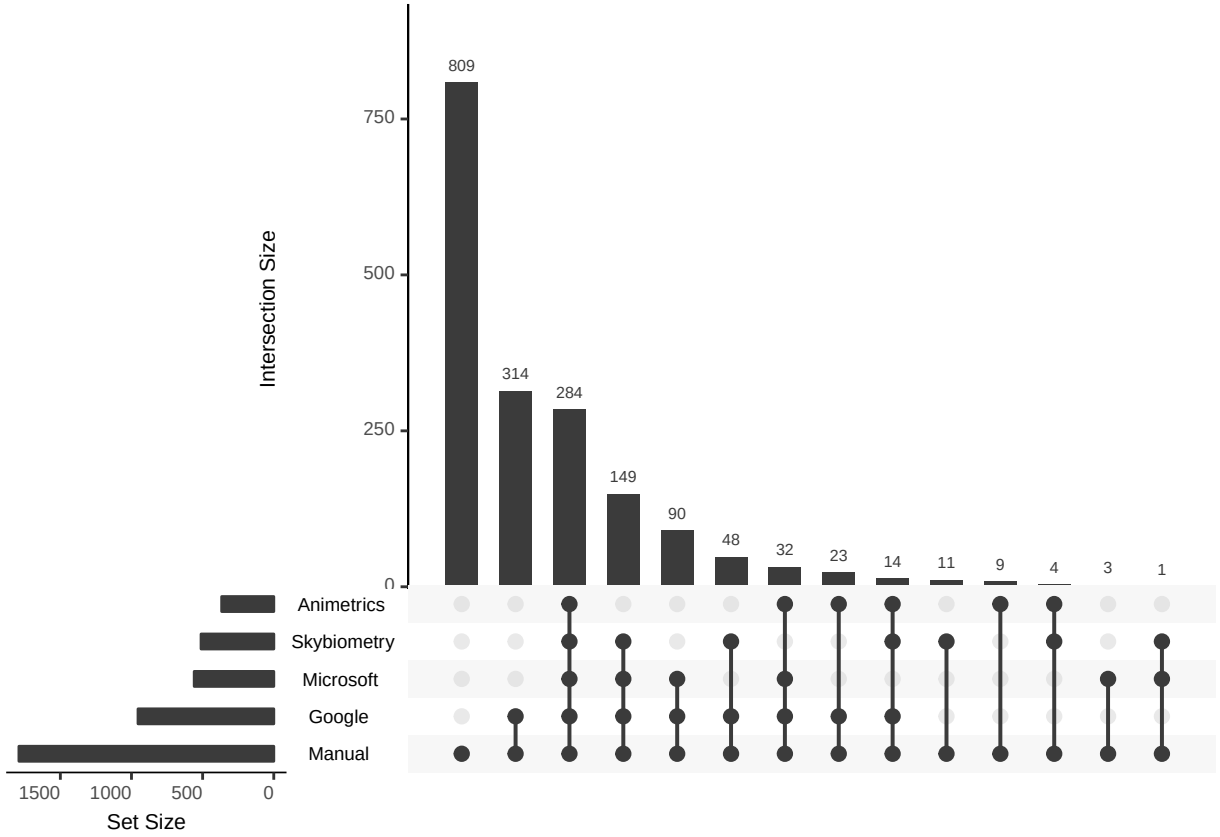
Detecting Facial Expressions in Professional Tennis Matches

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

November 19 2016

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 analysing 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 matchplay.

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. 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.

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 major aim 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. The performance of the evaluated software was compared against manual classification obtained notational 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 annontations as they were likely to be found by the softwares selected. To be able to contrast the abilities of the softwares and provide information on how to differentiate between players and other people for further research the sample was not filtered at this 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 softwares 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 selction 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 timeframe 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 realise this could present challenges given our dataset. It will also allow for an extension from detecting to recognising people in the dataset.

