

## Project Introduction

Sustainable Computing Research Lab has already built a project LabelLab, that has both web as well as mobile app with a great UI and functionality of letting users Login/Signup, View/Create projects, View/Edit images for labelling, selection using both bounding box and polygon and label them manually. However this webapp currently lacks Machine Learning models that can automatically generate labels by detecting objects in an image or multiple images on the fly.

There are various ML models present out there for object detection and labelling. Out of many such models, we will be using **Faster RCNN** and **YOLOv3** as our working models. We will also be using an image segmentation model called **Mask RCNN** for instance segmentation and masking purpose that is more meaningful and easier to analyse.

Currently the Machine Learning model integration in the project hasn't started. Therefore I wish to implement a complete functional and scalable Machine Learning model from scratch. Also along with revamped backend (later), Django which is built on python will act as boon for implementing ML models.

## Project Goals

- **Use PyTorch Deep Neural Network library for various object detection model implementation.**
  - PyTorch and TensorFlow, both are considered as a state of art Deep Neural Network library for object detection. TensorFlow despite being more mature and having larger community than PyTorch, there are few reasons why we should use PyTorch as DNN library for our project which are listed below:-
    1. Has a simple API.
    2. GPU enabled drop in replacement for NumPy.
    3. PyTorch code is known for executing faster and more efficiently.
    4. Better Graph construction and Debugging.
    5. Dynamic computation Graphs.
    6. Graphs can be built at runtime.
    7. Debugging PyTorch code is same as Python code as PDB-Python Debugger can be used.
- **Integrate optimal Machine Learning(ML) model out of many for object detection that balances between speed and accuracy.**
  - There are various ML models present out there for object detection and labelling. R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, SSD, FPN, RetinaNet, YOLOv3 etc. are among few of them. We will also be using a image segmentation model for masking purpose that is more meaningful and easier to analyze. So, out of these many models we will be using three best models that gives optimal speed and accuracy and is easier to analyze. The three models are:-
    1. **Faster RCNN** and
    2. **YOLOv3** for object detection using bounding box
    3. **Mask RCNN** for Instance Segmentation

- **ML models should not be limited to just binary classification, rather it should be able to detect objects of multiple classes and instances.**
  - Use of above three proposed model ensures the identification of object that can be of multiclass as well as single class and also of different instances.
- **ML models should be able to detect objects in images and label it in both boundary box and mask selection/segmentation.**
  - Use of above proposed models also ensures that a proper label name along with the identified object and its confidence is automatically generated.
- **Implementation of performance graph, considering speed and accuracy as parameters to compare results of generated labels in image using various models.**
  - After a user generates label using all or any of the above three models, its becomes very important to get a proper performance graph along that will have speed and accuracy as its parameters. So, this feature should be implemented.
- **ML models should be able to perform not only on static images but also in real-time scenarios like videos/webcam.**
  - Users might provide an image or a video as input to the label app. So it's important that the ML model should be able to perform detection task in real-time.
- **Feature to apply various image smoothing techniques, ranging from low to extreme, in order to suppress/reduce noise in images resulting in achieving accuracy.**
  - A feature of image smoothening along with their range from low to extreme should be implemented to reduce noise present in the images.
- **Generated labels after object detection by a model, should clearly mention the label name along with it.**
  - Use of Above proposed models also ensures that a proper label name along with the identified object and its confidence is automatically generated.
- **Automatically add detected new labels from the models to labels database**
  - New labels should be automatically added to the labels database whenever the model identifies new labels in the image. This gives user a concise list of all the labels detected by the model in various images.
- **View the previous list of images from database and apply ML model to it for label generation.**
  - Users should be able to view all the list of images present in the database and apply ML model to it at any point of time.
- **Integrate selection of various models and generation of labels accordingly in the label UI.**
  - Users should be able to select the above proposed models from the dropdown list generate labels with ease.
- **Feature to apply various ML model to multiple images at once.**
  - Users might upload multiple images at once. Therefore there must be a feature to label those multiple uploaded images at once.
- **Documentation using Sphinx.**
  - We will be using Sphinx for all the required documentation purpose.

## Why “Faster RCNN, YOLOv3 and Mask RCNN” models?

I've discussed earlier in the project goals about the presence of many models for object detection such as R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, SSD, FPN, RetinaNet, YOLOv3, Semantic segmentation, Instance segmentation, etc. So, now in this section I will be briefly explaining about why I think that my proposed model stands out from all these.

