DRAMS

Disaster Relief and
Management System

Team Numero Uno

Executive Summary

Introduction to DRAMS

- Impact of Natural Disasters
- Idea and High level Workflow

PS Requirements

Use Case - Flash Floods

- Detect (Pre Flood)
- Alert (Early Warning)
- Relief (Post Flood)

Use Case – COVID Prevention

- Methods Of Prevention
- Detect (Social Distancing Violations)
- Warn (Warning to Violators)
- Disperse (Police Deployment)

Use Case – City Fire

- Detect – Alert –Relief Model

Impact

Application Architecture & Deployment

- Application Control Flow
- Sensor Architecture (Edge Computing)
- Controller Architecture (Central Server)
- Kubernetes Cluster Architecture
- Database Architecture (MongoDB & Django)

Execution

Technical Details of Algorithms

- VGG-16 Fundamentals and Architecture
- Live Example Of Handwriting Classifier
- Architecture of Flood and Fire Networks
- Object Detection For COVID Prevention
- Efficient Routing using GRASP & VND
- DBSCAN based Clustering Demo
- Performance Comparison

Technology

UI, User Experience And GitHub Links

Introduction To DRAMS

Impact Of Natural Disasters

$232 Billion

Average annual economic loss world wide

20,000

Average casualties caused by Natural Disasters every year

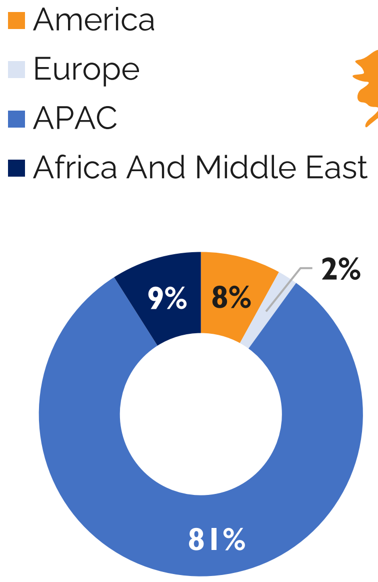
Major life and economic loss is caused

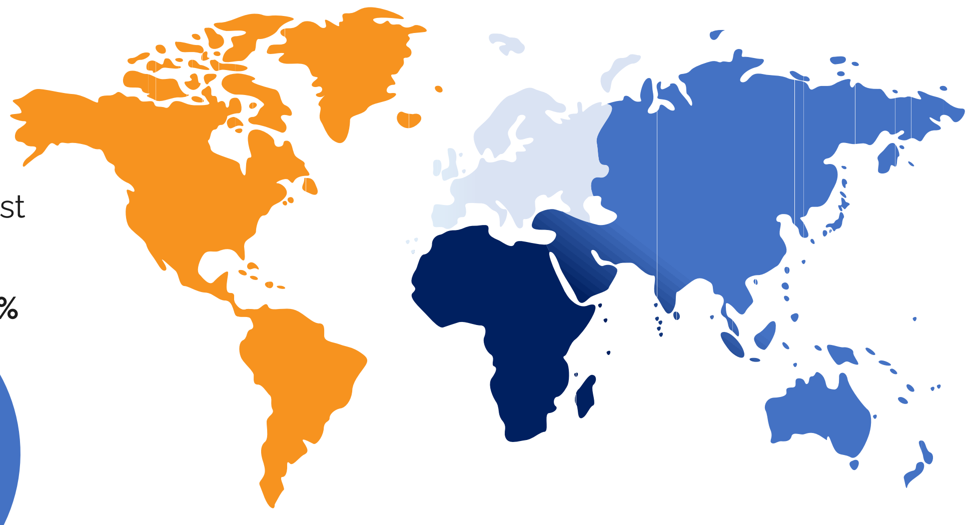
26 lakh Indians were affected by floods last year

Lack of real time data integration

Inefficient Response

Disaster Response teams lag behind in fast paced calamities like Flash floods & City Fires

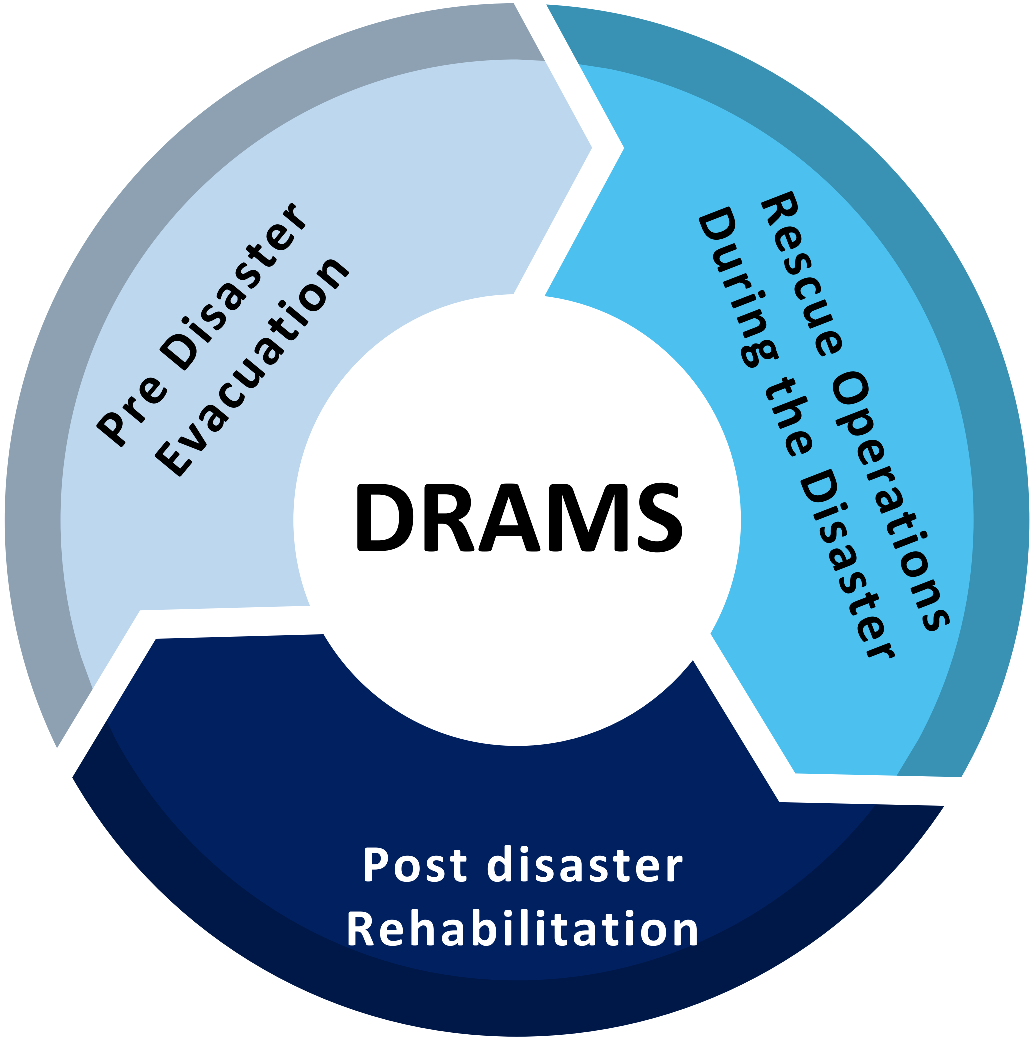




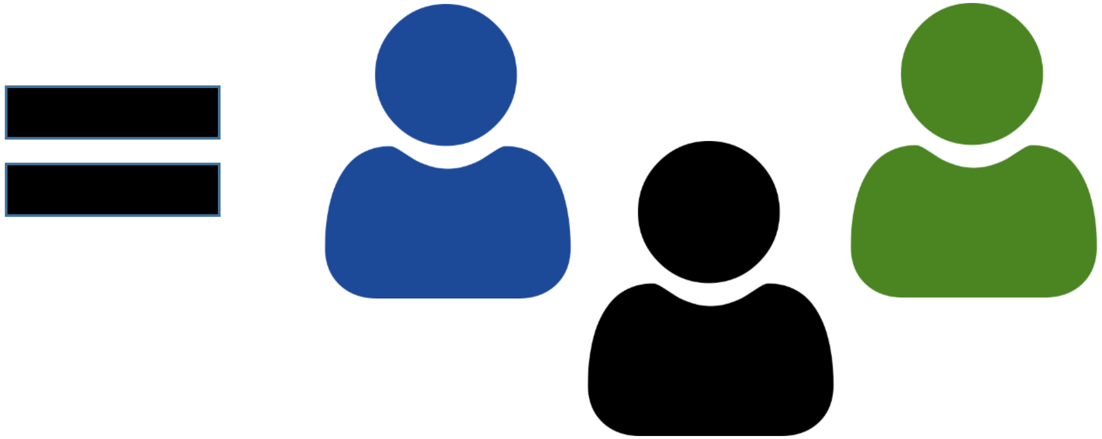
Worldwide Casualties due to Natural Disasters (2018)

