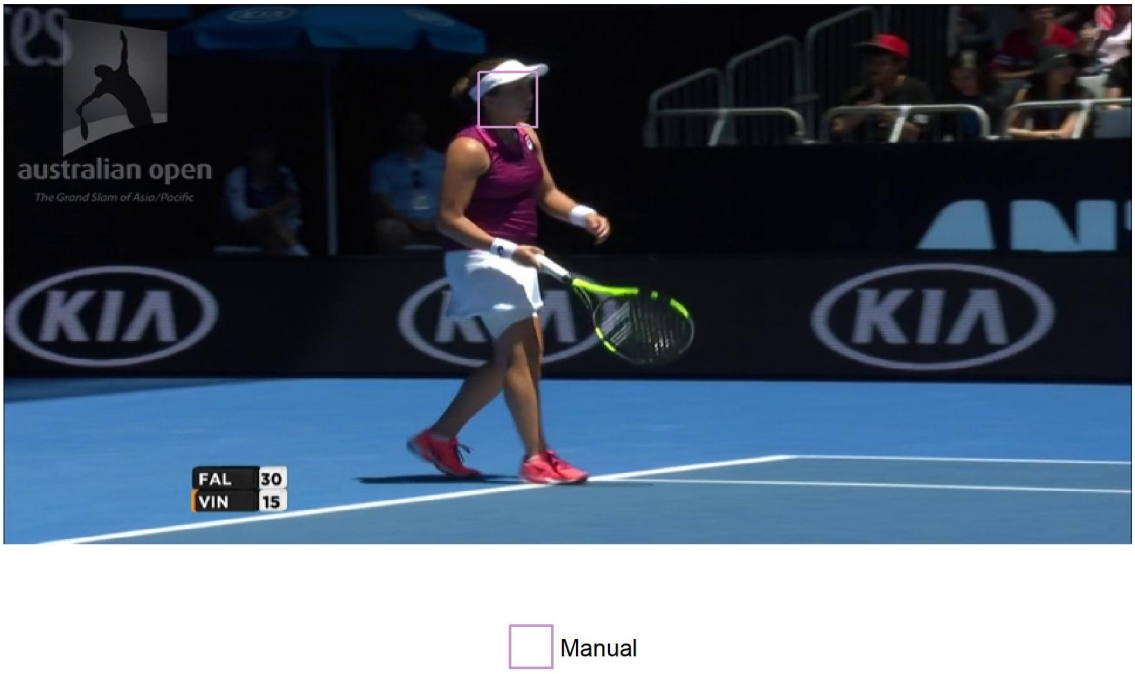

Figure 3: The face captured manually in this image was not captured by any API. The head angle, shot angle, lighting, wearing the visor and being captured across the court during the match all contribute to the struggle of the APIs.

