Detecting Facial Expressions in Professional Tennis Matches

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

2017

Contents

Introduction

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

In this report, we investigate the performance of several popular facial recognition software's through their Application Programming Interfaces (APIs), and evaluate their performance when applied to the broadcasted videos of elite tennis matches. Using the broadcasted videos gives an objective insight to emotions as the player progresses through a match. While it is impossible to know the thoughts and feelings of a player during a match, professionals may be able to infer this information through results produced by a recognition software. As opposed to the approach of previous studies that have used Player's recollections after a game to determine their emotions.

Making use of the recognition software's currently available presents a challenge as high performance sports are not the intended uses of such software's. Their capabilities are often limited to their intended security and surveillance uses. Barr [2014] addresses the 'lack of robustness of current tools in unstructured environments' that this paper faces and applies to a sports environment. This report aims to analyse the application of these software's to a broadcast to find a suitable software and API to use to analyse a pre-recorded tennis broadcast file.

Project Aim

The aims of the present study were to determine the feasibility of using currently available facial recognition algorithms for extracting facial information from players during broadcasts of professional matches by comparing the performance of several popular facial recognition APIs. This limited selection was based on accessible APIs that we believed would produce appropriate and useful facial recognition. The performance of the evaluated software was compared against manual classification obtained notation tool developed by the authors. In addition to looking at the overall performance, we also evaluated image factors that influences the performance of each service.

Sample and sampling approach

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

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

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

The sample included images that contained the faces of many people, this included players, staff on the court and fans in the crowd. These faces were included in the manual annotations as they were likely to be found by the software selected. We felt including these additional faces would not only increase the sample by which to judge the software's capabilities but also allow provision of information on how to differentiate between players and other people for further research. Therefore the sample was not filtered at this initial stage.

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

Selecting the software

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

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

From the software's analysed we considered availability for use within the time frame of the report. This led us to choose Animetrics FaceR. The report outlines that for Animetrics, 'one requirement is that image/face proportion should be at least 1:8 and that at least 64 pixels between eyes are

required'. We realize this could present challenges given our data set. It will also allow for an extension from detecting to recognizing people in the data set.

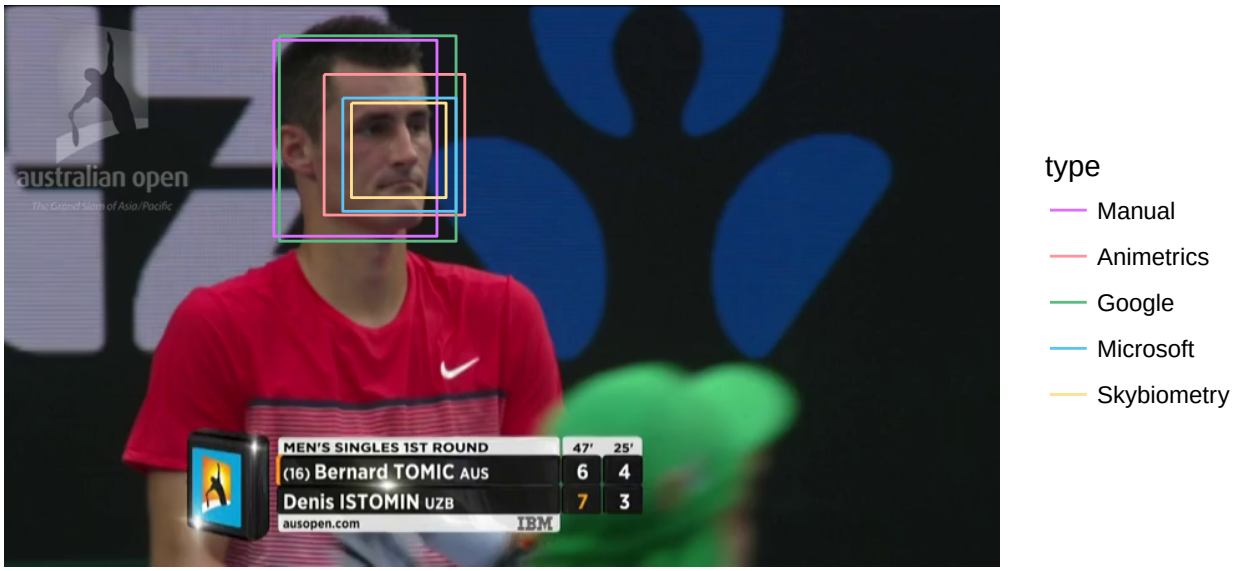
After considering several other off-the-shelf products, we did not choose any other software's from those analysed as they were not as readily available as other products on the market.

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

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

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

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



Methodology

An annotation tool was constructed to create a base for comparative analysis, we refer to this as Manual Classifications. These manual Classifications involved describing the features of the Scenes.

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

The aim was to collect specific information on each different face within the scene. To determine which of the sometimes many faces in the scene it would be reasonable for software to detect a standard was created for reasonable detection. This was based on the attributes provided in Table 1, below:

Table 2: Software solutions specifications.