APAC accounts for more than 81% of the Natural Disaster casualties worldwide

The Idea Behind DRAMS

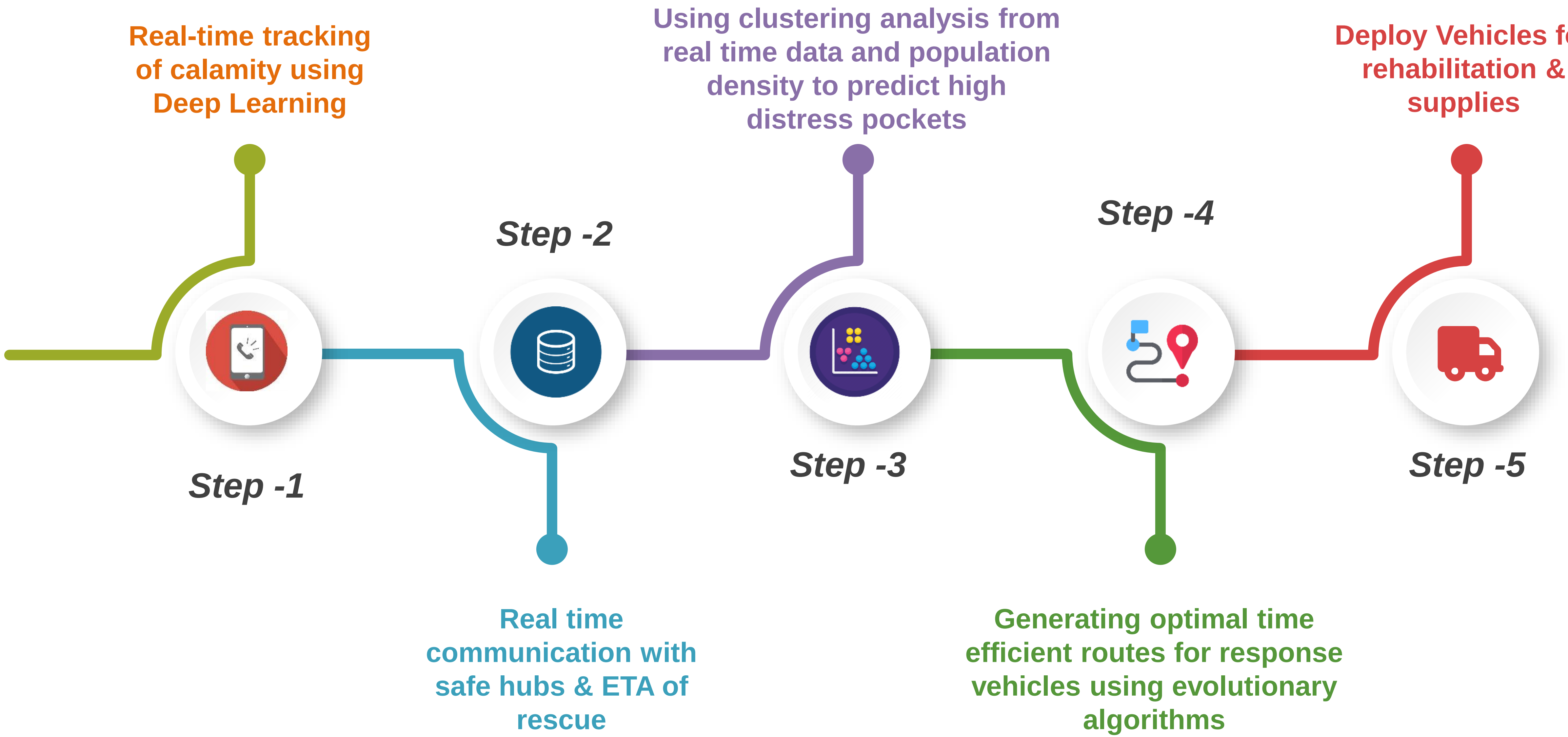


The basic idea is to develop a decision support system which can automate & optimize various operations during all stages of a natural calamity



Lives Saved

High Level Workflow



Use Case
Flash Floods

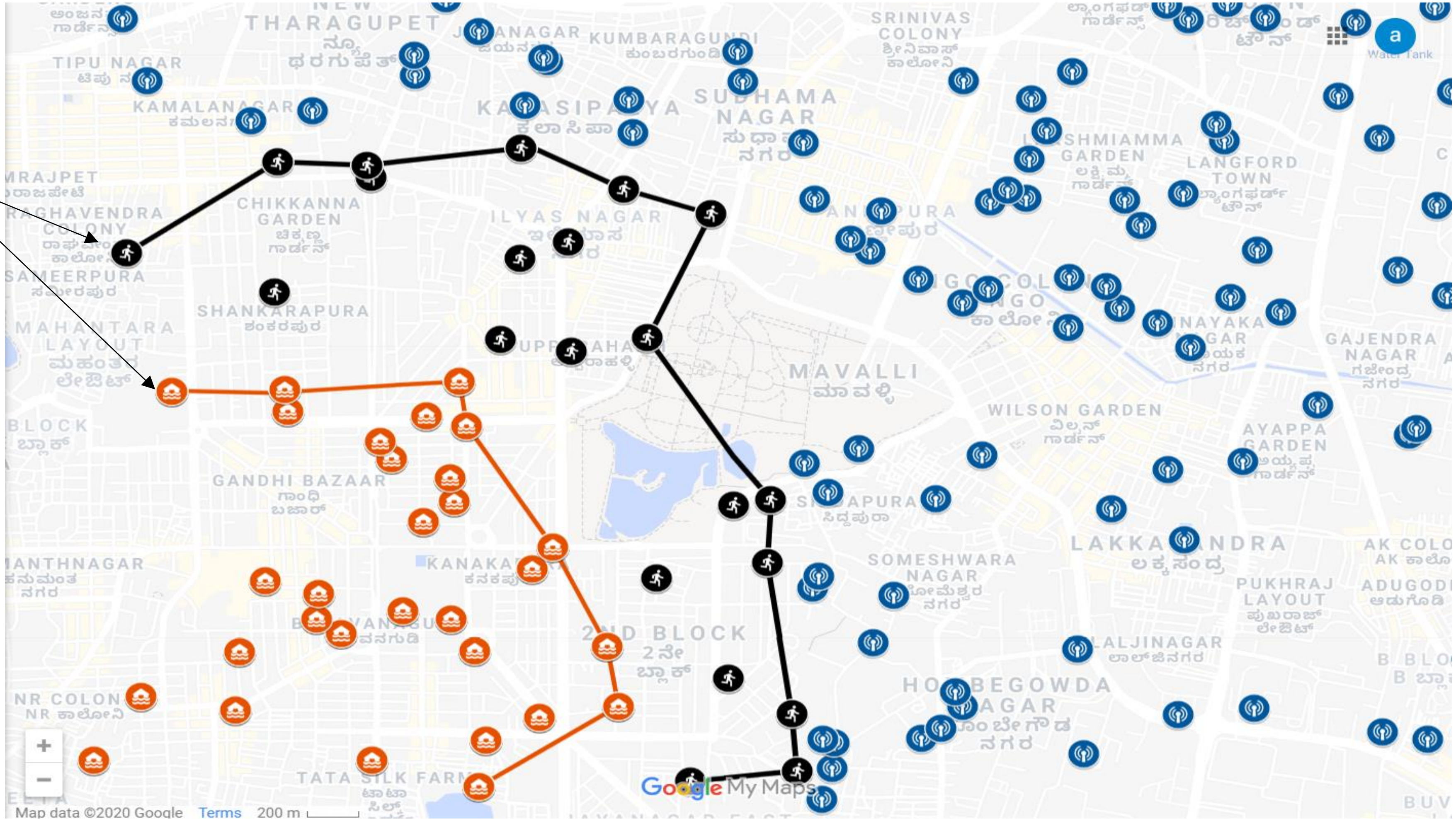
Pre Disaster Phase (Detect)

Disaster Detection & Tracking in Smart Cities

- Convolutional Neural Network Architecture is used to detect disasters from CCTV feed
- Live data showcasing areas witnessing the calamity is stored in database
- Evacuation Area is predicted using estimated time of impact, and population clusters
- Cameras installed through out the city are used as nodes for active data points

Example Flood Demonstration

CCTV Nodes



Boundary of Evacuation Zone

Affected Area

Live CCTV nodes to dynamically update the calamity affected area and rescue zone as the flood progresses

Disaster Phase (Alert)

Rescue Operations

Real time communication showcasing – early warning, nearest safe hub and ETA of rescue vehicles

We use a combination of static databases like Aadhar and PAN as well as dynamic location information from Aarogya Setu to send out accurate alerts to the end users

Post Disaster Phase (Rescue And Relief)

Re-establishing Connectivity and Planning Relief Operations

The major challenge after any natural calamity is to re-establish connectivity as soon as possible.

Our decision support system clusters the nodes received from live data points, population density & route condition. It then calculates the most efficient path to reconstruct roads and achieve full connectivity

Routing path of disaster relief vehicles would be based on output from **DBSCAN**, **GRASP** and **VND** algorithm leveraging the estimated time of impact, clusters of high population density and geo-location data

Prototype Demonstration 1

Use Case
COVID Prevention

COVID 19 Pandemic

A Glimpse At The Numbers

100,000 Cr
Average loss per month since this pandemic started

36,511
Deaths in India caused by this virus so far

1.7 M
Cases reported in the Indian subcontinent so far

4
Average infections per COVID patient with preventive measure in place

Preventive Measures

Regular Handwash

Social Distancing

Major life and economic loss is reported

Highly infectious disease with no drug or vaccine

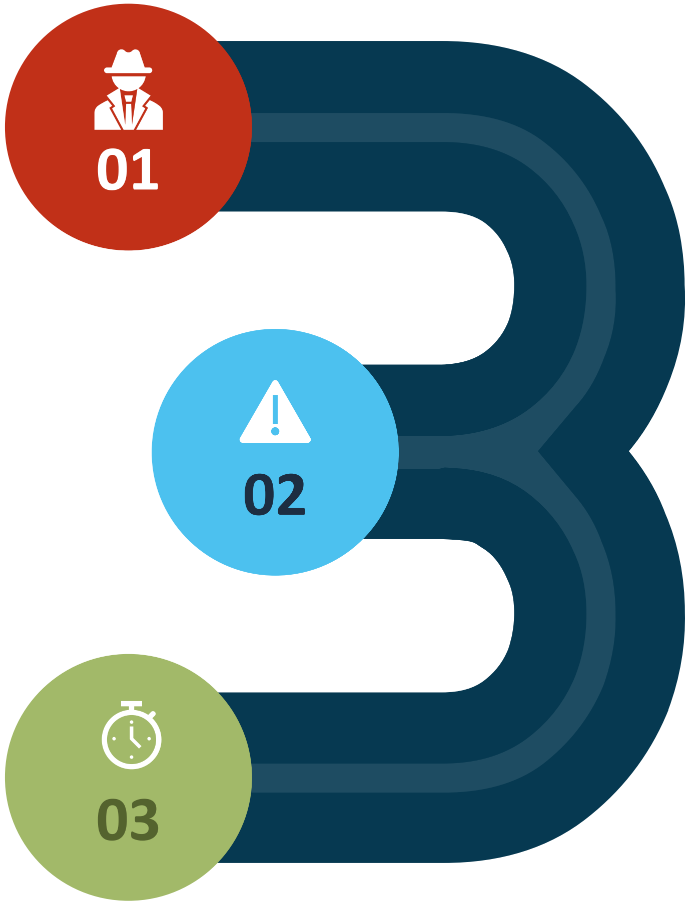
Lack of Proper Crowd Monitoring

Inefficient Resource Planning

Police Response teams lag behind in dispersing fast paced crowd activities and maintaining social distancing

It is possible to minimize the loss of life by proper social distancing and crowd regulation

Detect – Warn – Disperse Model



01

Detect

Our AI agent detects the crowd concentration in CCTV footage. In case the social distancing norms are violated, an alert is raised to the concerned authorities.

