

Traffic Violation Detection System

FINAL YEAR PROJECT [CSC-500]



PARTICIPANTS
MUHAMMAD KASHIF
[16SBSCSM081]
MUHAMMAD YASIR HASAN
[16SBSCSM017]

SUPERVISOR DR. SYED ASIF ALI

COURSE CO-ORDINATOR MR. BASIT HASSAN MS. WAJIHA NAIM

INTRODUCTION

Road accidents have been very common in the present world with the prime cause being the careless driving.

The necessity to check this has been very essential and different methods have been used so far.

However, with the advancement in the technology, different governing bodies are demanding some sort of computerized technology to control this problem of over speed driving.

INTRODUCTION



- We are proposing a system to detect traffic violations such as:
 - Not wearing seat belts

PROBLEM STATEMENT

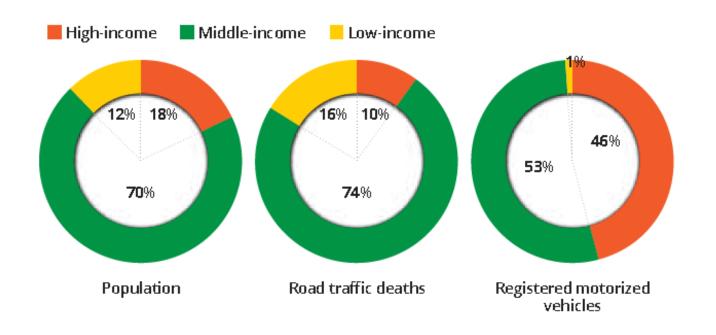
- develop a real-time traffic violation detection system that will work efficiently under the supervision of human
- Detect the objects that are merged into the background due to a temporary stop
- Adapting different traffic environment conditions, robustness against progressive or sudden illumination changes.

OBJECTIVES

- This work is aimed at developing an android application that depicts smart traffic offence analysis tool.
- •To develop an application that is faster, efficient and manageable.
- •To prepare complete, integrated solution for traffic offence management.

