Project Name: End-To-End ML with Deployment AIBasketball Prediction – Faster RCNN – Classification

Table of Contents

Demo

```python
app = Flask(__name__)
UPLOAD_FOLDER = './static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
#useless key, in order to use session
app.secret_key = "super secret key"

@app.route("/")
def index():
    return render_template("index.html")

@app.route('/detection_json', methods=['GET', 'POST'])
def detection_json():
    if request.method == 'POST':
        response = []
        f = request.files['image']
        # create a secure filename
        filename = secure_filename(f.filename)
        print("filename", filename)
        # save file to /static/uploads
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        print("filepath", filepath)
        f.save(filepath)
        detectionAPI(response, filepath)
        print(response)
        try:
            return jsonify(response), 200
        except FileNotFoundError:
            abort(404)
```

```python
@app.route('/sample_detection', methods=['GET', 'POST'])
def upload_sample_image():
    if request.method == 'POST':
        response = []
        filename = "sample_image.jpg"
        print("filename", filename)
        filepath = "./static/uploads/sample_image.jpg"
        print("filepath", filepath)
        get_image(filepath, filename, response)
        return render_template("shot_detection.html", display_detection=filename, fname=filename, response=response)

@app.route('/basketball_detection', methods=['GET', 'POST'])
def upload_image():
    if request.method == 'POST':
        response = []
        f = request.files['image']
        # create a secure filename
        filename = secure_filename(f.filename)
        print("filename", filename)
        # save file to /static/uploads
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        print("filepath", filepath)
        f.save(filepath)
        get_image(filepath, filename, response)
        return render_template("shot_detection.html", display_detection=filename, fname=filename, response=response)

@app.route('/sample_analysis', methods=['GET', 'POST'])
def upload_video():
    global shooting_result
    shooting_result['attempts'] = 0
    shooting_result['made'] = 0
    shooting_result['miss'] = 0
    if (os.path.exists("./static/detections/trajectory_fitting.jpg")):
        os.remove("./static/detections/trajectory_fitting.jpg")
    if request.method == 'POST':
        filename = "sample_video.mp4"
        print("filename", filename)
        filepath = "./static/uploads/sample_video.mp4"
        print("filepath", filepath)
        session['video_path'] = filepath
        return render_template("shooting_analysis.html")
```

```python
@app.route('/shooting_analysis', methods=['GET', 'POST'])
def upload_sample_video():
    global shooting_result
    shooting_result['attempts'] = 0
    shooting_result['made'] = 0
    shooting_result['miss'] = 0
    if (os.path.exists("./static/detections/trajectory_fitting.jpg")):
        os.remove("./static/detections/trajectory_fitting.jpg")
    if request.method == 'POST':
        f = request.files['video']
        # create a secure filename
        filename = secure_filename(f.filename)
        print("filename", filename)
        # save file to /static/uploads
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        print("filepath", filepath)
        f.save(filepath)
        session['video_path'] = filepath
        return render_template("shooting_analysis.html")

@app.route('/video_feed')
def video_feed():
    video_path = session.get('video_path', None)
    stream = getVideoStream(video_path)
    return Response(stream,
                    mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route("/result", methods=['GET', 'POST'])
def result():
    return render_template("result.html", shooting_result=shooting_result)

#disable caching
@app.after_request
def after_request(response):
    response.headers["Cache-Control"] = "no-cache, no-store, must-revalidate, public, max-age=0"
    response.headers["Expires"] = 0
    response.headers["Pragma"] = "no-cache"
    return response

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```

Live Web App Demo Link

Deployment on Heroku: https://dlaib.herokuapp.com/

Abstract

The purpose of this report will be to use the pre-trained models to analyse basketball shots and shooting pose depending on details provided by user. This can be used to gain insight into how and why shot is such at a given time. This can also be used as a model to gain a marketing advantage, by advertisement targeting those who have interest in sports or e-gaming industry. Shot Prediction is a classification problem, where using the various parameters or inputs from user model will supply result. This is an artificial intelligence application built on the concept of object detection. Analyze basketball shots by digging into the data collected from object detection.

This is diving into Analyses of Shot Detection through Machine Learning Concept.

End to End Project means that it is step by step process, starts with data collection, EDA, Data Preparation which includes cleaning and transforming then selecting, data resizing, data rescaling training and saving DL Models, Cross-validation and Hyper-Parameter Tuning and developing web service then Deployment for end users to use it anytime and anywhere.

This repository contains the code for Shot Detection using python's various libraries.

It used numpy, matplotlib, tensorflow, opencv, os, sys, time and flask libraries.

These libraries help to perform individually one particular functionality.

Numpy is used for working with arrays. It stands for Numerical Python.
Matplotlib is a plotting library.
TensorFlow is an open-source software library for high performance numerical computation.
Flask is classified as a microframework because it does not require particular tools or libraries.
This is an artificial intelligence application built on the concept of object detection. Analyze basketball shots by digging into the data collected from object detection. We can get the result by simply uploading files to the web App, or submitting a POST request to the API.
The purpose of creating this repository is to gain insights into Complete DL Project.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my DL repository.