- **Faster RCNN(Region-based Convolutional Neural Networks):-**

- Unlike its other predecessor RCNN, Fast-RCNN which uses selective search to generate regions and further generate predictions, Faster RCNN replaces the selective search method with region proposal network which has made the algorithm much faster.
- It maintains a good balance between accuracy and speed.
- It detects even very small objects in an image since it has nine anchors in a single grid.
- However, this model fails when performing real-time detection.

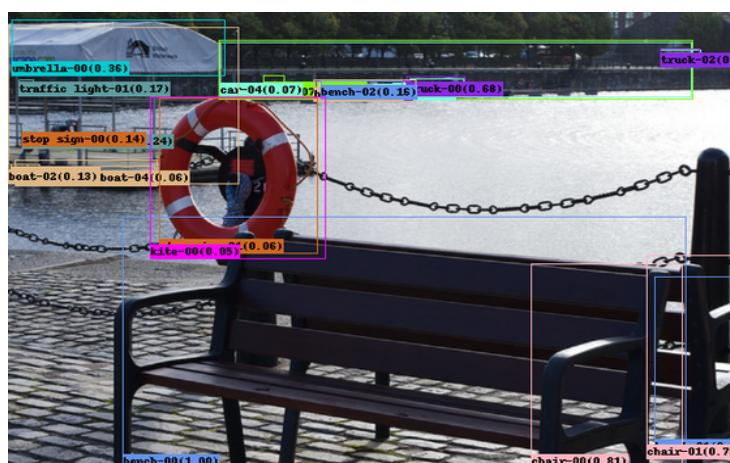


Fig:1 Faster RCNN in action

- **YOLO\_v3(You Only Look Once):-**

- All the features and advantages of YOLO model is same as faster RCNN, except for its ability to detect very well in real-time that Faster RCNN lacks at the moment.
- However, this model fails in detecting small and close to each other objects in an image due to only two anchor boxes in a grid predicting only one class of object.

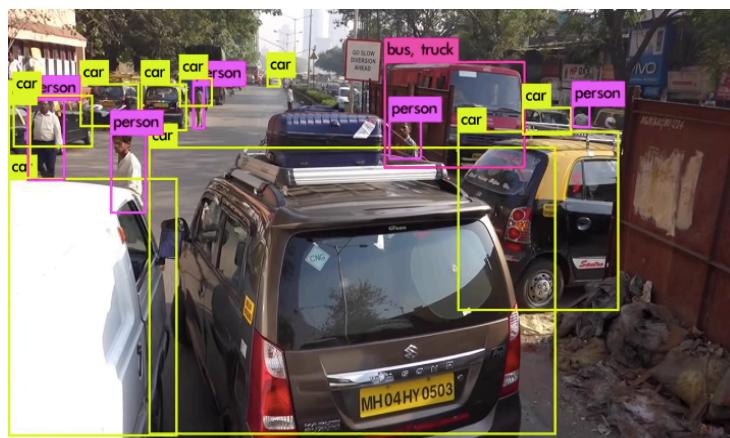


Fig: 2 YOLO\_v3 in action(real-time)

- **Mask RCNN:-**

- Unlike the other two models discussed above that can only detect objects in an image with rectangular boundary box, Mask RCNN can identify both the location of the object and each pixel that the object occupies in the picture.
- It can detect multiple instances of same classes that are overlapping and then classify the pixels' membership to only a particular instance.



Fig: 3 Mask RCNN in action

## Implementation

Initial phase of the project will be the integration of **PyTorch** Deep Neural Network library along with the revamped backend- Django.

Then the three proposed model “Faster RCNN, YOLOv3 and Mask RCNN” will be integrated along with the PyTorch Deep Neural Network and will be tested thoroughly with the wide range of images and videos.