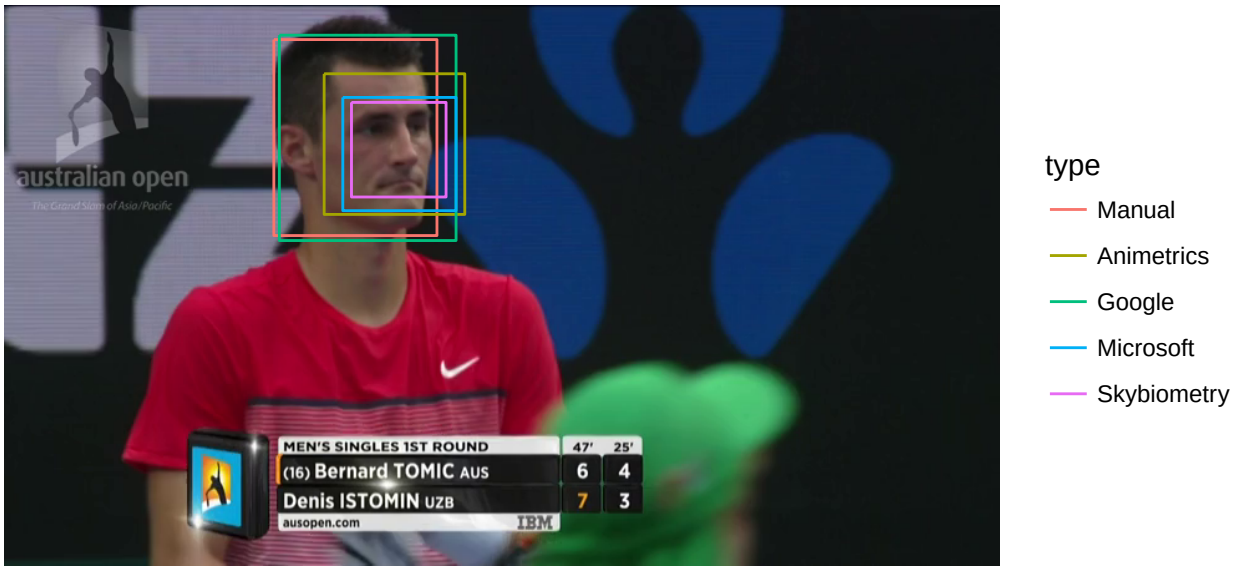
After considering several other off-the-shelf products, we did not choose any other software's from those analysed as they were not as readily available as other products on the market.

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 company's 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 softwares were viable, we used the following 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 Player, Tra

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 softwares 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 softwares. 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 selections of attribute values for each face and scene a Shiny Chang et al. (2016) App was created. We called this Application our ManualClassificationProgram. This helped to provide information for all attributes quickly and consistently. The program worked by presenting an image, using the imager (**???**)

package, from the sample of images that had not yet been considered, appearing underneath were a set of radio buttons corresponding to the Scene Attibutes in Table 2 as shown above.

If there was a face in the image the annotator was able to highlight the square 'Face Box'. This changed the display and presented a set of Attributes with radio buttons, this allowed information to be recorded for the specific face. 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[1].

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[2].

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.

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[3], Microsoft[4] and Skybiometry[5]. 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 anomoly 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 information. Some of the softwares also provided a more detailed level of information.

The collation of the results from the Manual Recognition Program created two CSVs, ManualClassifiedFaces[6] and ManualClassifiedScenes[7].

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[8], 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 indentified in the image by counting each additional recognised 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 softwares were recognising 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

