Springboard – Data Science Camp
Capstone Project 2

Fútbol Match Highlights

By Tom Widdows
April, 2020

Table of Contents

Introduction

Problem

College coaches require soccer players to submit highlight video and game film to be considered for an athletic scholarship. The issues facing student-athletes to provide highlight videos and game films include recording games is time-consuming, athletes must rely on someone else to take the video, most people would prefer to watch the game and not be a videographer, processing the game file is time consuming and the required hardware and software are expensive.

Stakeholders

Stakeholders include high school athletes, their parents and college sports recruiters.

Figure 1 – Athletes and Parents

Results

The goal is to identify a target player in a photo (ultimately a video) with a high degree of accuracy. By managing the quality throughout the process, the resulting accuracy of locating (and ultimately tracking) the target player should be very high.

Post project

Two videos will be stitched together, face images will be extracted from the videos (which are images), the face recognition utilized in this project will be applied to the video image and the target player identified allowing the software to effectively track the player.

Implementation Details

Additional implementation details and all code related to the project can be found on my GitHub repository:

https://github.com/8-Waste/Springboard/tree/master/Capstone%202%20-%20F%C3%BAtbol

Approach

Data Acquisition and Wrangling

Facial recognition requires a sequence of related steps including ensuring the source image is sharp, locating faces in the image, provide quality checks to ensure quality faces, centering and measuring unique facial features, comparing the target face to known faces and making a prediction.

The data source for this project is images of fútbol players taken by Tom Widdows. Although the images are currently from still photographs, the application can be easily modified to work with frames of a video. The images are primarily in a proprietary Canon format. After consolidating the photos, they are

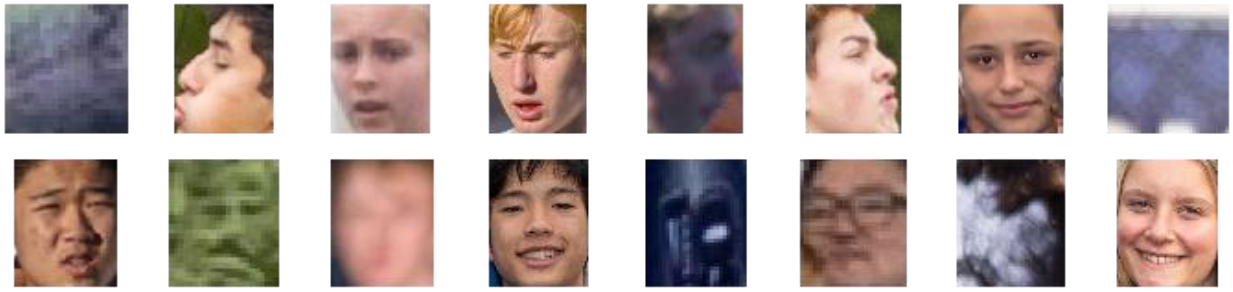
converted, if necessary, from the proprietary Canon format (.CR2) into JPG files so we can easily access them in python.

To evaluate image sharpness, A support vector machine (SVM) that uses labeled photos (sharp and blurred) with 12 features consisting of 3 measurements (mean, variance and maximum) each of 4 edge detection algorithms (laplace, sobel, roberts and canny) is fit and saved. The source images, with the same 12 calculated features, are processed through the trained model and identified as a sharp or blurry image.

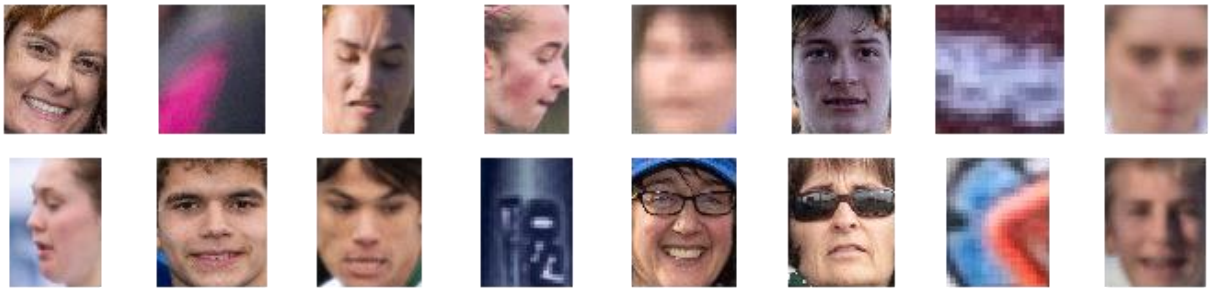
Storytelling and Inferential Statistics

The sharp images are processed through two separate face detection algorithms, a Harr-cascade classifier (HARR) included with OpenCV and a Multi-task Cascaded Convolutional Network (MTCNN).