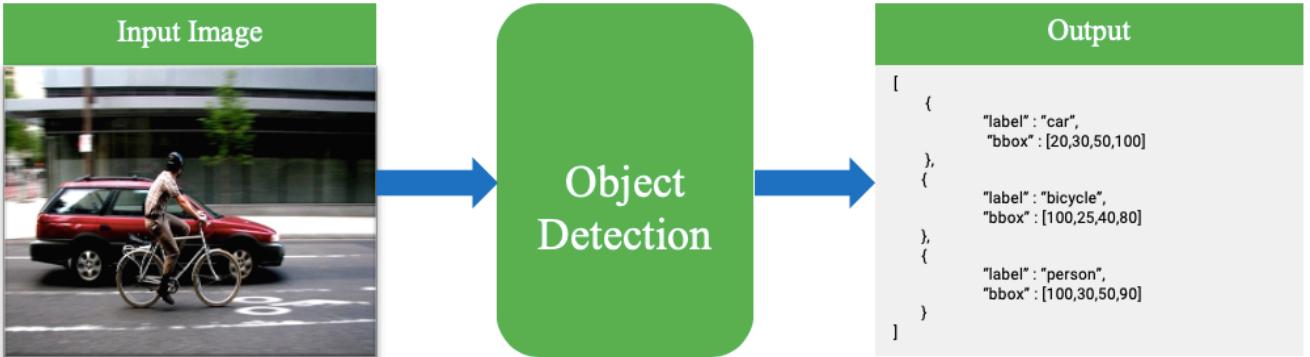
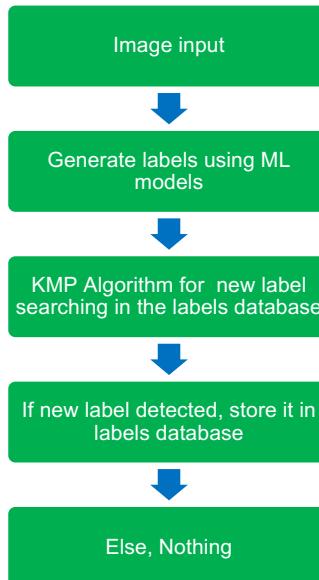


Fig: 4 Object detection mechanism

Performance graph visualization with speed and accuracy as parameters should be implemented in real time. PyTorch DNN comes handy for this purpose as it can handle dynamic graphs in real time with ease.

Identification and validation of all newly generated labels by a model with the existing labels present in the database should be implemented. For this purpose all the existing labels in the database are stored as strings in array data structure. Then we perform **Knuth–Morris–Pratt algorithm** on every string/labels present in an array and if the detected label is not found in the array, we append the label to the existing array and insert it in labels database. We will be using Knuth–Morris–Pratt algorithm for this purpose because the worst case complexity for searching a string in the data structure is  $O(n)$ . Therefore its highly efficient.



Flowchart: 1 Label searching

Maintaining balance between the speed and accuracy during the label generation is a must. A model should be able to generate highly accurate labels without compromising the speed. Therefore various digital image pre-processing techniques like smoothening, sharpening, etc. should be used for the reduction of noise, managing over exposure, etc. in the image.

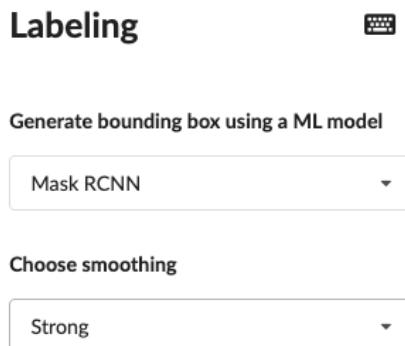


Fig: 5 Prototype for smoothing

Testing of model working and performance will be done with various types of images ranging from single class object to real-time video footage. This ensures the app reliability and versatility.

The final documentation will include projects readme files, user manual, test cases and API documentation for rest API.

Since the project is Machine Learning based, I believed in prototyping first and testing it thoroughly with different image inputs before pushing it to production code. Below are some of the prototype ML model Labelling web app images I've created:-