These above screenshots and video in Video_File Folder will help you to understand flow of output.

Motivation

The reason behind building this project is, because I like to play sports in my free time as we all know it has many benefits along with that it refreshes our mind. At the same time this deep learning concepts are more interesting to apply. So, I created AIBasketball Detection Project to gain insights from IT perspective to know working of the model. Hence, I continue to spread tech wings in IT Heaven.

The Data

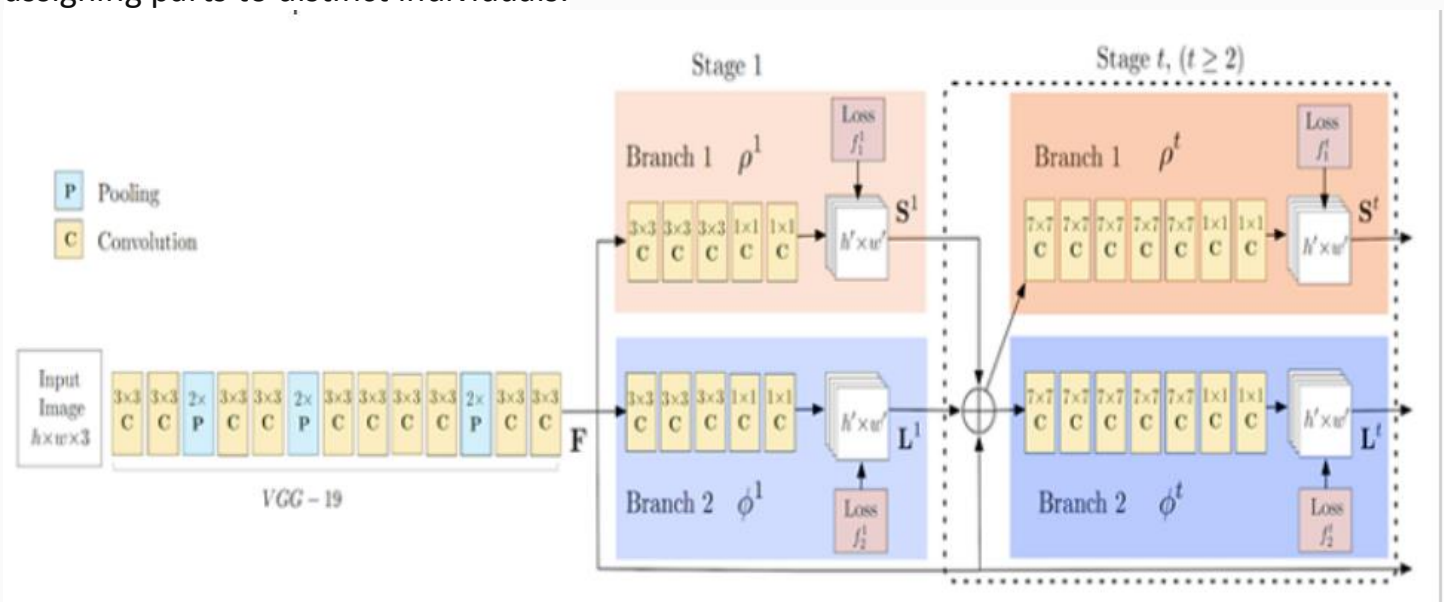
All the data for the shooting pose analysis is calculated by implementing OpenPose. Please note that this is an implementation only for non-commercial research use only.

Modelling

- Math behind the metrics

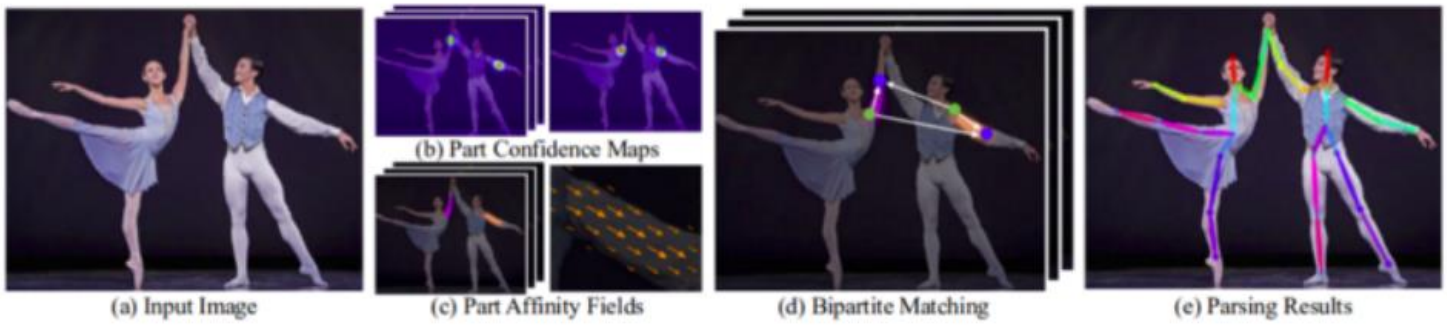
OpenPose:

A Human Pose Skeleton represents the orientation of a person in a graphical format. Essentially, it is a set of coordinates that can be connected to describe the pose of the person. Each coordinate in the skeleton is known as a part (or a joint, or a keypoint). A valid connection between two parts is known as a pair (or a limb). Note that, not all part combinations give rise to valid pairs.