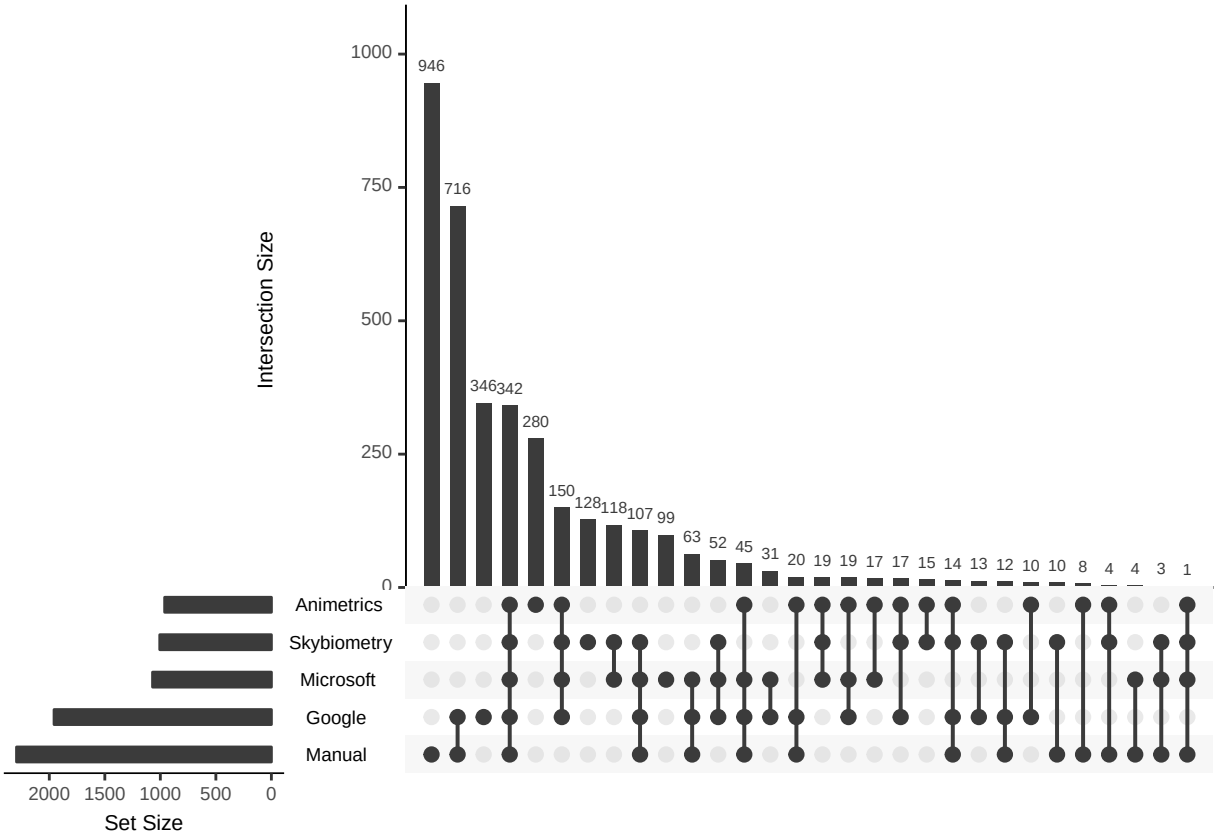

Figure 4: Set visualization, to examine intersections between API and manual face detections: bar charts show counts, by API (lower left) and for combinations (main part of plot), and dot display (bottom) indicates the group intersections. All five way intersections are examined, but only the first few 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 4: Capture rates of the APIs relative to manually tagged faces. Google tagged twice as many of the manual faces found than the next best API. It also had a larger proportion of Non-faces found, all of these faces were actual faces, usually of crowd members. However many of the faces found by other APIs were not actually faces, instead areas contained the net, clothing and logos.

	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

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.