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

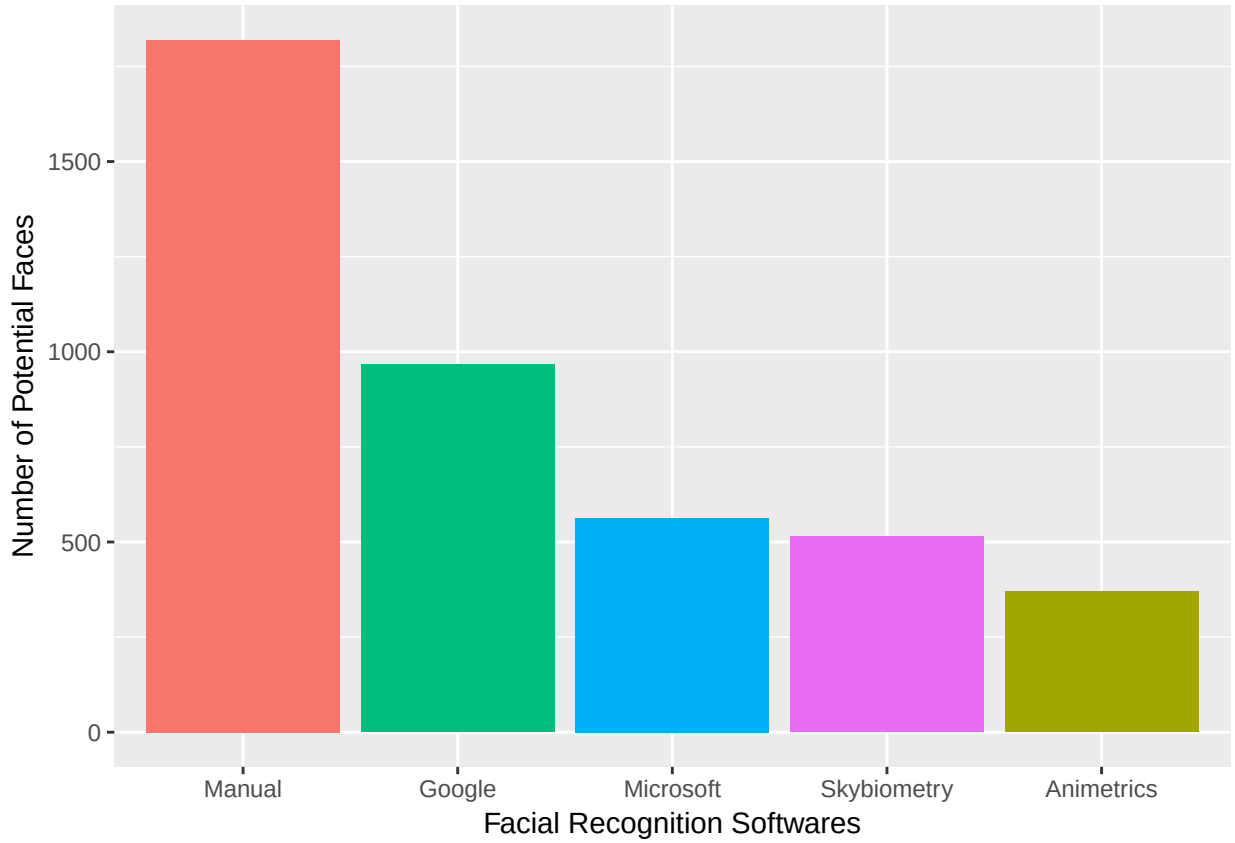
information of each face and compares the intersecting regions of the polygons created by the x,y coordinates of Manual Faces and other software's faces, to determine if the same face was recognised. We determined the ratio of intersecting area to total area must be greater than 0.1 to be considered 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[9].