Name	Input	FileType	ImageSizeMin	ImageSizeMax
Animetrics-FaceR	Images	NA	NA	NA
Google-Vision	Images	JPEG, RAW	NA	4MB
Microsoft	Images	JPEG,PNG	1KB	4MB
Skybiometry	Images	JPEG,PNG,BMP	NA	2MB

It was decided that it would be reasonable for a software to detect was decided to be a face size larger than 20 by 20 pixels, as it was specified that the software had minimum distances between the eyes on a face for recognition. (reference table with values) If it was the face of a player it was recorded if it obviously showed their face. The back of the head was not able to be picked up by any software, so after a demo trial these faces were classified manually but reclassified as other. Crowd shots provided difficultly in determining which faces were reasonable to classify. As these faces were not the intended targets of the recognition these faces were contributing to our understanding of the software. The same face size standard applied to crowd members, but focus was placed on the most prominent faces. For each of these faces, we collected information on the following attributes:

Table 3: Attributes of an individual face within an image. The most appropriate option from the list was selected for each attribute.

Attribute	Choices
Detectable Whose Face	Player, Other Staff Member (on court), 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

Manual Annotation

To record the details of attributes for each face and scene a Shiny [2016] App was created. We called this Application our ManualClassificationProgram[1]. This helped to provide information for all attributes quickly and consistently. The Shiny [shi, 2017] package provides a 'framework for building web applications using R'. It allows for the creation of web applications using R rather than 'requiring HTML, CSS, or JavaScript knowledge.'

The program we created was an interactive application that worked by presenting an image, using the imager Barthelme [2016] package, from the sample of images that had not yet been considered, appearing underneath were a set of radio buttons corresponding to the Scene Attributes in Table 2 as shown above.

If there was a face in the image the annotator was able to highlight a section of the image to create a square 'Face Box'. This changed the display and presented a set of Attributes with radio buttons, this allowed information to be recorded for the face in the specific 'Face Box'. This recorded the x and y coordinates of the corner points of a box drawn by the mouse, and when the save button was hit it saved all the radio button selections and the 'Face Box' coordinates to a CSV file[2].

When a face was not selected, the radio buttons showed the Scene attributes and the radio buttons with the possible selections the annotated was able to choose from. When in this display, selecting the save button would then save the Scene selections to a specific CSV file[3].

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

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

Software Recognition

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[4], Microsoft[5] and Skybiometry[6]. 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.

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

[1] https://github.com/mvparrot/face-recognition/blob/master/ManualClassificationProgram.R
[2] https://github.com/mvparrot/face-recognition/blob/master/ManualClassifiedFaces.csv
[3] https://github.com/mvparrot/face-recognition/blob/master/ManualClassifiedScenes.csv
[4] https://github.com/mvparrot/face-recognition/blob/master/SoftwareRequestScripts/animetrics.R
[5] https://github.com/mvparrot/face-recognition/blob/master/SoftwareRequestScripts/microsoftAPI.R
[6] https://github.com/mvparrot/face-recognition/blob/master/SoftwareRequestScripts/autoSkybiometry.R

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, ManualClassifiedFaces[7] and ManualClassifiedScenes[8].

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[9], 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 software Type the Face Bounding Box was determined by. 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[10].

Analysis

The statistical analysis conducted to summarize and assess the validity of the method. This method allowed all the software and the faces they found to be compared.

Using the data set[11] 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 2 .

However, individual faces are not beneficial if they do not correspond to the faces manually annotated. Figure 3 was created by defining groups depending on the softwares that recognized each particular face. The UpSetR [2017] package helps visualise set intersections. Where in this circumstance Face Bounding Boxes may overlap on the same face, each bar shows the number of individual faces that have Face Bounding Boxes resulting from each of the softwares highlighted below the bar.

Following, we considered the sizes of the Boxes created in Figure 4, this was an attempt to determine how the software performed when presented with smaller faces. The boxplot uses the size of the Face Bounding Boxes, and contrasts based on the software the Boxes were found from.