02

Warn

If the authorities feel, they have the option to put out a warning message to all the people in a specific GPS location. This is achieved by dynamic location tracking using Aarogya Setu’s API

03

Disperse

Efficient route planning according to the available resources in terms of cars, jurisdiction and fuel costs is automatically performed by the AI agent and the instructions are sent out to the responsible authorities

Use Case City Fires

Prototype Demonstration 2

Algorithms Explained

Fundamentals Of VGG-16

2015 Paper By Oxford Researchers

Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman**+
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

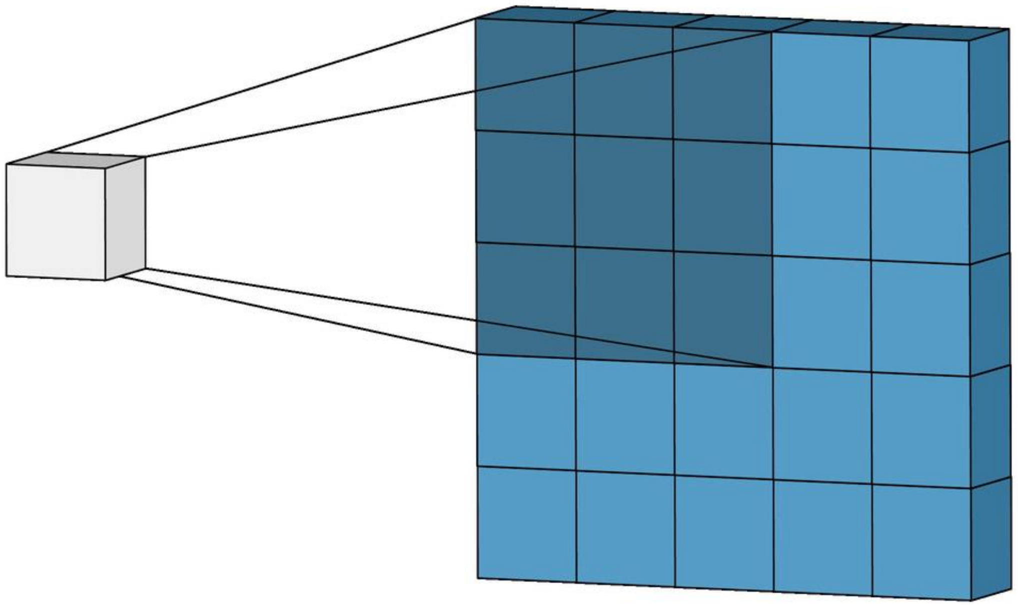
ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

1 INTRODUCTION

Convolutional networks (ConvNets) have recently enjoyed a great success in large-scale im-

2 Layer Network Example



Example of a Shallow Network

Workflow

Handwritten image is Fed to our Model

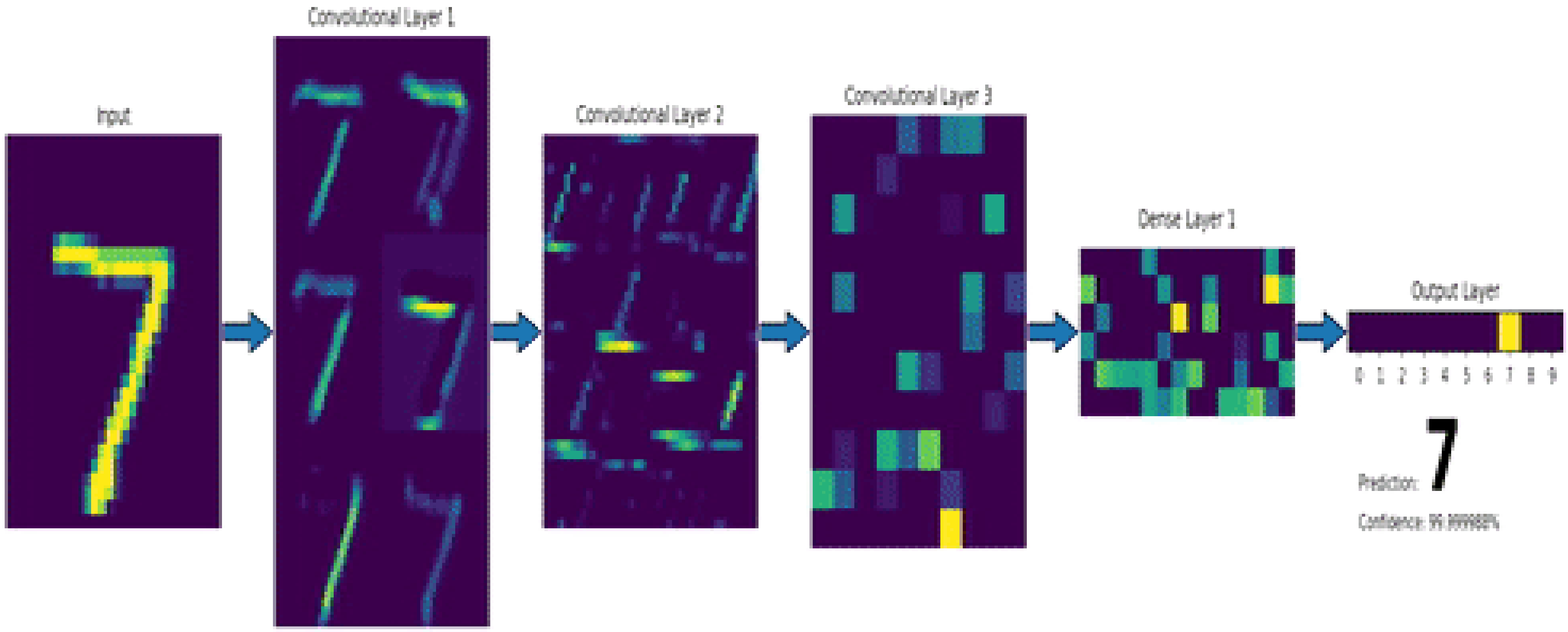
↓

The data is compressed and only the most critical information progresses further

↓

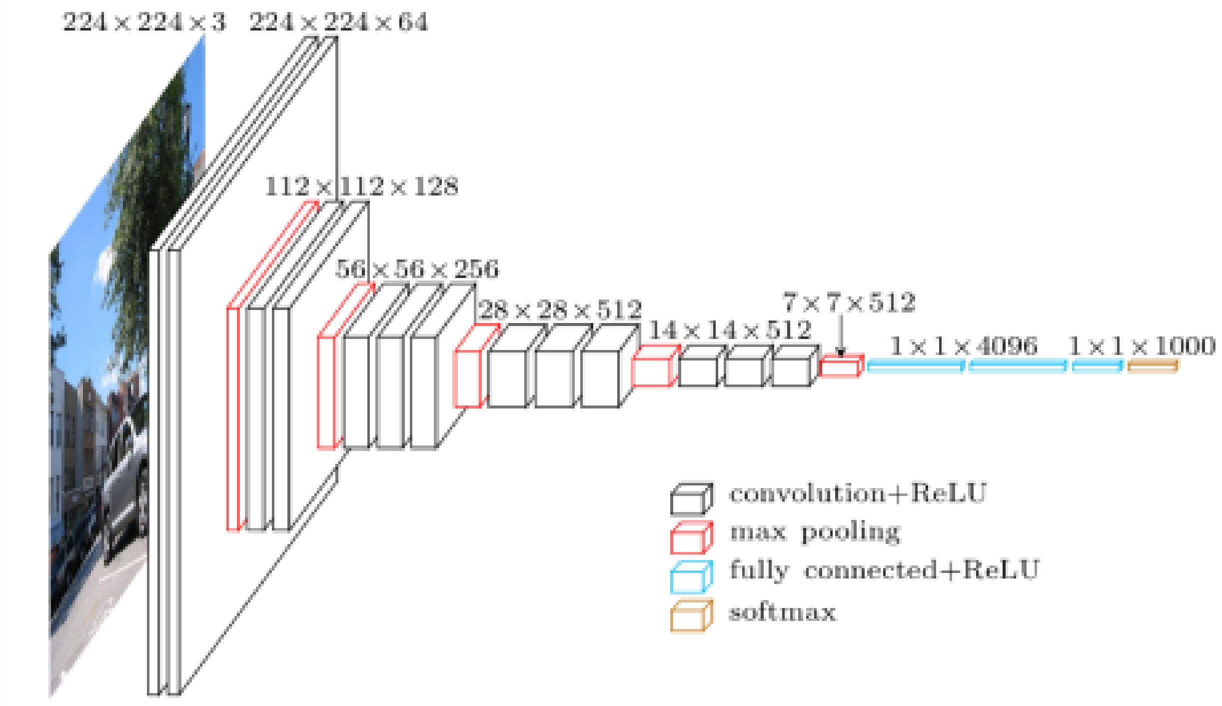
It is finally reduced to a single array of 9 numbers

An Example Demonstrating the Idea of Multilevel CNN Networks



Handwriting Classifier

The Deep Network Architecture We Use



Architecture of Our CNN System

$$x_{i,j}^{l} = \sum_{m}\sum_{n} w_{m,n}^{l} o_{i+m,j+n}^{l-1} + b_{i,j}^{l}$$

$$o_{i,j}^{l} = f(x_{i,j}^{l})$$

$$\delta_{i,j}^{l} = \frac{\partial E}{\partial x_{i,j}^{l}}$$

$$\frac{\partial E}{\partial x_{i',j'}^{l}} = \sum_{m=0}^{k_1-1}\sum_{n=0}^{k_2-1} \delta_{i'-m,j'-n}^{l+1} w_{m,n}^{l+1} f'\left(x_{i',j'}^{l}\right)$$

$$\frac{\partial E}{\partial w_{m',n'}^{l}} = \sum_{i=0}^{H-k_1}\sum_{j=0}^{W-k_2} \delta_{i,j}^{l} o_{i+m',j+n'}^{l-1}$$