Analysis

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

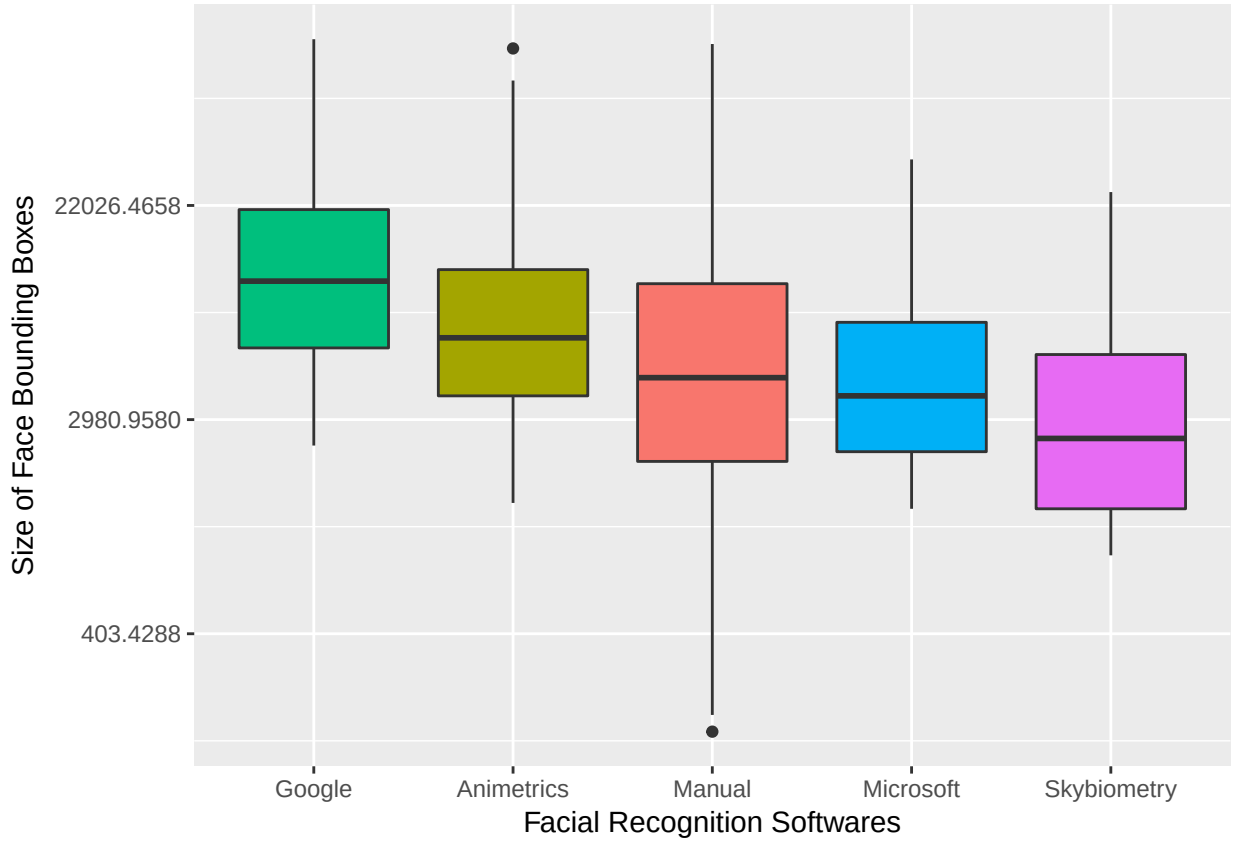
Using the data set[10] of the combined results, we were able to compare the performance of the softwares. Firstly, we considered how many individual faces the softwares were able to detect in Figure 2 .

Results



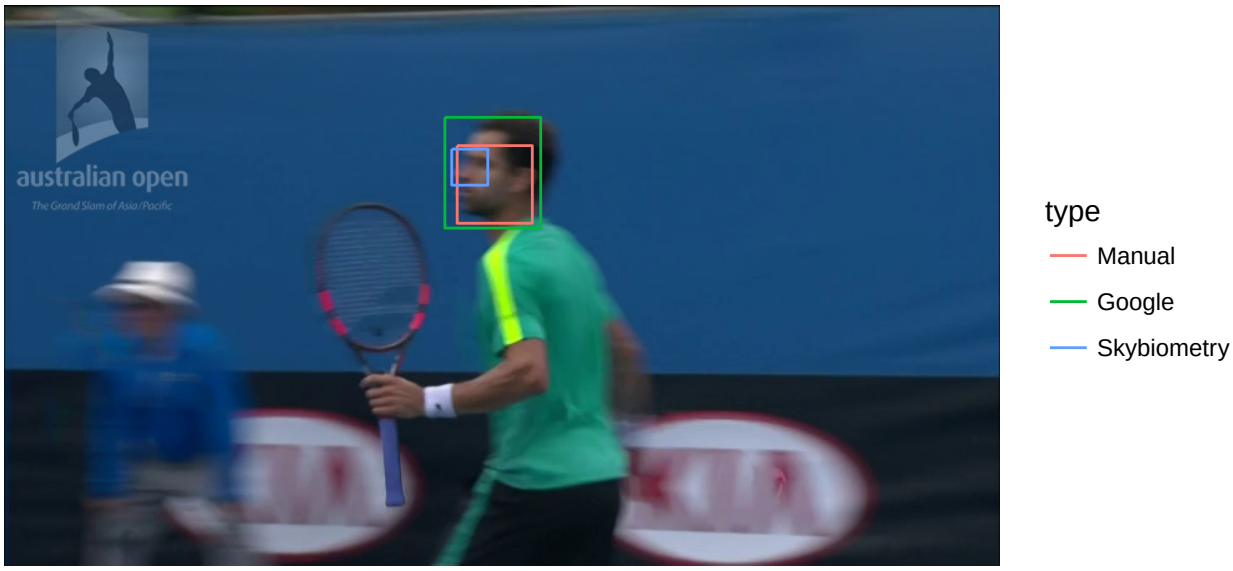
The bar chart in Figure 2 above shows the number of Bounding Boxes produced by each software, comparing the height of the bars indicates that Google's Facial Recognition software recognised almost 1000 more faces than the next best software, Microsoft.

