A
PROJECT REPORT
ON

Crowd Computing

Submitted by

Keval Padsumbiya(17IT405)
Maulik Beladiya(17IT409)
Gopal Sakhwala(17IT411)
Neel Savani(17IT469)

For Partial Fulfillment of the Requirements for Bachelor of Technology in Information Technology

Guided by
Prof. Prachi K. Shah

May, 2021

Information Technology Department
Birla Vishvakarma Mahavidyalaya Engineering College
(An Autonomous Institution)
Vallabh Vidyanagar – 388120
Gujarat, INDIA

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2020-21, Semester VIII

CERTIFICATE

This is to certify that the project work entitled **Crowd Computing** has been successfully carried out by **Keval Padsumbiya (17IT405), Maulik Beladiya (17IT409), Gopal Sakhwala (17IT411), Neel Savani (17IT469)** for the subject **IT442-Minor Project II** during the academic year 2020-21, Semester-VIII for partial fulfilment of Bachelor of Technology in Information Technology. The work carried out during the semester is satisfactory.

Prof. Prachi K Shah
IT Department
BVM

Dr. Keyur Brahmbhatt
Head, IT Department
BVM

ACKNOWLEDGEMENT

First of all, We are indebted to the GOD ALMIGHTY for giving us an opportunity to excel in our efforts to complete this project on time.

We are extremely grateful to **Dr. Indrajit Patel,** Principal, Birla Vishvakarma Mahavidyalaya Engineering College and **Dr. Keyur Brahmbhatt,** Head of Department, Department of Information Technology**,** for providing all the required resources for the successful completion of our project.

Our heartfelt gratitude to our project guide **Assistant Prof. Prachi Shah**, Associate Professor in Information Technology, for her valuable suggestions and guidance in the preparation of the progress report and implementation of project.

We express our thanks to co-ordinators**,** all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

We will be failing in duty if We do not acknowledge with grateful thanks to the authors of the references and other literatures referred to in this project.

Last but not the list; We are very much thankful to our parents who guided us in every step which We took.

Thanking You,
Keval Padsumbiya
Maulik Beladiya
Gopal Sakhwala
Neel Savani

ABSTRACT

In the current digital era, at many places crowd counting mechanisms still rely on old-fashioned methods such as maintaining registers, making use of people counters and sensors based counting at entrance. These methods fail in the places where the movement of people is completely random, highly variable and dynamic. These methods are time consuming and tedious. The proposed system is developed for situations where emergency evacuations are required such as fire outbreaks, calamitous events, etc. and making informed decisions on the basis of the number of people such as food, water, detecting congestion, etc.

A deep convolution neural network (DCNN) based system can be used for near real-time crowd counting. This document presents the development of crowd counting model using CNN. This model has wide feasibility like image size and colour and resolutions and it is also work with great accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1. Introduction

1.1 Brief overview of the work

- Nowadays, there are many problems with **manual monitoring system** in any occasion or meeting, like

  - Large ongoing staff training cost.
  - System is dependent on good individuals.
  - Reduction in sharing information and customer services.
  - Time consuming and costly to produce reports.
  - Lack of security.
  - Duplication of data entry.

- In addition to, due to pandemic **social distance and traffic control** is require and this requirement arises solution **like crowd counting or crowd monitoring system**, so based on research we plan to develop crowd counting model using CNN. Moreover, we also train the model so it work at highest accuracy and with wide range feasibility.

- Moreover, we also train the model so it work at highest accuracy and with wide range feasibility.

1.2 Objective

- The objective of application is to automate manual monitoring system in public places and meetings etc.
- Also, model counting output will help the organization or person in decision making system and it mainly helps in surveillance.

1.3 Scope

- Crowd counting is essential to serve many real-world applications, such as resource management, traffic control, security, disaster management etc
- In addition to, this model can be integrated with existing system to automate monitoring such as community halls or temples or gardens.

1.4 Project Modules

- Data Gathering
- Data Pre-processing
- Network Finding
- Model Generation
- Web App Development

1.5 Project Hardware/Software Requirements

- **Hardware requirements:**
  - **Client side:**
    - Recommended: Android phone 8.0+ (Oreo or higher) ▪
    - Recommended: Basic computer/laptop that can run browser
  - **Server side:**
    - Recommended: Basic computer/laptop with Win10/Linux with integrated graphics and python (Scipy, Pyplot, Django and other ML libraries)

- **Development tools (Recommended):**
  - Tensorflow
  - Spyder
  - Jupyter Notebook
  - VS Code/Py Charm

Chapter 2. Literature Review

2.1 Literature Review:

- Crowd scenes analysis can be classified into three categories**: detection-based methods**, **regression based methods**, and **density estimation-based methods**. By combining the deep learning, the CNN based solutions show even stronger ability in this task and outperform the traditional methods.

i. Detection-based approaches:

- Most of the early researches focus on detection-based approaches using a moving-window-like detector to detect people and count their number [4]. These methods re- quire well-trained classifiers to extract low-level features from the whole human body (like Haar wavelets [5] and HOG (histogram oriented gradients) [6]).
- However, they perform poorly on highly congested scenes since most of the targeted objects are obscured. To tackle this problem, researchers detect particular body parts instead of the whole body to complete crowd scenes analysis [7].

