Project Details

The project:

Counting animals from the Image

Used:

Framework: Django

Language: Python

Lybraries: opencv, keras

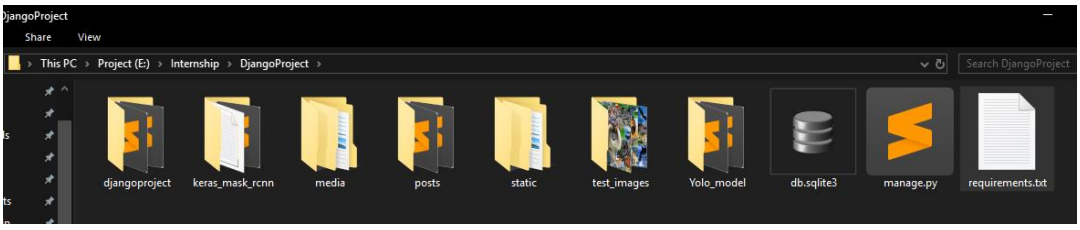
Techniques: Yolo, Maskrcnn

Install maskrcnn:

- ➔ **Go to** https://github.com/matterport/Mask_RCNN and download it.
- ➔ Extract the file and go the the **"Mask_RCNN-master"** folder and open terminal in that folder.
- ➔ Run the below command in the terminal.
  - Python setup.py install

How to Start the project:

- ➔ Extract Zip
- ➔ Set working directory to
  - DjangoProject

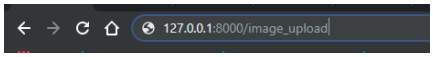


- ➔ Open "Requirements.txt" files and install Libraries
- ➔ Run server by below command
  - python manage.py makemigrations
  - python manage.py migrate