[9] https://github.com/mvparrot/face-recognition/blob/master/ALLmetIMG.csv
[10] ALLmetaIMG



The box and whisker plot in Figure 3 shows the differences between the sizes of the 'Face Bounding Boxes' produced by each software. Where a 'Face Bounding Box' refers to the size of the square the software provided as a location in the image of a face that was detected. On average, Google maps the largest boxes around faces, however Animetrics has the largest box recognised in the set. On average, the smallest faces are recognised by Skybiometry.

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

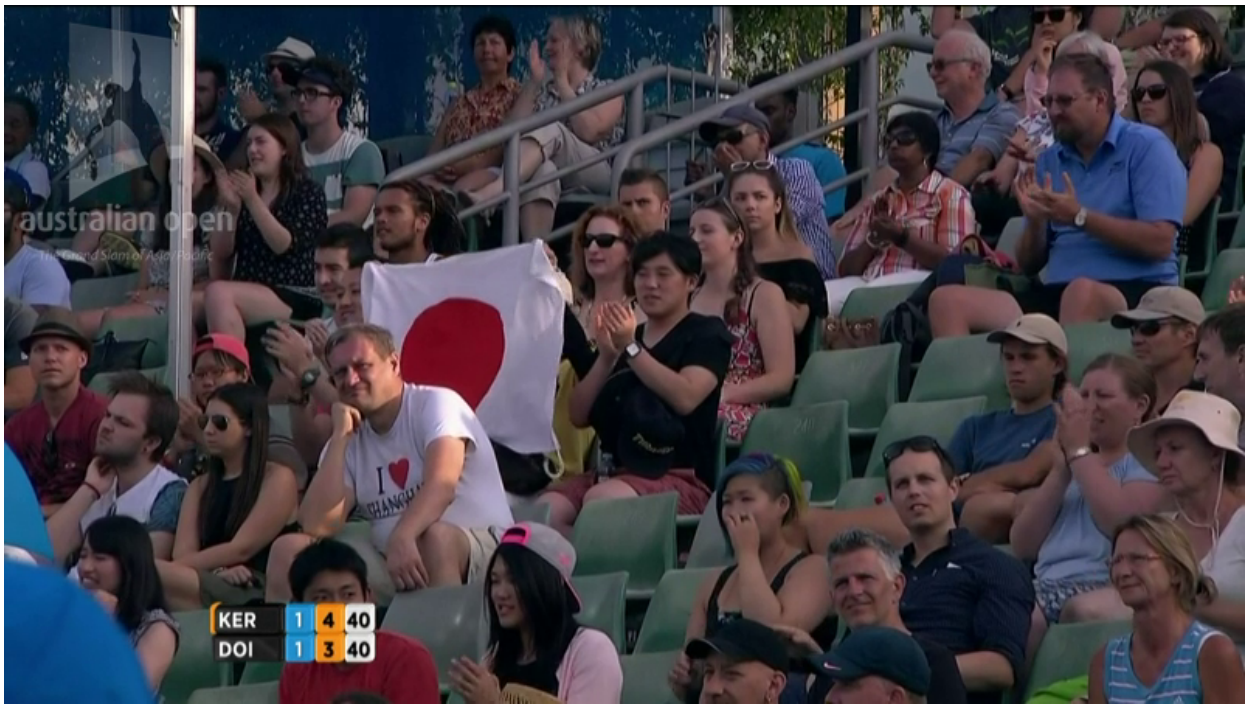


This image shows the smallest Face Bounding Box recognised. However visual inspection shows it only captures the player's eye and nose, not their whole face.

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

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

Image 5.2016_SC2_R01_AKerber_GER_vs_MDoi_JPN_WS1332_clip.0083.png



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

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

	Animetrics	Google	Microsoft	Skybiometry
TRUE	370	967	563	514

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 recognised by the four softwares.

To consider how many Type I errors occured, 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 softwares. This is especially high for Google with 289 faces identified.