ii. Regression-based approaches:

- Since detection-based approaches can not be adapted to highly congested scenes, researchers try to deploy regression-based approaches to learn the relations among extracted features from cropped image patches, and then calculate the number of particular objects. More features, such as foreground and texture features, have been used for generating low-level information [8]. Following similar approaches, Idrees *et al.* [3] propose a model to extract features by employing Fourier analysis and SIFT (Scale- invariant feature transform) [9] interest-point based counting.

iii. Density estimation-based approaches:

- When executing the regression-based solution, one critical feature, called saliency, is overlooked which causes in- accurate results in local regions. Lempitsky pro- pose a method to solve this problem by learning a linear mapping between features in the local region and its object density maps.
- It integrates the information of saliency during the learning process. Since the ideal linear mapping is hard to obtain, Pham use random forest regression to learn a non-linear mapping instead of the linear one.

iv. CNN-based approaches:

- Literature also focuses on the CNN-based approaches to predict the density map because of its success in classification and recognition. In the work presented by Walach and Wolf, a method is demonstrated with selective sampling and layered boosting. Instead of using patch-based training, Shang try an end-to- end regression method using CNNs which takes the entire image as input and directly outputs the final crowd count.

- Boominathan *et al*. [2] present the first work purely using convolutional networks and dual-column architecture for generating density map. Marsden explore single-column fully convolutional networks while Sindagi propose a CNN which uses the high-level prior in- formation to boost the density prediction performance.
- An improved structure is proposed by Zhang *et al*. [1] who introduce a multi-column based architecture (MCNN) for crowd counting. Similar idea is shown in Onoro where a scale-aware, multi-column counting model called Hydra CNN is presented for object density estimation.

Chapter 3. System Analysis & Design

3.1 Comparison of Existing Applications

- In Exiting system , There are crowd scientists who count the number of people in certain parts of an image and then extrapolate to come up with an estimate.
- More commonly, we have had to rely on crude metrics to estimate this number for decades.
- Crowd Counting is a difficult problem especially in dense crowds due to two main reasons:
    1. There is often clutter, overlap and occlusions present.
    2. In perspective view it is difficult to take into account the shape and size of object present with respect to the background.
- While we don't yet have algorithms that can give us the exact number, most computer vision techniques can produce impressively precise estimates.
- **Limitations of the state-of-the-art approaches:**

- Most recently, Sam propose the Switch-CNN using a density level classifier to choose different regressors for particular input patches. Sindagi present a Contextual Pyramid CNN, which uses CNN networks to estimate context at various levels for achieving lower count error and better quality density maps. These two solutions achieve the state-of-the-art performance, and both of them used multicolumn based architecture (MCNN) and density level classifier.
- However, we observe several disadvantages in these approaches: Multi-column CNNs are hard to train according to the training method described in work . Such bloated network structure requires more time to train. Multi-column CNNs introduce redundant structure. Different columns seem to perform similarly without obvious differences. Both solutions require density level classifier before sending pictures in the MCNN.
- However, the granularity of density level is hard to define in real-time congested scene analysis since the number of objects keeps changing with a large range. Also, using a fine-grained classifier means more columns need to be implemented which makes the design more complicated and causes more redundancy. These works spend a large portion of parameters for density level classification to label the input regions instead of allocating parameters to the final density map generation. Since the branch structure in MCNN is not efficient, the lack of parameters for generating density map lowers the final accuracy. Taking all above disadvantages into consideration, we propose a novel approach to concentrate on encoding the deeper features in congested scenes and generating high quality density map.

3.2 Project Feasibility Study

- **Technical feasibility:**

- This project is web based so it can feasible with any android or ios or pc devices with browser support but it require higher computing power.

- **Economic feasibility:**

- The project is economically feasible because it can run on any devices whatever client already has.

- **Operational feasibility:**

- It will need an device with camera that can capture good quality of image and video.