OpenPose first detects parts (key points) belonging to every person in the image, followed by assigning parts to distinct individuals.



Flowchart of the OpenPose architecture. (Source)

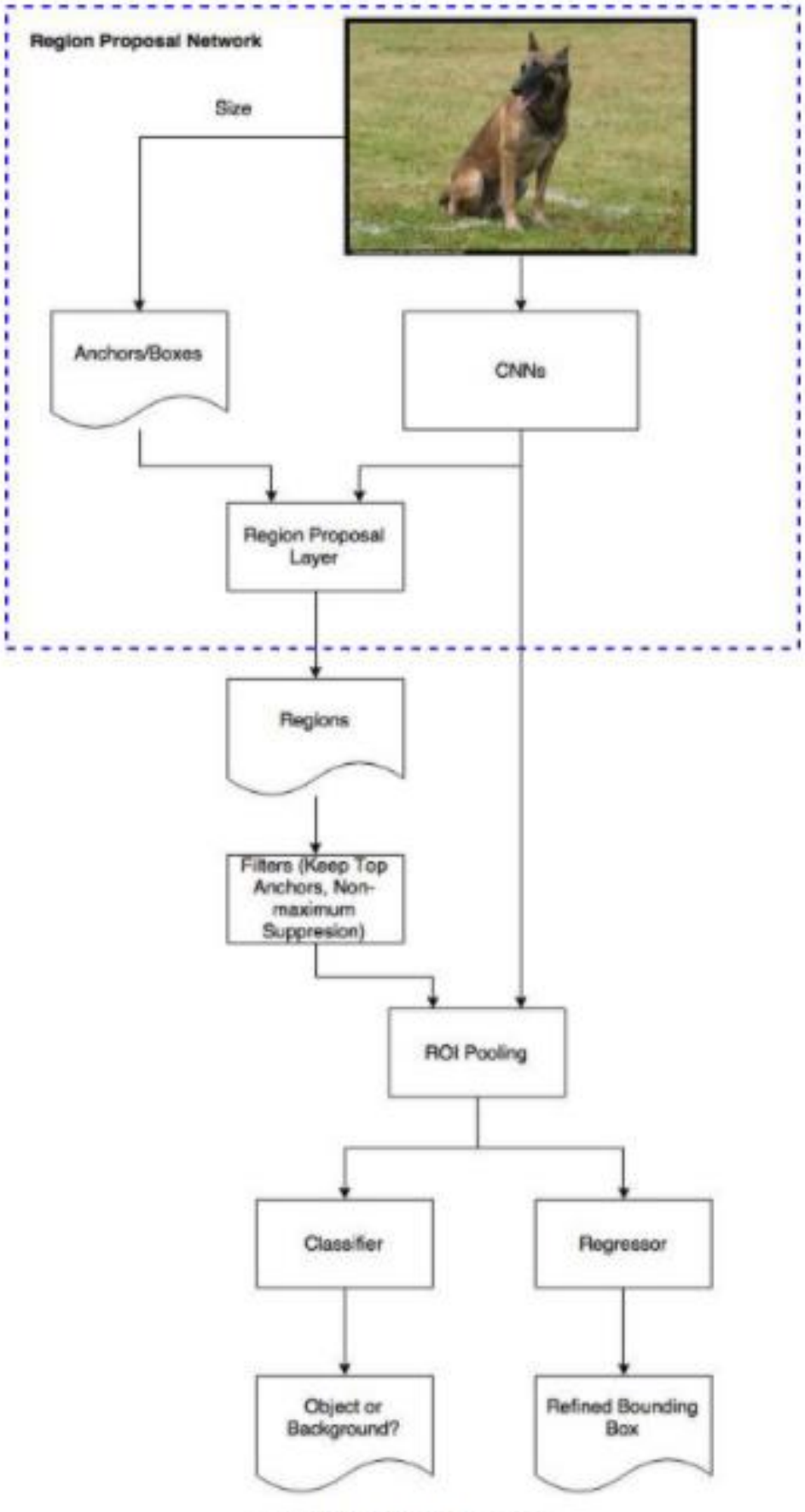
The OpenPose network first extracts features from an image using the first few layers (VGG-19 in the above flowchart). The features are then fed into two parallel branches of convolutional layers. The first branch predicts a set of 18 confidence maps, with each map representing a particular part of the human pose skeleton. The second branch predicts a set of 38 Part Affinity Fields (PAFs) which represents the degree of association between parts.



