**Emergency Vehicle Identification**

**Abstract**

Convolutional Neural Networks (CNNs) are Artificial Intelligence algorithms based on multi-layer neural networks that learn relevant features from images, being capable of performing several tasks like object classification, detection, and segmentation (Todt and Krinski, 2019). The identification of an emergency vehicle is discussed in this report

**1.0. Introduction**

A bigger problem occurs when there are accidents or life-threatening emergencies but emergency vehicles encounter traffic and as a result, they get delayed and danger could not be prevented or averted. The problem discussed is addressed by training a Convolutional Neural Network on a dataset of vehicle images covering everyday cars, ambulances, police cars, and fire-brigade vehicles in models that employs image processing for the detection of emergency.

The report will provide information on the steps considered before building a model, the effects of increasing the model layers, check for the occurrence of overfitting in the model, and the performance of the models built.

**2.0. Methodology**

**2.1. Steps Considered prior to building a model**

1. Imported all the key libraries
2. Artificial transformations to the raw dataset in order to make the dataset cleaner, learnable and in a uniform format and read the picture dataset stored in a CSV format.
3. The test train CSV file was assigned 0s and 1s for non-emergency and non-emergency vehicles, respectively.
4. Decode and reshaped the data JPEG content to RGB grids of pixels with channels.
5. The data is split into training, validation, and test sets
6. Converted data into tensors, which are multi-dimensional arrays of numerical values to floating-point tensors for input to neural network
7. Convert these into floating-point tensors for input to neural nets.
8. Training data augmentation by using random rescaling, rotation range, horizontal flips, sharing, stretching, width and height range, as well as random cropping
9. Converting the labels to a one-hot encoded format by normalizing the pixel values which ranges between 0 and 255 to the range [0, 1], the activation functions can operate more effectively, which can improve the performance of the neural network
10. Converted the labels to a one-hot encoded format: In many cases, the target labels will be represented as a class index (e.g. 0, 1, 2, etc.), rather than as a vector of probabilities. In this case, it is necessary to convert the class indices to a one-hot encoded format
11. The ImageDataGenerator class is used to load and preprocess images on the fly, which can be useful for large datasets that do not fit into memory. The batch\_size parameter specifies the number of images to be processed at once, and the target\_size parameter specifies the dimensions to which the images will be resized. The class\_mode parameter specifies the type of classification problem being solved, in this case, it is set to 'binary' for a binary classification problem.

**2.2. Increasing the number of layers**

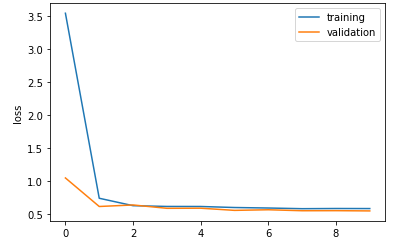
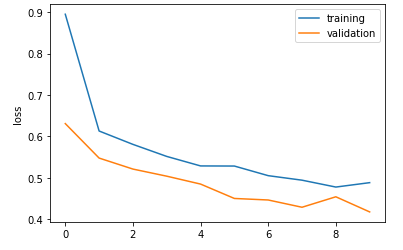
Doing this in a model basically enhances its performance by allowing the model to learn more complex and abstract patterns in the dataset which leads to better accuracy and other metrics, such as lower error rates, higher precision and recall.

Increasing the number of layers also increases the number of computational resources required to train and evaluate the model, which can make the training process slower.

Lastly, using a large number of layers can make the model more prone to overfitting, which can degrade its performance on unseen data.

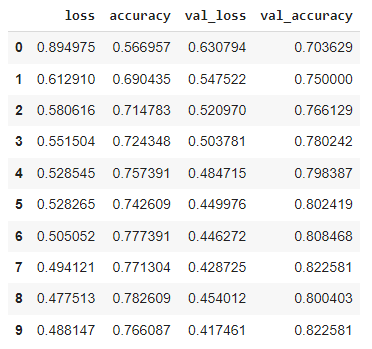
Therefore, there is a trade-off between model performance and training time, and the optimal number of layers for a given model will depend on the specific details of the dataset and the task at hand.

**2.3. Overfitting**

For model one, there was no case of overfitting. The training loss is not extremely low and there is no high testing loss. The line plots below show that the training loss is dropping to coincide with the line plot for the testing loss. For model two, A gap between the training and validation metrics may indicate overfitting, because the has a good performance on the training data but not on test data.

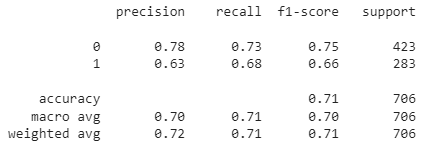
**3.0. Results**

**3.1. The performance measures most interested in and the reasons**

From the table for model\_2 metrics, we can see that the training and validation loss both decrease as the epoch number increases, which indicates that the model is improving its ability to predict the correct labels for the input data. Additionally, the training and validation accuracy both increase over time, it started learning at the rate of 56% and learned about 76% which indicates that the model is making more and more accurate predictions. Overall, this table provides a summary of the model's performance during training, which can be useful for diagnosing issues and monitoring progress.

**3.2. Classification report**

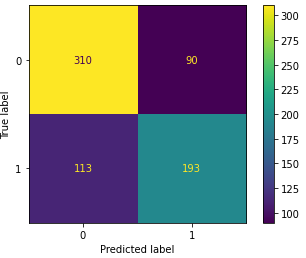
The classification report visualised the metrics used to access the quality of the model. The global accuracy score reveals that the model identified 74% of all the images correctly.

The precision score shows that for class 0, 78% of the images predicted to be non-emergency vehicles were actually non-emergency vehicles. For class 1, 63% of the images predicted to be emergency were actually emergency vehicles.

The recall score shows that 73% of the actual non-emergency vehicles classified were correctly predicted. For class 1, the model predicted the actual identification correctly for 68% of the images identified.

The overall performance of the model with the F1 score is 0.71 F1 score for the non-emergency vehicles (0.78) is much closer to 1 than the F1 score for emergency vehicles(0.66). This tells us that the model is doing a good job of identifying emergency vehicles than non-emergency vehicles. Although the current f1-score for emergency vehicles is okay it can be better. Finally, the support values show that 423 sample images were identified for non-emergency vehicles and 283 images were sampled for emergency vehicles.

**3.3. Confusion Matrix**



The confusion matrix 2x2 array obtained from the trained model gives a lot more than the accuracy of the model. It shows that there are many misclassifications for emergency vehicles (113) and it predicted 193 emergency vehicles correctly.

For non-emergency vehicles, it predicted 310 non-emergency vehicles correctly and misclassified 90 images as emergency vehicles.

**4.0. Conclusion**

The results of the models used for the classification of emergency vs non-emergency vehicles were discussed. The data preprocessing steps, model performance and model accuracy were covered.

References

Todt, E. and Krinski, B. (2019). *Introduction CNN Layers CNN Models Popular Frameworks Papers References Convolutional Neural Network -CNN*. [online] Available at: https://www.inf.ufpr.br/todt/IAaplicada/CNN\_Presentation.pdf [Accessed 13 Dec. 2022].