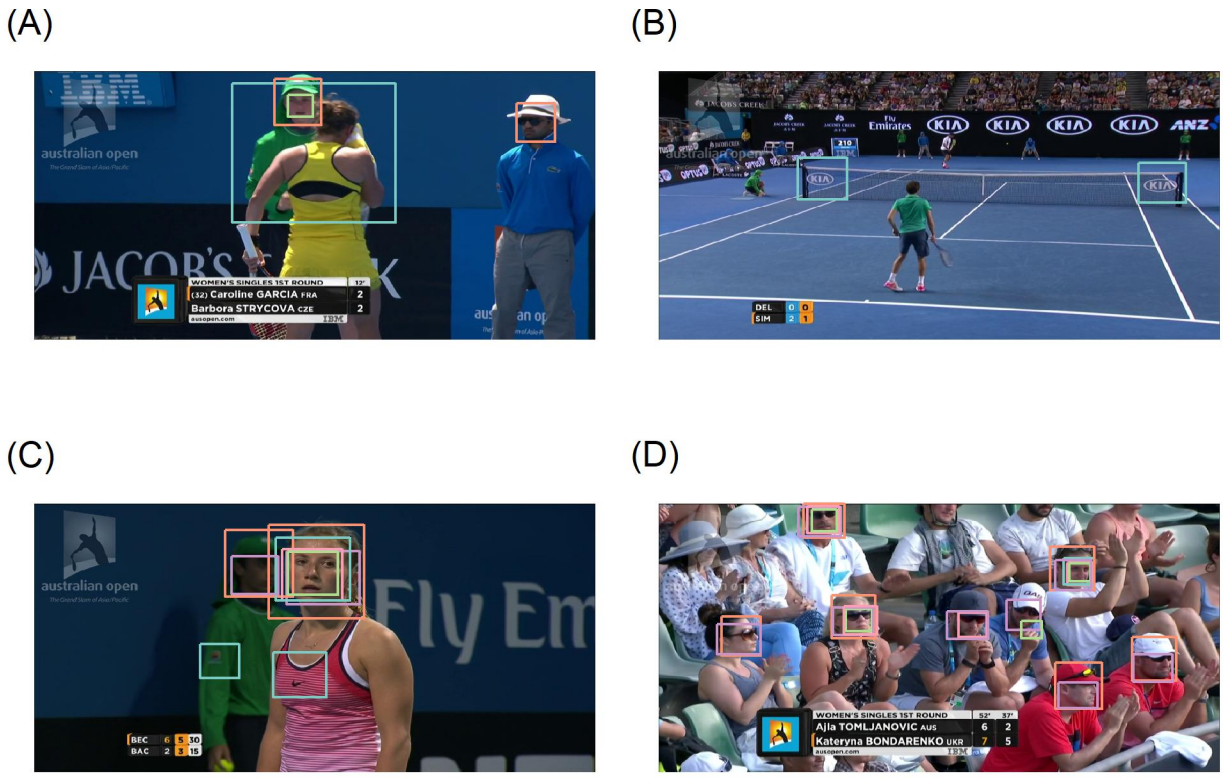

Figure 5: Four images in which Animetrics located unusual faces.