## 1. Detection using Faster RCNN model:-

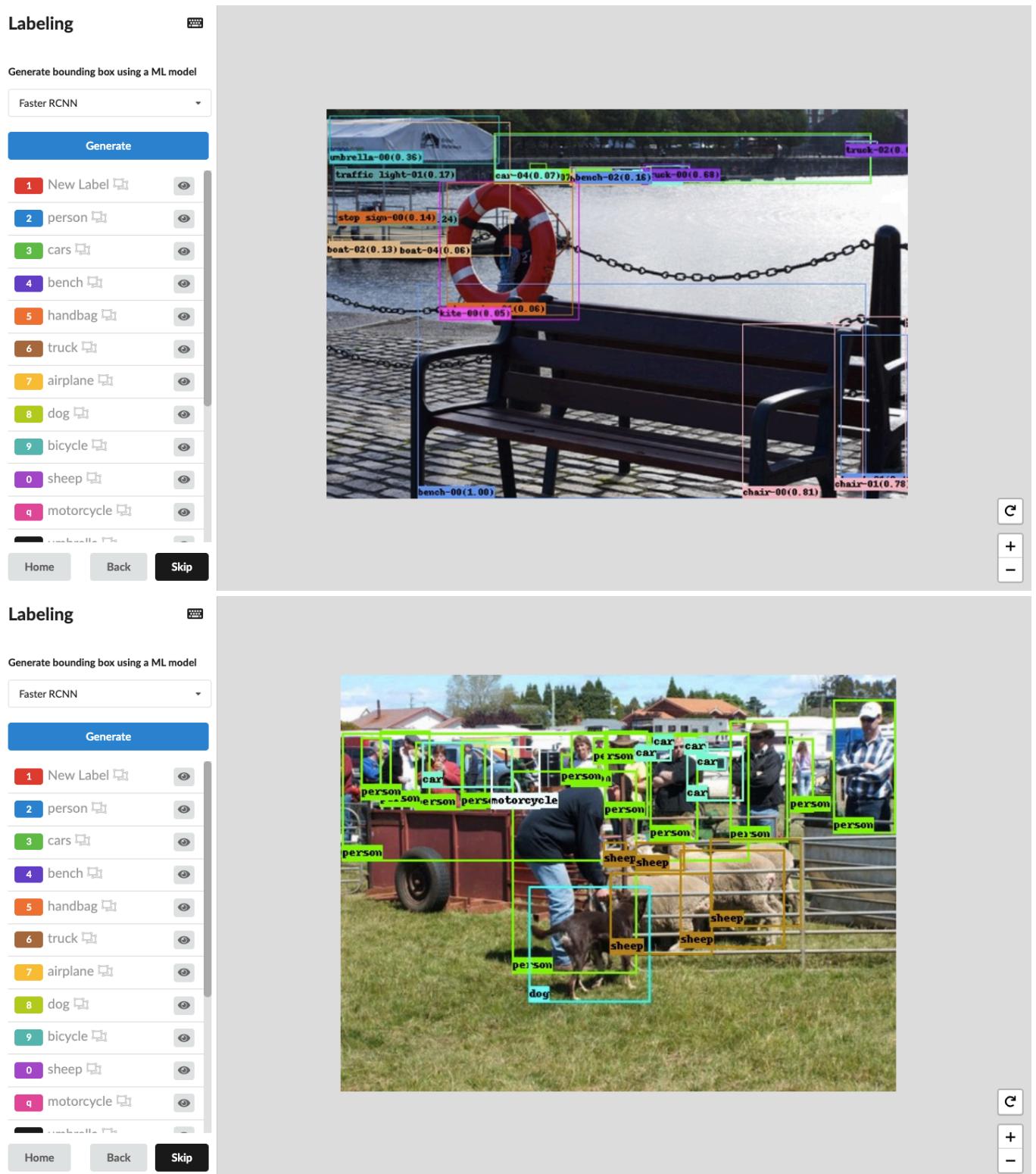


Fig:6 Detection using Faster RCNN model

## 2. Detection using YOLOv3 model(Real-Time):-

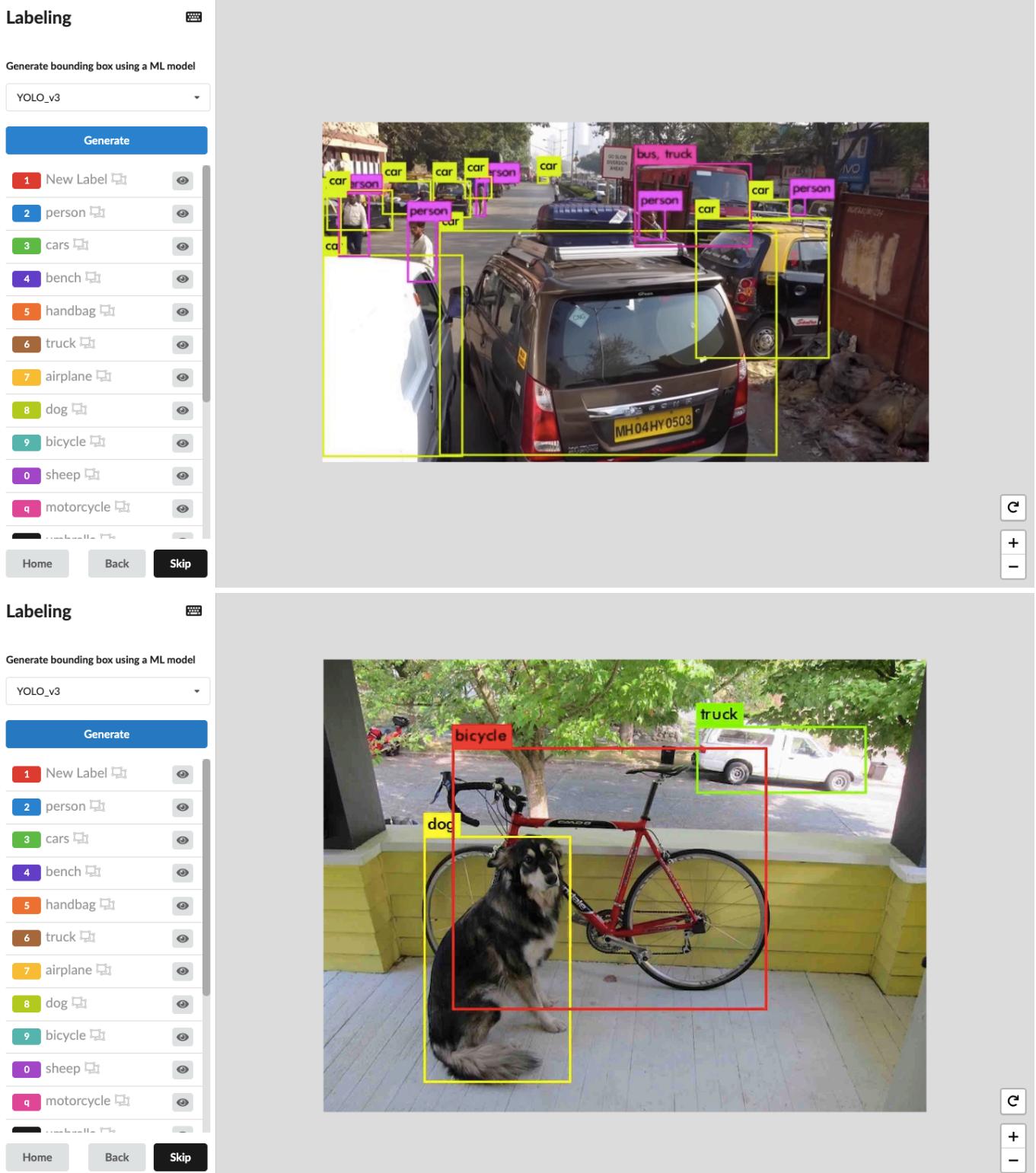


Fig:6 Detection using YOLOv3 model

### 3. Detection using Mask RCNN model(Instance segmentation):-

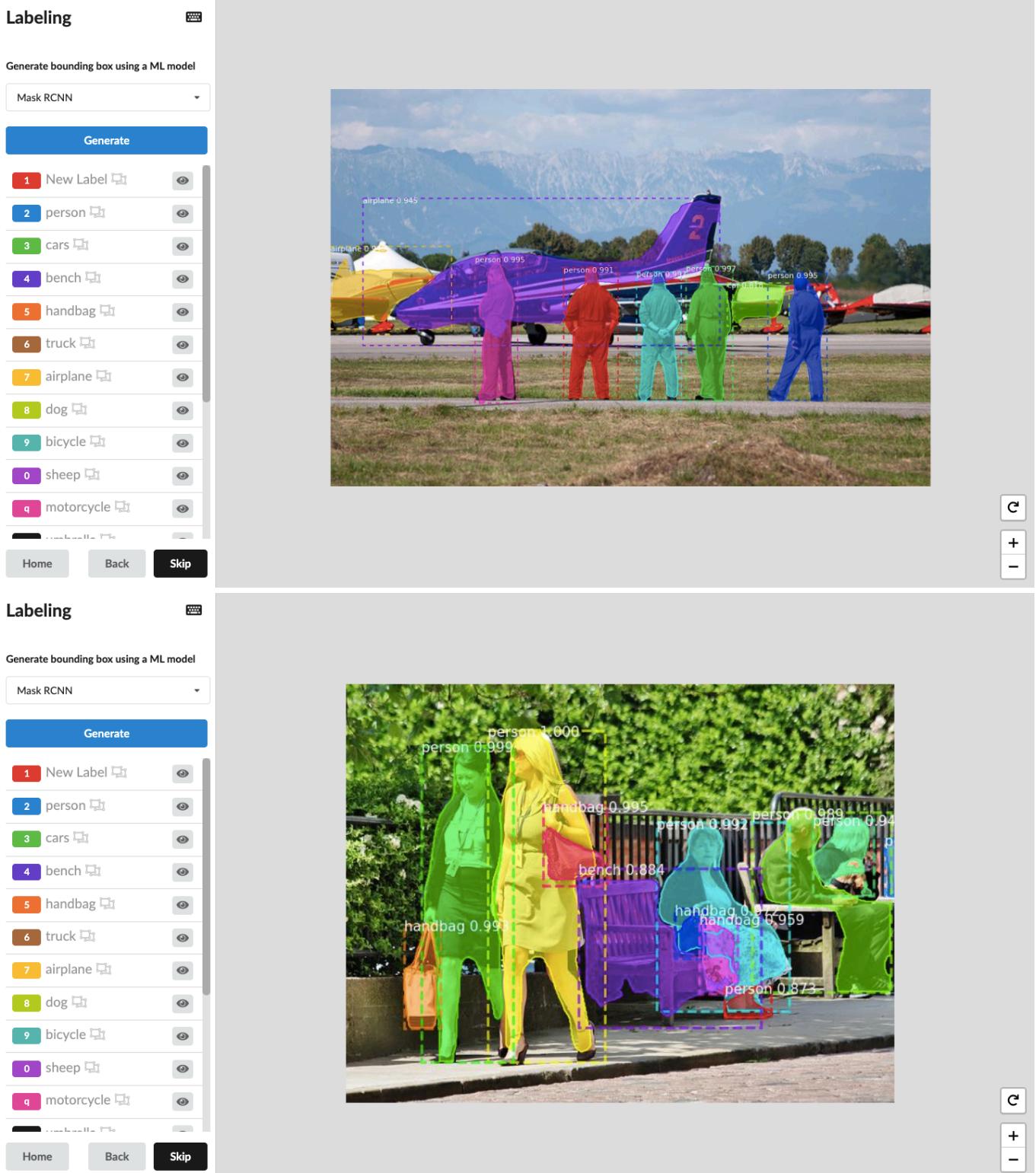


Fig:6 Detection Mask RCNN model

## Timeline

Milestone	Title	Start Date	End Date	Achievement
GSoC	Community Bonding	04/05/2020	01/06/2020	
M1	Research and prototype work	01/06/2020	28/06/2020	Research on different ML models and its implementation
M1.1	Get know about the community better			
M1.2	Discuss and finalize the backend implementation of revamped LabelLab web app			
M1.3	Decide on best Deep Neural Network library suitable with the project requirement		Easier to use, understand and maintain in long run	
M1.4	Decide on the best possible ML models			Ensures balance between speed and accuracy
GSoC'20	Phase 1 Evaluation	29/06/2020	03/07/2020	