3.3 Project Timeline chart

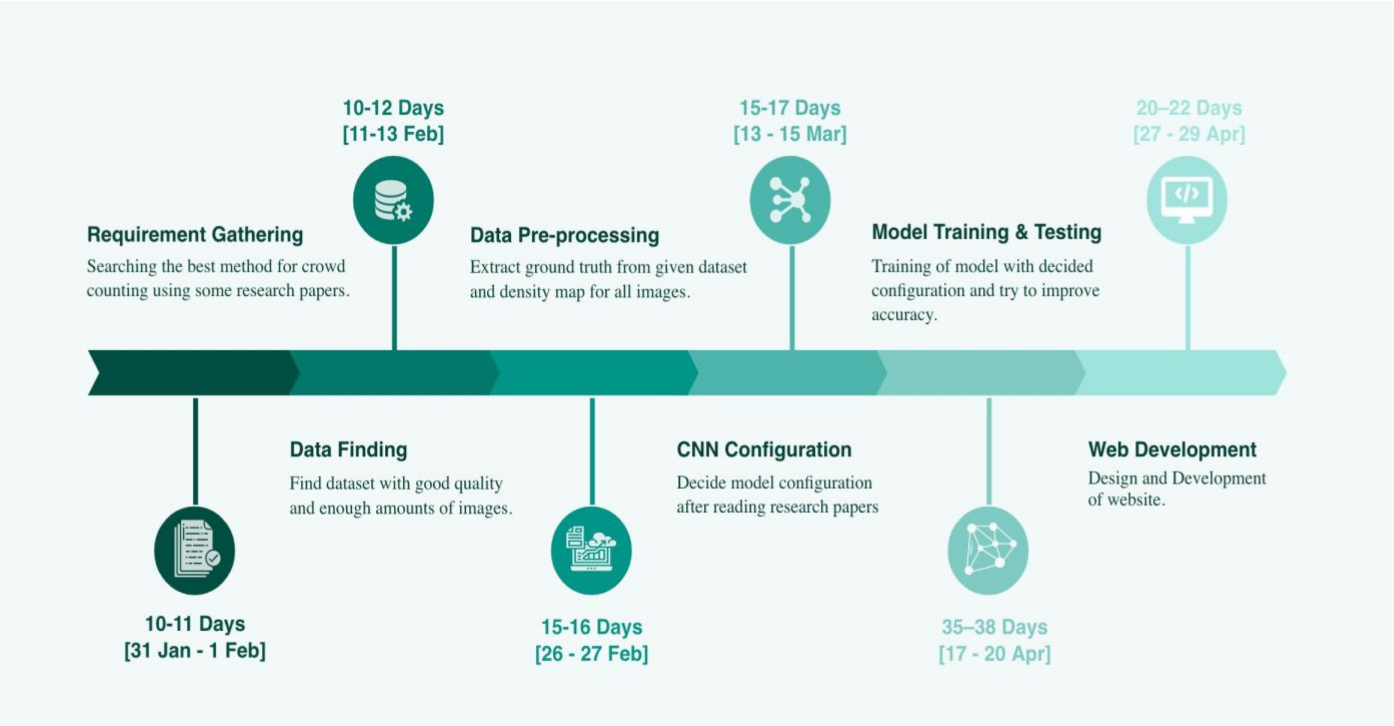

Figure 1 Timeline Chart

3.4 Detailed Modules Description

1. Data Gathering

- This Module used to find dataset with enough amount of good quality images and also setting resolution.

2. Data Pre-processing

- This Module used to dataset cleaning and transformation and also adjust quality pixel base and density map generation.

3. Network Finding

- This Module used to find best neural network for given dataset and finding best hype parameters for training model.

4. Model Generation

- This Module used to generate model with higher accuracy based upon training that model and export model for web app.

5. Web App Development

- This Module cover integration of model in web app and real time web cam based monitoring functionality.

