CONVOLUTIONAL NEURAL NETWORK MODEL

ABSTRACT

This report explores the application of a Convolutional Neural Network (CNN) model for image classification. The CNN was developed to accurately detect emergency vehicles (e.g., ambulances, fire trucks etc). The dataset consisted of separate test sets and training set which included labelled images categorized as 0 or 1, representing emergency or non-emergency vehicles. The architecture consists of multiple convolutional layers, pooling layers, and fully connected layers, optimizer (Adam optimizer). The model achieved a peak accuracy of 80% on the test dataset.

1.0 INTRODUCTION

Convolutional Neural Networks (CNNs) have become a key tool for tasks involving image classification, and recognition. It has emerged as an e6ective class of models for comprehending the content within images, leading to significant advancements in image recognition, segmentation, detection, and retrieval (Sharma, Jain & Mishra, 2018). CNNs also make strong and mostly accurate predictions about the nature of images and have much fewer connections and parameters and so they are easier to train, their ability to automatically learn spatial hierarchies features from input images has made them a preferred choice for tasks in image classification (Krizhevsky, Sutskever and Hinton, 2012).

2.0 METHODOLOGY

This study uses a CNN model to classify binary images of a dataset comprising images categorized as "0" and "1," the model leverages the hierarchical feature learning capability of CNNs to achieve a solid classification performance using a dataset of labelled images. The CNN architecture consisted of convolutional, pooling, and fully connected layers, with ReLU activation functions. The model was optimized using the Adam optimizer and binary cross-entropy as the loss function. Evaluation was done using accuracy, precision, recall, F1-score, and a confusion matrix to assess the model's performance.

3.0 ANALYSIS AND RESULTS

The analysis commenced by reading the CSV file and loading the training and test images. Followed by label mapping to map the binary classification 0: non_emergency and 1: emergency. The dataset was explored through visualizations and methods to assess the labelled images in the dataset.

Figure 1: Displaying the images

Architecture of the CNN Model 1

Table 1

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu

Kernel size	3,3
Layer	MaxPooling
dropout rate	50%
Padding	Same
output layer	1 using sigmoid activation function
input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	512
optimizer	SGD (learning rate = 0.01)
Batch size	32
Epochs	20

