COMPONENT 3
EMERGENCY VEHICLE IDENTIFICATION

Abstract
The primary aim of this report is to classify vehicles into emergence and non-emergence, with the help of identification of the vehicles' characteristics and components.

Introduction
Emergency vehicles such as police vans, ambulance, and fire trucks, were identified through lights and siren alerts. Identifying emergency vehicles and allowing them free road access in cases of emergencies, will be of great help. Therefore, it is of utmost importance that vehicles be classified into categories, emergency and non-emergency based on noticeable characteristics, to aid in avoiding confusion and delay during the case of an emergency.

Data Pre-processing and Features
The data used in this project, was obtained from an online source. It consists of the train and test sets dataframes having key columns; 'image_names' as vehicle labels and 'emergency_or_not'. With the latter stating if a vehicle is emergence (i.e.,1) and not emergence (i.e.,0). The pictures of these vehicles were also available in train and test sets too.
In the Jupyter notebook, the directories of the images and dataframes were read in and for the dataframes, each of the columns were converted to 'object' data type. The shape of these images was confirmed to be (224, 224,3) during overview.

Prior to building the model, the keras image data augmentation (Image Data Generator) was used to transform the images of the vehicles' and generate batches by specifying the required arguments (Keras, 2022). The next step taken was splitting into 70% train, test and 30% validation sets.

Convolutional Neural Network
Convolutional neural network (CNN) is a part of machine learning that uses deep learning, to imitate the way human thinks and learns. These algorithms are designed to recognize patterns and operate like the human brain neurons. They have the tendency to classify and identify objects. For example, Image and Facial Recognition (Rahul, 2022).

Constructor Stage
A CNN consists majorly of convolution layers. These layers comprise of the input, hidden and output layers. The filters are the features passed into this network to be identified in the kernel. The hyperparameters that were used for 32 and 64 filters are:

- Kernel Sizes (3x3) and input shape (224,224,3) as shape of images.
- The stride length which describes the steps which the kernel takes as it moves over the image to identify its features. This was set to default (i.e., 1)
- Padding set as same assuming the stride is 1 which helps to keep the calculations of a convolutional layer together (Ojie, 2022) and valid so that layers will only access the important features.
- The activation function which introduces non-linearity to the kernel. The rectified linear unit activation function (ReLU) was used.

- The sub-selection of samples using ‘MaxPooling’ which at each step around the kernel, filters the maximum number (pixel).
- The dropout and batch normalization helps to regularize this process to avoid overfitting or underfitting.
- The sigmoid activation function was used at the output layer.

During this process, the weights of the features are examined to extract the most useful information based on these hyperparameters.

Compilation and Training Stage

The aim in the compiling stage is to minimize loss while increasing the training weights. This is achieved through backpropagation. The optimizer ‘Adaptive Moment Estimation (Adam)’ and loss quantifier ‘binary cross entropy’ were used to optimize weights and quantify loss (Kiefer *et.al.*, 2019).

The train and validation sets were then passed to the model to train with 10 epochs.

Evaluation Stage

Model evaluation was then done on the test data set to know how much the

Results

The performance metrics ‘accuracy’ was used to determine how often the predictions of test set equals the vehicle features.

While fitting the sequential model on the train and validation data sets with all the parameters described earlier, it was observed that during the first convolutions, the train and validation loss was reducing as accuracy increases. Introducing the second layers, showed a slight increase in loss even as the accuracies increased. The complete convolutional process ended on the third layer. Here, there was a high variance between the train and validation losses and accuracies.

This concludes that as the layers of a convolutional neural network increases, there is a tendency of overfitting.

References

Keras(2022). *Keras Documentation*. Available Online: https://keras.io/ja/preprocessing/image/ [Accessed: 14/12/2022].

Husniza, R., Rusyaizila R., Mohammed H. A. *Emergency Vehicle Recognition and Classification Using HSV Color Segmentation*. Conference Paper · February 2020 DOI: 10.1109/CSPA48992.2020.9068695.

Kiefer, J. and Wolfowitz, J., 1952. *Stochastic Estimation of Maximum of a Regression Function. The Annals of Mathematical Statistics*, pp.462-466.

Rahul, A. (2022), *Convolutional Neural Network (CNN)*. Available Online: https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network [Accessed: 14/12/2022].