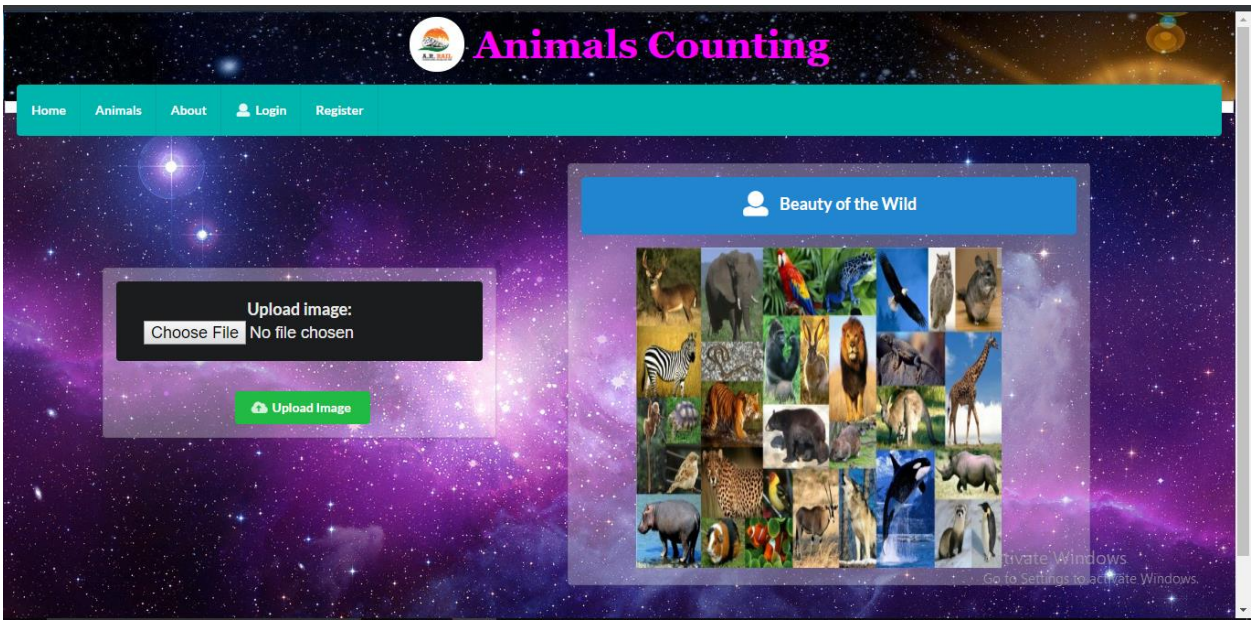
- python manage.py runserver

➔ Open browser(any) and enter below URL in URL-Box

- 127.0.0.1:8000/image_upload

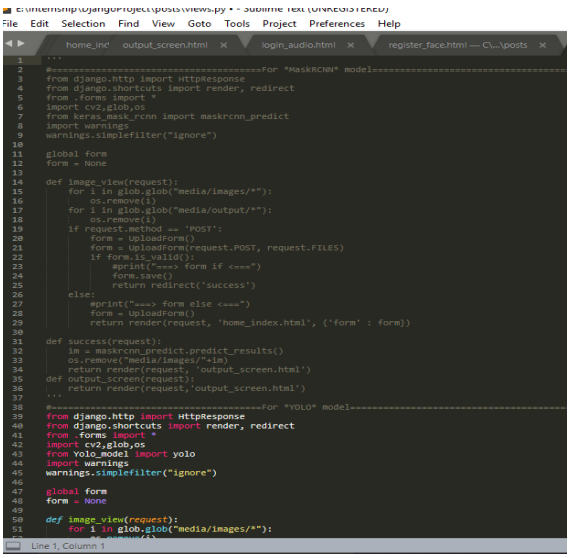


⇨ Home Screen

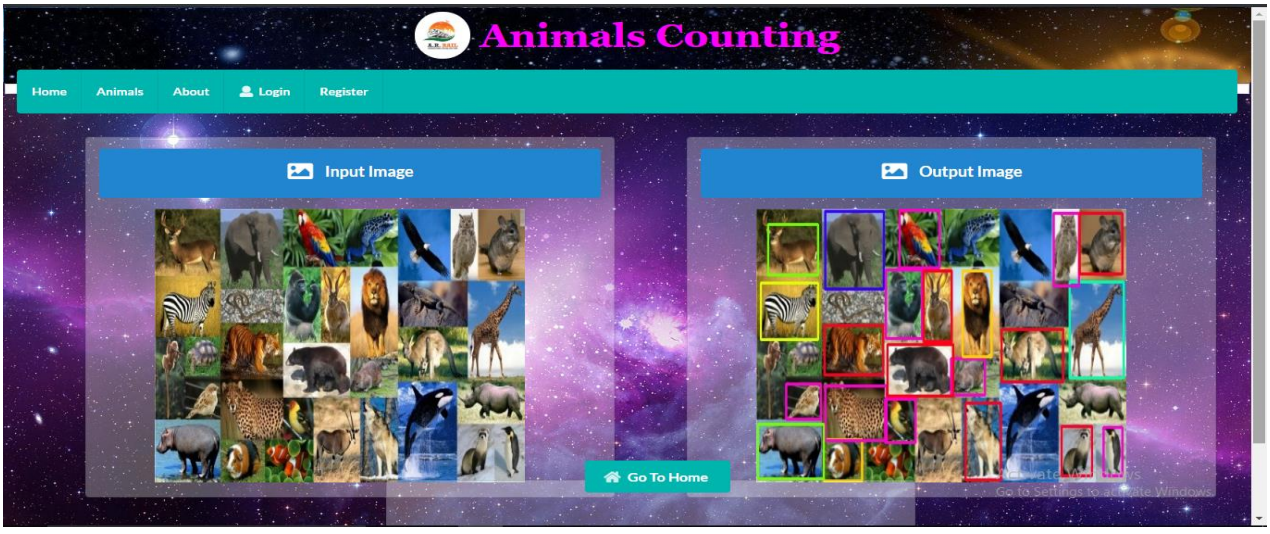


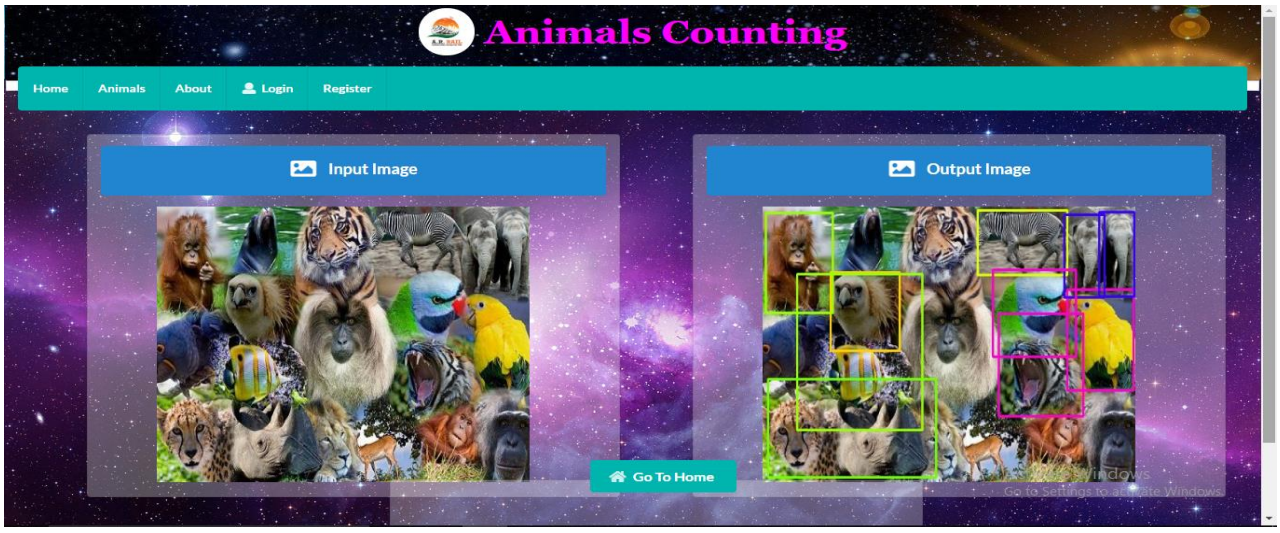
- Choose a file with and click on “Upload Image”
- Now Process will starts
  - Two Techniques are used (Yolo, MaskRCNN)
  - **“Yolo”** will takes 10 seconds to detect all objects(animals) from the image where **“MaskRcnn”** will takes 30 to 40 seconds on My ***CPU* system.**
    - **It will fast on detecting on *GPU* system.**
    - We can decrease the prediction time also by loading the model only once.
- Default Technique is ***MaskRCNN***
  - If you want to activate ***YOLO***, just activate commented code and deactivate default activated code(by adding and remove comments in appropriate places).***Refer below image.***