M2	Functional/Core Development	29/06/2020	26/07/2020	Implementation of Deep Learning network along with different finalized models integration and performance graph
M2.1	Implement finalized Deep Neural Network integration from M1.3			
M2.2	Implement finalized Machine Learning models integration from M1.4			
M2.3	Implement performance graph of different models used			Ensures better visualization of speed and accuracy
M2.4	Implement UI for all the above functionalities to be available for end users			Users can finally apply ML models to automatically generate labels in an image
GSoC'20	Phase 2 Evaluation	27/07/2020	31/07/2020	
M3	Feature Development, Testing, Code Review and Documentation	27/07/2020	23/08/2020	Assures overall better user experience Increase in speed and accuracy of models High quality code and better documentation
M3.1	Implement feature of labelling multiple images at once and add newly detected label after generating labels to the labels database.			

M3.2	Implement various image pre-processing techniques like smoothening, sharpening, etc.			Reduces noise, manages over exposed images, etc. to get better accuracy without compromising on time
M3.3	Training and Testing ML models with numerous datasets			Makes ML model work more efficient when given with different set of ranges of images  Through testing of model ensures reliability and versatility
M3.4	Code Review, Documentation using Sphinx			High quality code  Finished documentation for the project
GSoC'20	Final evaluation	24/08/2019	31/08/2019	
Post GSoC'20	Continue to work on the project			Work on extra milestones of the project more than the proposed one and keep maintaining the codebase in regular intervals

## SCoRe Contributions

I joined SCoRe community during GSoC'19 last year in which I almost built complete prototype of LabelLab web app frontend using ReactJS as mentioned in the challenging task. But I couldn't complete it fully/submit it as my internship at Bangalore started. However this year for GSoC'20 I chose kind of different and interesting project than last year in which I'm genuinely interested.