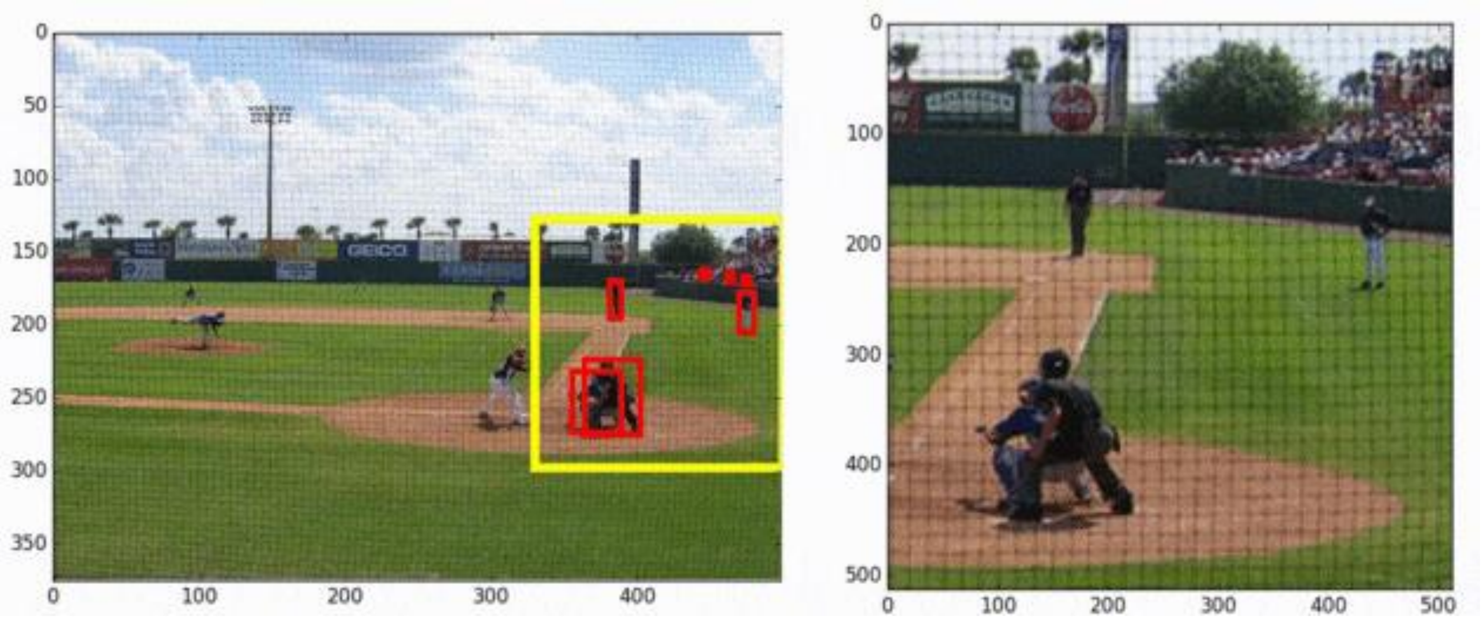
Gradient Optimizer

Object Detection For COVID Explained

In this algorithm, we choose a grid cell of a specific size. Let us choose the grid cell of size 2x2.

We pass the above grid cell through the image and convolute the part of the image in the grid cell and predict the output.

Then we slide the grid cell through stride-2 and then convolute the next part of the image.



In this way, we go cover the whole image. We repeat the same procedure with different grid cells size.

Protype Demonstration 1

Our Routing Algorithm Explained

Algorithm For Handling Rescue And Relief

Multi-objective particle swarm optimisation based integrated production inventory routing planning for efficient perishable food logistics operations

Felix T.S. Chan [a], Z.X. Wang [b,c*], A. Goswami [a,d], A. Singhania [a,d] and M.K. Tiwari [d]

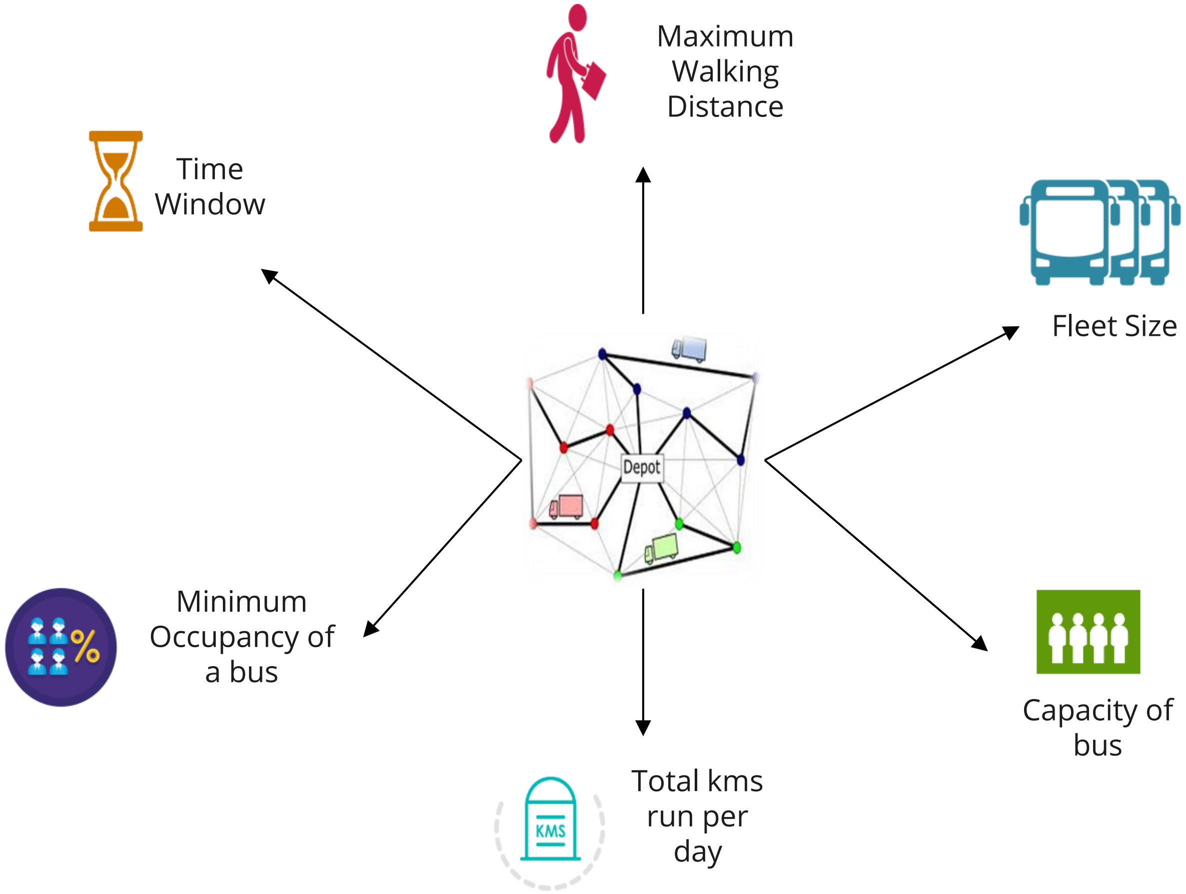
[a] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong;* [b] *School of Business Administration, Dongbei University of Finance and Economics, Dalian, People's Republic of China;* [c] *Institute of Systems Engineering, Dalian University of Technology, Dalian, People's Republic of China;* [d] *Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India*

(*Received 27 June 2019; accepted 24 November 2019*)

Sustainable and efficient food supply chain has become an essential component of one's life. The model proposed in this paper is deeply linked to people's quality of life as a result of which there is a large incentive to fulfil customer demands through it. This proposed model can enhance food quality by making the best possible food quality accessible to customers, construct a sustainable logistics system considering its environmental impact and ensure the customer demand to be fulfilled as fast as possible. In this paper, an extended model is examined that builds a unified planning problem for efficient food logistics operations where four important objectives are viewed: minimising the total expense of the system, maximising the average food quality along with the minimisation of the amount of CO_2 emissions in transportation along with production and total weighted delivery lead time minimisation. A four objective mixed integer linear programming model for intelligent food logistics system is developed in the paper. The optimisation of the formulated mathematical model is proposed using a modified multi-objective particle swarm optimisation algorithm with multiple social structures: MO-GLNPSO (Multi-Objective Global Local Near-Neighbour Particle Swarm Optimisation). Computational results of a case study on a given dataset as well as on multiple small, medium and large-scale datasets followed by sensitivity analysis show the potency and effectiveness of the introduced method. Lastly, there has been a scope for future study displayed which would lead to the further progress of these types of models.

