AUTOMEDIC

Prepared by : Project Phoenix

Product Pitch Framework

ESSENTIAL POINTS TO COVER

The Problem

The Solution

Market Research

How to Use

Business Model

The Problem

AMBULANCES GETTING STUCK IN TRAAFIC

James Douglas had once said that just a Few Seconds Can Mean the Difference Between Life and Death. And he was absolutely correct. Ambulances typically provide acute care in emergency cases such as a cardiac arrest or an accident, however, getting stuck in traffic congestion makes it is difficult for an emergency vehicle to cross the roads.

The Solution

WHAT DO WE DO?

In the instances mentioned in the previous slide, traffic lights play a key role in traffic management to control the traffic on the road. Therefore AUTOMEDIC has designed and developed a simple yet highly intricate python-based sensor to help emergency vehicles cross the road at traffic light junctions during emergency situations by stopping their stops. The main objective of the project is to develop an autonomous system to avoid the need for a timer-based system in India.

TECHNOLOGIES USED

The following work has been done in the arena:

- An AI model is trained using TensorFlow
- Emergency vehicles (such as ambulances) have been detected using OpenCV for better automated prioritization of vehicles.
- Traffic light system was designed using Turtle


```
5   wn.bgcolor("black")
6
7   #Draw a box around the light
8   pen=turtle.Turtle()
9   pen.color("yellow")
10  pen.width(3)
11  pen.hideturtle()
12  pen.penup()
13  pen.goto(-30,60)
14  pen.pendown()
15  pen.fd(60)
16  pen.rt(90)
17  pen.fd(120)
18  pen.rt(90)
19  pen.fd(60)
20  pen.rt(90)
21  pen.fd(120)
22
23  #Red Light
24  red_light=turtle.Turtle()
25  red_light.shape("circle")
26  red_light.color("grey")
27  red_light.penup()
28  red_light.goto(0,40)
29
30  #Yellow Light
31  yellow_light=turtle.Turtle()
32  yellow_light.shape("circle")
33  yellow_light.color("grey")
```

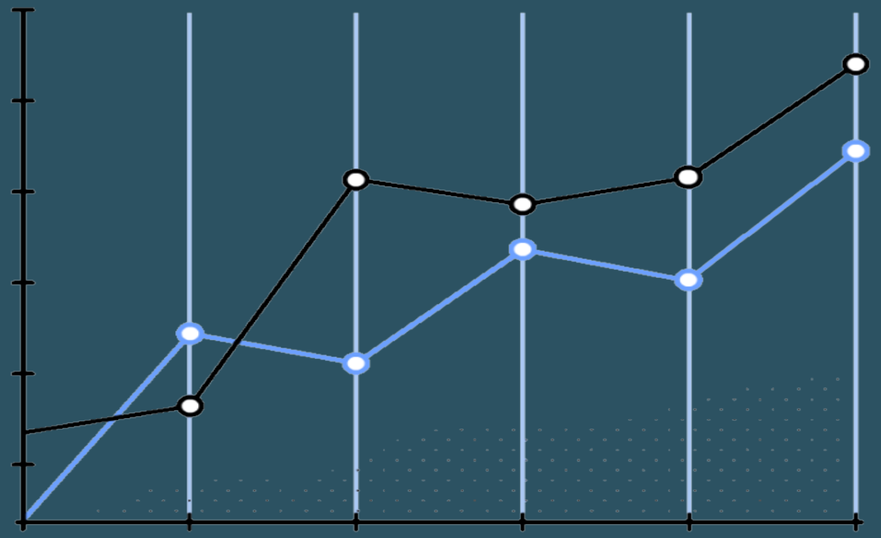
USP & FEATURES

- Can detect an emergency vehicle from 500m, thus providing enough time for the traffic lights to change safely.
- 95% accuracy
- Enhance the pre-existing CCTVs cameras
- No upfront cost

Market Validity & Target Market

Nowadays, most industrialized countries are using fixed-time strategies for urban traffic control. Vehicles at the traffic lane wait until the green light is on. Usually, during a traffic jam, the emergency vehicle, such as an ambulance, will be stuck at the traffic light junction. This is because even if the road users are willing to move out of the way they are still stuck waiting for the traffic light to turn green.