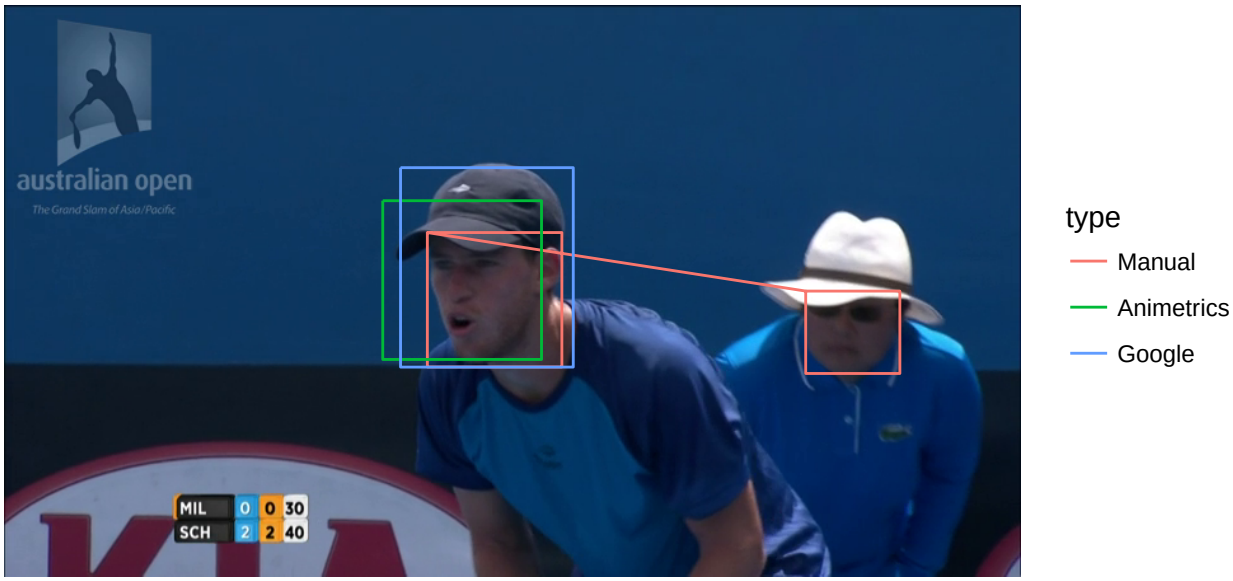
All the Face Bounding Boxes that Google found which do not match manually annotated faces were correctly identifying faces. This exhibits the occurences of errors.

We then considered the characteristics of the images that the softwares found Potential Face Bounding Boxes[11] in.

situation	bg	shotangle	detect	count
Court player close-up	Logo wall	Player Shoulder Height	Player	485
Court in play	Logo wall	Player Shoulder Height	Player	226
Crowd	Crowd	Upward Angle	Fan	138
Court player close-up	Court	Birds Eye	Player	105
Court in play	Logo wall	Player Shoulder Height	Other staff on court	81
Off court close up of player	Logo wall	Player Shoulder Height	Player	80
Court player close-up	Logo wall	Player Shoulder Height	Other staff on court	77
Off court close up of player	Crowd	Player Shoulder Height	Player	55
Crowd	Crowd	Player Shoulder Height	Fan	53
Court in play	Logo wall	Upward Angle	Player	43

Figure 4 displays the feature combinations that produced the most potential face Bounding Boxes recognitions by all four softwares. The most common shot is a crowd shot. The second row in the table with 830 faces recognised 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:



person	situation	bg	shotangle	count
NA	NA	NA	NA	NA

[11] These boxes represent an area of pixels that are a potentially recognised 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

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 recongition 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 recognitions of undesirable faces.