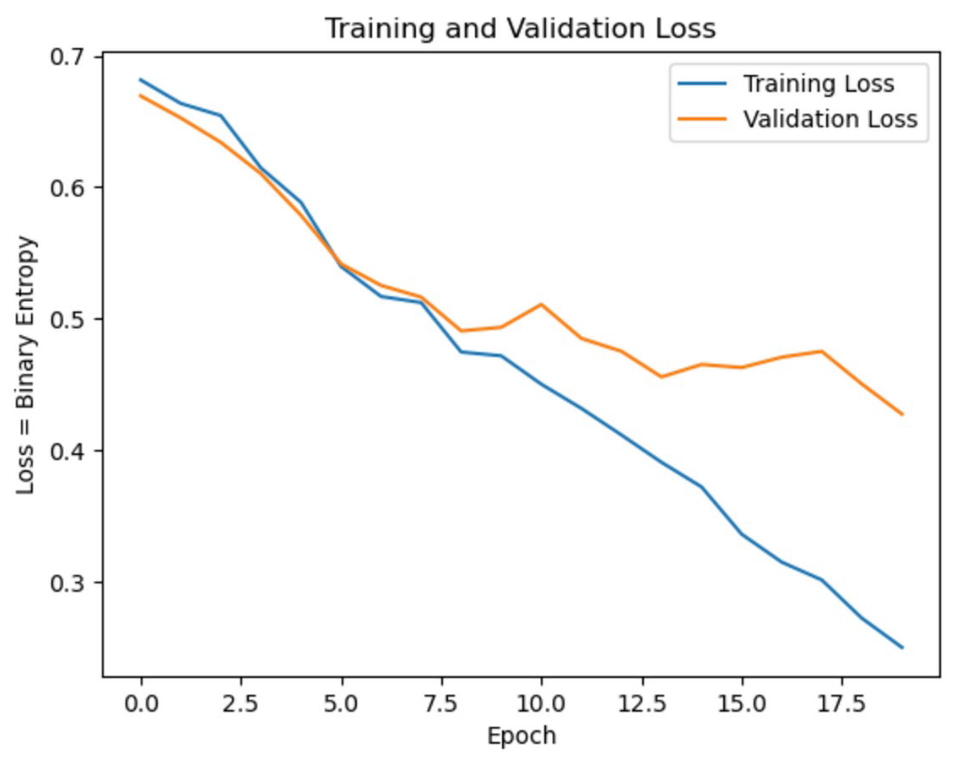

Figure 2: Training and Validation Loss of Model 1

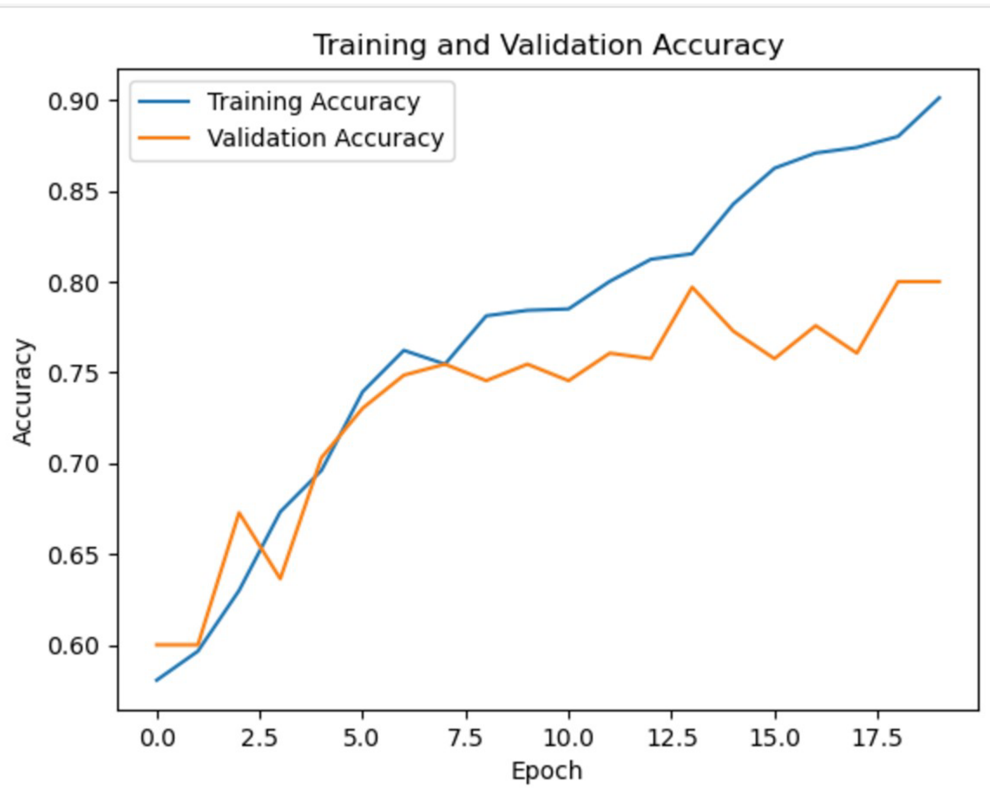

Figure 3: Training and Validation Accuracy of Model 1

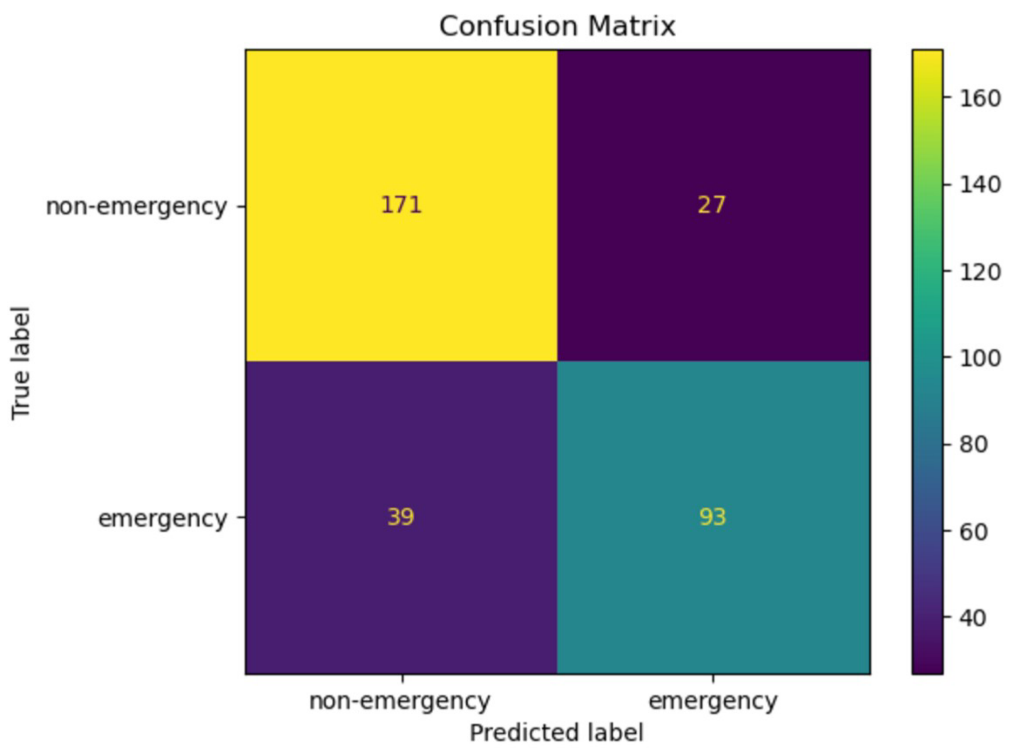

Figure 5: Confusion Matrix of Model 1

Table 2: Model 1 Evaluation Metrics

	precision	recall	f1-score	support
0	**0.81**	**0.86**	**0.84**	**198**
1	0.78	0.70	0.74	132
accuracy			0.80	330
macro avg	0.79	0.78	0.79	330
weighted avg	0.80	0.80	0.80	330

In table 2 we observe that the model performs well on the non-emergency class (Class 0) but struggles with the emergency class (Class 1), evident from its lower recall of 0.70 which is not a bad result. This suggests that the model struggle to identify emergency vehicles, this can be improved.

Architecture of the CNN Model 2 Table 3

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu
Kernel size	3,3
Layer	MaxPooling
dropout rate	30%

Padding	Same
output layer	1 using sigmoid activation function
input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	512
optimizer	SGD (learning rate = 0.001)
Batch size	32
Epochs	20

