**Salus**

**Final Project Definition**

**By:**

Eden Mironi

Maya Rachmani

Sandra Mashinsky

Vasily Kluchnikov

Gil Ben Basat

Imri Kalabrino

**Supervisor:**

Moshe Butman

**Table of Contents**

[1. Project description 3](#_Toc91524069)

[2. Related Work 3](#_Toc91524070)

[3. Functional Description / Requirements 5](#_Toc91524071)

[4. Architecture 6](#_Toc91524072)

[5. Work plan 9](#_Toc91524073)

[6. Client side 11](#_Toc91524074)

[7. Server Side 13](#_Toc91524075)

# Project description

Our project addresses the problem of identifying road accidents using security camera footage. In each frame we work on, we will identify whether an accident occurred, the part of the image in which it occurred and all those involved in the accident - is it a motorcycle accident, a car collision, or a pedestrian?

We will solve the problem with the help of Object Detection and Object Classification.

The product of the project will be a web app will be connected to live broadcast of security cameras - in Israel or around the world, and it will alert in real time of an accident. The alert could reach the police and the rescue forces who can call for help if needed.

For example, we can connect to the "Netivei Israel" website, which broadcasts security footage of the country's roads live, and from there alert the rescue forces.

With implementing the project, we will maximize the security given to drivers and all road users and add a layer of defense to deal with emergencies on Israel's highways. The overarching goal is to work to significantly reduce the number of fatalities and the severity of injuries in road accidents - especially in cases where speed of response to help and rescue are critical things.

# Related Work

To date, number of studies have been conducted on this subject on which we will detail.

In 2016, a study was conducted (https://yuxng.github.io/chan\_accv16.pdf) aimed at predicting accidents in advance. The study included about 680 high-quality dashcams videos, with the videos manually divided into 1750 short videos.

In some of the videos the objects (vehicles, pedestrians, bicycles, etc.) were manually tagged with bounding boxes, while in the rest of the videos an algorithm for calculating the location of objects (Faster R-CNN) was activated.

The deep learning methods used in this study are

Dynamic-Spatial-Attention (DSA) and Recurrent Neural Network (RNN).

For the frames in the video think what the probability of an accident is, when the probability has passed a certain threshold, it has been announced that an accident will occur.

The study managed to reach 80% recall and 56% accuracy in predicting accidents 1.8 seconds before the accident on average.

In 2018, another study was conducted (https://arxiv.org/abs/1809.05782) with a similar goal of predicting accidents, as well as identifying them.

The study contains a new data set which contains about 1400 videos of accidents alone, with about 200 videos manually tagged with bounding boxes. The study does not contain a data set of negatives but uses another public data set (https://detrac-db.rit.albany.edu/) which contains a very large amount of data of videos without an accident (dozens of videos lasting 10 hours each).

The study is based on the previous study we mentioned from 2016 with a prediction model of the accident, but it uses an improved R-CNN algorithm for object identification:

Faster R-CNN with Context Mining / Augmented Context Mining

This algorithm has mostly improved human identification, but it has reached very similar recall and precision results compared to the 2016 study.

In 2021 a study was conducted (https://www.semanticscholar.org/paper/Automatic-Detection-of-Traffic-Accidents-from-Video-Robles-Serrano-Sanchez-Torres/e8c135640fdbe672c688004c677c39c5af7295e1)

(https://www.mdpi.com/2073-431X/10/11/148/pdf)

with the aim of identifying accidents by security camera footage, similar to our study.

The study uses the previous data set (from the study cited from 2018), and a public image repository for improving the identification of objects in the image.

In the study they trained 2 different models, one for the identification of the objects, and the other for features derived from the video, and finally combined them.

The first model was created using the InceptionV4 architecture, and the second with two layers of ConvLSTM (a special type of CNN).

The new study was able to achieve 98% accuracy in binary identification of videos with an accident.

# Functional Description / Requirements

The system will consist of two main modules:

- A part where we will train our model according to the database we collected

- A second part which will be the application in which the prediction will be performed and will consist of a client-server architecture.

The server will receive information live from an external API of [Netivei Israel](https://www.iroads.co.il/%D7%AA%D7%99%D7%A7%D7%99%D7%99%D7%AA-%D7%9E%D7%A6%D7%9C%D7%9E%D7%95%D7%AA/), which broadcasts from dozens of cameras deployed throughout the country, and with the help of the model we trained, each frame will be classified as to whether an accident occurred.