3.5 Project SRS

3.5.1 Use Case Diagram

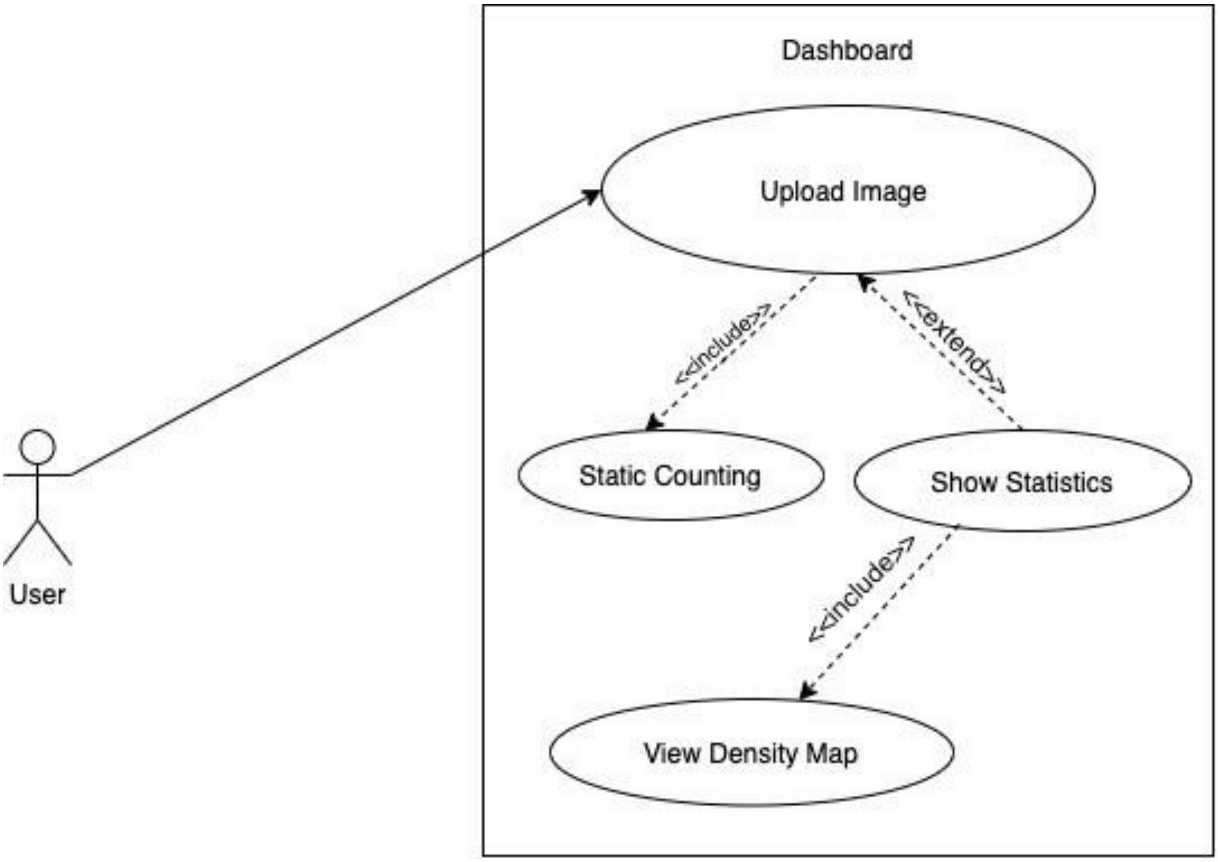

Figure 2 Use Case Diagram

3.5.2 Class Diagram

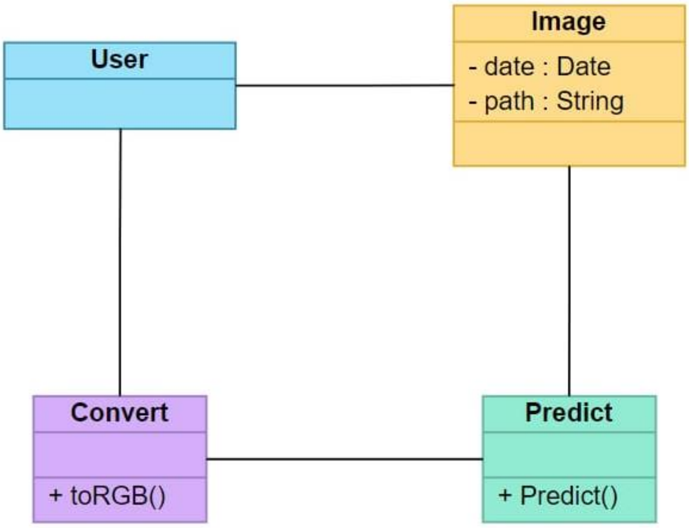

Figure 3 Class Diagram

3.5.3 Sequence Diagram

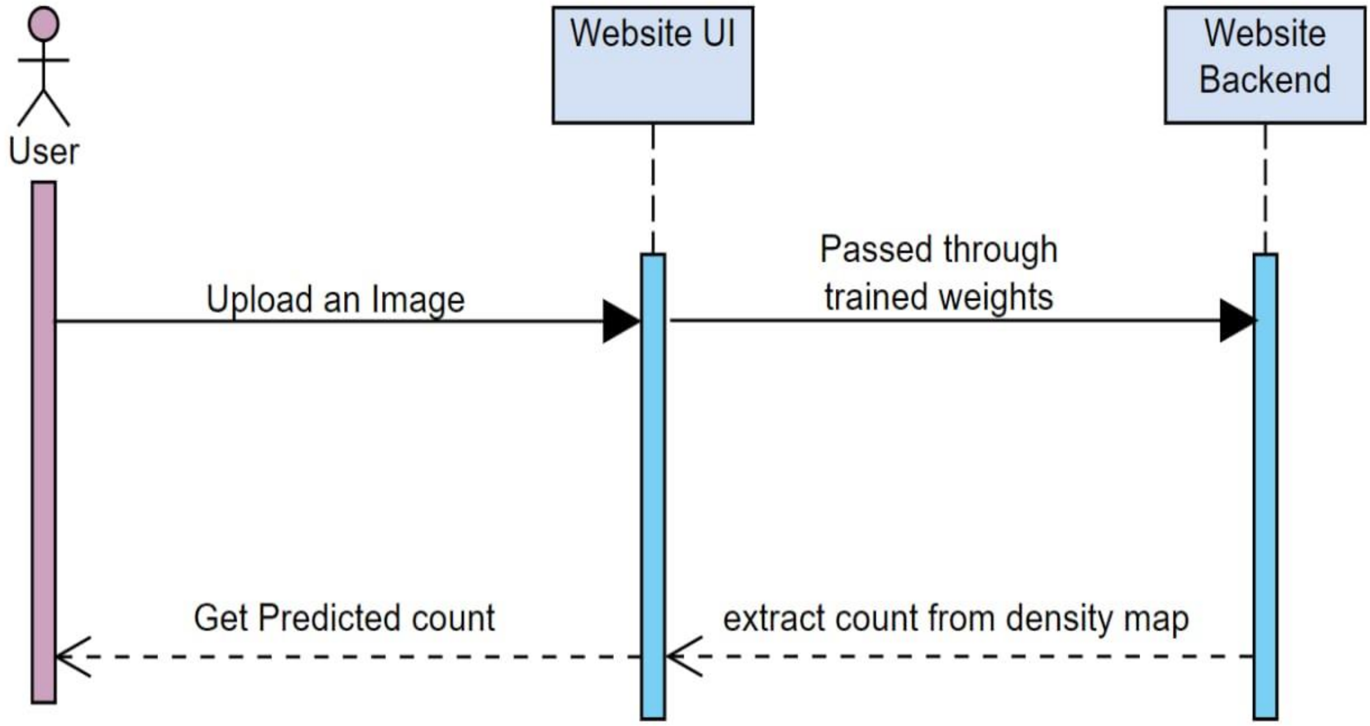

Figure 4 Sequence Diagram

3.5.4 State Diagram

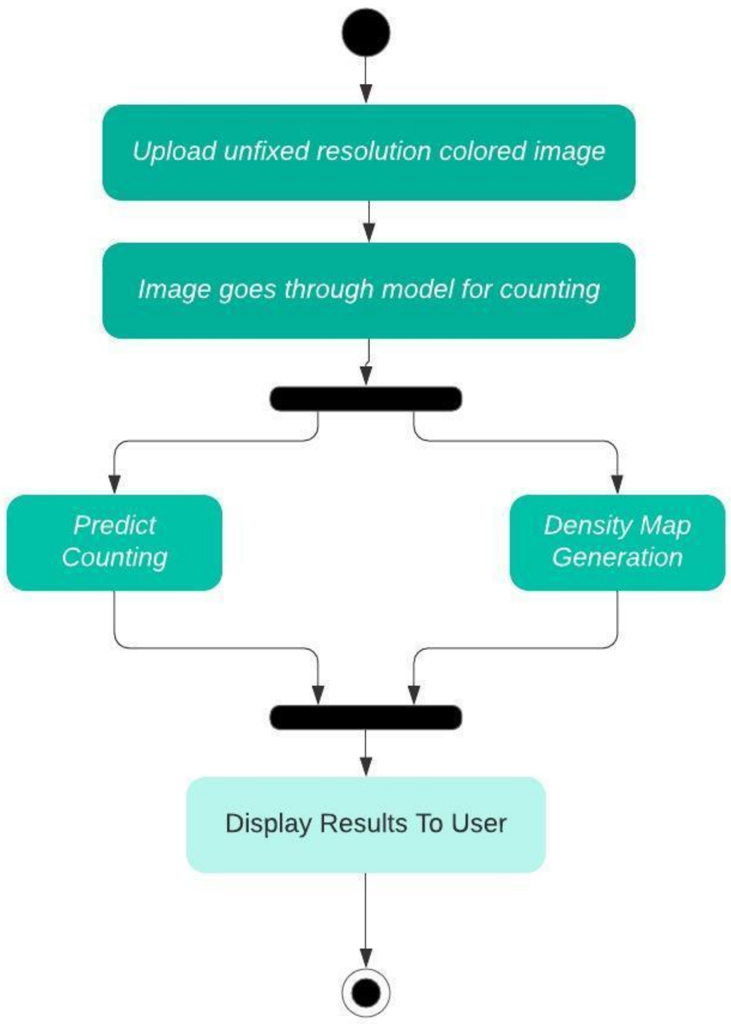