Figure 6: Training and Validation Loss of Model 2

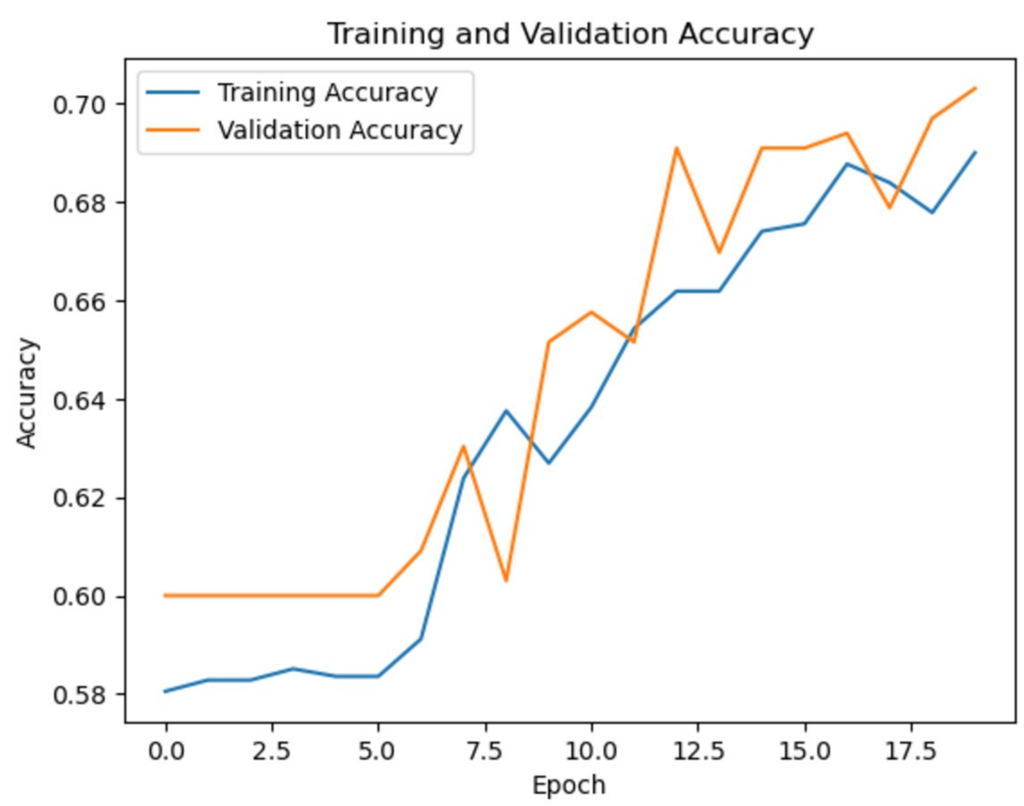

Figure 7: Training and Validation Accuracy of Model 2

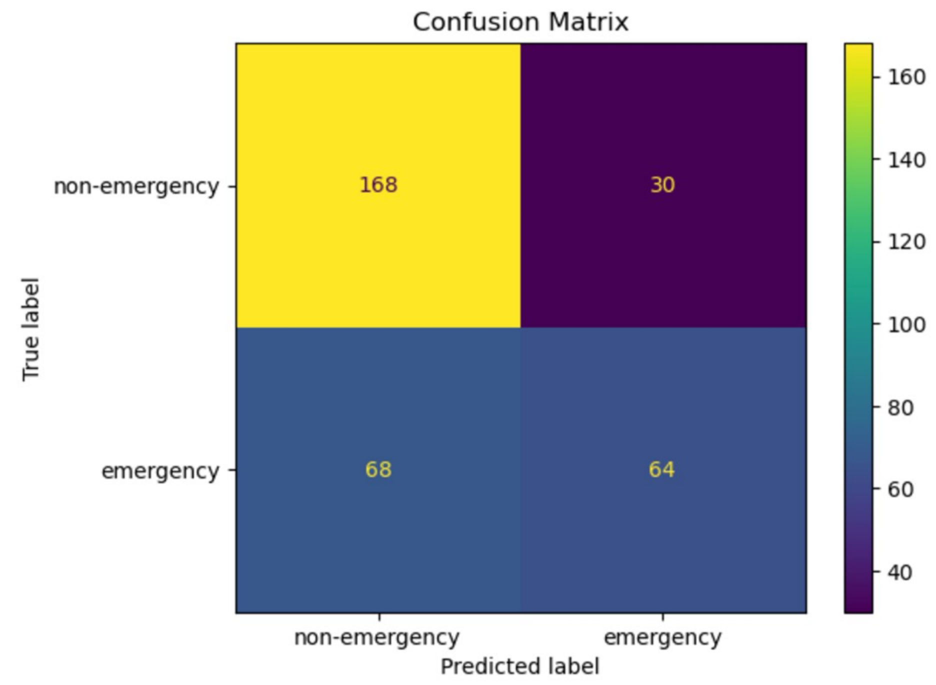

Figure 8: Confusion Matrix of Model 2

Table 4: Model 2 Evaluation Metrics

	precision	recall	f1-score	support
0	**0.71**	**0.85**	**0.77**	**198**
1	0.68	0.48	0.57	132
accuracy			0.70	330
macro avg	0.70	0.67	0.67	330
weighted avg	0.70	0.70	0.69	330

In Table 4 the model performs above well on the non-emergency class (Class 0) but struggles with the emergency class (Class 1), evident from its lower recall of 0.57. This suggests that the model is missing many actual emergency cases (false negatives)

Architecture of the CNN Model 3 Table 5

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu
Kernel size	3,3
Layer	MaxPooling
dropout rate	30%
Padding	Same
output layer	1 using sigmoid activation function

input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	128
optimizer	SGD (learning rate = 0.001)
Batch size	128
Epochs	20