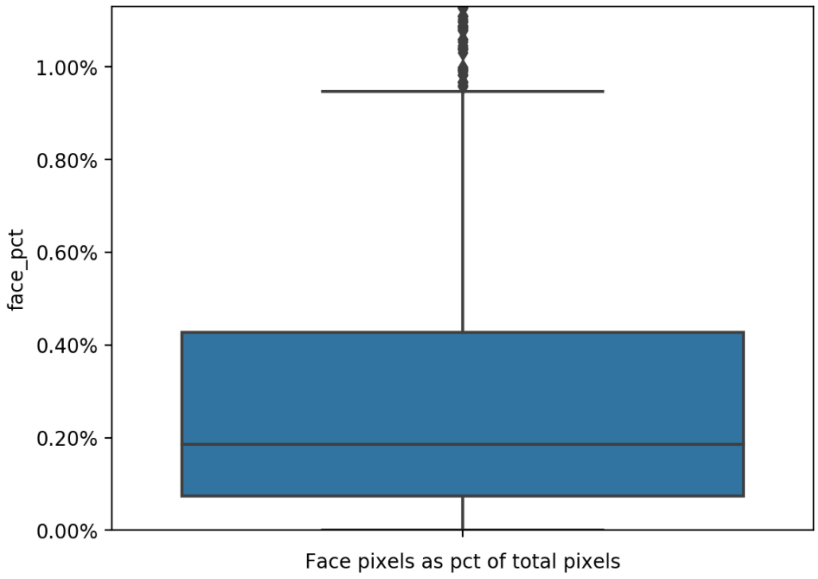
The images below were created using the HARR classifier for face detection. The sample images below certainly show the HARR classifier identified many non-faces as faces (false positives).



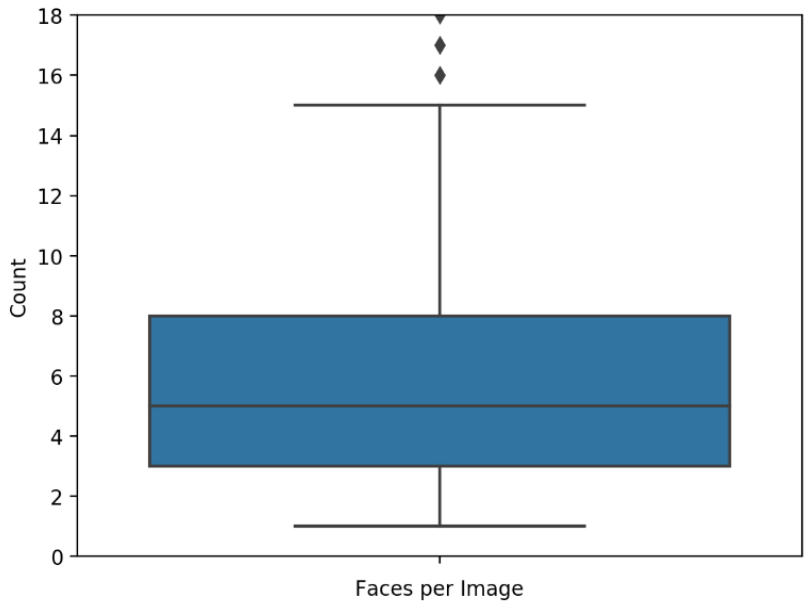
The images below were created using a Multi-task Cascaded Convolutional Network (MTCNN) for face detection. The MTCNN identified many non-faces as faces (false positives) but at first look, appears to have performed better than the HARR classifier.



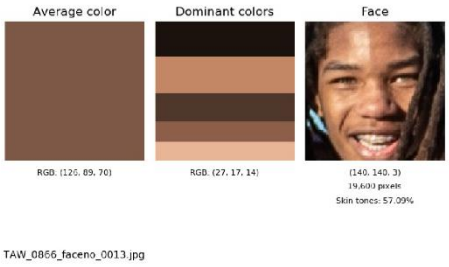
A face in an image is rarely more than 1% of the entire image (calculated using square pixels) and the median percentage is 0.20%.