Steps involved in human pose estimation using OpenPose. (Source)

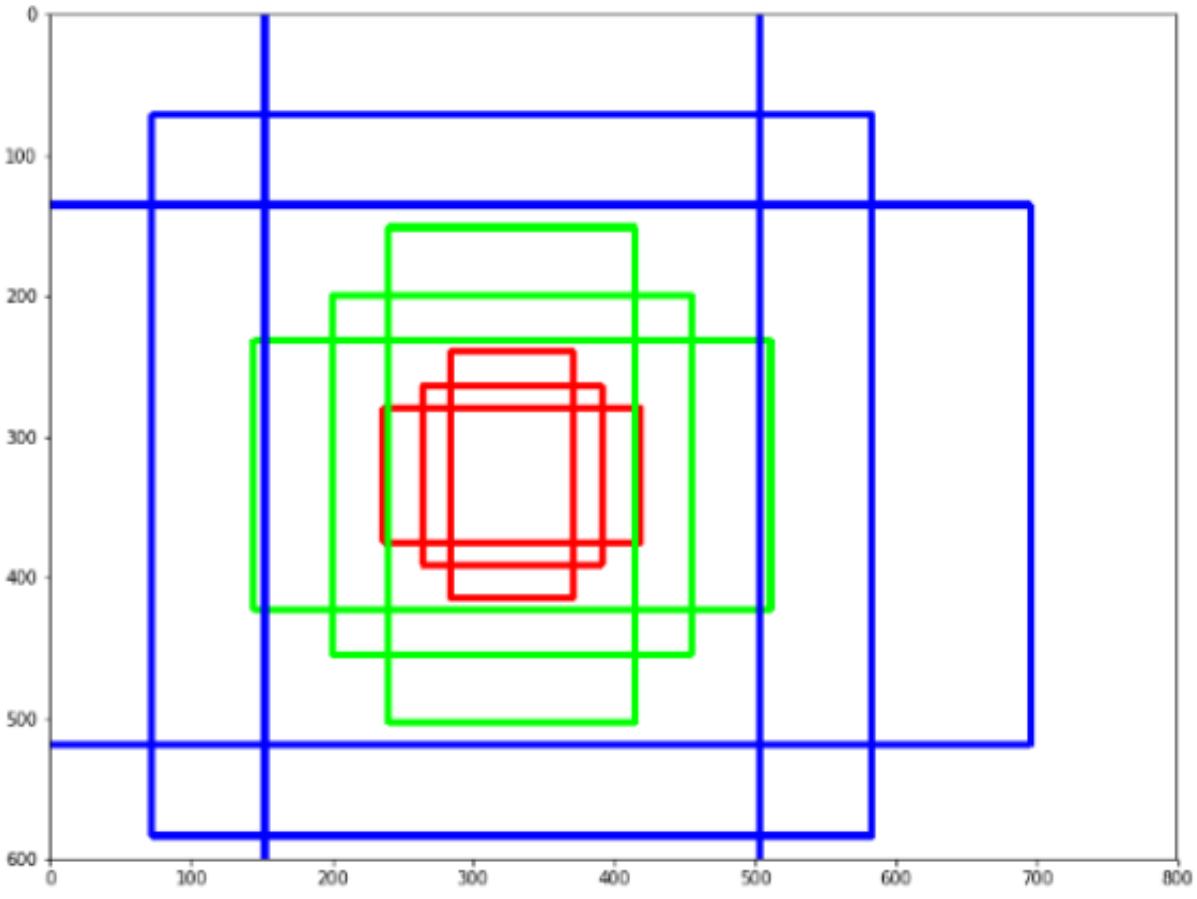
Successive stages are used to refine the predictions made by each branch. Using the part confidence maps, bipartite graphs are formed between pairs of parts (as shown in the above image). Using the PAF values, weaker links in the bipartite graphs are pruned. Through the above steps, human pose skeletons can be estimated and assigned to every person in the image.

Faster RCNN:
Faster R-CNN has two networks: region proposal network (RPN) for generating region proposals and a network using these proposals to detect objects. The main different here with Fast R-CNN is that the later uses selective search to generate region proposals. The time cost of generating region proposals is much smaller in RPN than selective search, when RPN shares the most computation with the object detection network. Briefly, RPN ranks region boxes (called anchors) and proposes the ones most likely containing objects. The architecture is as follows.



The Architecture of Faster R-CNN

Anchors play an important role in Faster R-CNN. An anchor is a box. In the default configuration of Faster R-CNN, there are 9 anchors at a position of an image. The following graph shows 9 anchors at the position (320, 320) of an image with size (600, 800).



Anchors at (320, 320)

Let's look closer:

1. Three colors represent three scales or sizes: 128x128, 256x256, 512x512.

2. Let's single out the red boxes/anchors. The three boxes have height width ratios 1:1, 1:2 and 2:1 respectively.

If we choose one position at every stride of 16, there will be 1989 (39x51) positions. This leads to 17901 (1989 x 9) boxes to consider. The sheer size is hardly smaller than the combination of sliding window and pyramid. Or you can reason this is why it has a coverage as good as other state of the art methods. The bright side here is that we can use region proposal network, the method in Fast RCNN, to significantly reduce number.

If you follow the process of labelling anchors, you can also pick out the anchors based on the similar criteria for the regressor to refine. One point here is that anchors labelled as background shouldn't include in the regression, as we don't have ground-truth boxes for them. The depth of feature map is 32 (9 anchors x 4 positions).

The paper uses smooth-L1 loss on the position (x ,y) of top-left the box, and the logarithm of the heights and widths, which is as the same as in Fast R-CNN.