Figure 5 State Diagram

3.5.5 Activity Diagram

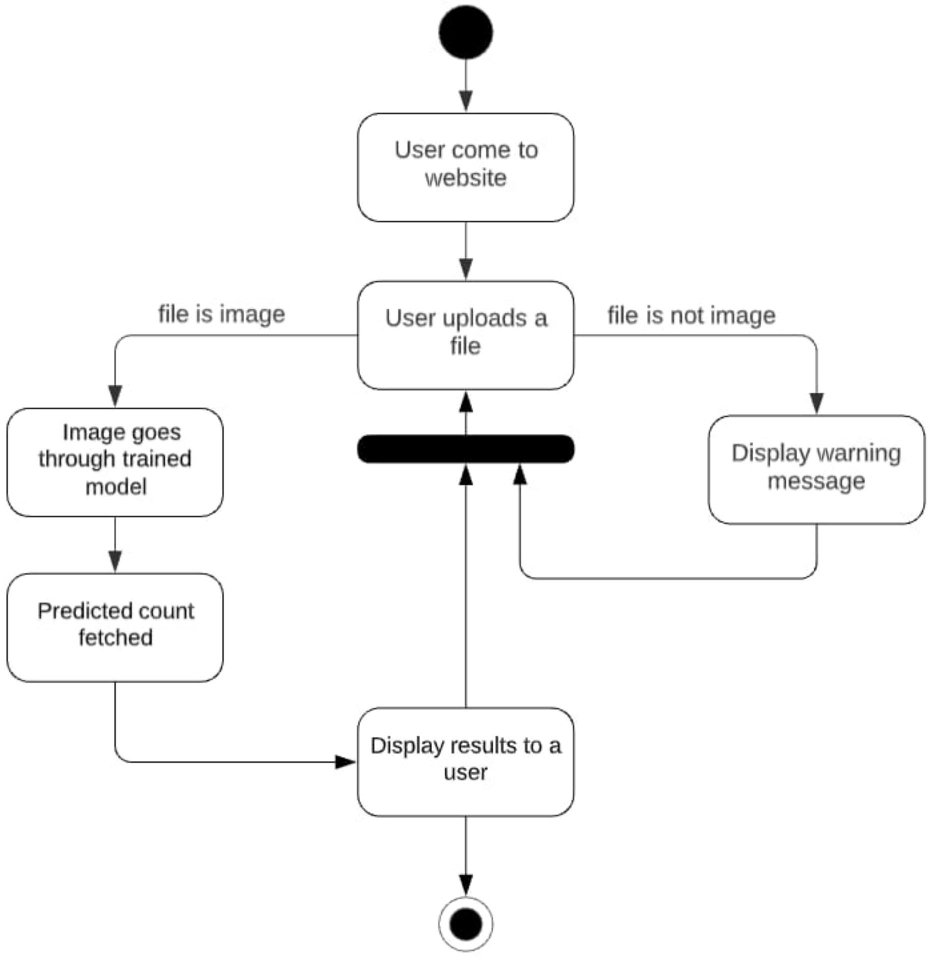

Figure 6 Activity Diagram

3.6 Data Dictionary

Sr No.	Name	Data Type	Description	Constraints
1	Pk	Int	Primary Key	PRIMARY KEY, NOT NULL, AUTO INCREMENT
2	Photo	Image Field	Uploaded Image	
3	Date	Date	Date And Time Of Uploaded Image	

Table 1 Data Dictionary Table