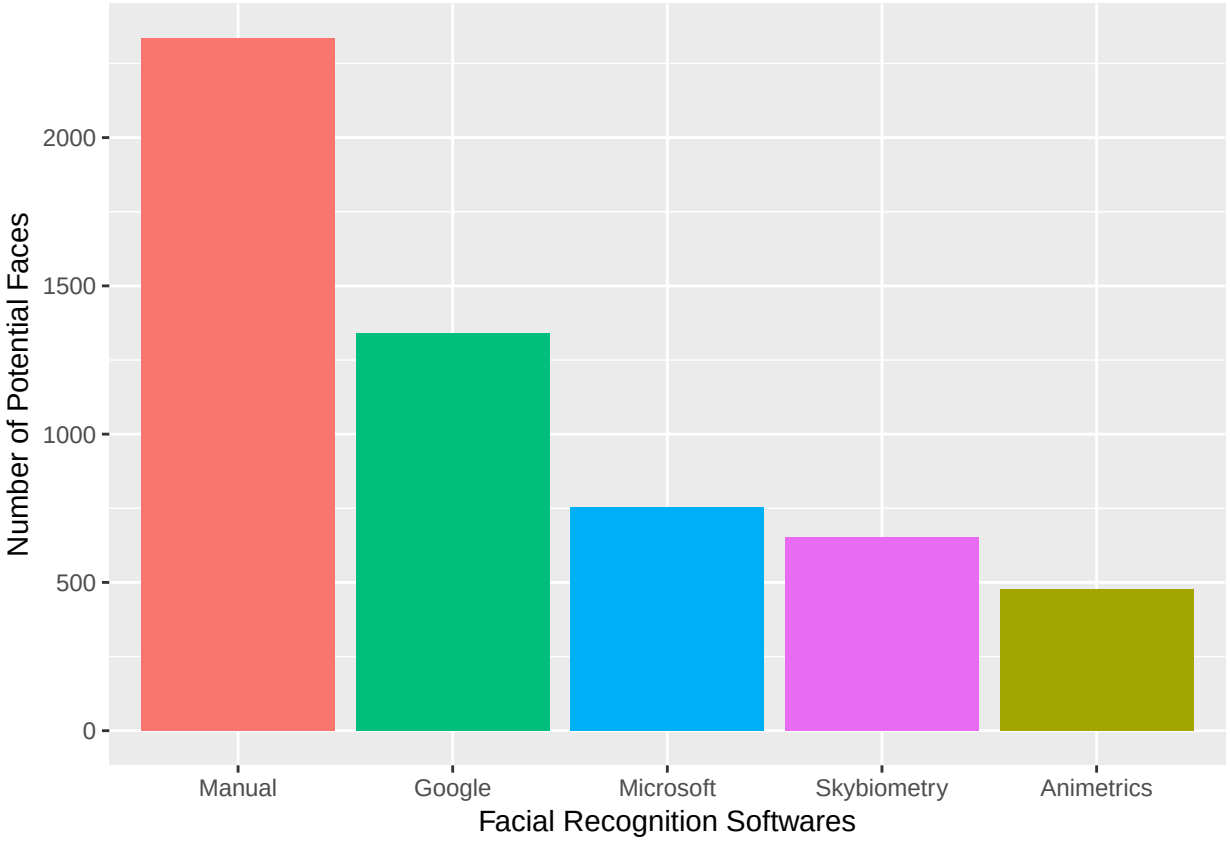
To determine how each software reacted to the same face we looked at an image that contained only

[7] https://github.com/mvparrot/face-recognition/blob/master/ManualClassifiedFaces.csv
[8] https://github.com/mvparrot/face-recognition/blob/master/ManualClassifiedScenes.csv
[9] https://github.com/mvparrot/face-recognition/blob/master/ALLmetIMG.csv
[10] https://github.com/mvparrot/face-recognition/blob/master/ALLmetIMG.csv
[11] ALLmetaIMG

finding faces that were recognized and grouping them by the softwares

Skybiometry results accounted for the 255 smallest recognized faces. However this is not necessarily a benefit to this research, as these are not all faces.

Results



The bar chart in Figure 2, Figure 4 above shows the number of Bounding Boxes produced by each software, comparing the height of the bars indicates that Google's Facial Recognition software recognized almost 1000 more faces than the next best software, Microsoft.