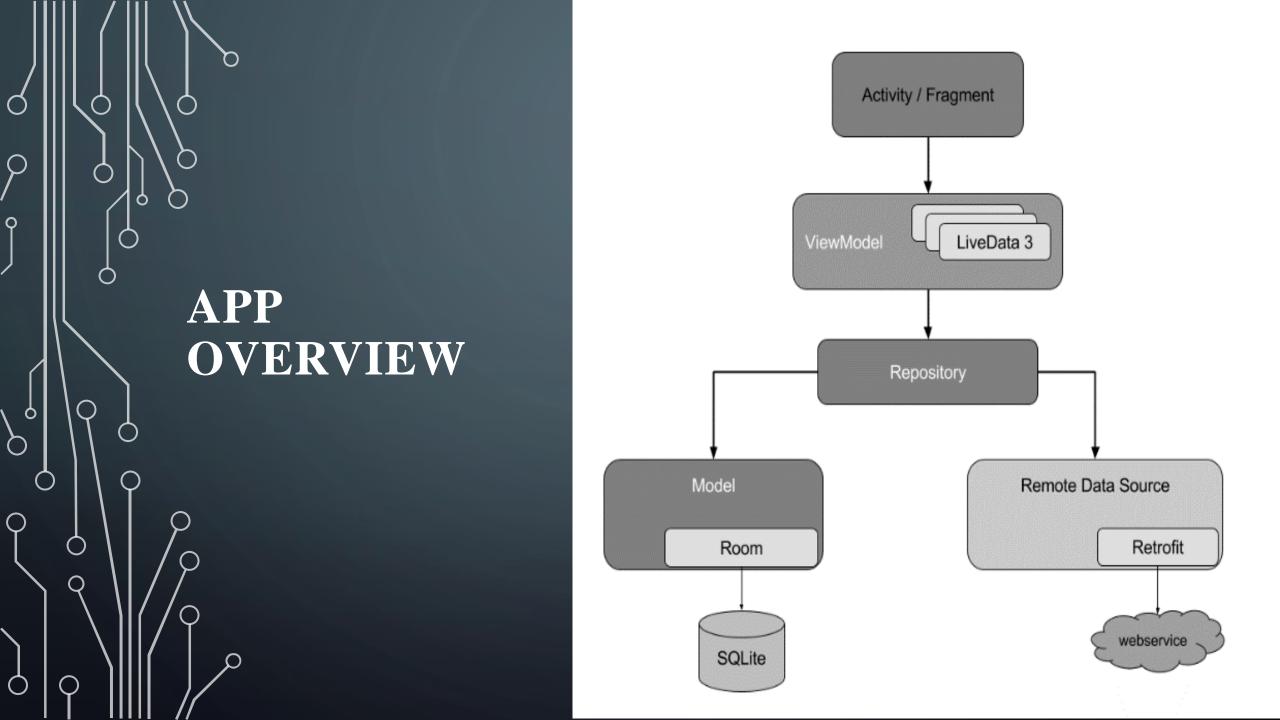
POPULATION, ROAD TRAFFIC DEATHS AND REGISTERED VEHICLES

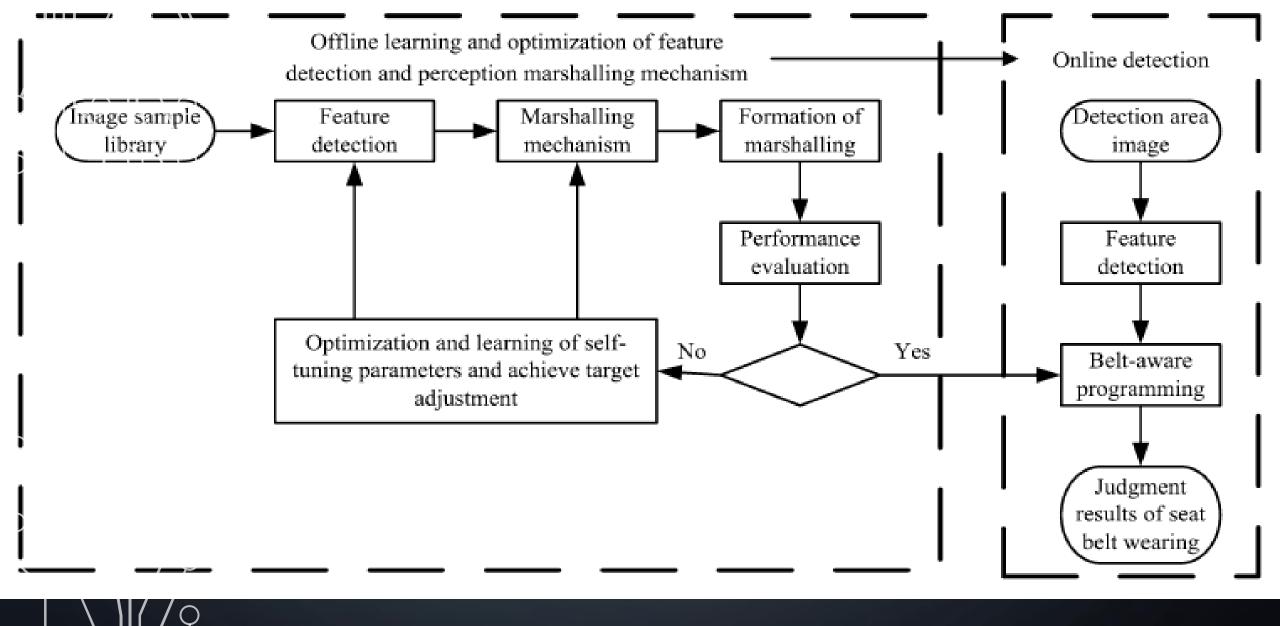
- Ninety-percent of road traffic deaths occur in low- and middle-income countries.
- These countries account for 82% of the world's
- Population
- A disproportionate number of deaths relative to their level of motorization (see next slide)