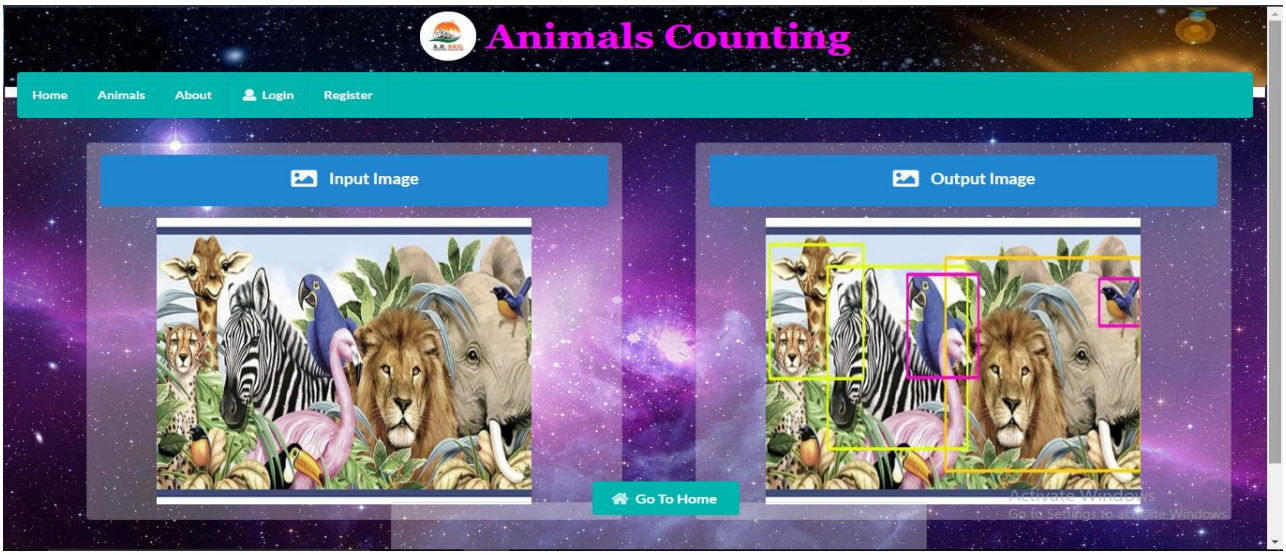
➔To➔



- **After prediction, the results**



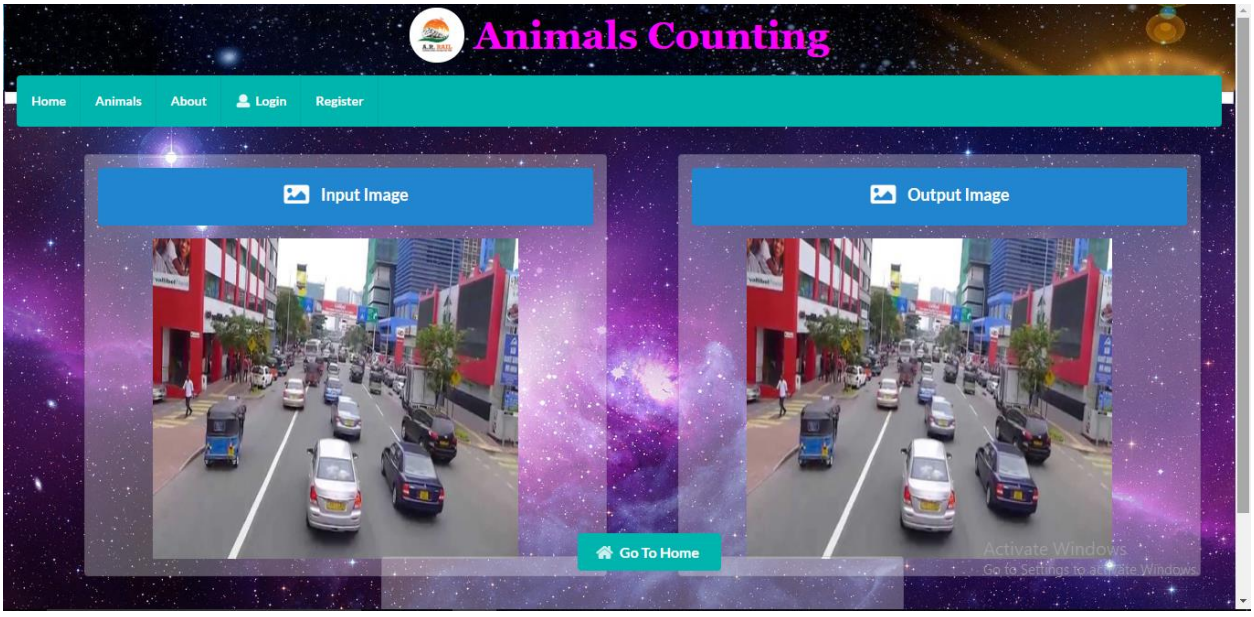




Above screens, output has some bounding boxes on it.Those are prediction results(localization of the animals).

➔ ***Count of animals* = *count of bounding boxes****

Note:*Please wait until Prediction results are out.(takes a long time based on techniques that I have used.)*



Special Case: For above input image, there is no bounding boxes in the *output image*.because this project only detects animals and gives bounding boxes to them. Rest of the objects will be ignored.

==➔ Improvements:

➔Decreasing prediction time

- **GPU systems**
  - For now time complexity is quite high because of I'm having CPU system. This project can run fastly in GPU systems.
- **Loading Trained model Only Once**
  - Every time model is to be loaded. Because, Django will not clears sessions automatically. We have to do some ground work on Django, then we can able to decrease time for predictions.
- **We can use other Multi label detectors like**
  - SSD, fast_rcnn, fast_rcnn for more accurate results.
- **Training on own dataset**
  - We can gather our own dataset related to all animals and train it by any one of the "Multi object detectors" like SSD,YOLO,Faster-RCNN for more accuracy.

Contact Details:

Name: Rajashekhar.G

Linkedin prof: https://www.linkedin.com/in/rajashekhar-gugulothu-26b01112a

Email: rajharry418@gmail.com

Contact No: 9550706607