In each short video we get from [Netivei Israel](https://www.iroads.co.il/%D7%AA%D7%99%D7%A7%D7%99%D7%99%D7%AA-%D7%9E%D7%A6%D7%9C%D7%9E%D7%95%D7%AA/) API there are 15 frames every second,

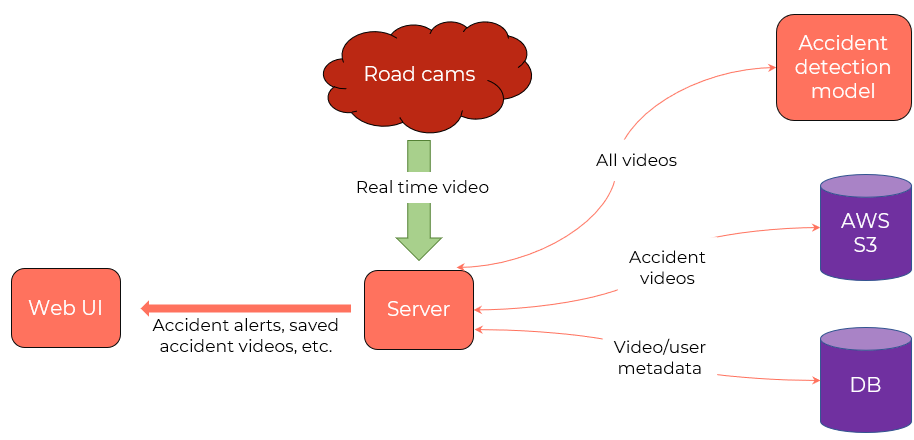
And the resolution is 1280\*720. We are planning to research and explore for the best use in the data and consider using less frames and different resolution, in order to get the best results for real time calculations.

The client side will display dashboards that will show the user past accidents that we have foreseen. In the event of an accident in real time, a message will pop up to the user and he will choose what the desired course of action is, whether the rescue and security forces should be called or whether it is a false alarm.

Our model will be built and trained in the google colab environment using GPU on a dataset consisting of 1,400 videos collected from YouTube. We will use the TensorFlow and Keras libraries.

We will apply state-of-the-art object detection models such as Faster R-CNN and accident forecasting models to our dataset and show their results.

# Architecture



**Application features**

Downloading real-time footage from CCTV cameras across the country, saving them, and running our accident prediction model on each video.

If the video does not contain an accident, it is deleted by the server shortly after.

If the video does contain an accident, then the video of the accident, and at least 30 seconds of footage before and after the accident, are also permanently saved.

In order to save video location for deletion or for a permanent save, the server uses a DB.

Allowing users to watch suspected accident videos, mark videos as true positive / false positive, watch previous accidents, view current cameras and their location, and more.  
In order to present the data to the user, and to make it possible for them to mark videos of suspected accidents as TP/FP, we will also use a DB in this part of the application.

**Technology choices**

Server - We chose a python http server. Python is a simple language, allowing us to write code quickly. We are also experienced in Python since it is used to build the model.

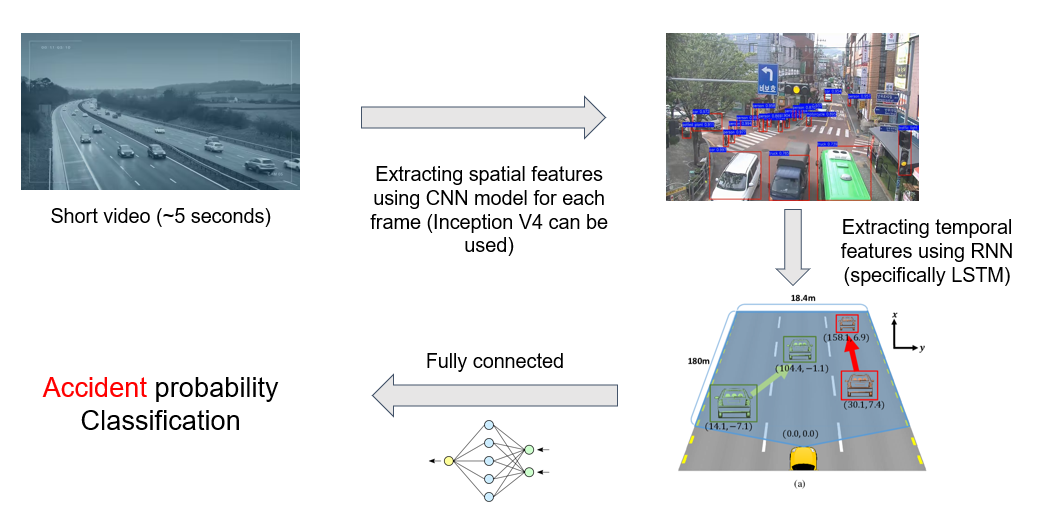
Client - We chose a web application, written using the React framework, utilizing JavaScript, html, and CSS. Web client works on all platforms and is the common way to write clients nowadays.