My contributions this year has just been limited to raising few issues that includes of bug reports and feature request as integrating a Machine Learning model to a project requires a lot of research and prototypes. Below are the few issues I've raised recently:-

- <https://github.com/scorelab/LabelLab/issues/397> (Bug Report)
- <https://github.com/scorelab/LabelLab/issues/396> (Bug Report)
- <https://github.com/scorelab/LabelLab/issues/390> (Feature Request)
- <https://github.com/scorelab/LabelLab/issues/389> (Feature Request)

## Personal Information

### Basic:

**Name:** Nikhil Gupta

**Gender:** Male

**Nationality:** Nepal  | (GMT+ 5.45)

### Education:

**Institute:** [NIT, Rourkela](#)

**Degree:** B. Tech

**Major:** Computer Science & Engineering

**Graduation:** 2020

### Social:

**GitHub:** [nguptaa](#)

**LinkedIn:** [Nikhil Gupta](#)

### Contact:

**Email address:** [116CS0183@nitrkl.ac.in](mailto:116CS0183@nitrkl.ac.in)

/ [nguptaa183@gmail.com](mailto:nguptaa183@gmail.com)

**Phone number:** +91 9901408565  /

+977 9842027627 

**Website:** <https://ngupta.me>

## About me

I am Nikhil Gupta, final year undergrad at National Institute of Technology, Rourkela. I've always been a technology enthusiast and is still actively involved in various projects and researches throughout my academic years at my institute. My most recent project include a "web app that takes image as an input, identifies and extract text in it and provides output in speech". Also my ongoing research project at NIT Rourkela includes "Efficient timeout based flow table management in OpenFlow Enabled SDN using Machine Learning".

I have experience working with Python, NodeJS, ReactJS, C/C++, HTML, CSS, JavaScript, PHP, DevOps, VCS/Git, Bootstrap. My field of interest includes but not limited to Machine Learning/AI, Data Science, Web Development, Microservices, Scalability.

My previous work experiences, skills and leadership has prepared me to be a valuable asset to the SCoRe community and therefore I look forward to be a valued contributor to this amazing team.

## Reference

1. <https://www.scorelab.org/gsoc-2020/#labellab03>
2. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
3. [https://everitt257.github.io/post/2018/08/10/object\\_detection.html](https://everitt257.github.io/post/2018/08/10/object_detection.html)
4. <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>
5. <https://pjreddie.com/darknet/yolo/>
6. [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)
7. [https://docs.google.com/document/d/1ErDhg\\_8wRGGVPyBeuOYdzSzKVp2ZmzAXwt5zOL6zjsE/edit#](https://docs.google.com/document/d/1ErDhg_8wRGGVPyBeuOYdzSzKVp2ZmzAXwt5zOL6zjsE/edit#)

## Questions

### 1. Are you a SCoRe contributor/ Have you contributed to SCoRe before?

No, as I haven't contributed any commits to the SCoRe codebase.

Yes, as I've been actively researching and working on prototyping various Machine Learning models required to generate labels in images in LabelLab project maintaining balance between speed and accuracy.

### 2. How can we reach you if we have questions about your application?

Email address: [116CS0183@nitrkl.ac.in](mailto:116CS0183@nitrkl.ac.in) / [nguptaa183@gmail.com](mailto:nguptaa183@gmail.com)

Phone number: +91 9901408565  / +977 9842027627 

### 3. What is your GitHub username?

nguptaa

## Project Specific Questions

### 4. Which SCoRe GSoC project are you applying for?

LabelLab - Integrate ML module for LabelLab

### 5. What do you plan to accomplish over this summer for this project?

#### a. What project you want to work on?

LabelLab - Integrate ML module for LabelLab

#### b. How you will approach that project portion (with your milestones)?

1. Integrate Deep Neural Network- PyTorch with the revamped backend Django
2. Through testing of different Machine Learning models using multiple images
3. Implement model selection dropdown in frontend
4. Feature to label multiple images at once
5. Add new labels to database after detection
6. Implement performance graph that shows results of different models and has speed and accuracy as parameters
7. Implement few image pre-processing techniques like smoothing to reduce noise in images.
8. Drafting the documentation.

### 6. If you have your own project to propose, please describe it here:

My own project proposal was to revamp the backend of webapp by replacing NodeJS with Django as backend language. Django is built in python and there are literally hundreds and thousands of python libraries dedicated to machine learning. From TensorFlow, PyTorch, Keras to OpenCV we can use any heavy computing libraries in backend Django. With all high quality tools and libraries of python language we get a very enriching experience to do almost any stuff and coupled with Django it's even better, we can deploy our own trained ML models easily.

## 7. Projects related details

Since the backend of project has been written in NodeJS, it by no means is a bad language. But since our LabelLab project emphasize on generating labels, image pre-processing, etc. I found problem finding a good working Machine Learning model and also very little community support for NodeJS language. With NodeJS it won't be an enriching experience implementing Deep Neural Network along with various ML models. Also deploying our own trained ML models becomes very cumbersome task.

Therefore I suggest to use Django as backend language as it is built in python and Machine Learning in python can do wonders and won't have any boundaries.

## 8. List down any plans you have during this summer

I don't have any specific plans for this summer, however I've got my end semester examination postponed to last week of June cause of Covid-19. I can dedicate 35 hours per week for my GSoC project.