-Google more than others

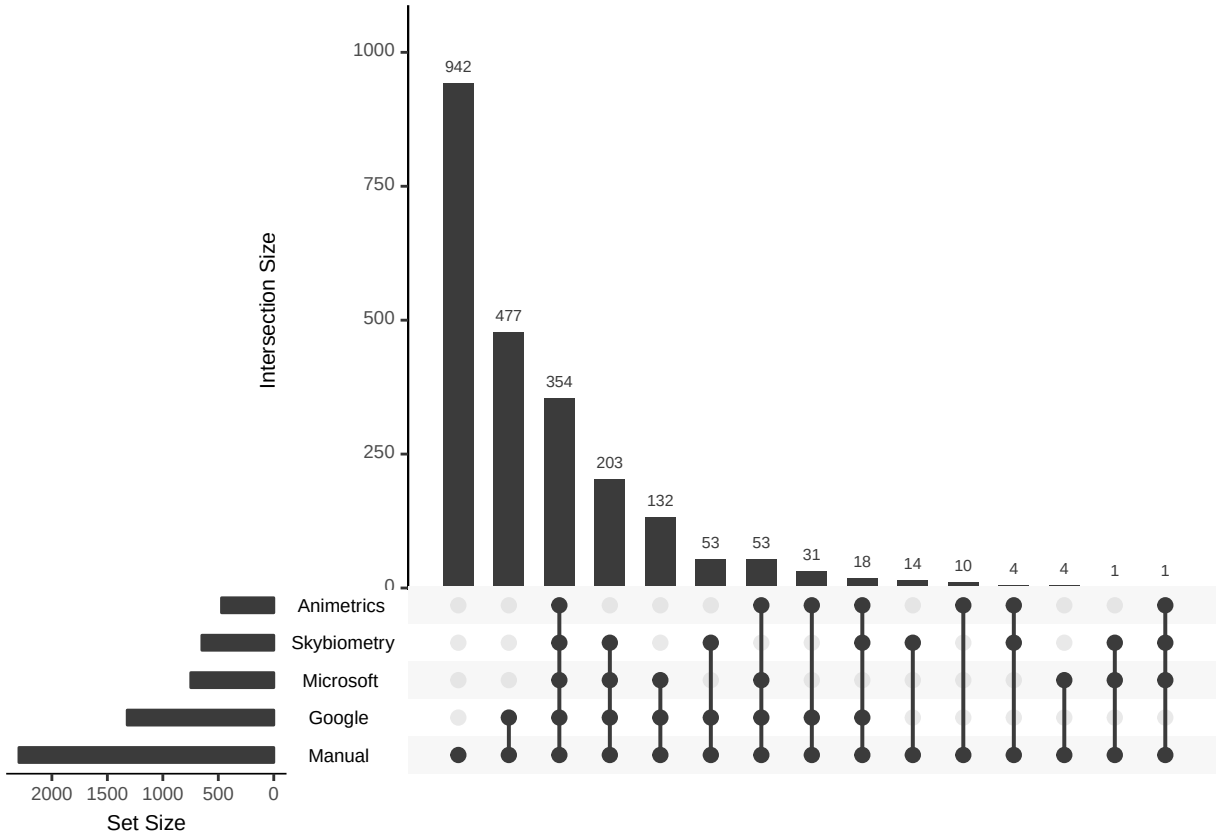
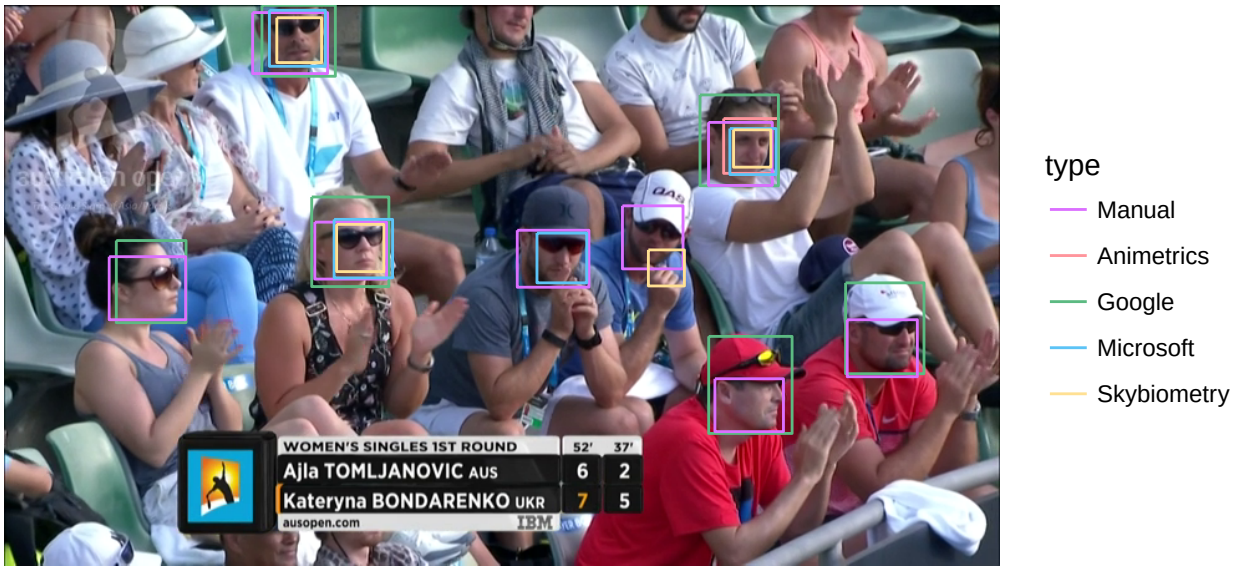


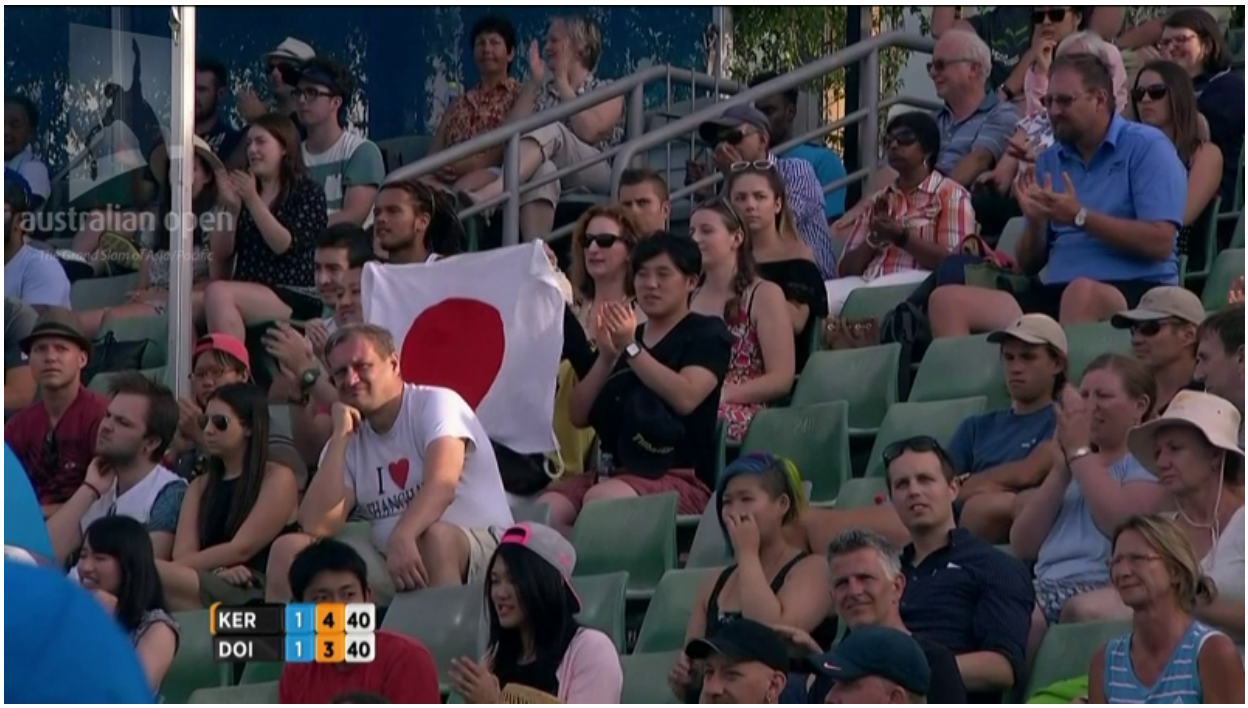
Figure 5 Bar Chart shows the Face Bounding Boxes that were recognized by multiple software. The largest group with 809 faces is Face Bounding Boxes only found by Manual annotations. The following group was Boxes representing the same faces being recognized by both Manual annotations and the Google API.