Video storage - We chose AWS S3 as our storage for videos, as it is a very popular and generic cloud storage, suited well for video files. Another option is to just save the files locally on the server, if the server storage is large enough.

Database - We chose PostgreSQL. It is free, easy to set up, and it uses a familiar SQL syntax.

Neural Network Model architecture

We will approach the model development with two possible architectures and will explore both to get the best results.



One model architecture is based on previous research and state of the art neural networks, used for the same problems as accident detection, or similar problems such as accident prediction.

We want to extract both spatial features and temporal features from the video, in order to predict the accidents.

**Spatial features** include objects and their types (such as cars, pedestrians).

**Temporal features** include the movement and the speed of the objects in the video over time.

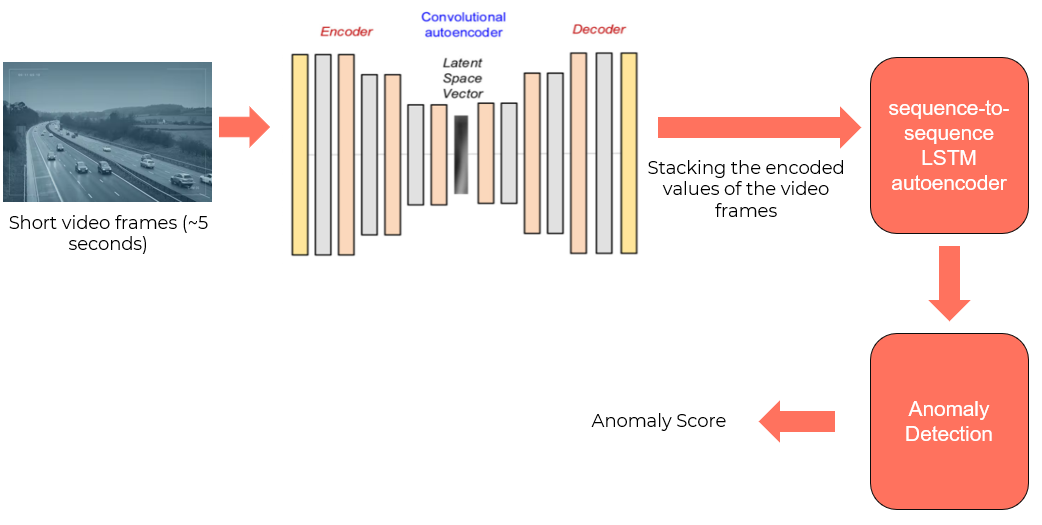
Our model has an input of a short video, consisting of many frames (images).

Then we extract spatial features of each frame using a CNN.

The next layers of our model extract temporal features, using an RNN.

The last layers of our model consist of fully connected layers, finally connecting a single neuron which contains the probability of the accident for the given video input.

The CNN used will be **Faster R-CNN with Inception V2**, which is a very fast neural network for classifying and detecting objects. A fast model is a must for video real-time analysis.

The RNN used will be **ConvLSTM**, a neural network which combines CNN and RNN techniques to extract temporal features, a well-suited model for running on image sequences inputs (which are extracted from the video using the previous CNN).

The other possible model architecture is based **anomaly detection** by using deep learning approach based on spatio-temporal autoencoder and sequence-to-sequence long short-term memory (LSTM) autoencoder. We assume that by following unusual activity of vehicles or pedestrians on the road we will detect an accident. The proposed model applies convolutional autoencoder and LSTM with sequence-to-sequence learning ability in a cascaded manner. The crux of the proposed approach is that it learns in an unsupervised manner alleviating the need to generate labeled​ anomalous road accident data – compared to the first proposed model.