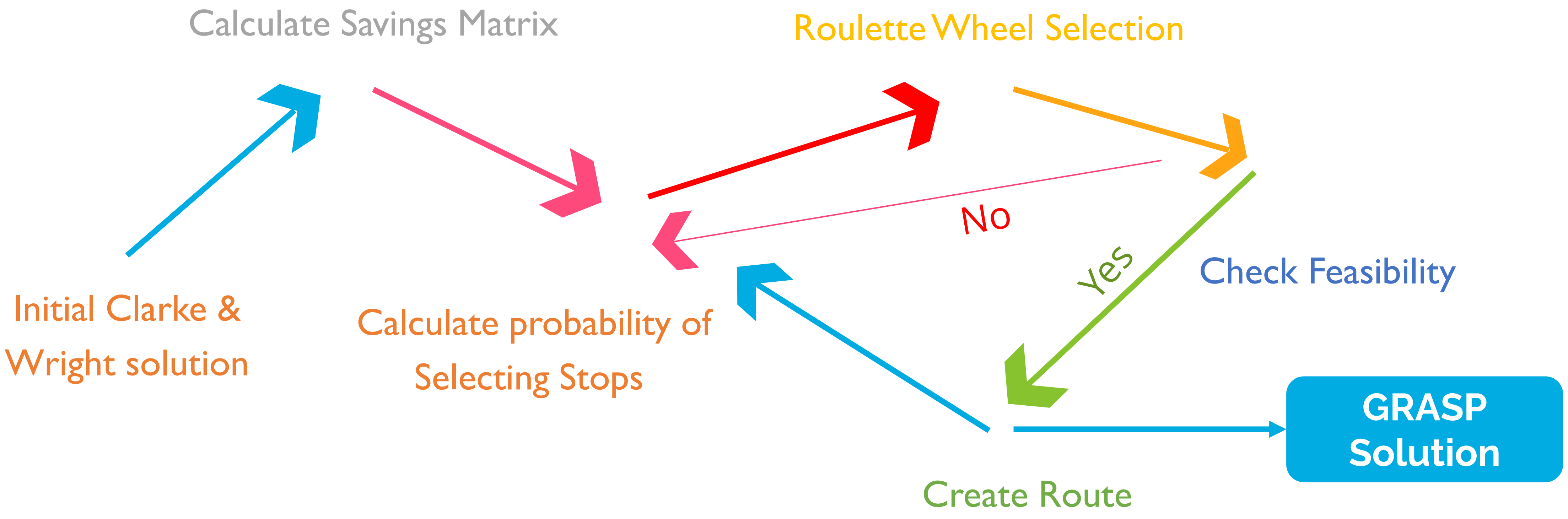
Keywords: multi-objective optimisation; particle swarm optimisation; food quality; perishable product; intelligent food logistics operations; integrated outlining