On average, Google maps the largest boxes around faces, and has the largest box recognized in the set. On average, the smallest Face Boxes are created by Skybiometry.



As before, the 'face' is not as we would hope it would be classified. Just right of the center, the smaller box actually captures a fist, not a face.

I4 shows the sensitivity of the software. We may consider this too sensitive as it recognized the eye and nose of the ball boy on the court.



As seen in this image of the crowd, the software is performing reasonably well. It does recognize the faces, yet only one not recognized by the other software, this may be due to the sunglasses.

This image demonstrates that the it is not necessarily the smaller faces that Skybiometry recognizes, but it puts smaller bounds on the facial features than other software.

	Animetrics	Google	Microsoft	Skybiometry
TRUE	476	1340	753	651

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.

The sample used contains all the manually annotated faces and all the faces recognized by the four software.

To consider how many Type I errors occurred, where a face was detected incorrectly, we look at the Bounding Boxes that do not match manually annotated faces.

Table 4 shows whether the potential Face Bounding Boxes match faces that were annotated during the manual classifications. Where the FALSE row denotes where software's Face Bounding Boxes do not coincide with manually annotated faces. The tables shows that Google found 38.70% of the 90.34% of the Faces found that matched Faces also identified manually.

A potential face detected that does not match a face manually annotated occurs for 9.66% of all the

Faces detected by the software. This is especially high for Google with 289 faces identified.

All the Face Bounding Boxes that Google found which do not match manually annotated faces were correctly identifying faces. This exhibits the occurrences of errors.

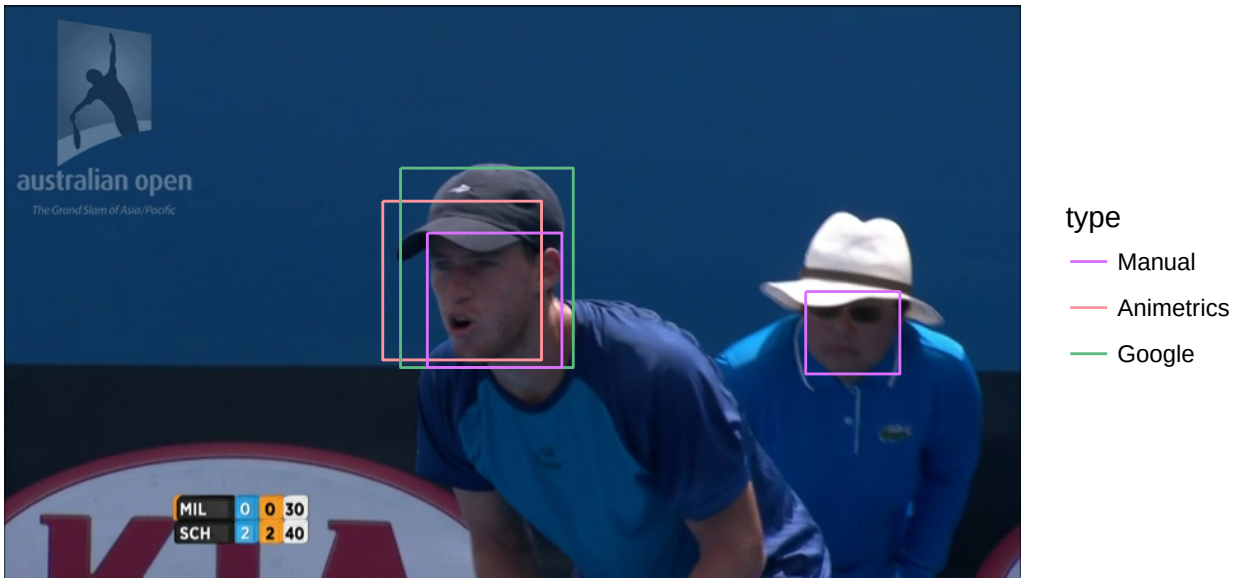
We then considered the characteristics of the images that the software found Potential Face Bounding Boxes[12] in.