## 9. Education:

### a) What year are you in school?

4<sup>th</sup> year

### b) What programming courses have you taken?

- i. Basic programming in C – C
- ii. Object Oriented Programming in C++ - C++
- iii. Learn Git by Codeacademy – Git
- iv. Complete Pythonv3 bootcamp by Jose Portilla – Pythonv3
- v. Machine Learning by Andrew Ng
- vi. Python for Data Structures, Algorithms, and Interviews by Jose Portilla
- vii. The Web Developer Bootcamp by Colt Steele
- viii. PHP for Beginners - Become a PHP Master - CMS Project by Edwin Diaz
- ix. The Complete Node.js Developer Course (3rd Edition) by Andrew Mead, Rob Percival
- x. Modern React with Redux [2019 Update] by Stephen Grider

### c) What is your major?

Computer Science & Engineering

### d) Have you done group projects (programming or otherwise)?

#### i. University Projects/Research:

- Efficient timeout based flow table management in OpenFlow Enabled SDN using Machine Learning (Group Research) – *Python, Ryu Controller, Mininet, Machine Learning models*
- Revamped website of Treasherlocked, India's biggest online cryptic hunt event hosted by Microsoft Campus Club of which I served as Technical Coordinator (Group Project) – *HTML, CSS, JavaScript, PHP*
- Imgeech, online image to text and extracted text to speech (Group Project) – *HTML, CSS, JavaScript, NodeJS*

ii. Individual Projects:

- **Sentiment Analysis(Text Mining):** A python script written out of a curiosity to know the reaction of peoples of my country Nepal on newly elected PM. The script basically scrapes all the top tweets on PM topic and based on the user response it performs sentiment analysis on it and determines whether the opinion is positive, negative or neutral.
- **Movie Tv show Notifier(Web Scraping):** A python script to send a user an email that contains information like release dates of the entered Movie/TV Show. It's written in python 3. All you need is to run the script, provide your email address and list of Movie/TV Shows you want to get notified.
- **Schedule Bot(Natural Language Processing):** Schedule Bot is a messenger based NLP(Natural Language Processing) bot which I made during my freshman year when I was too annoyed with my paper timetable as I have to see it every other hour. This bot helps a person by letting them know their particular subject at a particular time when provided with respective section and lab group.
- **Microsoft Campus Club(Full Stack Website):** Developed a full stack website for the club throughout the summer from scratch. Temporarily it's hosted at <https://msclubnitrkl.000webhostapp.com> Soon it will be live on NIT Rourkela subdomain. HTML, CSS, JavaScript are used in the frontend part, PHP in the backend and MySQL in the database.
- **NIT Rourkela Voice(Content Management System):** NITRkl-voice is a full stack content management system which I made for my DBMS project under Prof. Korra Sathya Babu. Temporarily it's hosted at <https://nikhilgupta.000webhostapp.com>. HTML, CSS, JavaScript are used in the frontend part, PHP in the backend and MySQL in the database.

e) What was your primary contribution to/role in the group?

Currently I'm working as undergraduate Researcher working on the domain of Software Defined Networking(SDN) using Machine Learning Techniques. My contribution to/ role as a researcher is to implement recent papers written on SDN and apply my own custom ML models that results in efficient flow table management in OpenFlow enabled SDN.

I was project leader of both the group projects mentioned in the university projects therefore crafting entire project from scratch using mentioned programming language was entirely done by me.

## 10. Do you have work experience in programming? Tell us about it.

### **Summer Intern at PropellD, Bangalore**

*(May 2019-June 2019)- 2 months*

Created a complete well documented service called cache service and integrated it with frontend for visualization. Also integrated consul services for health check-up and service discovery in few services used, and remotely contributed in few API's documentation of different services used.

### **Technologies Used:-**

NodeJS, React, Python, Consul, Swagger for API documentation.

## 11. Do you have previous open source experience? Briefly describe what you have done.

### **Winter Open Source Intern at KWoC(Kharagpur Winter of Code), IIT Kharagpur**

*(Dec 2018-Jan 2019)- 2 months*

Contributed in developing student information portal of National Institute of Technology, Durgapur. Also added continuous integration and code coverage in the GitHub repository for better integration.

**Blog Link:-** [Nikhil Gupta-KWoC](#)

### **Technologies Used:-**

HTML, CSS, JavaScript, Python, Django, Flask.

## 12. Tell one interesting fact about yourself.

Coding stuffs related to problem around us and watching them in action is my adrenaline rush.