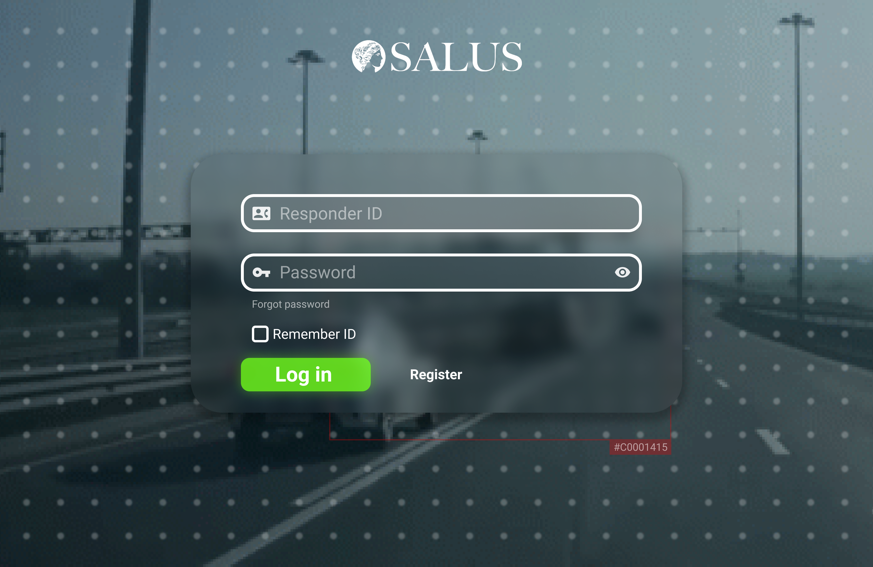
# Work plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topic** | **Task** | **Start Date** | **Finish Date** | **Number Of People Assigned** |
| **Data Collection** | Check if the current data from our dataset is good for us and similar to the videos of "Netivi Israel" | 31.12 | 31.12 | 1 |
| Get more videos of accidents | 31.12 | 7.1 | 1 |
| Get more videos of non-accidents | 31.12 | 7.1 | 1 |
| **Infrastructures** | repository for DB - PostgreSQL | 31.12 | 7.1 | 1 |
| repository for Cloud - S3 | 31.12 | 7.1 | 1 |
| Check for computer power for our model calculation | 31.12 | 7.1 | 1 |
| **CLIENT** | Basic app template | 2.1 | 9.1 | 1 |
| Dashboard page | 10.1 | 31.1 | 1 |
| Archive Page | 10.1 | 24.1 | 1 |
| **SERVER** | Connection between "Netivi Israel" live videos and out model | 18.1 | 25.1 | 1 |
| Connection to the DB | 10.1 | 17.1 | 1 |
| Connection to the Cloud | 10.1 | 17.1 | 1 |
| Notifications system | 10.1 | 24.1 | 1 |
| Client functionality | 10.1 | 24.1 | 1 |
| **Model** | Anomaly detection model | 25.1 | 15.2 | 3 |
| model based Faster R CNN + ConvLSTM architecture | 25.1 | 15.2 | 3 |
| **Integration** | Connection between the finished models and the server | 16.2 | 20.2 | 1 |
| Client-Server connection | 16.2 | 20.2 | 1 |

# Client side

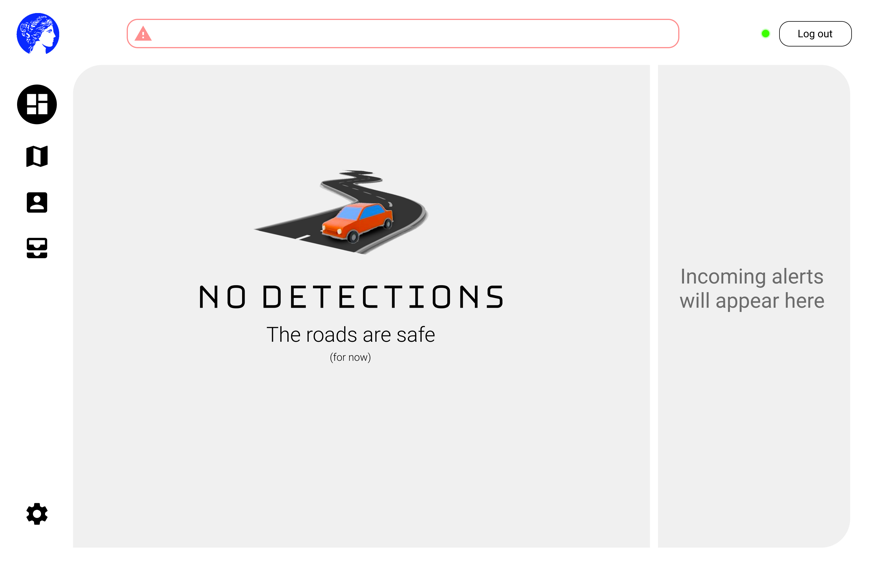
**Login**

Basic login screen that makes the system accessible solely for representatives of the transportation department or the police force to ensure information security.