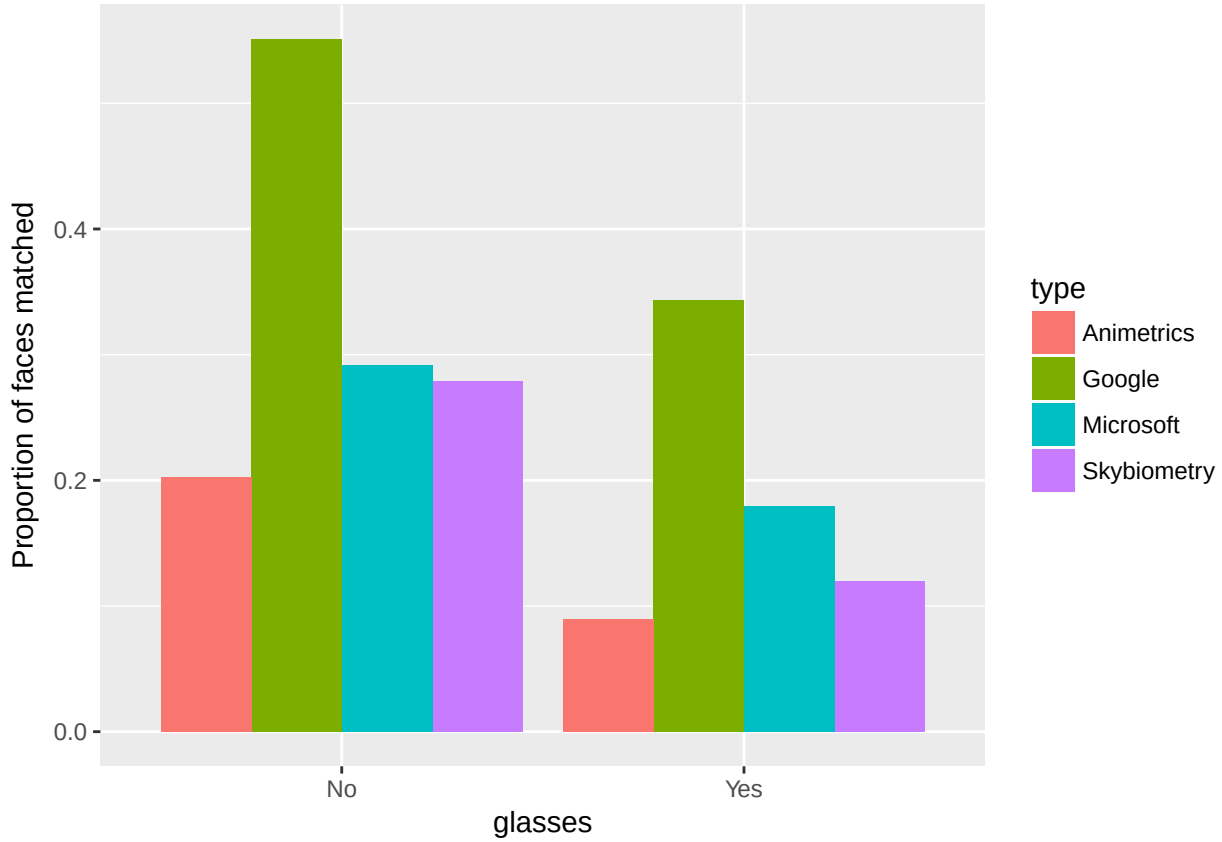
visorhat	glasses	headangle	lighting	obscured	detect	count
No	No	Other	Partially shaded	No	Player	124
Yes	No	Other	Partially shaded	No	Player	79
No	No	Other	Partially shaded	Yes	Player	63
No	No	Profile	Partially shaded	No	Player	61
Yes	No	Other	Partially shaded	Yes	Player	45
Yes	No	Profile	Partially shaded	No	Player	30
No	No	Profile	Partially shaded	Yes	Player	27
Yes	No	Front on	Partially shaded	No	Player	27
No	No	Profile	Shaded	No	Player	25
Yes	No	Other	Shaded	No	Player	24

table 7 utilises 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'[12] for nine of the top ten facial characteristic combinations.