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

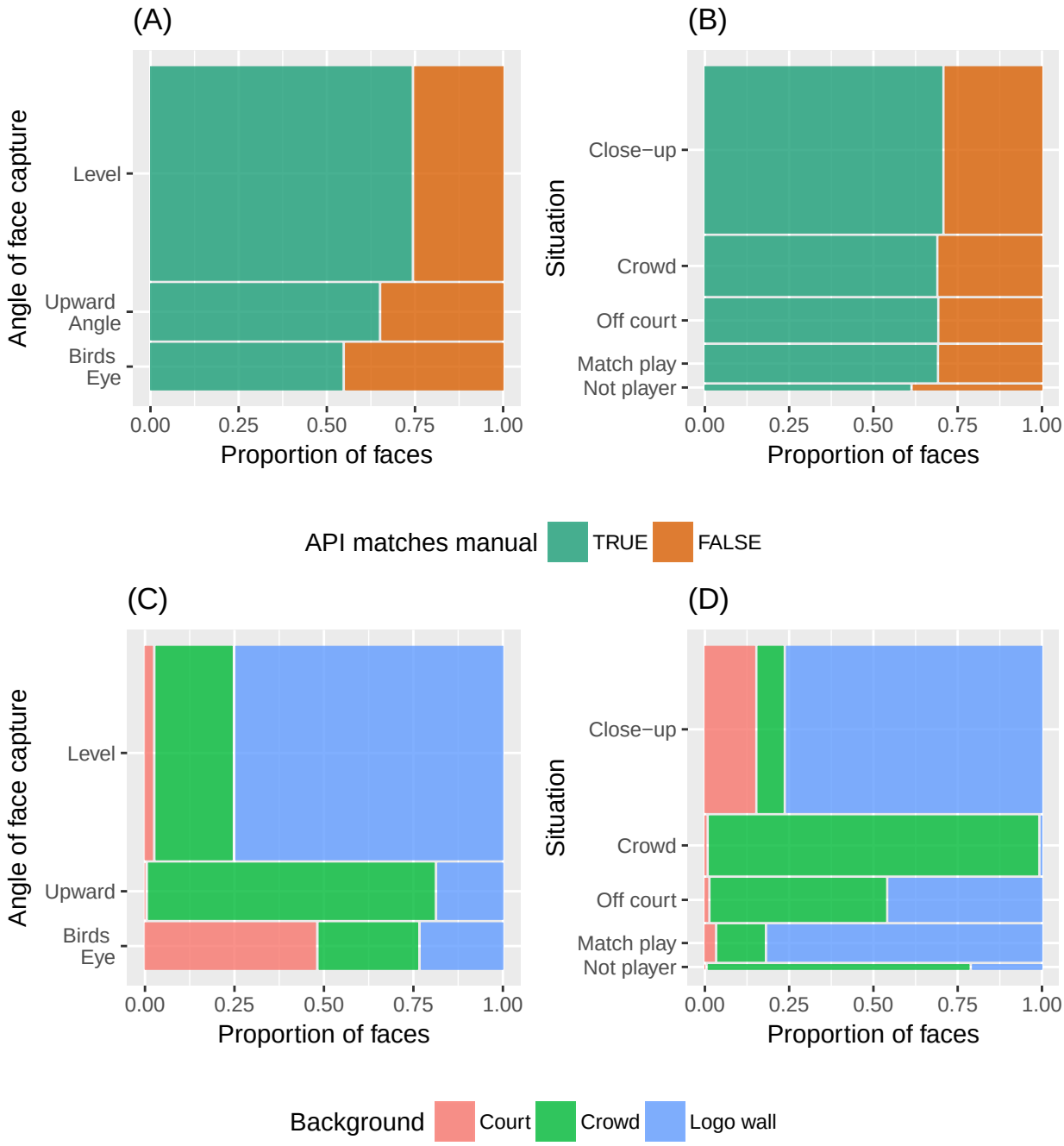

Figure 6: Relationships between court and face characteristics, and API detection, using mosaic plots: (A) Angle of face capture, (B) court situation, (C) angle of capture by background, and (D) situtation of capture by background. Most face captures occurred at a level angle, and this produces slightly better detection. Most situtation 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. Color 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. Most players do not have glasses, but head wear, either visors or hats, are very common.

Table 7 compares the face capture rate of the different APIs for the player accessories, as compared to the manually tagged face attributes. Conditional proportions are calculated two ways: (1) proportion of the manually tagged faces, (2) proportion detected with the

Table 6: Usage of head wear and glasses by players, as manually tagged. 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

Table 7: Detection rates of the APIs for different accessories, as measured against the manual tagging. Rate is the proportion of detected with the accessory relative to without. Google has higher detection rates than the other APIs when a player has head wear or glasses.

	Head wear			Glasses		
API	Yes	No	Rate	Yes	No	Rate
Google	0.48	0.69	0.70	0.43	0.60	0.72
Microsoft	0.19	0.32	0.59	0.14	0.26	0.54
Skybiometry	0.16	0.28	0.57	0.12	0.23	0.52
Animetrics	0.11	0.27	0.41	0.08	0.21	0.38

accessory relative to without. From the former conditional probability it can be seen that Google detects more faces regardless of the accessory, with slightly higher proportion when the player has no head wear or glasses. From the latter calculation, Google has less difference between the accessory or not detection rate. Overall, Google detects a higher proportion of faces, but they also have better detection rates if faces of players wearing head wear and glasses. Animetrics performed the worst when a player was wearing head wear or glasses.

3 Future Work

This paper has described the results of a survey of commonly available software for face detection when applied to stills from tennis match video. A long term goal is to detect

emotional state of the player, and track this through a match, to study the effect on performance. Several of the face recognition APIs also report an emotional state, but our analysis of this data suggests it is unreliable.

One of the problems encountered with the emotion reporting, was that individual player's faces were repeatedly tagged with one emotional state, which looked more related to the basic facial features of the player. To accurately detect an individual player's emotions would require getting something like a baseline measurement on what their face looks like in a neutral emotional state, and then measuring differences from that. To study emotions and performance, we would recommend creating a training data set similar to the face recognition data, where emotional state is manually tagged. It would also be important to take comprehensive video footage, to track a player throughout the tournament, rather than isolated images.

The result of the face recognition survey clearly points to a winner: Google's API. It detected a higher proportion of faces under different conditions and with different accessories.

Acknowledgements

The authors wish to thank Tennis Australia for access to the data. The analysis was conducted using R(R Core Team 2017), and the following packages: bookdown (Xie 2017*a*), descr (Aquino 2016), EBImage (Pau et al. 2010), ggthemes (Arnold 2017), ggmosaic (Jeppson et al. 2017), grid (R Core Team 2017), gridExtra (Auguie 2016), gtable (Wickham 2016*a*), kfigr (Koohafkan 2015), knitcitations (Boettiger 2015), knitr (Xie 2017*b*), pander (Darczi & Tsegelskyi 2015), readr (Wickham et al. 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 contains the data collected on each face by manual tagging.
- ManualClassifiedScenes.csv contains scene information about each image.

The web app for manual tagging, called taipan, is available from `https://github.com/srkob1/taipan`. It can be installed using the devtools (Wickham & Chang 2017) package function 'install_github'.

