Patrick O.
Regan M.

Module 1 Project Report:

Summary:

In this project, we will be creating a program that detects whether a soccer ball has crossed over a goal line. This goal line technology is used to determine points in a soccer game, and can decide the victor from the loser, potentially determining the lives of the players and managers, and the funding that a club can receive. To introduce more fairness and objectiveness, goal line technology is implemented to make sure goals are properly identified. Our program became as we added more images of different environments, and we ended with a solid 80% accuracy.

Introduction:

The FeaRDeClass that we have chosen to analyze lies in the field of soccer. It is used to determine if a goal has been scored in a soccer game, and helps when the ref is unable to indicate a clear goal. This technology can decide the course of a game, and can also decide the fate of the players and managers involved, as funding can be directly tied to winning or losing games.

The algorithm itself will involve machine learning, and training our program through the use of projection. After taking several pictures of a soccer ball in conjunction to its positioning from the goal line, we will vectorize the images and use vector projection to predict if another set of images are goals or not. After that, the program will compare the prediction with the reality, and provide the percentage of the correct guesses. With enough test images, we should be able to correctly identify if a goal is scored.

This technology can help make soccer games more fair, and allow for an objective look at if a goal has been scored. This can alleviate the anger and hatred directed towards refs who may make the wrong calls from time to time, as it provides a way for a goal to be very clearly defined. At the same time, this technology also provides an absolute, and that means that teams can lose out on money or promotions, where before they may have had a chance due to incorrect calls.

Our main question is can we create an algorithm that predicts goals successfully and can it be good enough for use on the soccer field?

Context

For context,we should define what a goal is and what isn’t a goal, according to the FIFA organization.

According to FIFA.com, “A goal is scored when the whole of the ball passes over the goal line, between the goalposts and under the crossbar, provided that no offence has been committed by the team scoring the goal.”

Therefore for a game to be registered, the whole ball has to pass over the goal line, for it to be registered as an official goal, and if the whole ball doesn’t cross the line, it is not registered.

As a result, there are moments in official games, where the ball is heading towards the goal at high speed, and the opposition attempts to scramble the ball away on the goal line, and it’s times like these that we don’t know whether the ball has crossed the line before the clearance has occurred, or hasn’t. The human eye is too slow to catch these types of situations, and it is up to technology to determine whether the ball has crossed the line or not, which is where goal line technology comes in.

Methods:

Data collection was done out on the Olin soccer field. We used the same soccer ball for all pictures, and our methodology looked only at if the ball was fully in the frame of the picture or not. If the ball wasn’t fully in the picture, that counted as a no-goal (as in soccer, the ball has to fully cross over the line).

After taking the pictures, we then started our MatLab code.

Here are sample pictures used in our code

Goal: No Goal:

We are using the process of vector projection--A^TAx = A^Tb. Our first step is to upload the images into MatLab. We used a for loop that added each image and resized them each to an 84 by 112 pixel image. Next, we changed the images into vectors, converting the images into single vectors with the reshape function in MatLab. From there, we set up the equation A^TAx = A^Tb with our training data set. We solved for x, and projected this new weighted value onto the test data set, creating a prediction of b, with a column vector of 60 rows, representing our 60 test images. We then compared the prediction of b to the actual b value, and found the percentage that the program got right. This process was repeated several times with added data in order to train the program. *See appendix for the full code

Detailed Findings:

The first time the program ran, we used 60 pictures for our training set, and kept a constant 60 test images, and it was only 40% successful at predicting if there was a goal scored. The data set that we used for that test run was taken around 4pm, and the sun had created very long shadows. To see if the problem was the shadows, we decided to collect more data in order to train our program more. The second data set, which brought our total images to 200, was taken around 11am, and after these images were added, our program's accuracy jumped to 51%. After adding more images, which brought the total to 471, also taken around the time around 11am, it jumped up to 68%. We finally ended up with 884 images for our training set, which brought our accuracy to 80%. With more pictures taken at different times within different environments, we can weed out the possibility of our program identifying an image wrong due to shadows or lighting, and hopefully increase the accuracy of predicting whether a goal has been scored or not.

Recommendations:

Goal line technology can be useful, and should always be improved upon. Eliminating human error within soccer games can help make them fairer, and have results that accurately reflect the skill level of the players. This program should be improved upon before being used in actual soccer games, as it's definitely still prone to error, just like humans.

References:

QEA Book (Vector projection)

FIFA Official Rulebook (What counts as a goal): https://www.fifa.com/mm/Document/FootballDevelopment/Refereeing/02/36/01/11/LawsofthegamewebEN_Neutral.pdf

Appendix:

```
% Read 1.jpg through m.jpg.
% Files are in the "yourFolder" directory.

for k = 1:884
  tiffFilename = sprintf('%d.tiff', k);
  fullFileName = fullfile("C:\Users\pogunbufunmi\Documents\GLT_PHOTOS", tiffFilename );
  if exist(fullFileName,'file')
    imageData = imread(fullFileName);
    imageData = rgb2gray(imageData);
    imageData = imresize(imageData,[3024/36 4032/36]);
    imageData = reshape(imageData,[],1);
  end
  glttraindata(:,:,k)=imageData;
end
```

```
% Read 1.jpg through m.jpg.
% Files are in the "yourFolder" directory.

for k = 1:60
  tiffFilename = sprintf('%d.tiff', k);
  fullFileName = fullfile("C:\Users\pogunbufunmi\Documents\GLT_PHOTOS_TEST", tiffFilename );
  if exist(fullFileName,'file')
    imageData = imread(fullFileName);
    imageData = rgb2gray(imageData);
    imageData = imresize(imageData,[3024/36 4032/36]);
    imageData = reshape(imageData,[],1);
  end
  glttestdata(:,:,k)=imageData;
end
```

ML_Face_Test.mlx | dsknasA.m | testtttttttttttttttttttttt.m | workspace_glt.m | livescriptglt.mlx

```
B=[1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
Ball_Check_test=[1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1
%Ax=B
Ball_train_reshape=reshape(glttraindata,[(3024/36)*(4032/36) siz
Ball_test_reshape=reshape(glttestdata,[(3024/36)*(4032/36) size(
BTrain=double(Ball_train_reshape);
BTest=double(Ball_test_reshape);
%x=AtA*BAt
x=(BTrain'*BTrain)\(BTrain')*B;
BallPred=BTest*x;
BGoal=BallPred<150;
GoalComparator=BGoal==Ball_Check_test;
answer=mean(GoalComparator);
newOrth=reshape(x,84,112);
see=imagesc(newOrth);colormap('gray');
```