Figure 9: Training and Validation Loss of Model 3

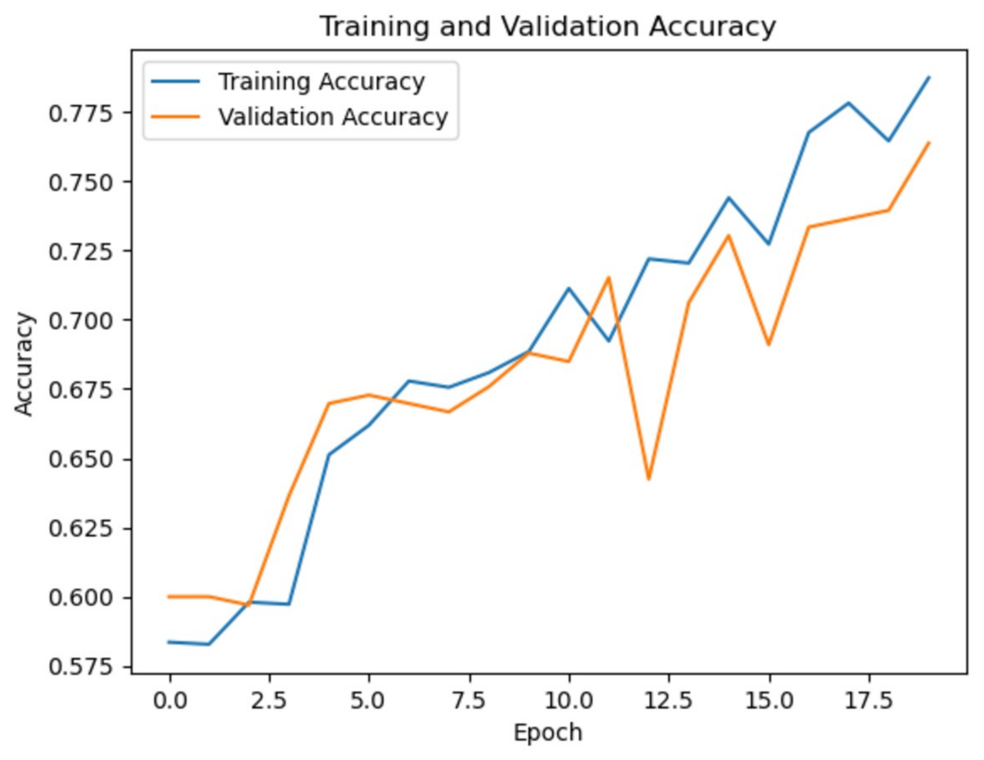

Figure 10: Training and Validation Accuracy of Model 3

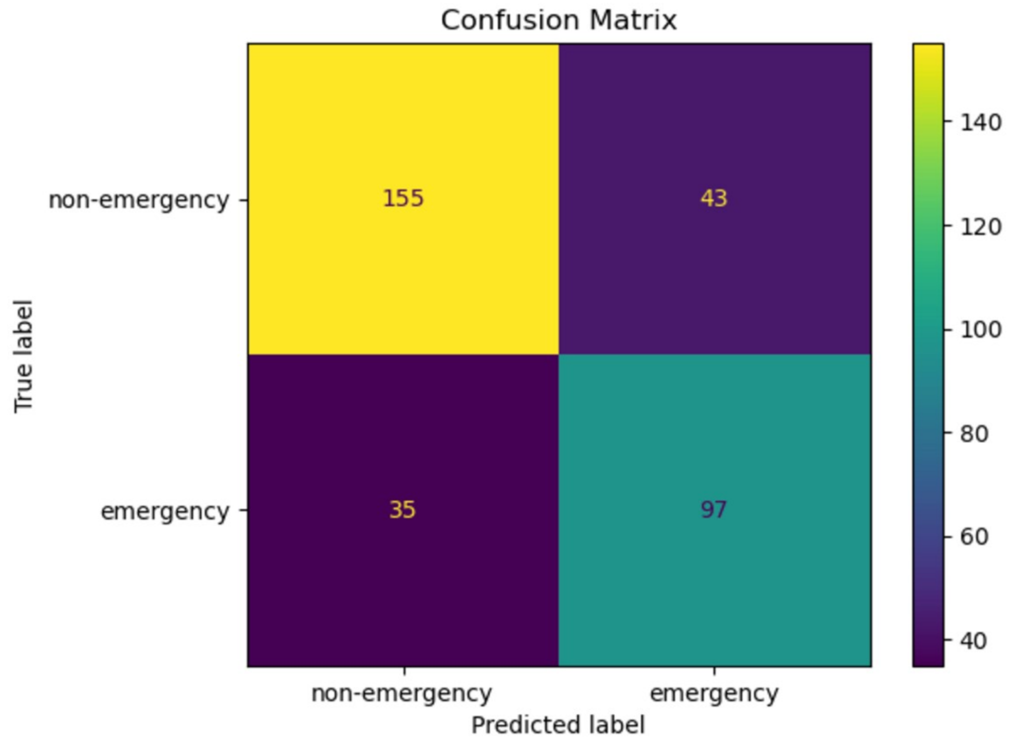

Figure 11: Confusion Matrix of Model 3 Table 6: Model 3 Evaluation Metrics

	precision	recall	f1-score	support
0	**0.82**	**0.78**	**0.80**	**198**
1	0.69	0.73	0.71	132
accuracy			0.76	330
macro avg	0.75	0.76	0.76	330
weighted avg	0.77	0.76	0.76	330

The model performs relatively well and shows a bit of balance on both the non-emergency class (Class 0) and the emergency class (Class 1), This this can still be improved.