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i), \qquad (2)$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \qquad (3)$$

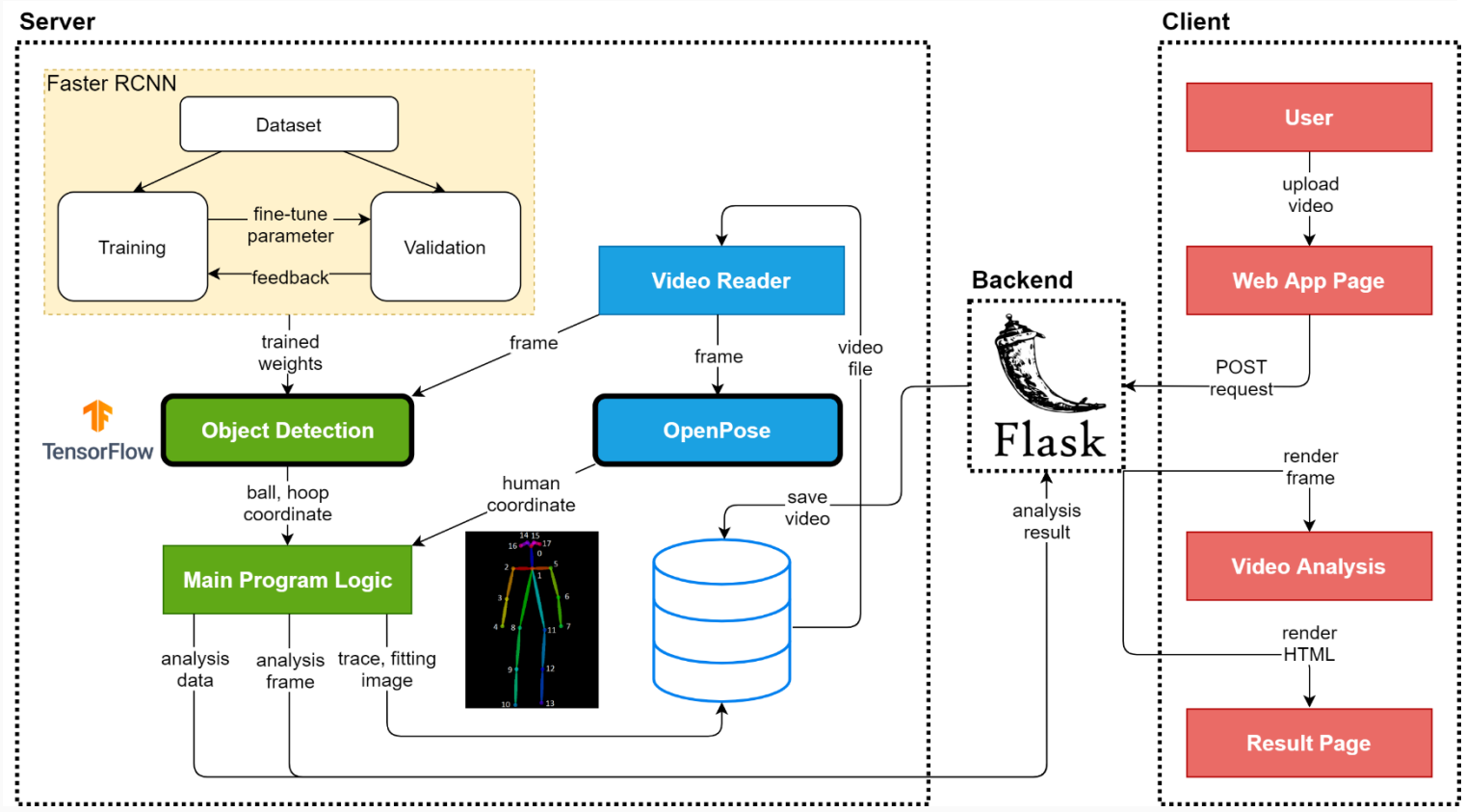
Loss Function of the Regressor

The overall loss of the RPN is a combination of the classification loss and the regression loss.
https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/

- The Model Analysis

The difference between object detection algorithms and classification algorithms is that in detection algorithms, we try to draw a bounding box around the object of interest to locate it within the image. Also, you might not necessarily draw just one bounding box in an object detection case, there could be many bounding boxes representing different objects of interest within the image and you would not know how many beforehand. The reason “Fast R-CNN” is faster than R-CNN is because you don't have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolution operation is done only once per image and a feature map is generated from it. Faster RCNN is an object detection architecture presented by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015, and is one of the famous object detection architectures that uses convolution neural networks. Faster R-CNN is a single-stage model that is trained end-to-end. It uses a novel region proposal network (RPN) for generating region proposals, which save time compared to traditional algorithms like Selective Search. It uses the ROI Pooling layer to extract a fixed-length feature vector from each region proposal. OpenPose is the first real-time multi-person system to jointly detect human body, hand, facial, and foot key-points (in total 135 key-points) on single images.

Project Architecture Process Through Visualization



Project Features

This project has three main features, shot analysis, shot detection, detection API.