situation	bg	shotangle	detect	count
Court player close-up	Logo wall	Player Shoulder Height	Player	735
Court in play	Logo wall	Player Shoulder Height	Player	241
Crowd	Crowd	Upward Angle	Fan	151
Court player close-up	Court	Birds Eye	Player	133
Court player close-up	Logo wall	Player Shoulder Height	Other staff on court	119
Off court close up of player	Logo wall	Player Shoulder Height	Player	116
Crowd	Crowd	Player Shoulder Height	Fan	111
Court in play	Logo wall	Player Shoulder Height	Other staff on court	81
Off court close up of player	Crowd	Player Shoulder Height	Player	65
Crowd	Crowd	Birds Eye	Fan	55

Figure 6 displays the feature combinations that produced the most potential face Bounding Boxes recognition by all four software. The most common shot is a crowd shot. The second row in the table with 830 faces recognized is more interesting than the first result. This useful scene is an image of a Court Player Close-Up in front of a Logo Wall, taken at Player Shoulder Height.

These attributes typically represent an image similar to the following:

[12]These boxes represent an area of pixels that are a potentially recognized face.



person	situation	bg	shotangle	count
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA

table 6 shows that the potential Face Bounding Boxes Google returned were found in images that had the same characteristics of those that were manually annotated for faces. Without looking at each individual image this Table confirms that the potential Face Bounding Boxes Google located will likely be reasonable, and actually contain faces.

The best scenes for facial recognition have been found, given this information the following table 7 considers the Characteristics of the individual faces found within those scenes. For this section we chose to consider only the faces that were manually annotated as players, with the intention of not basing results on recognition of undesirable faces.

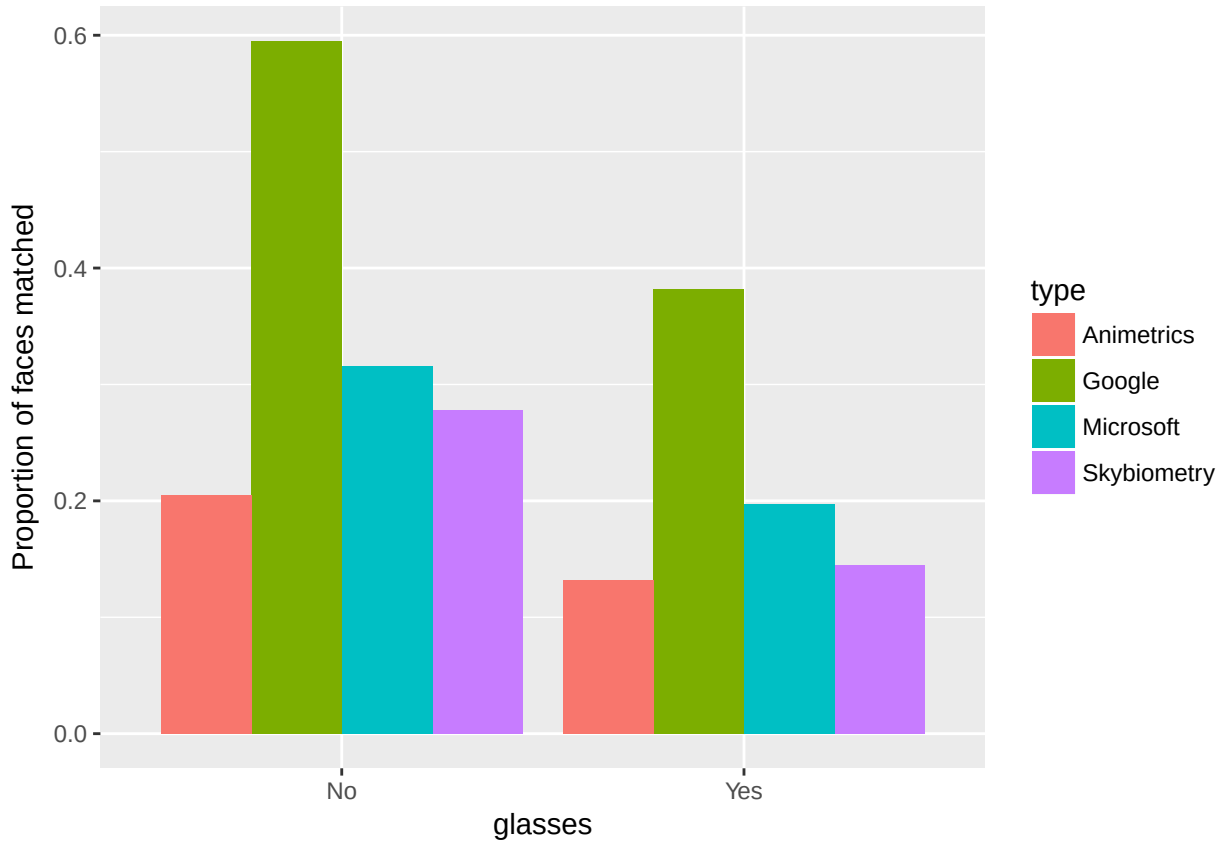
visorhat	glasses	headangle	lighting	obscured	detect	count
No	No	Other	Partially shaded	No	Player	166
Yes	No	Other	Partially shaded	No	Player	111
No	No	Profile	Partially shaded	No	Player	102
No	No	Other	Partially shaded	Yes	Player	85
Yes	No	Other	Partially shaded	Yes	Player	58
Yes	No	Profile	Partially shaded	No	Player	54
No	No	Profile	Partially shaded	Yes	Player	38
No	No	Profile	Shaded	No	Player	36
Yes	No	Other	Shaded	No	Player	32
Yes	No	Front on	Partially shaded	No	Player	30

table 7 utilizes a set of faces that were Manually annotated and also found by Google. Nine of the top ten facial characteristic combinations contained faces that were not wearing glasses. The head angle describing the face angle was 'Other'[13] for nine of the top ten facial characteristic combinations.