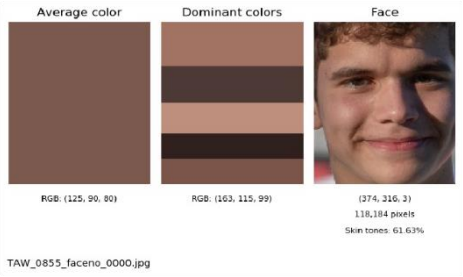


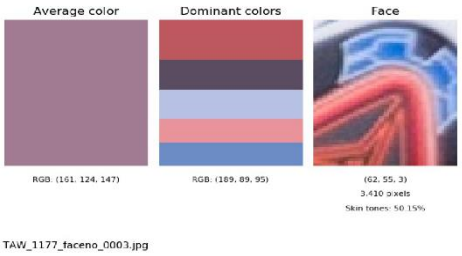
An image rarely contains more than 15 faces and the median number of faces in an image is five.



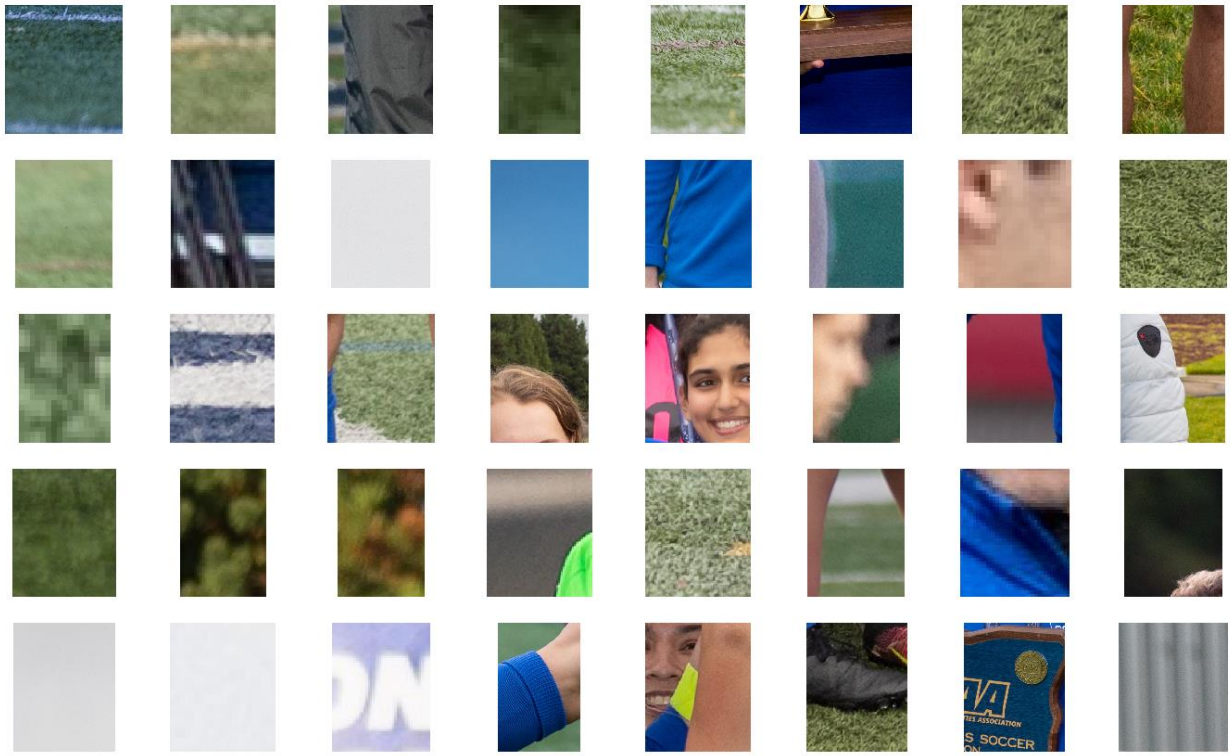
Now let's examine some of the color properties of an identified face. We will calculate the average color of the face, the top 5 dominate colors in the face and the percent of skin tones in the face. Below are 3 random faces with the calculated color values:



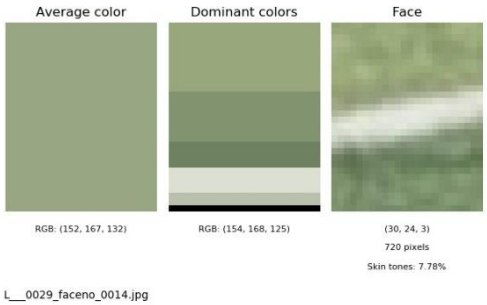


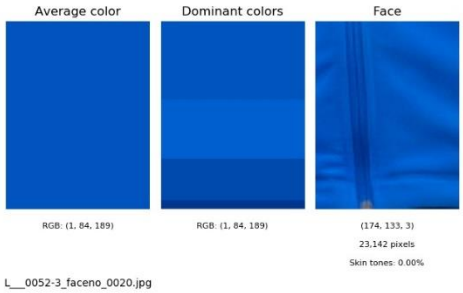


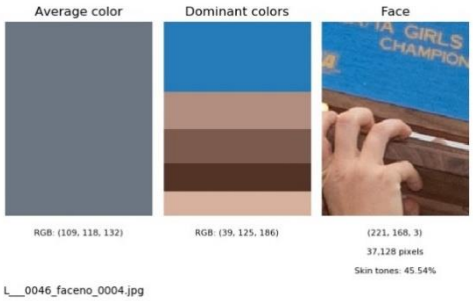
Let's calculate if the difference in skin tones between face and non-face images is statistically significant. In each image, we will programmatically count the number of faces and take an equal number of non-faces (or more accurately a random area the same size as the face). Since the median face covers only 0.20% of an image and rarely covers over 1% of an image, the result should include very few faces and it would be unlikely the face would be centered. Below is a sample of the calculated non-faces:



Now let's examine some of the color properties of a non-face. Again, we will calculate the average color of the non-face, the top 5 dominate colors in the non-face and the percent of skin tones in the non-face. Below are 3 random non-faces with the calculated color values:







Let's test if the means of the percent of skin tones re identical. The hypothesis test is to test if the difference in percent skin tones in a face image is statistically significant. The null and alternative hypothesis are as follows:

H_o: Percent Skin Tones has no effect on face identification, means are same

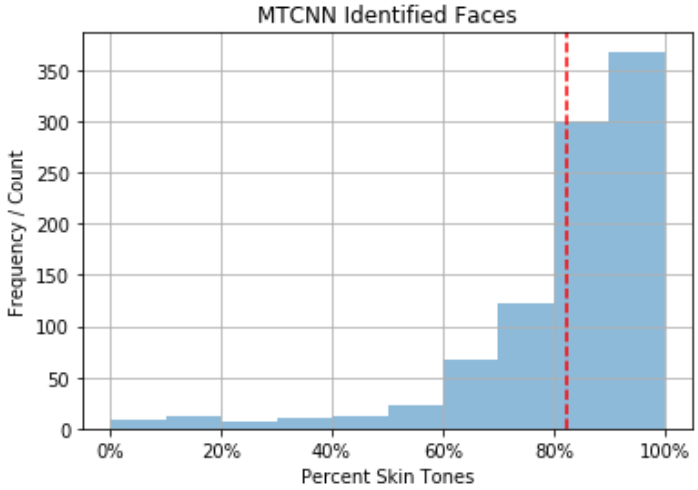
H_1: Percent Skin Tones has effect on face identification, means are different

Samples: **931**

Mean: **82.39% (dashed red line)**

Std Dev: **16.86%**

Median: **87.69%**

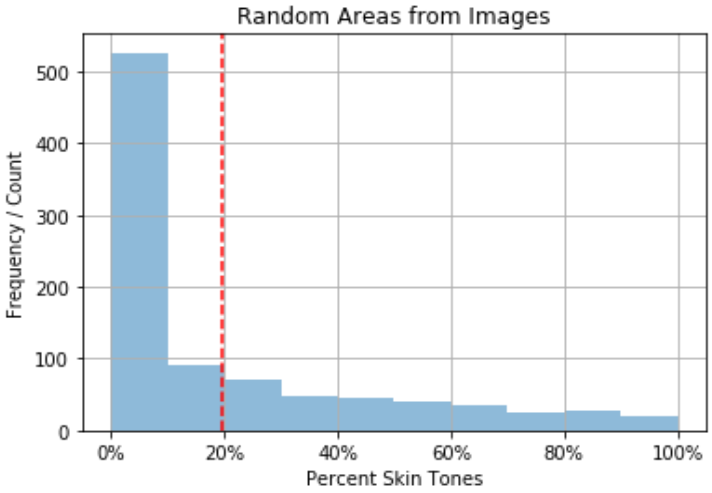


Samples: **931**

Mean: **19.72% (dashed red line)**

Std Dev: **26.71%**

Median: **5.98%**



I calculated a two-sided t-test for the null hypothesis that the samples have identical average (expected) values utilizing a significance level of .01. The p-value was 0.0000000 so **I rejected the null hypothesis in favor of the alternative hypothesis.**

H_1: Percent Skin Tones has effect on face identification, means are different

Baseline Modeling (Pending)

Extended Modeling (Pending)

Findings (Pending)

Conclusions and Future Work (Pending)

Consulted Resources (Pending)