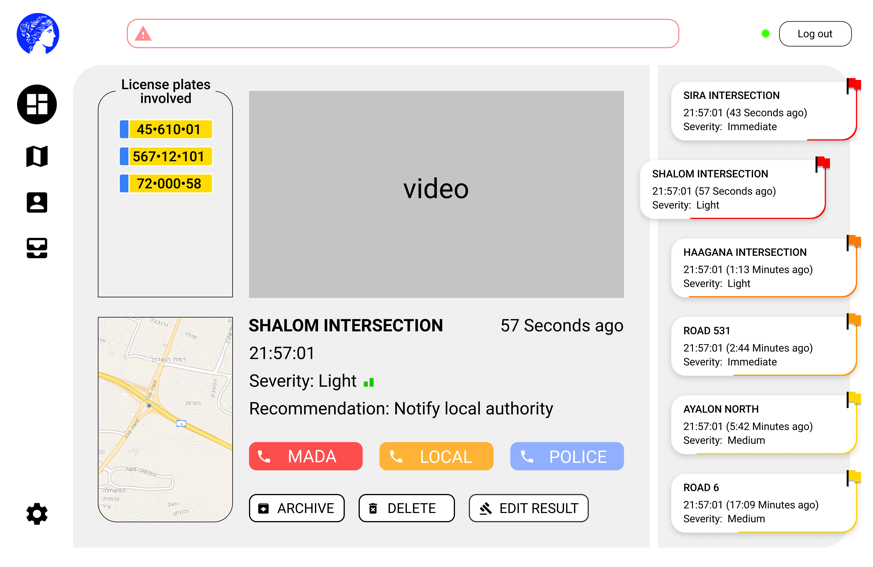


Dashboard display, showing the relevant alerts regarding detected accidents and a response action bar to allow users to call, alert and notify local authorities and police. The user can see alert details, edit, archive, and delete each alert.

(Potential here to extend the use of the system to allow analysis of severity per alert and creation of customized recommendation regarding to action-taking for the user)