Architecture of the CNN Model 4 Table 7

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu
Kernel size	5,5
Layer	MaxPooling
dropout rate	30%
Padding	Same
output layer	1 using sigmoid activation function
input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	128

optimizer	SGD (learning rate = 0.001)
Batch size	128
Epochs	20

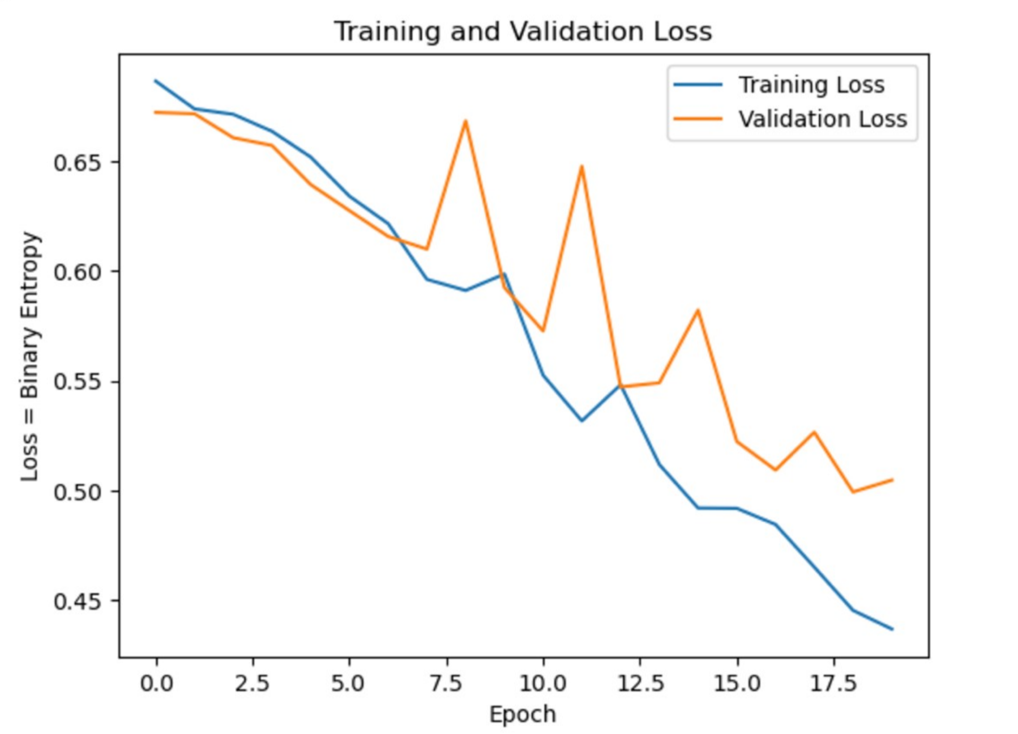