PROBLEM Constraints

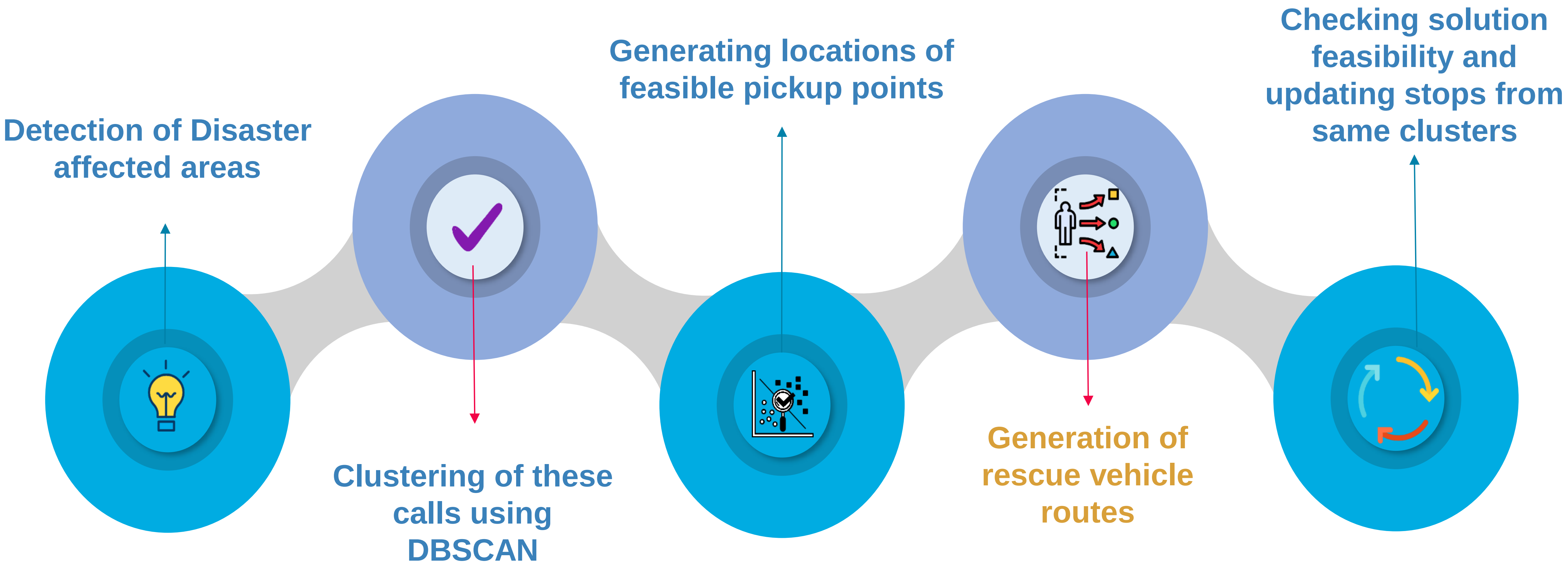


Our Routing Algorithm Explained

Greedy Randomized Adaptive Search Procedure

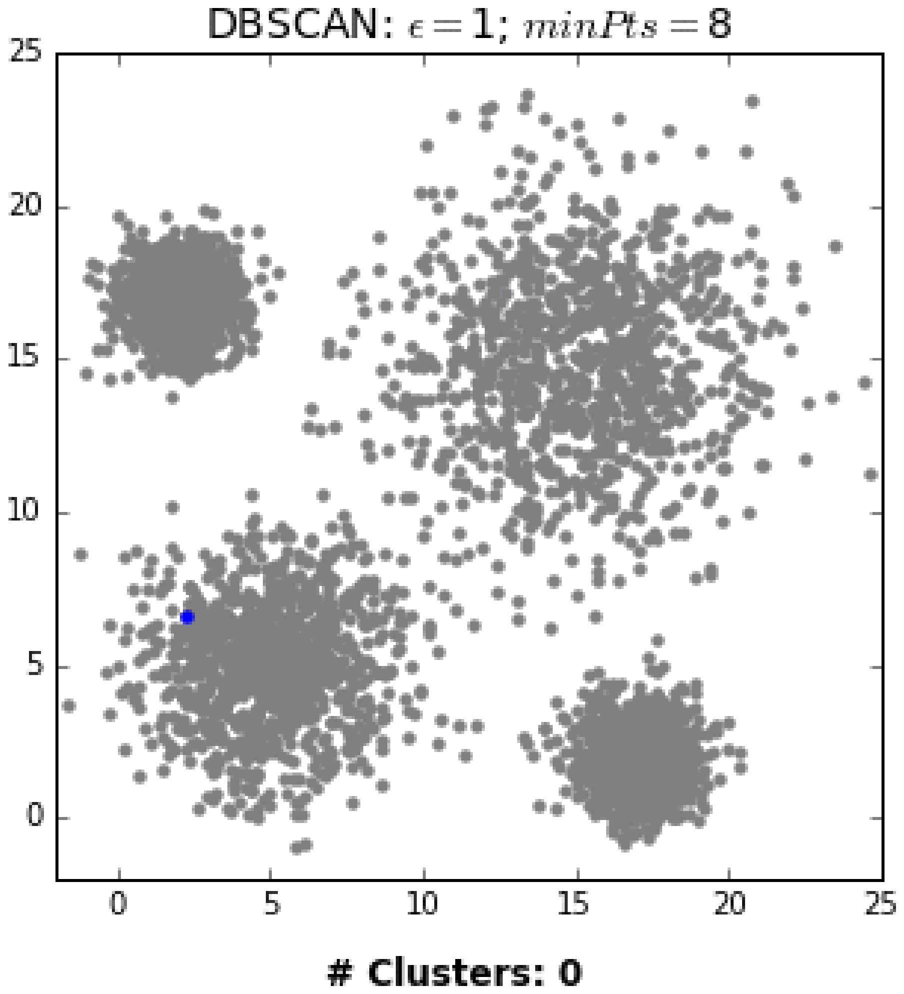


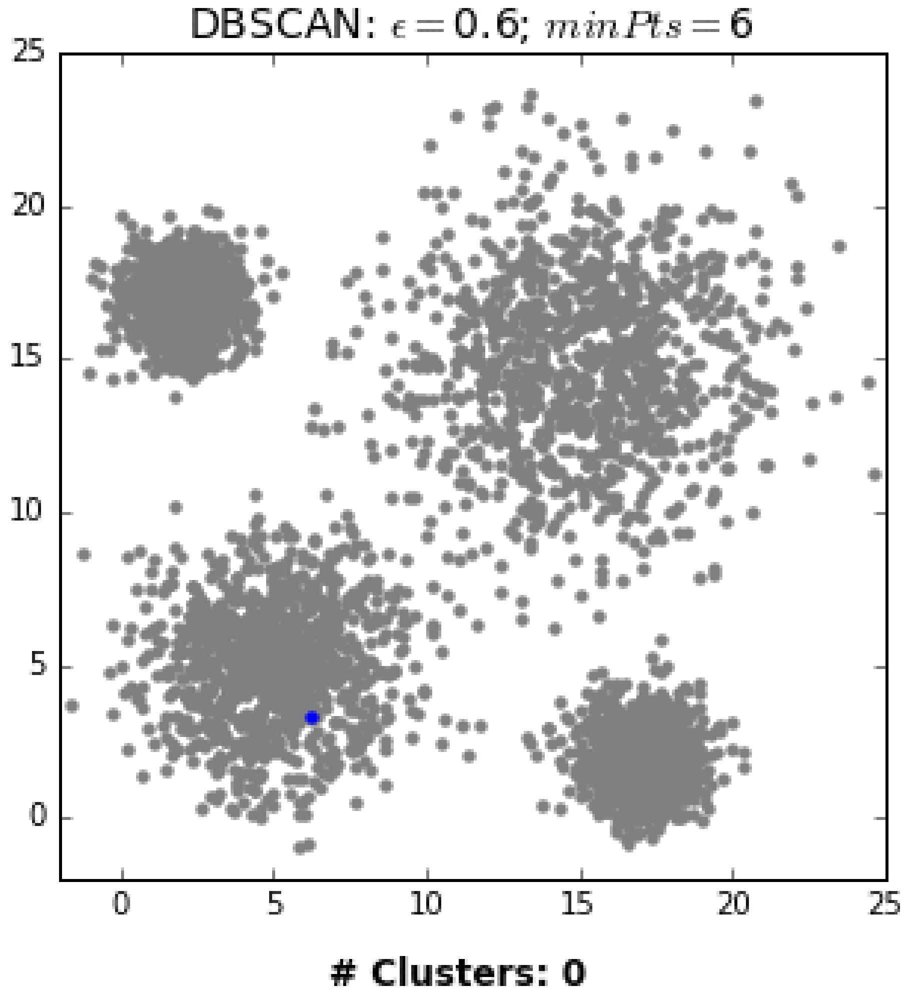
Our Routing Algorithm Explained



Clustering Module

Our Clustering Algorithm





DB Scan In Action

Our Routing Algorithm Explained

Initial GRASP Solution

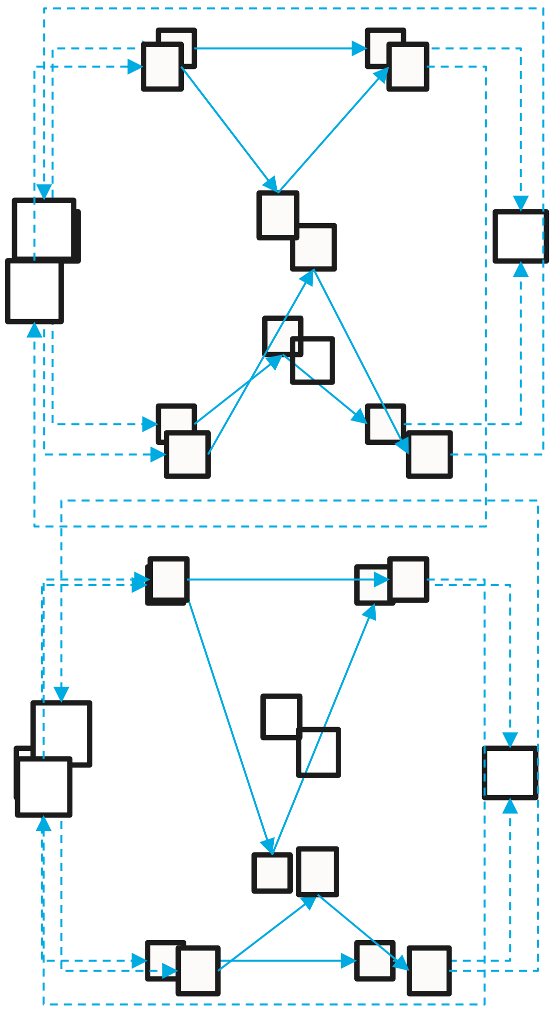
Remove-Insert Within Route

Remove-Insert Between Routes

Replace

Remove

Variable Neighborhood Descent

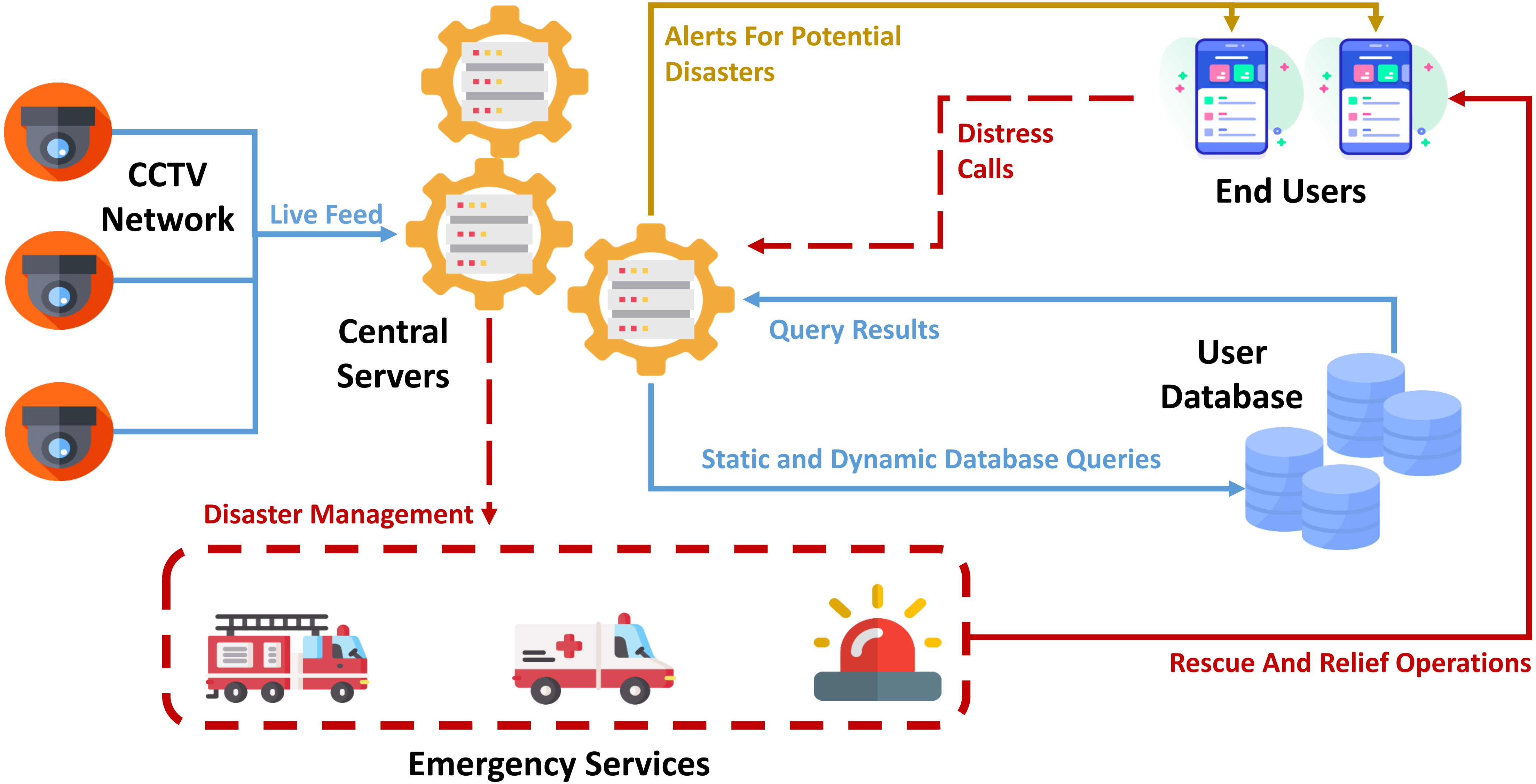


Performance Comparison

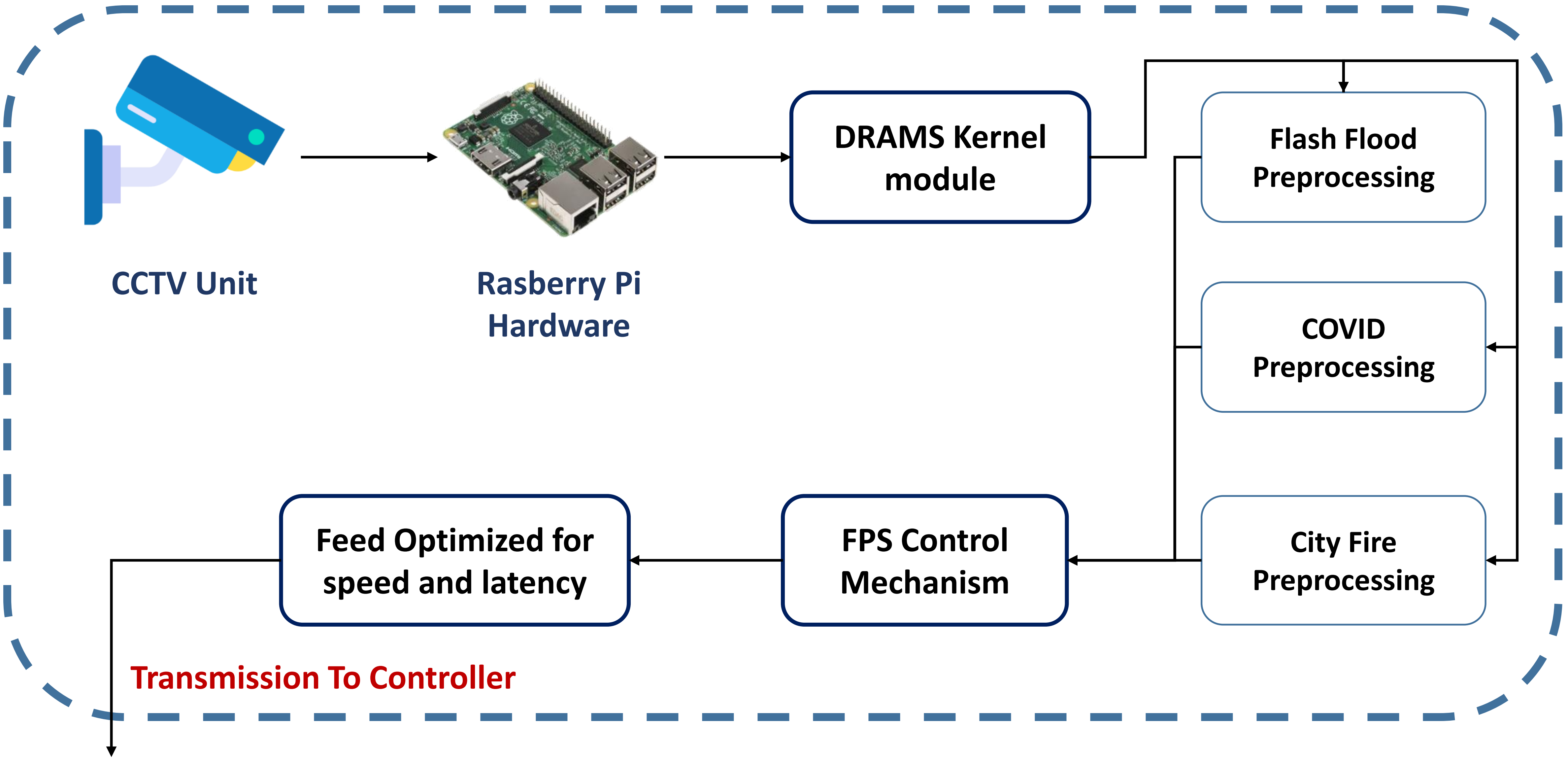
Problem Classes	Reinforcement Learning	Evolutionary Algorithm + Local Search Methods	Exact Methods	Heuristics	Our Approach
Handling Large Dataset	✓	✓	✗	✓	✓
Low Processing Power	✗	✓	✗	✓	✓
Handling Local Minima	✓	✓	✓	✗	✓
Parameter Free	✗	✗	✓	✗	✓
Constraint Handling	✓	✗	✓	✓	✓
Near Optimal Solution	✓	✗	✓	✗	✓