Chapter 4. Implementation and Testing

4.1 User Interface and Snapshot

➢ Web App UI

- **Home Page UI**
  - Home Page UI consists of links to different section of web app and basic introduction of crowd predictor

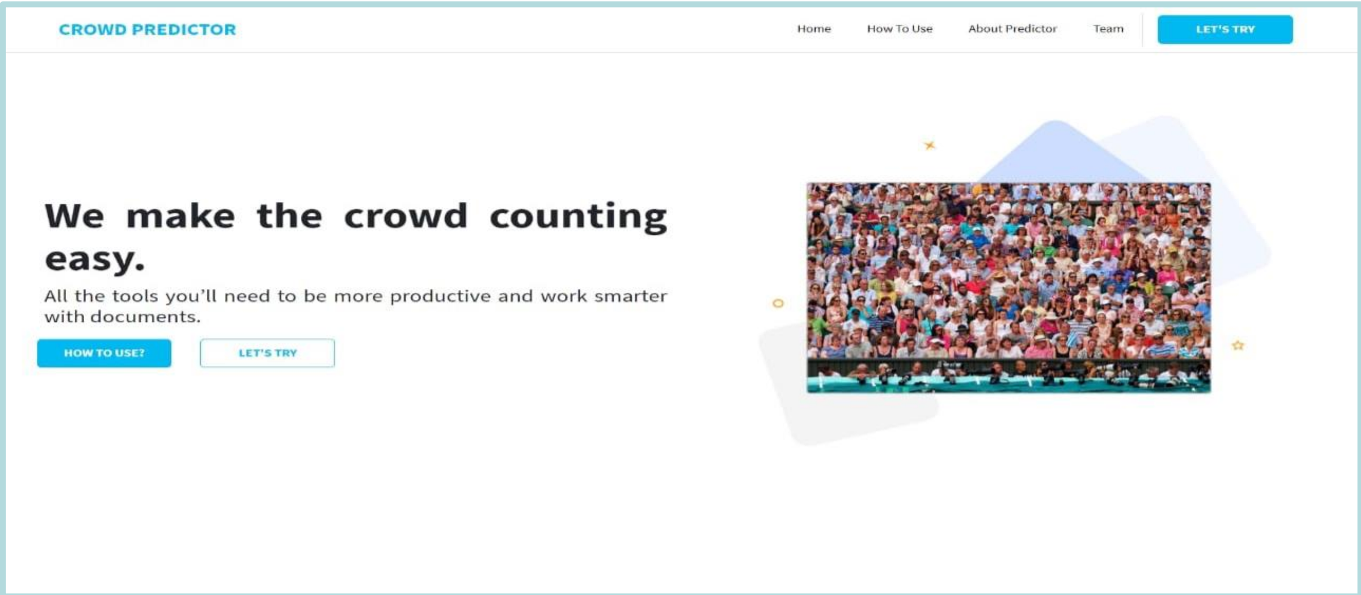

Figure 7 Home Page UI

  - This part of Home Page UI explains that How a user can use crowd predictor for static counting.