Our objective will be to implant our system in every traffic light junction of the country, starting with the metro cities



Prototype Implementation

```
Go  Run  Terminal  Help                                   model.py - a - Visual Studio Code
model.py 1 ×
project > model.py > ...
train_generator=train_data.flow_from_directory(
    traindir,
    target_size =(224, 224),
    class_mode='categorical',
    batch_size=32
    )

test_generator=test_data.flow_from_directory(
    testdir,
    target_size =(224, 224),
    class_mode='categorical',
    batch_size=32
    )

validation_generator=test_data.flow_from_directory(
    valtdir,
    target_size =(224, 224),
    class_mode='categorical',
    batch_size=32
    )

class_weights = class_weight.compute_class_weight('balanced', np.unique(train_generator.classes),train_generator.classes)


target_labels = next(os.walk(traindir))[1]
```

```
Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600      ['block_16_project_BN[0][0]']

Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0][0]']

out_relu (ReLU)                (None, 7, 7, 1280)   0           ['Conv_1_bn[0][0]']

==================================================================================================
Total params: 2,257,984
Trainable params: 2,223,872
Non-trainable params: 34,112
__________________________________________________________________________________________________
2022-02-12 15:56:27.000256: I tensorflow/stream_executor/cuda/cuda_dnn.cc:368] Loaded cuDNN version 8200
2022-02-12 15:56:36.964663: I tensorflow/stream_executor/cuda/cuda_blas.cc:1786] TensorFloat-32 will be used for the matrix multiplication. This will only be logged once.
E:\CN\Hackathon\a\env\lib\site-packages\keras\optimizer_v2\adam.py:105: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
Epoch 1/5
E:\CN\Hackathon\a\env\lib\site-packages\tensorflow\python\util\dispatch.py:1082: UserWarning: "`categorical_crossentropy` received `from_logits=True`, but the `output` argument w
as produced by a sigmoid or softmax activation and thus does not represent logits. Was this intended?"
  return dispatch_target(*args, **kwargs)
11/11 [==============================] - 18s 1s/step - loss: 9.3664 - accuracy: 0.2308 - val_loss: 8.7581 - val_accuracy: 0.1429
Epoch 2/5
11/11 [==============================] - 12s 1s/step - loss: 10.0845 - accuracy: 0.1331 - val_loss: 9.0336 - val_accuracy: 0.0714
Epoch 3/5
11/11 [==============================] - 11s 951ms/step - loss: 10.8388 - accuracy: 0.0976 - val_loss: 9.0015 - val_accuracy: 0.0714
Epoch 4/5
 6/11 [===============>..............] - ETA: 4s - loss: 11.7843 - accuracy: 0.1250
```

```
File  Edit  Selection  View  Go  Run  Terminal  Help                model.py - a - Visual Studio Code
EXPLORER                                  model.py 1 ×
A                                         project > model.py > ...
  Checkpoint_ver2                         IMG_SIZE = (224,224)
  env                                     IMG_SHAPE = IMG_SIZE + (3,)
  new_checkpoint                          base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
  project                                                                                include_top=False,
    OIDv6                                                                                weights='imagenet')
    main1.py
    main2.py                              base_model.summary()
    model.py                              len(base_model.layers)
    main.py
    THE_model


                                          image_batch, label_batch = next(iter(train_generator))

                                          feature_batch = base_model(image_batch)


                                          base_model.trainable = False


                                          model = tf.keras.Sequential([
                                            base_model,
                                            tf.keras.layers.GlobalAveragePooling2D(),
                                            tf.keras.layers.Dropout(0.2),
                                            tf.keras.layers.Dense(256, activation='relu'),
                                            tf.keras.layers.Dropout(0.2),
                                            tf.keras.layers.Dense(5, activation='softmax')
                                          ])



                                          base_learning_rate = 0.0001
                                          model.compile(optimizer=tf.keras.optimizers.Adam(lr=base_learning_rate),
                                                        loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True),
                                                        metrics=['accuracy'])


                                          initial_epochs = 5
OUTLINE                                   history = model.fit(train_generator
Python 3.9.7 ('env': venv)  ⊗ 0 ⚠ 1
```

Business Model

- Can be used in collaboration with government.
- Can be used by private healthcare enterprises.
- This technology can even be used in traffic management.
- Cost efficient as compared to other alternatives in the market

HOW IT WORKS

Three easy steps

STEP 1. DETECT EMERGENCY VEHICLE FROM APPROIPATE DISTANCE.

STEP 2.TURN THE TRAFFIC LIGHT GREEN.

STEP 3: ALLOW THE AMBULANCE TO PASS!

THANK YOU