Application Architecture and Deployment

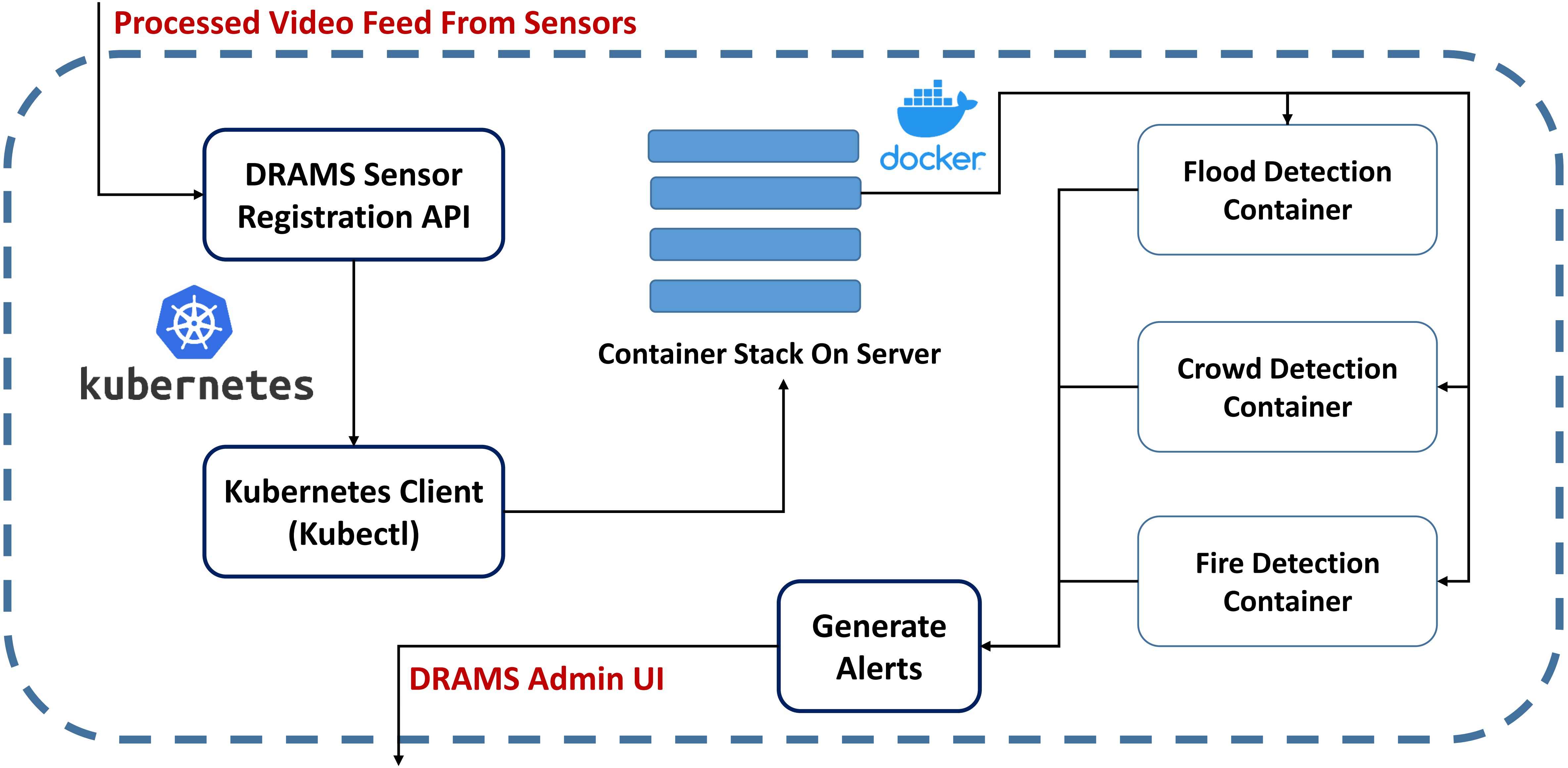
Application Architecture



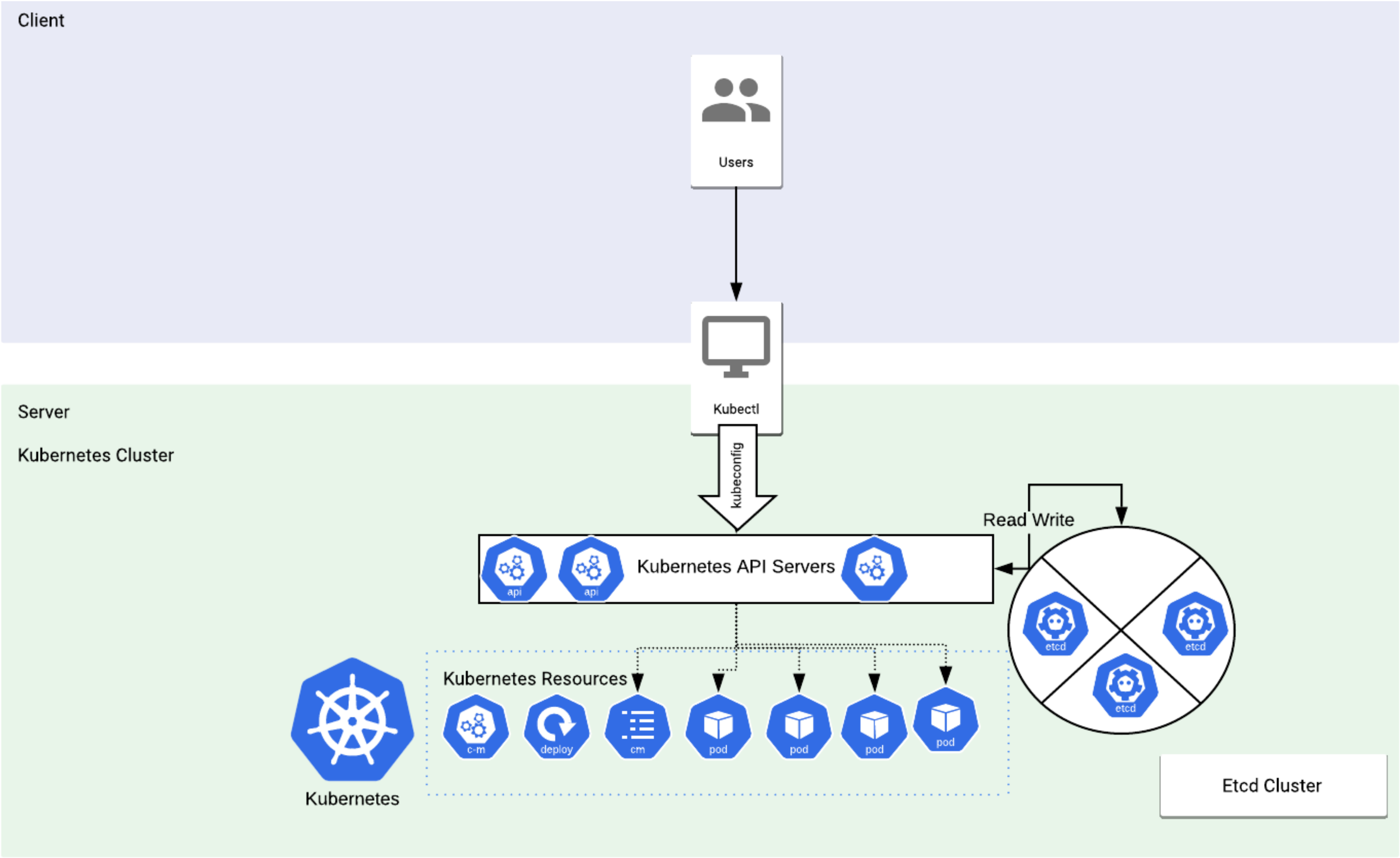
Sensor Architecture



Controller Architecture



Kubernetes Cluster Architecture



Django Database Architecture

Class Entity Relationship

Class City

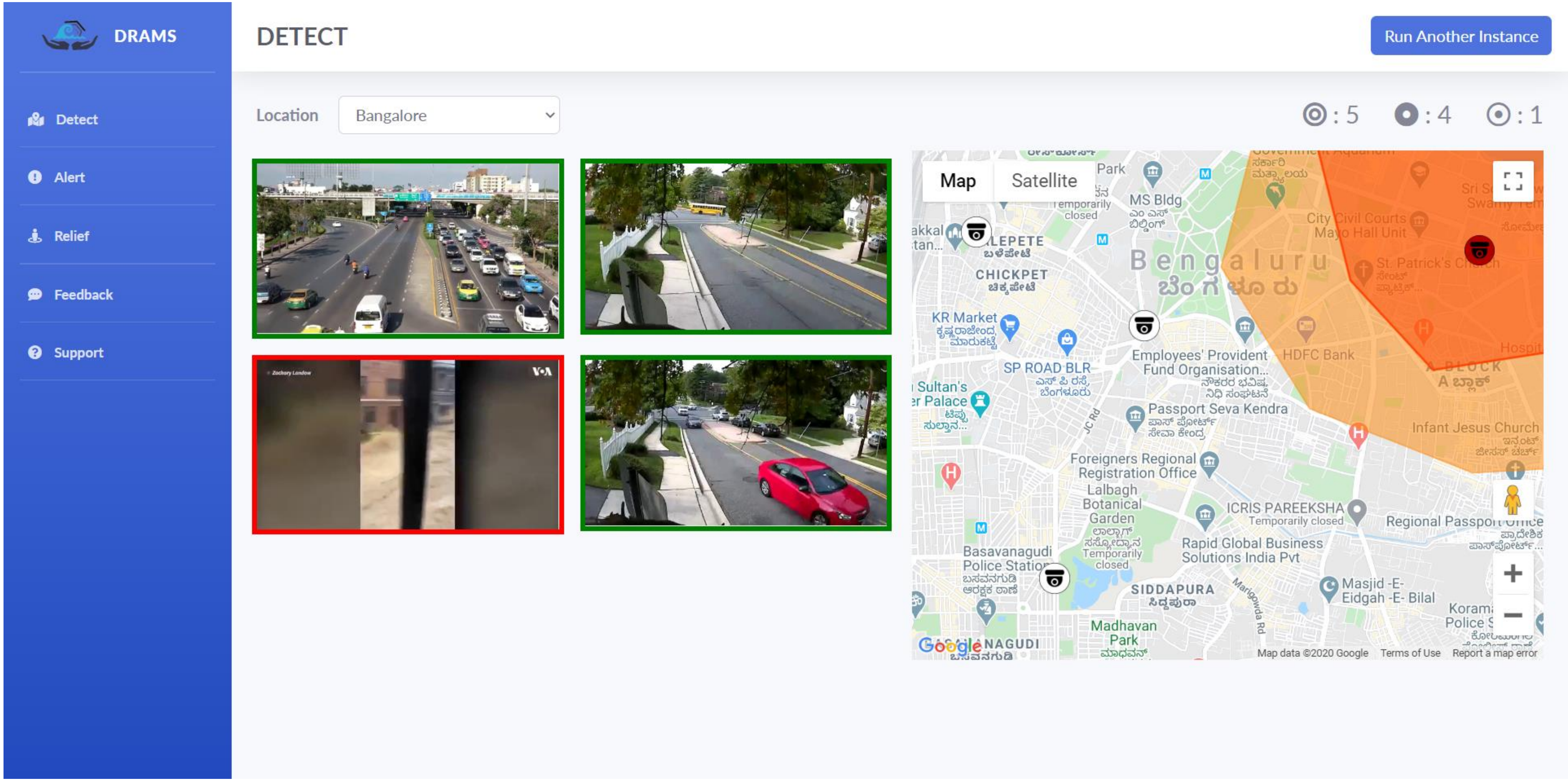
- Name
- Total Nodes
- Co-ordinates
- Live Nodes