Shot and Pose analysis:

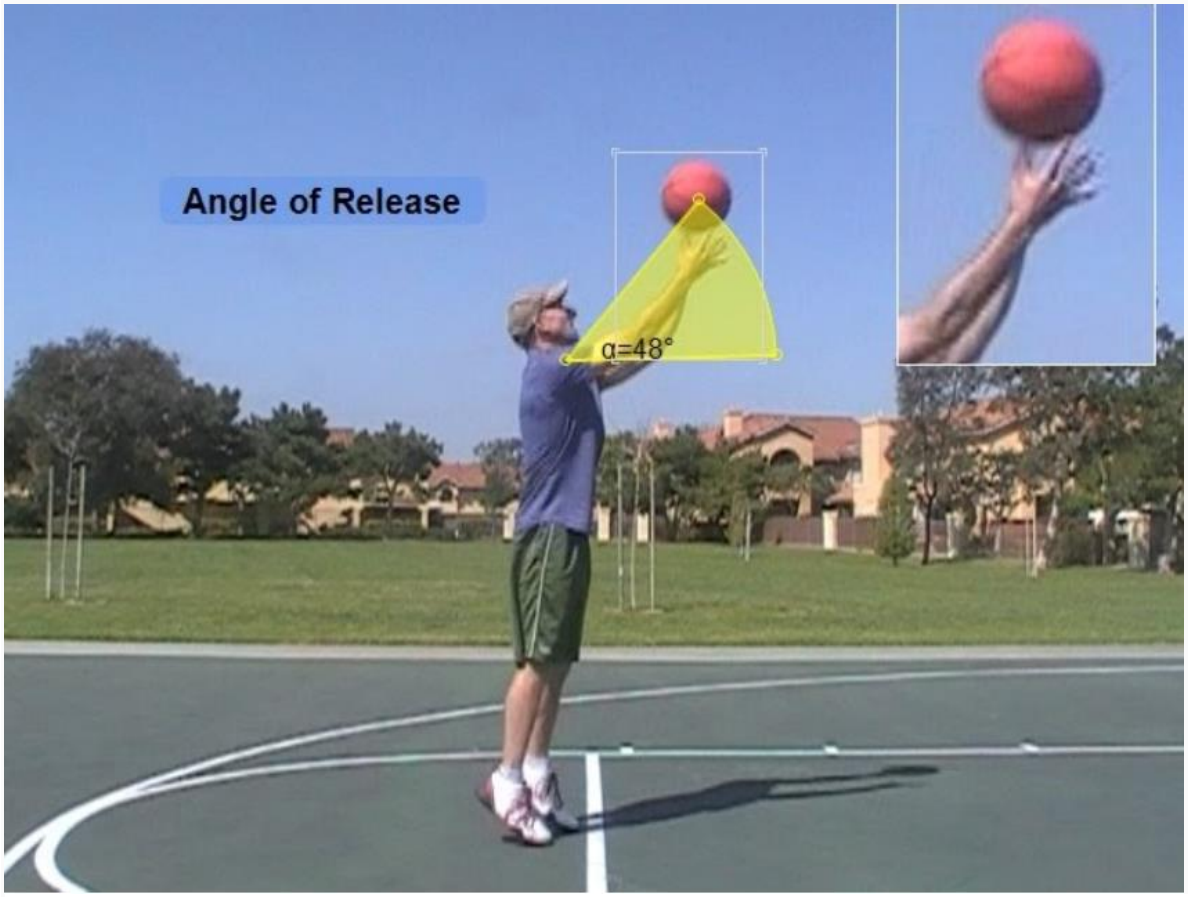
Shot counting

Counting shooting attempts and missing, scoring shots from the input video. Detection keypoints in different colors have different meanings listed below:

- **Blue:** Detected basketball in normal status
- **Purple**: Undetermined shot
- **Green:** Shot went in
- **Red:** Miss

Pose analysis

Implementing OpenPose to calculate the angle of elbow and knee during shooting.