4 References

References

Animetrics Inc (2016), *Animetrics Face R API.*
URL: *http://api.animetrics.com/documentation*

Aquino, J. (2016), *descr: Descriptive Statistics.* R package version 1.1.3. Package includes source code and/or documentation written by Dirk Enzmann and Marc Schwartz and Nitin Jain and Stefan Kraft.
URL: *https://CRAN.R-project.org/package=descr*

Arnold, J. B. (2017), *ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'.* R package version 3.4.0.
URL: *https://CRAN.R-project.org/package=ggthemes*

Auguie, B. (2016), *gridExtra: Miscellaneous Functions for "Grid" Graphics*. R package version 2.2.1.
URL: *https://CRAN.R-project.org/package=gridExtra*

Barr, J., Bowyer, K. & Flynn, P. (2014), 'The Effectiveness of Face Detection Algorithms in Unconstrained Crowd Scenes', *IEEE Winter Conference on Applications of Computer Vision (WACV)* .

Boettiger, C. (2015), *knitcitations: Citations for 'Knitr' Markdown Files*. R package version 1.0.7.
URL: *https://CRAN.R-project.org/package=knitcitations*

Chang, W., Cheng, J., Allaire, J., Xie, Y. & McPherson, J. (2016), *shiny: Web Application Framework for R*. R package version 0.14.2.
URL: *https://CRAN.R-project.org/package=shiny*

Dahl, D. B. (2016), *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-2.
URL: *https://CRAN.R-project.org/package=xtable*

Darczi, G. & Tsegelskyi, R. (2015), *pander: An R Pandoc Writer*. R package version 0.6.0.
URL: *https://CRAN.R-project.org/package=pander*

Gehlenborg, N. (2017), *UpSetR: A More Scalable Alternative to Venn and Euler Diagrams for Visualizing Intersecting Sets*. R package version 1.3.2.
URL: *https://CRAN.R-project.org/package=UpSetR*

Google Cloud Platform (2016), *Google Cloud Vision API*.
URL: *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, Canada Border Services Agency, DRDC Centre for Security Science, Ottawa ON Canada K1A 0L8.

Hintze, J. L. & Nelson, R. D. (1998), 'Violin Plots: A Box Plot-Density Trace Synergism', *The American Statistician* **52**, 181–184.

Jeppson, H., Hofmann, H. & Cook, D. (2017), *ggmosaic: Mosaic Plots in the 'ggplot2' Framework*. R package version 0.1.2.
URL: *https://CRAN.R-project.org/package=ggmosaic*

Koohafkan, M. C. (2015), *kfigr: Integrated Code Chunk Anchoring and Referencing for R Markdown Documents*. R package version 1.2.
URL: *https://CRAN.R-project.org/package=kfigr*

McLean, M. W. (2014), *Straightforward Bibliography Management in R Using the RefManager Package*. arXiv: 1403.2036 [cs.DL]. Submitted.
URL: *http://arxiv.org/abs/1403.2036*

Microsoft Cognitive Services (2016), *Microsoft Face API*.
URL: *https://docs.microsoft.com/en-us/azure/cognitive-services/face/*

Pau, G., Fuchs, F., Sklyar, O., Boutros, M. & Huber, W. (2010), 'Ebimage—an r package for image processing with applications to cellular phenotypes', *Bioinformatics* **26**(7), 979–981.

R Core Team (2017), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Skybiometry (2016), *Skybiometry API*.
URL: *https://skybiometry.com/documentation/*

Wickham, H. (2016*a*), *gtable: Arrange 'Grobs' in Tables*. R package version 0.2.0.
URL: *https://CRAN.R-project.org/package=gtable*

Wickham, H. (2016*b*), *httr: Tools for Working with URLs and HTTP*. R package version 1.2.1.
URL: *https://CRAN.R-project.org/package=httr*

Wickham, H. (2017), *tidyverse: Easily Install and Load 'Tidyverse' Packages*. R package version 1.1.1.
URL: *https://CRAN.R-project.org/package=tidyverse*

Wickham, H. & Chang, W. (2017), *devtools: Tools to Make Developing R Packages Easier*. R package version 1.13.2.
URL: *https://CRAN.R-project.org/package=devtools*

Wickham, H., Hester, J. & Francois, R. (2017), *readr: Read Rectangular Text Data*. R package version 1.1.1.
URL: *https://CRAN.R-project.org/package=readr*

Xie, Y. (2017*a*), *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.4.
URL: *https://github.com/rstudio/bookdown*

Xie, Y. (2017*b*), *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.17.
URL: *https://yihui.name/knitr/*