Class Location

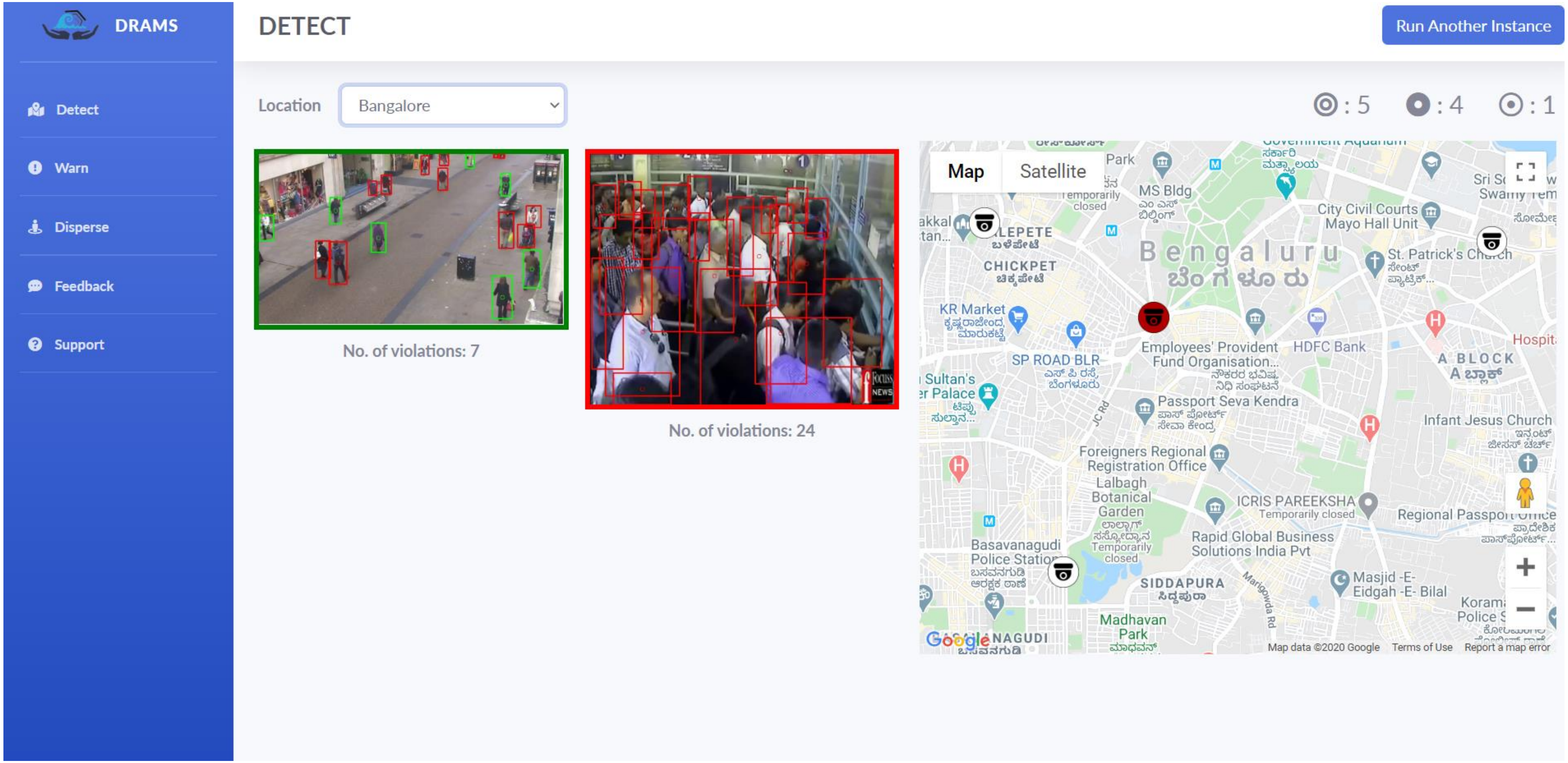
- Name
- Co-ordinates
- City - Linked
- Population
- Nodal Center
- Center Category
- CCTV URL
- Severity
- Alert Status

UI/UX Demonstration

Solution Appearance



User Experience – COVID Prevention



User Experience – Routing for Relief

- Detect
- Alert
- Relief
- Feedback
- Support

TIME
00:07:26

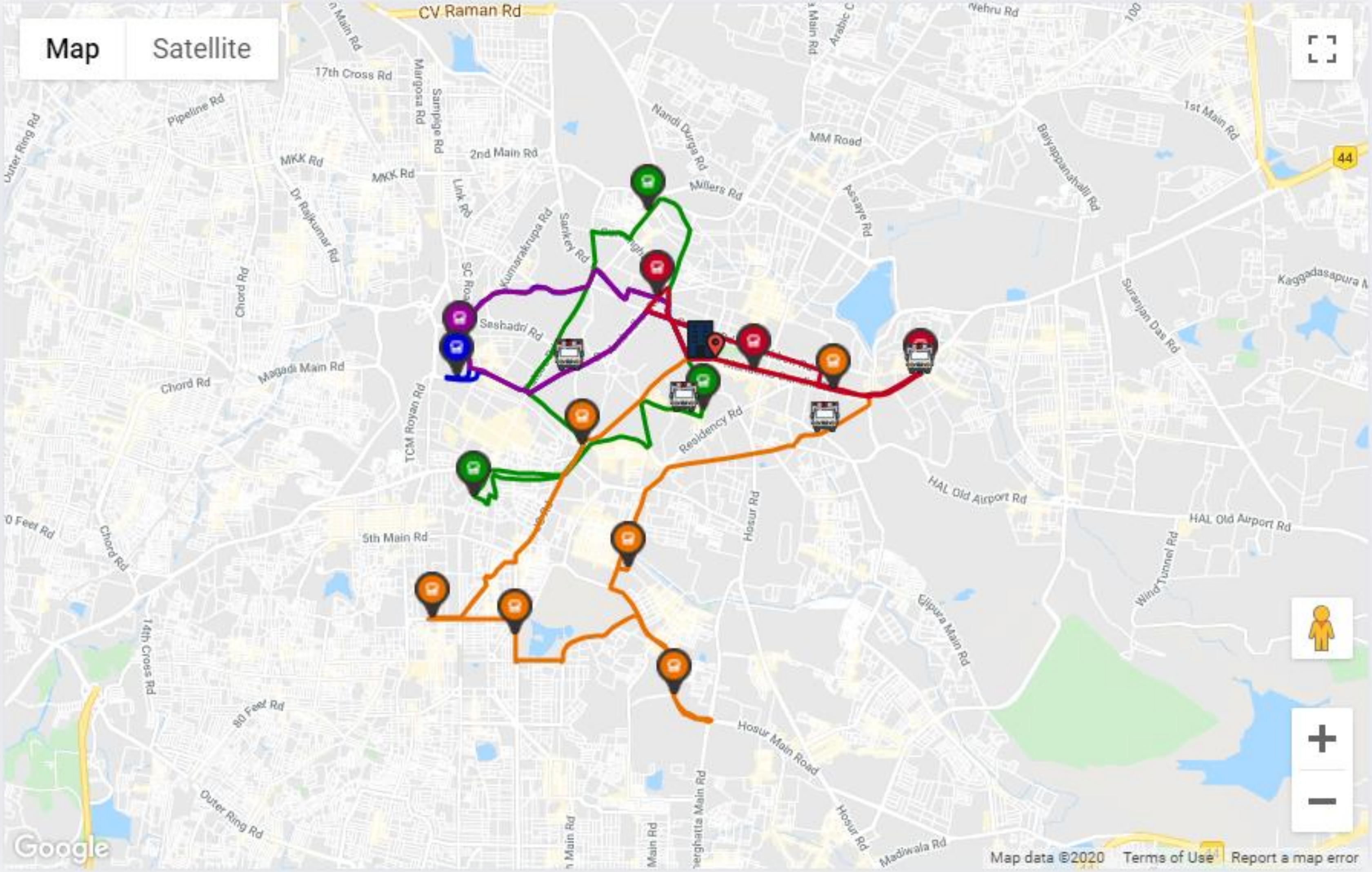
VEHICLE CAPACITY
10

NO. OF VEHICLES
5

MAX. OCCUPANCY
80%

KMS PER TRIP
60

MAX. WALKING DISTANCE
1 km



Thank You