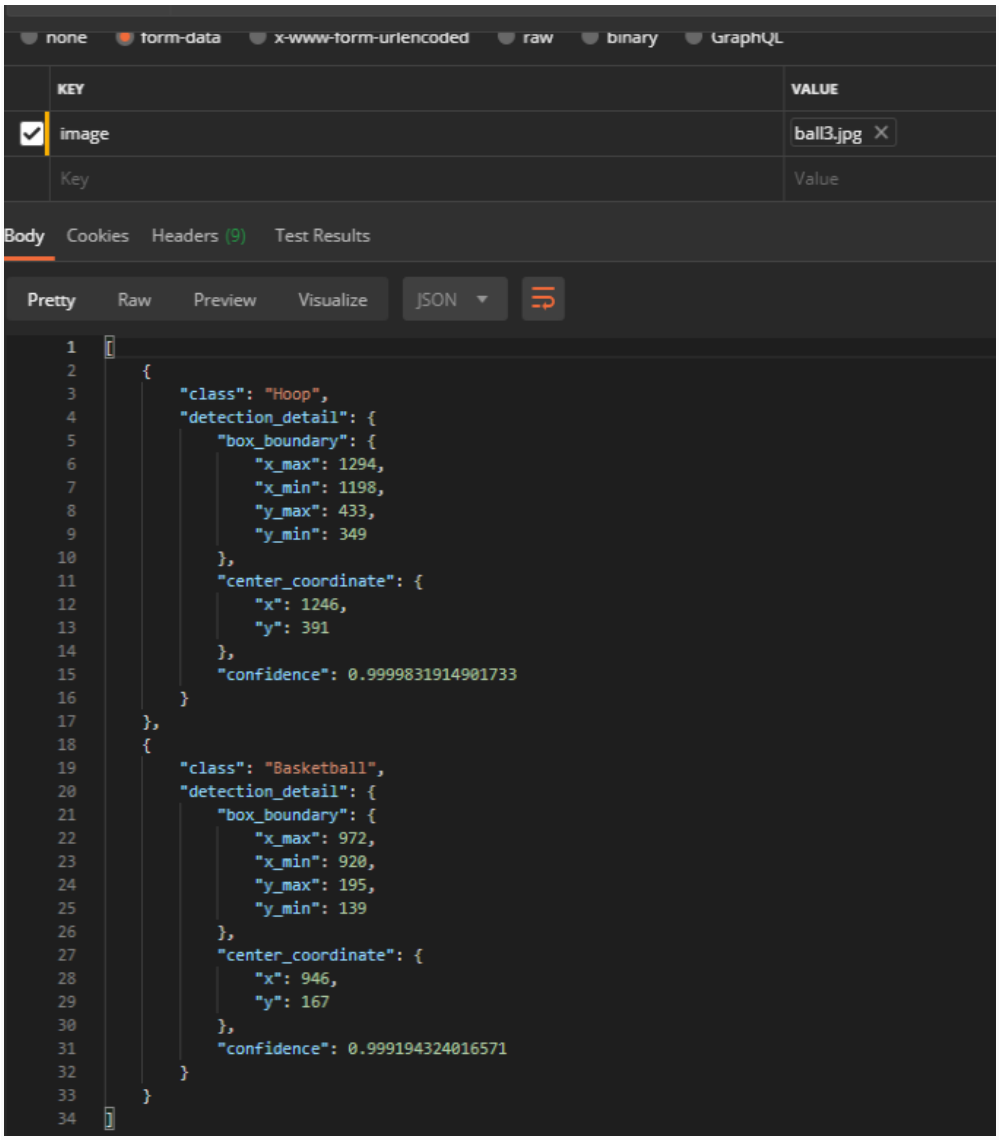


Release angle and release time are calculated by all the data collected from shot analysis and pose analysis. Please note that there will be a relatively big error for the release time since it was calculated as the total time when the ball is in hand.

Shot detection:

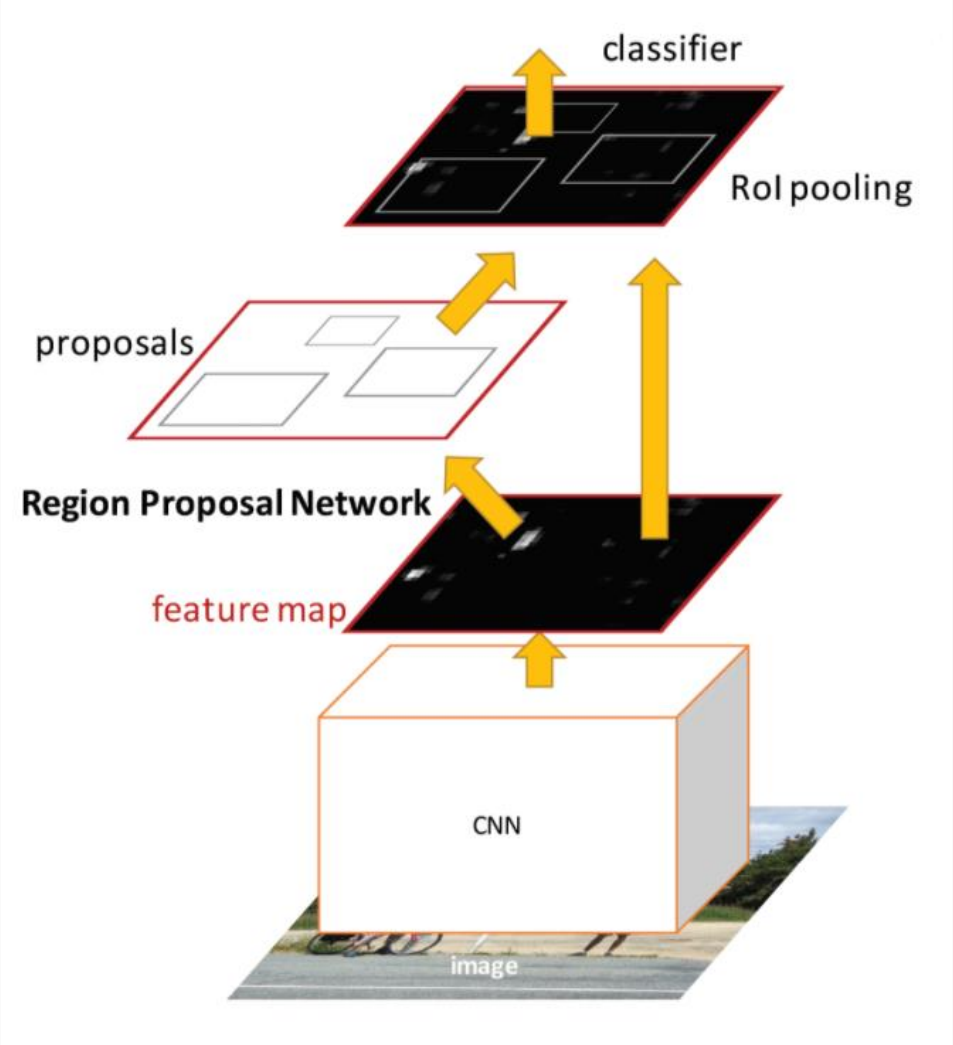
Detection will be shown on the image. The confidence and the coordinate of the detection will be listed below.