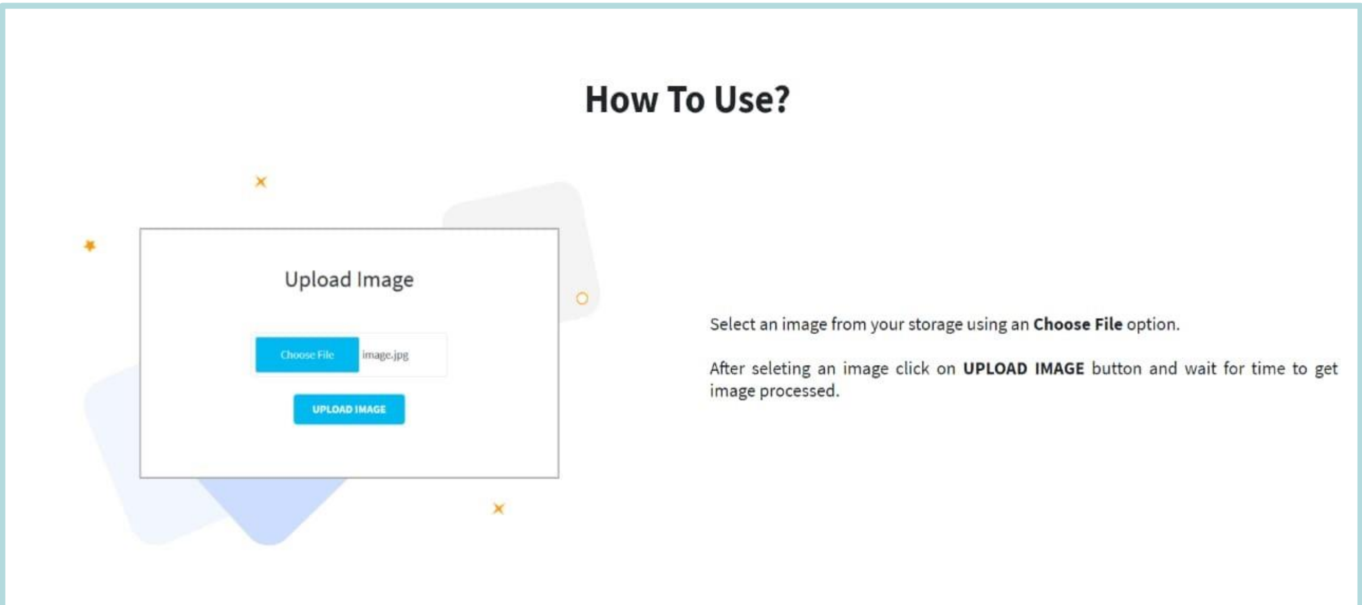

Figure 8 Illustration Of How To Use Predictor

- This part of Home Page UI explains about some technical terms like density map and predicted count.

Figure 9 Illustration Of Density Map & Predicted Count

4.2 Testing using Use Cases

➢ **Test-Case 1: Uploading Image:**

- This functionality use to upload image for counting and its consists of file type check so user can't upload wrong file like pdf, docx etc.

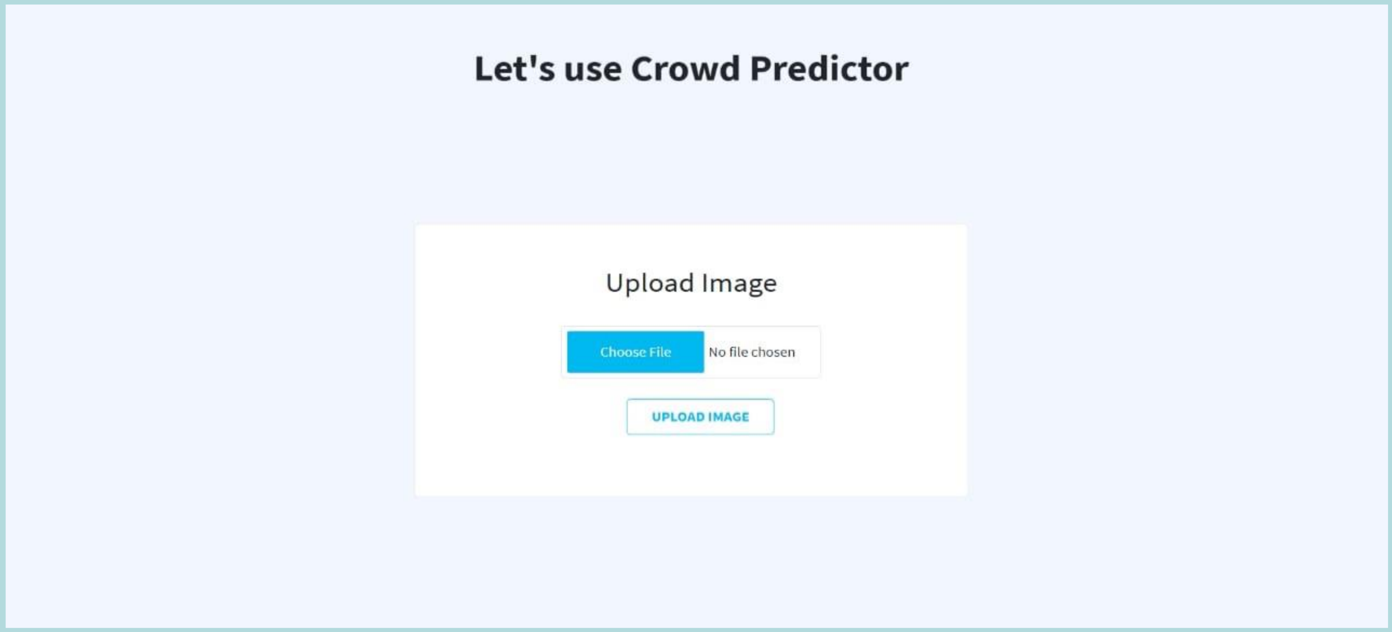

Figure 10 Upload Image UI

- This functionality warn user if he/she does not upload any image for counting.

Figure 11 Upload Image Validation

- This is UI to show crowd counting process in backend and when user upload image it pass through filter so it can work with our trained model

Figure 12 Backend Processing UI

- This is image of result which is generated after backend processing and it also contains density map.
- Moreover, user can download density map.