No	Yes
659	23

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

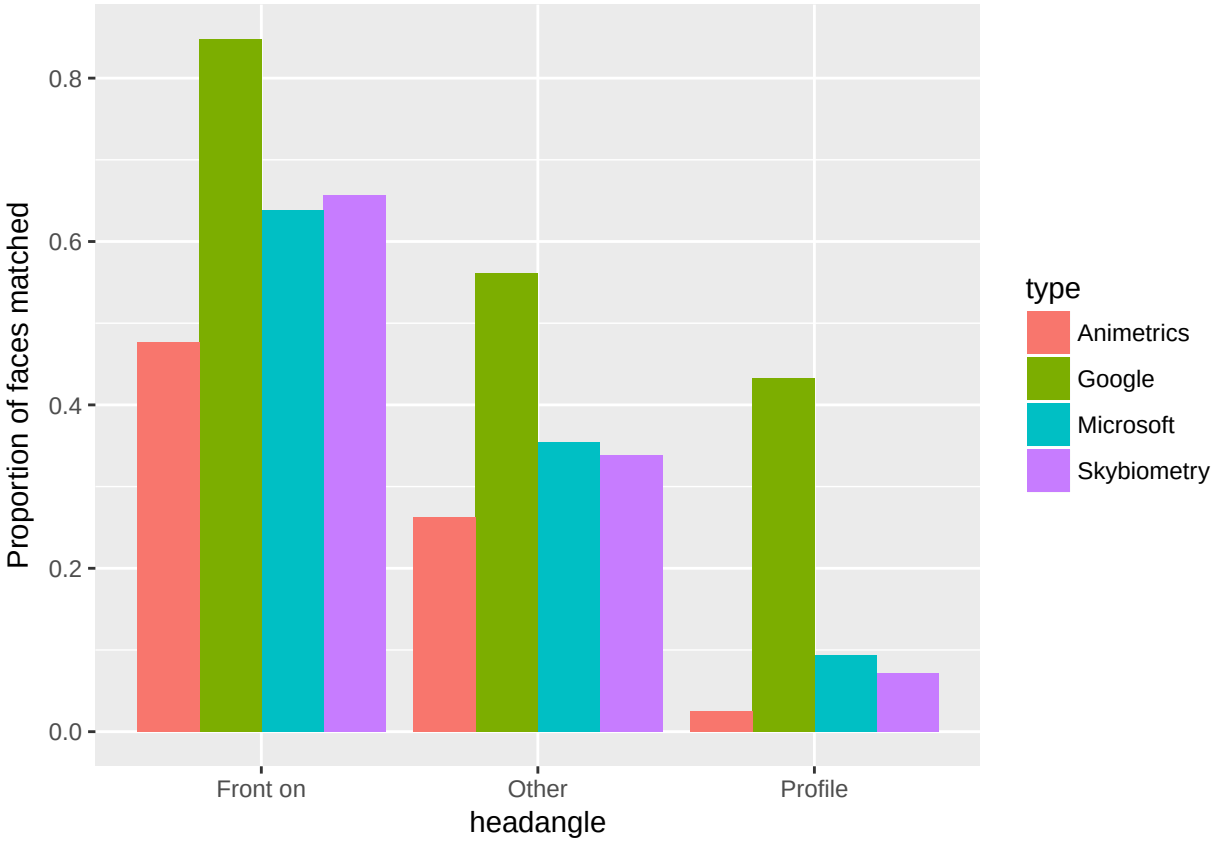
[12]Definition for headangle factor level 'Other' found in..



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

Front on	Other	Profile
89	400	193

The amount of faces found by Google with the headangle “Other”, is much larger than the amount of faces with the headangle “Profile” or “Front On”. Therefore this should also be considered propotionately.



Graph 3 diplays that Google performs much better in comparison to the other softwares when the headangle is Profile. This is outperforming unusually well, however this could also be due to the poor performance of the softwares 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	1820	370	967	563	514

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 recognised twice within the same image. This was a very unusual result and is notable as a point of interest but given it only occured 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	357
Person	Court player close-up	Logo wall	Upward Angle	Player	21

person	situation	bg	shotangle	detect	count
Person	Court player close-up	Logo wall	Birds Eye	Player	15

table 8 shows how many images with potential player's faces Google recognised. This diplays 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 softwares. 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 recognised, this is both helpful and unhelpful as it shows a strong ability of Google's algorithm to recognise 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 recognised and allow both the identity and emotion of the players to be recognised.

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 memeber behind him on the court.

While Google's recognitions 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 occured 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 unexplainable face Bounding Boxes as some of the other softwares 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[13] 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 recognitions for faces not Manually annotated thesse 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

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

accessories may contribute to the performance of a recongition 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 recognised, by both the Google recnogitions 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 prescence of Glasses on faces annotated manually did affect recognition by Google's alrgorithm, while it outperformed the other softwares 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 boxplot 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 recognise in these shots than in shots where there is only a player, and therefore many more faces recognised. 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 datacleaning and help group characterstics

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

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 softwares focussed 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 recognise 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 Facical 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 recognitions 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

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.

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.