Detection API:



Detection Model

The object detection model is trained with the Faster R-CNN model architecture, which includes pretrained weight on COCO dataset. Taking the configuration from the model architecture and train it on my own dataset.



Creation of App

Here, I am creating Flask App. Input image or video is taken from the user. Then it calls respective py file with its method calculation in it. Then render respective .html page for solution.

Technical Aspect

Numpy used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. It contains a multi-dimensional array and matrix data structures. It can be utilised to perform a number of mathematical operations on arrays.

Matplotlib is used for EDA. Visualization of graphs helps to understand data in better way than numbers in table format. Matplotlib is mainly deployed for basic plotting. It consists of bars, pies, lines, scatter plots and so on. Inline command display visualization inline within frontends like in Jupyter Notebook, directly below the code cell that produced it.

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

TensorFlow is a Python library for fast numerical computing. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation. It has a unique approach that allows monitoring the training progress of our models and tracking several metrics.

SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.

The OS module in Python is a part of the standard library of the programming language. When imported, it lets the user interact with the native OS Python is currently running on. In simple terms, it provides an easy way for the user to interact with several os functions that come in handy in day-to-day programming.

The python sys module provides functions and variables which are used to manipulate different parts of the Python Runtime Environment. It lets us access system-specific parameters and functions. The first advantage the sys module offers to us is its independence from the host machine Operating System. This means that this module can work the same even if it is working on Windows or Macintosh or Linux or any given OS.

The Python time module provides many ways of representing time in code. provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

cd <PATH>
pip install numpy
pip install tensorflow==1.15.2

pip install matplotlib
pip install opencv-python
pip install flask
Note: If it shows error as 'No Module Found' , then install relevant module.
You can also create requirement.txt file as, pip freeze > requirements.txt
Create Virtual Environment:
conda create -n aib python=3.6
y
conda activate aib
cd <PATH-TO-FOLDER>
run .py or .ipynb files.
Paste URL to browser to check whether working locally or not.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.
eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\
23)End_To_End_Projects\Project_10_DL_FileUse_End_To_End_AIBasketball
conda create -n aib python=3.6
y
conda activate aib
In Anconda Prompt, pip install -r requirements.txt to install all packages.
In Anconda Prompt, write python app.py and press Enter.
Paste the URL 'http://127.0.0.1:5000/' to browser to check whether working locally or not.
Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder]
[cd means change directory]

Directory Tree/Structure of Project

Folder: AI_Basketball > AIB
requirements.txt
Procfile
app.py
Aptfile

AI_Basketball > AIB > templates
index.html
layout.html
result.html

shooting_analysis.html
shot_detection.html

AI_Basketball > AIB > static
AI_Basketball > AIB > static > css
main.css
AI_Basketball > AIB > static > detections
.jpg

AI_Basketball > AIB > static > img
.png

AI_Basketball > AIB > static > uploads
.mp4 and .jpg
AI_Basketball > AIB >src
app_helper.py
config.py
utils.py
AI_Basketball > AIB > OpenPose

AI_Basketball > AIB > OpenPose > models
Coco, body_25 and mpi folder
.bat and .sh file
AI_Basketball > AIB > OpenPose > openpose
__init__.py

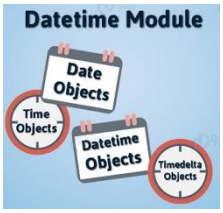
AI_Basketball > AIB > openPose > Release
.dll and .exe files

AI_Basketball > AIB > inference_graph
frozen_inference_graph.pb

To Do/Future Scope

Can also deploy on AWS and Google Cloud.

Technologies Used/System Requirements/Tech Stack

Download the Material

You can Download Entire Project from here: https://github.com/monicadesAI-tech/Project_81

You can Download App_File from here: https://github.com/monicadesAI-tech/Project_81/blob/main/app.py

You can see Detailed Project at Website here: https://github.com/monicadesAI-tech.github.io/aibasketball.html

Download dependencies from here: https://github.com/monicadesAI-tech/Project_81/blob/main/requirements.txt

Download images to test app from here: https://github.com/monicadesAI-tech/Project_81/tree/main/Test%20App%20Images%20%26%20Videos

Credits

Chonyy

Paper Citation

https://dl.acm.org/doi/10.5555/2969239.2969250

https://ieeexplore.ieee.org/document/8496641

https://arxiv.org/pdf/1812.08008.pdf