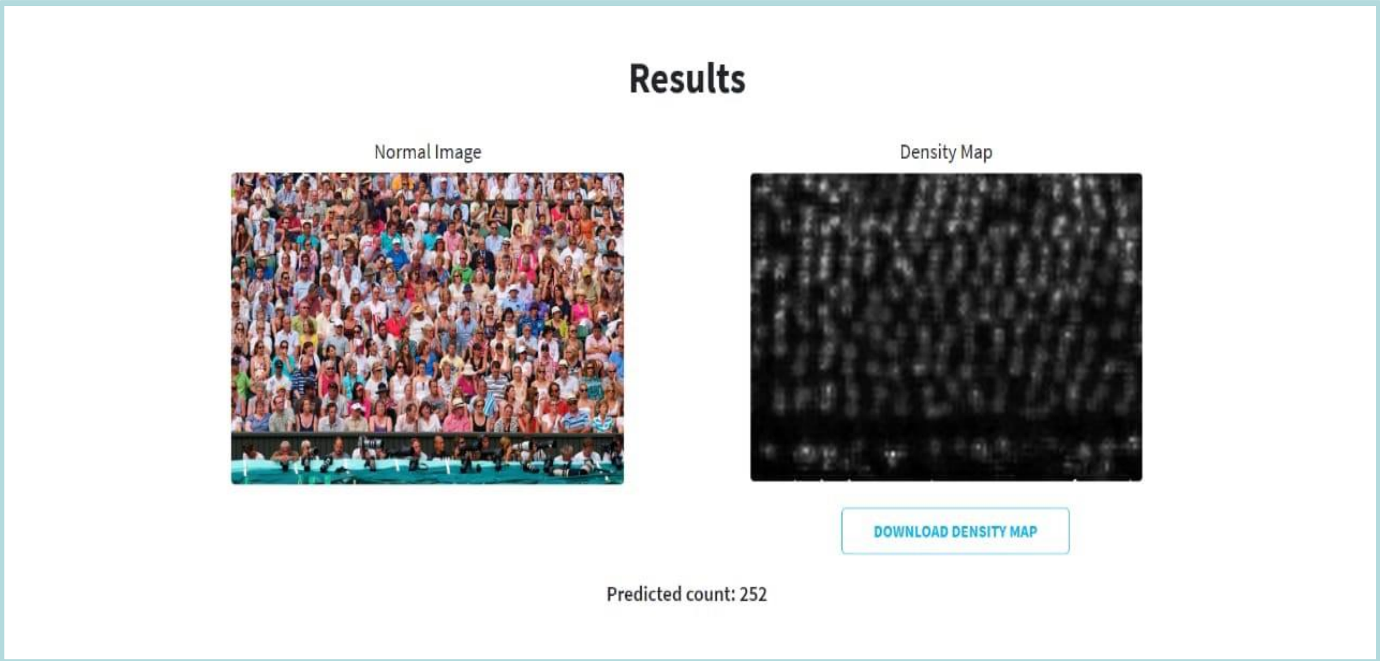

Figure 13 Result With Density Map

- **Test Case 2:Test Data Analysis**

- This analysis part shows how model gives high accuracy by training it with different weights and also provide some additional information like total number of images and total persons all over dataset.

```
######################## Analysis of part A test images ########################

total persons over all images : 78741
total number of persons predicted over all images : 75805
Accuracy in percentage : 96.2713198968771
Total number of test images : 182
Average number of persons in each image is : 432.64285714285717
Mean absolute error : 66.18681318681318
```

Figure 14 Model Accuracy Measures

Chapter 5. Conclusion & Future Work

5.1 Conclusion

- The proposed system performs admirably in situations where manual counting is simply not possible. Deep learning also enables the system to perform in versatile environments. The Experimental results reveal that the proposed methodology achieves promising crowd count predictions almost as good as ground truth.

5.2 Future Work

- The project has a very vast scope in the future and can be implemented on satellite footage in future. The project is flexible in terms of expansion and can be expanded to trace or study the movement of the crowds which could be helpful in managing riots, rallies etc. The proposed system architecture can also be used in monitoring real time traffic by creating density maps for cars.

Chapter 6. References

6.1 References

[1]. Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.

[2]. Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: a deep convolutional network for dense crowd counting. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 640–644. ACM, 2016.

[3]. Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547– 2554, 2013.

[4]. Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[5]. Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137– 154, 2004.

[6]. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pat- tern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7]. Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[8]. Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 545–551. IEEE, 2009.

[9]. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150– 1157 vol.2, 1999.

➢ **Useful Links**

[10]. https://www.coursera.org/learn/neural-networks-deep-learning?specialization=deeplearning#instructors

[11]. https://www.geeksforgeeks.org/vgg-16-cnn-model/

[12]. https://www.kaggle.com/blurredmachine/vggnet-16-architecture-a-complete-guide

[13]. https://neurohive.io/en/popular-networks/vgg16/

[14]. https://matplotlib.org/stable/tutorials/index.html

[15]. https://www.tensorflow.org/tutorials

[16]. https://keras.io/guides/

[17]. https://youtu.be/zfiSAzpy9NM

[18]. https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

Guide Remark/Suggestion:

Guide Signature:______________________