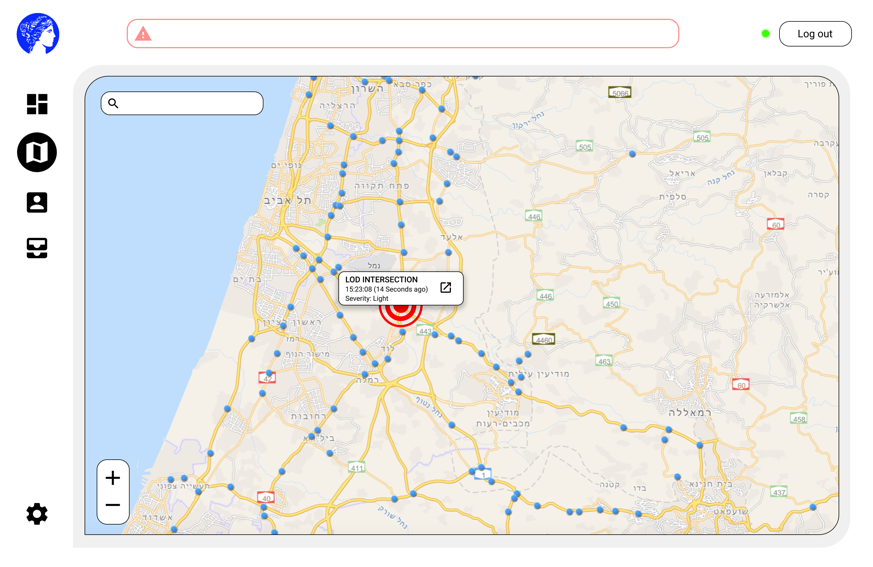


**Dashboard**

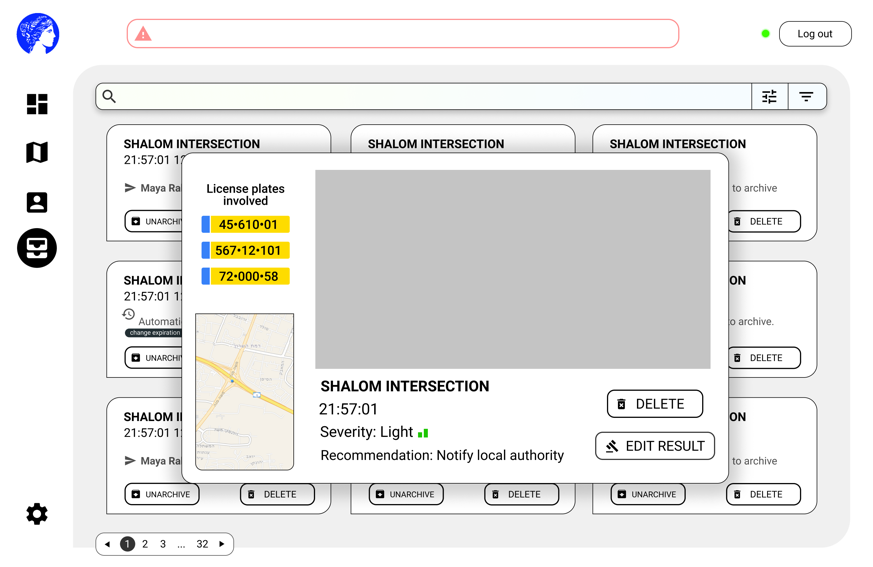
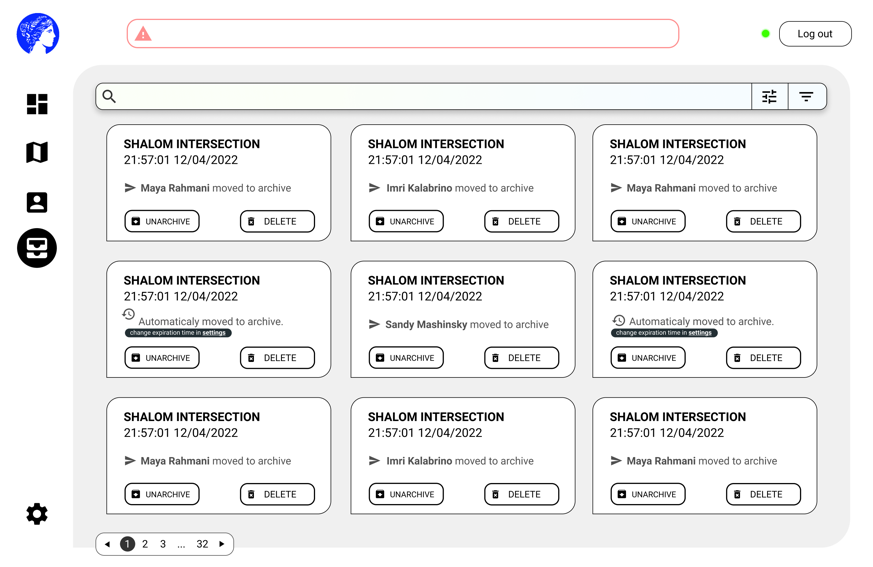


\

Conceptually similar to the dashboard, except in the form of a map – providing a broader real-time snapshot of Israeli roads. When an accident is detected, the alert shall pop on screen immediately.



**Map view**



**Archive**

UI display made to view and manage past predictions and alerts that have been moved to "the archive – which displays data no longer relevant or urgent.

# Server Side

Our server consists of two points of view.

One, deals with our training process. The model is based on state-of-the-art object detection models such as Faster R-CNN. Faster R-CNN is a deep learning architecture for object detection in still images, it extracts deep features of each proposal regions using a deep learning backbone such as ResNet-50. An important designing aspect of Faster R-CNN is its two-stage design: after features are extracted for proposals, they are classified and regressed to match the anchor boxes.

Our model will be trained at the google colab environment using GPU and we will use TensorFlow and Keras libraries.

The other point of view is the server side of the app we will develop. We will create a web app that will display dashboards of past accidents that we successfully forecast. In the event of an accident in real time, which we will discover using [Netivei Israel](https://www.iroads.co.il/%D7%AA%D7%99%D7%A7%D7%99%D7%99%D7%AA-%D7%9E%D7%A6%D7%9C%D7%9E%D7%95%D7%AA/) API, a message will pop up to the user and they will choose what the desired course of action is, whether the rescue and security forces should be called or whether it is a false alarm.

We will also maintain a DB (PostgreSQL) that will contain all the past accidents we have witnessed and their videos content location on the cloud we will use (S3).

On the server side we will call the prediction functions that will be based on the model we developed to check whether the video contains an accident.