No	Yes
928	29

The extreme disparity between the amount of faces with glasses to without them means that the occurrence of these attributes across the software must be considered proportionally.

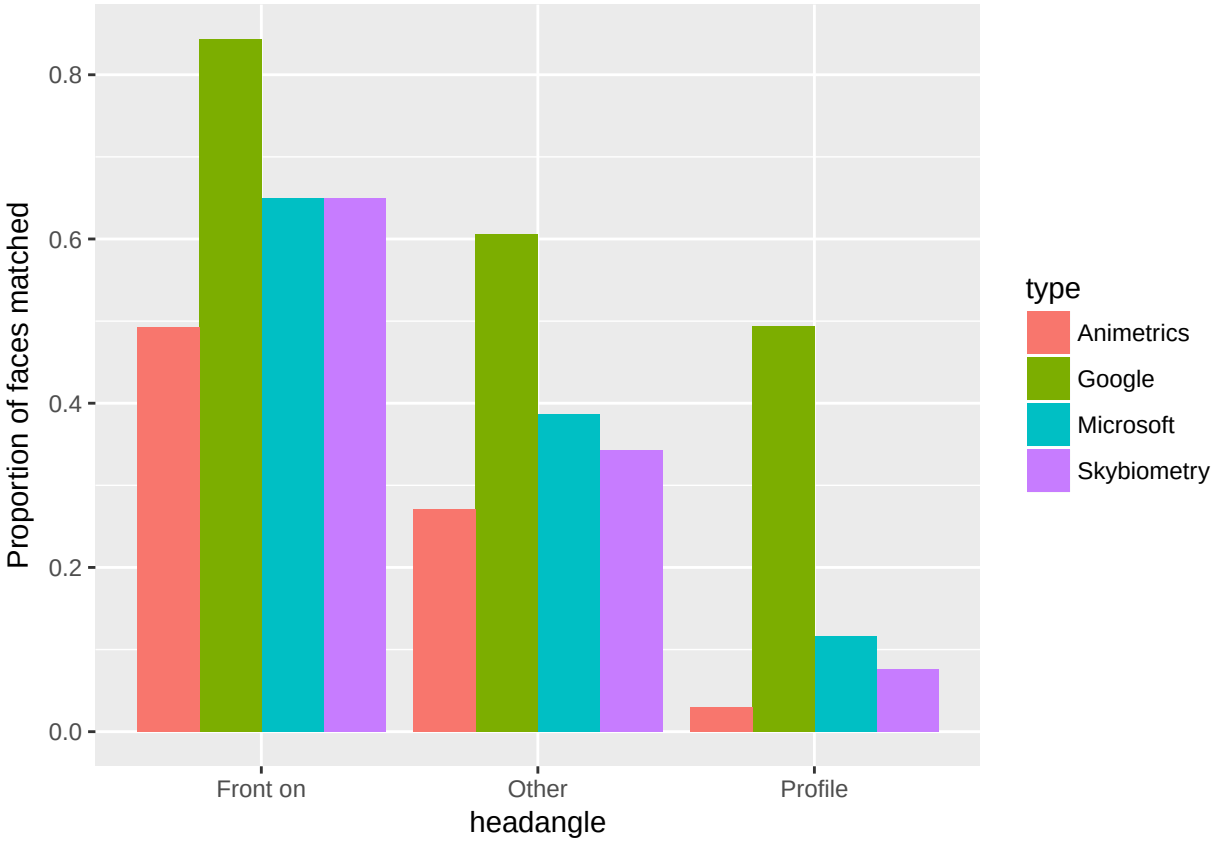
[13]Definition for head-angle factor level 'Other' found in..



Graph 2 above shows that Google outperforms the other software, with or without glasses. When there are no glasses worn Google finds over 60% of the manually annotated faces.

Front on	Other	Profile
113	551	293

The amount of faces found by Google with the head-angle "Other", is much larger than the amount of faces with the head-angle "Profile" or "Front On". Therefore this should also be considered proportionately.



Graph 3 displays that Google performs much better in comparison to the other software when the head-angle is Profile. This is outperforming unusually well, however this could also be due to the poor performance of the software in this circumstance.

Table 2 = Describe the images/boxes identified by the algorithms–what are they typically like? what is the area represented?

Table 3= Evaluate the performance: what is the overall accuracy of each algorithm (sample should be annotated faces + all boxed identified by algorithms) How often is type I error made (a face detected incorrectly)?

