Basketball Shot Detection with Deep Learning

Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Emmanuel Sedicol, BSc(Hons) in Applied Computing
Department of Computing and Mathematics, School of Science and Computing, Waterford Institute of Technology
Student number: 20072377

Introduction

With the rapid growth of modern technology such as Artificial Intelligence (AI), automated player tracking and ball detection in sports is growing high in demand. As more advance applications emerges not only league owners or coaches are slowly becoming keener into investing on the idea of data driven decision in sports but also the athletes themselves. A 2019 study showed that the global sports analytics market is expected to grow at a compound annual rate (CAGR) of 22% as more sports analysts are able to take data and create insightful yet simple visualizations to communicate to other key decision makers of a team.

Inspiration and Build Overview

Data recorded from every game gives us, the audience a glimpse of how good a player is on the court. This opened up the idea of building an artificial intelligent application using neural networks to automatically parse an input video to detect every shot attempt of an athlete and extract location of attempt on the court. The build overview shown below is the exact step process of how I plan to tackle the build.

Build Overview:

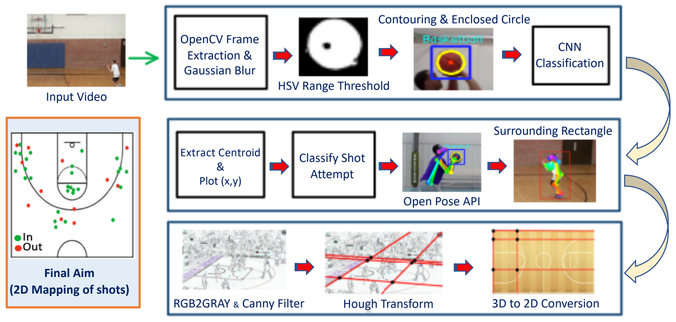


Fig.1 Build Process Blueprint

Technologies

OPENCV
- ✓ open source programing library that offers multiple functions aimed at computer vision, image processing and machine learning.
- ✓ used in this project to extract video frames, perform image processing and display detected contours.

TensorFlow
- ✓ open source deep learning library developed by Google's Brain Team which is aimed at handling numerical computation using data flow graphs

Keras
- ✓ high level neural python library used for building prediction models and neural networks

Aims and Questions

Aims
- ✓ build a prediction model using a two-dimensional Convolutions to classify if the detected contour is a basketball.
- ✓ map player location into a two-dimensional court image.
- ✓ visually and statically represent accuracy of shots

Questions
- ✓ how big is the margin of error when extracting player position
- ✓ how to accurate classify between a make and missed shot

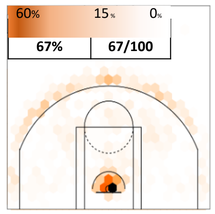


Fig.2 Aimed visual and statistical representation of shot attempts

Methodology

Contours and Centroids:

- for η distict points in countounrs $(x_1 \ldots x_n)$, centroid is given by: $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$
- enclose contour with a min-fit circle from the centroid
  ((x, y), radius) = cv2.minEnclosingCircle(c)

Fig.3 Plotting the centroid

Convolutional Neural Network Model:

- two dataset (basketball and basketball hoops)
- transform to grayscale for a 2D CNN
- uses of Sequential() model for a linearly stack of layers
- trained the model using a sigmoid (0, 1) activation
- loss function equals to binary cross entropy
- Model Trained and Validated for 8 epochs

Fig.4 Dataset from CNN

given $H(\rho, q) = \sum_x p(x) \log q(x)$ is minimized through the standard ADAM optimization

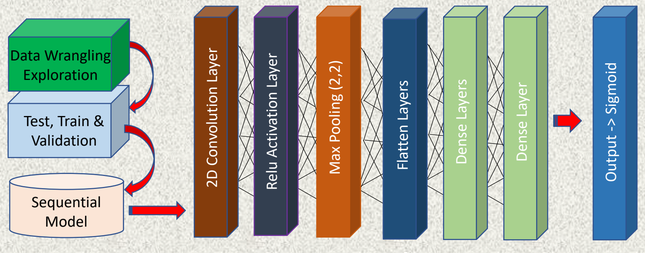
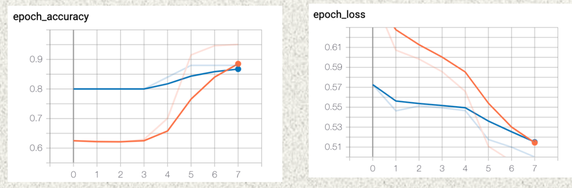


Fig.5 Structure of my CNN



- the epoch graph show that most suitable epoch value is 8 as it is not over fitting or under fitting the mode

Using the Model:

- apply to CNN model to frame to classify if contour retrieve is a ball
- if *True* store centroid in array, plot shot trajectory and label basketball using an enclosed box/circle

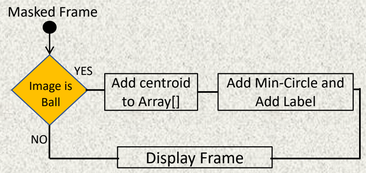


Fig.6 Decision flow for model

Pose Estimation:

- imported TfPoseEstimator API transform
- create function to feed image or frame into the estimator module return an image
- use new function in main file, feed frame and enclosed with a bounding rectangle

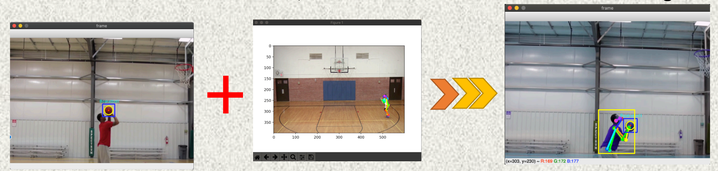


Fig.7 Adding two feature into one

3D-2D Mapping

Image Processing:

- applying multiple filter on the image increase the accuracy of detecting court sidelines
- apply Hough algorithm for edge detection and filter out unwanted line
- grab coordinate of the corners

Player Position :

- using the pose-estimate module draw surrounding rectangle on player
- calculate moments and retrieve centroid of rectangle
- apply cv2 circle to in place temporary marker on image

Image Transformation:

- measure coordinates of 2D image
- implement homograph image transformation to overlay the 3D on the 2D
- apply mask using the colour used on markers
- detect player position and insert in player position array

Convolutional Model Results

- perform extra test to ensure model efficiency
- over fitting on a model can be see in figure 7, where accuracy epoch stays constant while loss epoch is decreasing linearly.

Fig.8 Over Fitting the model

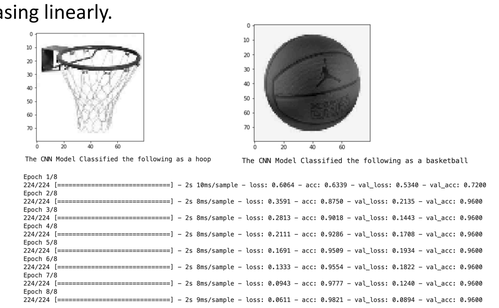


Fig.9 Visual Test on Model

Conclusion and Discussion

- the proposed idea of automated player and ball detection can now be done a lot faster using different APIs in the web, but it can be concluded that its possible to perform such idea with OpenCV and colour segmentation.
- Future work for this project would involve multidimensional CNN for miss and make shot classification.
- also future work includes better colour segmentation thresholds to enhance the accuracy of our tracking system.

Technologies

TensorFlow™ · Neural Networks · python™ · Keras · OpenCV