POPULATION - DEATH – REGISTERED IN COUNTRIES WITH INCOME STATUS

Start Ís Object a Vehicle? No YES, Violating **CCTV** Footage any rule? No SYSTEM YES **OVERVIEW** Take Actions Moving Object against Detection Object Classification End





DETECTION MODEL

- 1 import os
- 2 import numpy as np
- 3 from keras.models import Sequential
- 4 from keras.layers import Activation, Dropout, Flatten, Dense
- 5 from keras.preprocessing.image import ImageDataGenerator
- 6 | from keras.layers import Convolution2D, MaxPooling2D, ZeroPadding2D
- 7 from keras import optimizers
- 8 import tensorflow as tf

Using TensorFlow backend.

LIBRARIES USED

```
1 # used to rescale the pixel values from [0, 255] to [0, 1] interval
   datagen = ImageDataGenerator(rescale=1./255)
   # automagically retrieve images and their classes for train and validation sets
   train_generator = datagen.flow_from_directory(
           train_data_dir,
           target_size=(img_width, img_height),
           batch size=16,
           class_mode='binary')
   validation generator = datagen.flow from directory(
12
           validation_data_dir,
13
           target size=(img width, img height),
           batch size=32,
           class_mode='binary')
15
```

Found 1996 images belonging to 2 classes. Found 530 images belonging to 2 classes.

TRAINING TENSORS

```
1  nb_epoch = 1
2  nb_train_samples = 2048
3  nb_validation_samples = 624
```

```
model.fit_generator(
train_generator,
samples_per_epoch=nb_train_samples,
nb_epoch=nb_epoch,
validation_data=validation_generator,
nb_val_samples=nb_validation_samples)
```

GENERATING MODEL

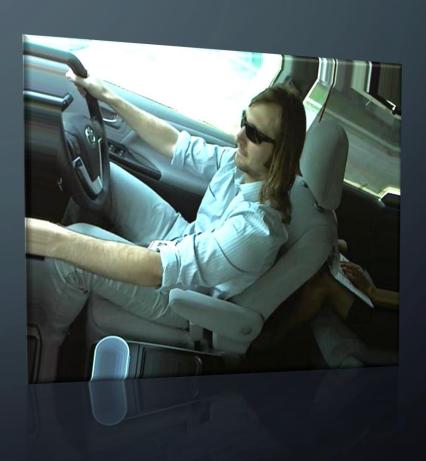
TRAINING DATA (SEATBELT)





TRAINING DATA (NO SEATBELT)





Learn more about these metrics and graphs

Score threshold

Average precision

Precision

Precision

97%

105ms

On Google Pixel 1

MODEL EVALUATION

EVALUATION FOR SEATBELT



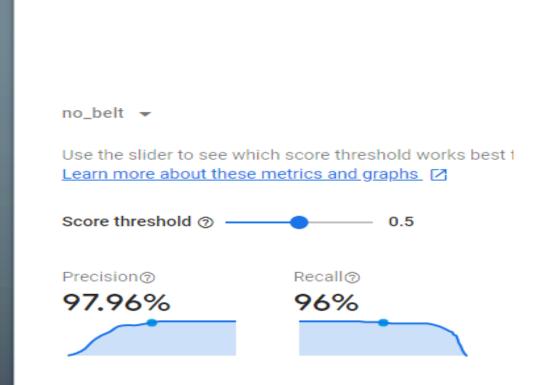
Use the slider to see which score threshold worl Learn more about these metrics and graphs [2]







EVALUATION FOR NO SEATBELT





Predictions

Only the top 10 labels are shown

no_belt 74.1% seatbelt 26.3%

MODEL EVALUATION (NO SEATBELT)



Predictions

Only the top 10 labels are shown

seatbelt no_belt

07.070

30.2%

MODEL EVALUATION (SEATBELT)



USE CASES