	Manual	Animetrics	Google	Microsoft	Skybiometry
TRUE	2335	476	1340	753	651

How often type II error (a face is incorrectly NOT detected)?
3 of the four software's

It should be noted that there were results where a single face was recognized twice within the same image. This was a very unusual result and is notable as a point of interest but given it only occurred for Animetrics, it is not worth basing decisions on this unusual result.

Table 4 = Identify possible explanatory factors to performance; Does accuracy vary by lighting conditions? face size? obscuring factors? angle? etc.

person	situation	bg	shotangle	detect	count
Person	Court player close-up	Logo wall	Player Shoulder Height	Player	544
Person	Court player close-up	Logo wall	Birds Eye	Player	25
Person	Court player close-up	Logo wall	Upward Angle	Player	21

table 8 shows how many images with potential player's faces Google recognized. This displays a vast gap between the amount of potential faces found given the different shot angles. Shoulder height is an optimum angle for a Face Bounding Box.

Considering only the Shoulder Height Angle, the following Graph looks at how accessories affected Google's recognition.

Discussion

Graph 1, the Bar Chart of the Face Bounding Boxes promotes Google as the best possible software for a facial recognition application in tennis. According to Table 4 Google had 38.23% of the 9.66% potential face boxes not annotated manually. It was considered that this may have shown Google's API may have been finding more unwanted faces than the other software. However, visual inspection showed that these were actually faces. This is considered in Table 6, which showed that some of the faces were found in a crowd setting. Therefore some of these could be considered irrelevant, possibly being crowd faces. These were deemed unlikely to be detected and neglected during manual annotation. Interestingly, Animetrics had the least amount of matches to Manually annotated faces. Looking into this further showed that Animetrics results contained potential Face Bounding Boxes that did not contain faces.

The images were all considered manually. The scene information was recorded and the combinations were shown to find how many potential face Bounding Boxes were found with the combination of scene attributes. This showed that crowd members faces were often recognized, this is both helpful and unhelpful as it shows a strong ability of Google's algorithm to recognize faces, even when these faces are not the goal of the research.

Image 1 shows the images preferable for future research. Where the faces will be recognized and allow both the identity and emotion of the players to be recognized.

Google gave the optimum results in this image as it found the face of the player, but did not locate the face of the staff member behind him on the court.

While Google's recognition's mostly matched the Manually annotated set of faces, there were some that did not. These were all actually faces and were missed during manual annotations.

Table 6 shows that 190 of the faces that were not found manually occurred in the scene of a Court player close-up, with a background of a logo wall, where the shot was taken at player shoulder height.

This shows it was performing extremely well and not resulting in unexplained face Bounding Boxes as some of the other software were. This is a strong indicator that applying Google's software for further research would result in the recognition of desired faces.

These results have contributed to the choice of Google given the optimum scene as described above. Implementing a filtering process, either using current or alternative footage[14] would allow Google to provide Tennis Australia with the most applicable results.

We moved to considering the characteristics of the faces. This helped to distinguish where Google performed well in comparison to the other software options.

Table 7 showed the combinations of attributes that were found for each face[Given that information was not recorded where Google provided facial recognition for faces not Manually annotated these could not be considered.]. The use of accessories, Glasses and Visors or Hats, was considered as the Australian Open takes place on both indoor and outdoor courts. To apply this research all courts that elite Tennis players compete on had to be included. It was assumed that outdoor courts would led to the use of these accessories and these accessories may contribute to the performance of a recognition software. It may be implied by the table that Glasses prohibits recognition as all but one of the combinations have 'No' for the Glasses variable. However, we are cautious of validating this as Table 8 tells that there are many more faces recognized, by both the Google recognition's and manual annotations, that do not have Glasses. This disproportionate sample of faces with Glasses means that we considered it proportionally rather than as a total.

Graph 2 demonstrates that the presence of Glasses on faces annotated manually did affect recognition by Google's algorithm, while it outperformed the other software in both instances, faces were identified more often if the person did not wear glasses.

Moving to looking at the characteristics that were considered manually shows that the use of glasses by players coincided with less faces being annotated.

The box-plot in graph encourages our comparisons to not consider the size of face bounding box as a measure of how the software performs on small faces.

Challenges

It is understandable that there would be many more faces to recognize in these shots than in shots where there is only a player, and therefore many more faces recognized. This provides many faces to sort through to find emotions of a player.

We faced the challenge of accessing usable images of players, and specifically their faces. - Availability of software - Using the software - Time constraints

Method, automated the process to reduce data cleaning and help group characteristics

Pricing

Conclusions

Employ the Google Vision API, which would allow the use of still images, or video (TEST VIDEO) files, reducing the need for stills. This product - cost in relation Ease of access - API calling

[14] See future research for further information on these options

Future Work

The Long Term goal of this research 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 used in this research which produced the best recognition in images does not yet have the potential to detect faces and emotions in a video.

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

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

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

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

References

References

Shiny server: Easy r web apps, jan 2017. URL https://www.rstudio.com/products/shiny-2/.

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

Simon Barthelme. *imager: Image Processing Library Based on 'CImg'*, 2016. URL https://CRAN.R-project.org/package=imager. R package version 0.31.

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2016. URL https://CRAN.R-project.org/package=shiny. R package version 0.14.2.

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