Interim report on Automotive. Surveillance.

[**Section 2**](#_en9h09cfhql7) **2**

[**Approach to EDA**](#_4z4l1n7lmv0x) **2**

[Class having minimum number of images](#_64our4e8wrub) 2

[Class having maximum number of images](#_ttgahy73agmf) 2

[Class distribution](#_jmvsk42fk5kd) 2

[Maximum Size of the image available in dataset](#_8fb478tv8beu) 2

[Minimum Size of the image available in dataset](#_p2v4tw4axlk2) 2

[Checking bounding boxes on the image](#_kmfffocirvdh) 2

[Image Pre-processing](#_b5wje35dgnz7) 2

[Creating dataset from images folder](#_jxbk1y7upmnr) 2

[Extraction image height and width](#_82swsbkgfjuy) 2

[Merging annotations dataset, image size dataset with images folder](#_ywldlw9def3x) 2

[**Section 3**](#_r6j2x08ngtfd) **2**

[Deciding Models](#_upkvdog3qwlx) 2

[Model Building](#_oc6vzlis9nr2) 2

[**Section 4**](#_15nj0oncsz5b) **2**

[Model performance](#_jupbdc5hc4zt) 2

[Data Augmentation](#_hen9d444i2g3) 2

[Transfer learning](#_lte178u1k4ue) 3

[Last layer Classification and regression](#_r12v1hdqa9px) 3

## Section 2

#### Approach to EDA

###### Class having minimum number of images

###### Class having maximum number of images

###### Class distribution

###### Maximum Size of the image available in dataset

###### Minimum Size of the image available in dataset

###### Checking bounding boxes on the image

#### Image Pre-processing

###### Creating dataset from images folder

###### Extraction image height and width

###### Merging annotations dataset, image size dataset with images folder

## Section 3

~~Based on the nature of the problem, decide what algorithms will be suitable and why?~~

~~Experiment with different algorithms and get the performance of each algorithm.~~

#### Deciding Models

The problem statement states that it is a Automotive. Surveillance problem and the context states that computer vision can be used to automate supervision and triggering events for images of interest.

By understanding these statements, we can conclude that the problem can be solved by using deep learning techniques. In Deep learning techniques, we can look into particular sections of convolution neural networks (CNNs). Convolutional Neural Networks (CNN or ConvNet) are complicated feed forward neural networks used in machine learning. Because of its great accuracy, CNNs are employed for image classification, image localization , image detection , etc. The CNN uses a hierarchical model that builds a network, similar to a funnel, and then outputs a fully-connected layer in which all neurons are connected to each other and the output is processed. The benefit of using CNNs is their ability to develop an internal representation of an image by looking at only a subset of pixels in the images.

We cannot use a Dense neural network for computer vision tasks in deep learning. The fundamental difference between the Convolutional and Dense layers is that the Convolutional layer requires fewer parameters because the input values are forced to share the parameters. The Dense Layer employs a linear operation, which means that the function generates each output based on each input. An output of the convolution layers is formed by just a small size of inputs which depends on the filter's size and the weights are shared for all the pixels

For this problem statement we have decided to use transfer learning and use a pre-built model in tensorflow. Few layers in pre-built models can be trained to get good accuracy and best prediction for the bounding box. Transfer learning is adaptable, allowing pre-trained models to be used directly as feature extraction preprocessing or integrated into completely new models.

We have evaluated a few pre-trained models - MobileNet, ResNet50, VGG and efficient net as part of the experiment. Among these evaluations we have got the best result for an efficient net - efficientnet-b5.

#### Model Building

Efficientnet-b5 model with ImageNet weights is used for developing network for classification and regression tasks for the given problem statement. The efficientnet-b5 model is one of the EfficientNet models designed to perform image classification. This model was pretrained in TensorFlow\*. All the EfficientNet models have been pre trained on the ImageNet\* image database. Below mentioned are result obtained after training the model.

## Section 4

#### Model performance

~~What are the approaches you can take to improve your model~~

~~Can you do some feature selection, data manipulation and model improvements.~~

###### Data Augmentation

One of the ways of improving the model performance is to train the model on more images. The training of deep learning models usually necessitates a large amount of data. In general, the more data there is, the better the model will perform. The issue with a paucity of data is that our deep learning model may not be able to learn the pattern from the data, and so may result in poor performance on test data.

Instead of collecting more data, augmentation techniques can be applied to generate as much data as required. Some of the commonly used augmentation techniques are rotation, shear, flip, etc. While applying data augmentation techniques for the given problem statement, we have to modify the bounding box coordinates as well. For this purpose, we can make use of imgaug or chitra libraries which are readily available and apply few inbuilt functions on images for data augmentation.

One of the augmenter which is available in the library is Affine. Affine transformations involve Translation (“move” image on the x-/y-axis),Rotation, Scaling (“zoom” in/out), Shear (move one side of the image, turning a square into a trapezoid). This Augmenter will affect bounding boxes and hence we need to use BoundingBoxesOnImage function on the bounding box coordinates.

Horizontal flips, Resize and change brightness can also be applied on the images and these function can be wrapped in function which can then be applied which using batch generator for the images.

###### Transfer learning

Unfreezing a portion of a model and retraining it with a very low learning rate on the new data would give significant improvement in model accuracy. Since the dataset which is provided to us has 8144 images and data similarity is quite low, freezing initial layers of the pretrained model and re-train just the remaining layers will help in accuracy improvement.

Here it is recommended to keep the learning rate very low since the layers are unfreezed and model weights are trainable. Because the training will be done on a larger model it's also crucial to utilise a very modest learning rate at this stage. There are substantial weight changes happening here and there is always a danger of overfitting. So incremental readjustment of the pre-trained weights are crucial.

###### Last layer Classification and regression

If the model is overfitting for given data, dropout layer can be introduced between last dense layers which will help in reduction of overfitting. Dropout is a training strategy in which randomly selected neurons are rejected.They are “dropped-out” randomly. This means that on the forward pass, their contribution to the activation of downstream neurons is removed temporally, and on the backward pass, any weight updates are not applied to the neuron.

Using Batch normalization helps in reduction of general errors. Batch normalisation is a technique for standardising network inputs that can be applied to either the activations of a previous layer or the inputs themselves. Batch normalisation reduces generalisation error by speeding up training (in some situations by halving or bettering the epochs) and providing some regularisation.