Figure 12: Training and Validation Loss of Model 4

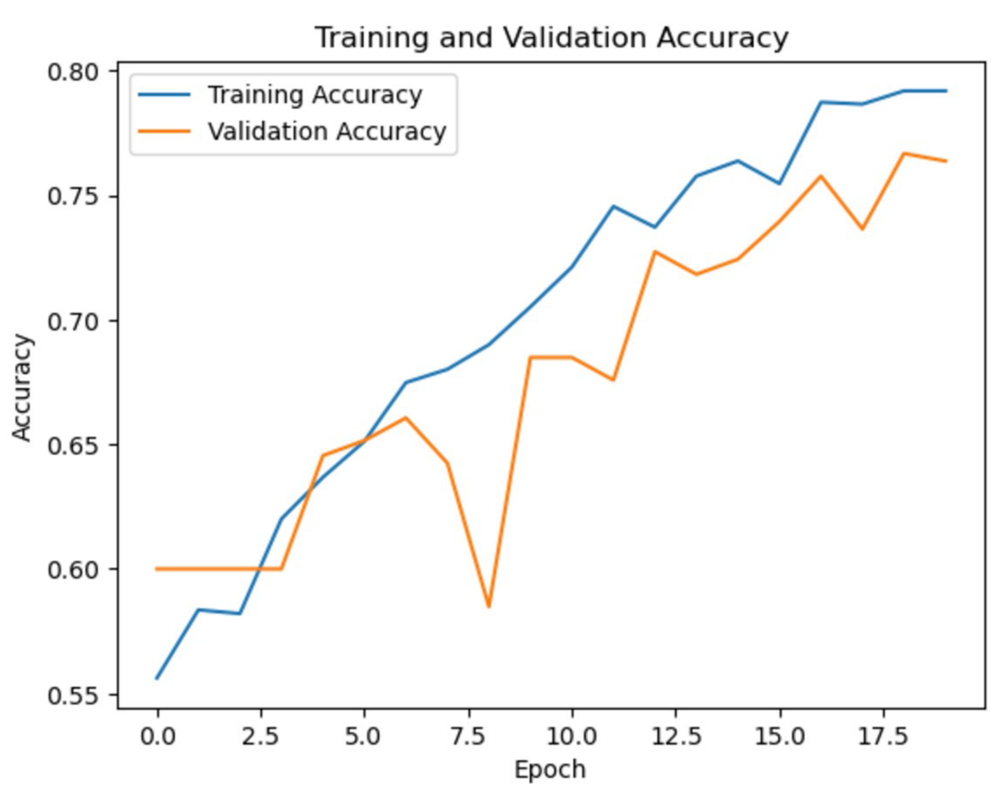

Figure 13: Training and Validation Accuracy of Model 4

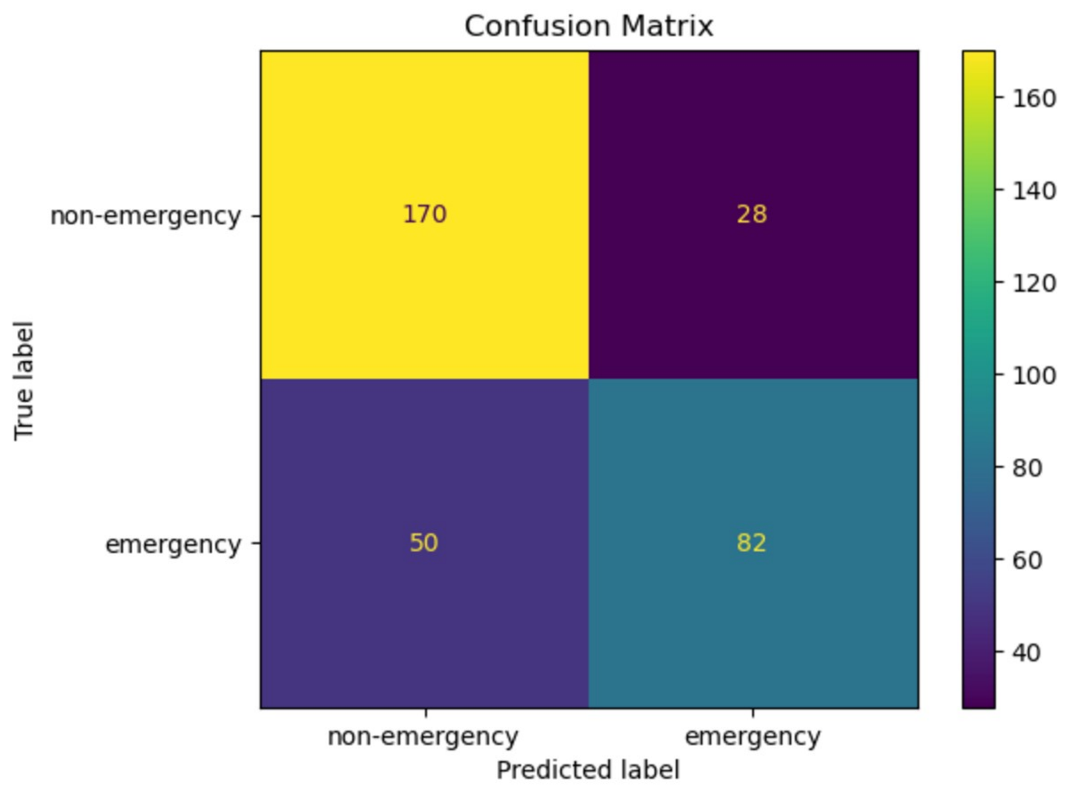

Figure 14: Confusion Matrix of Model 4

Architecture of the CNN Model 5 Table 8

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu
Kernel size	5,5
Hyperparameter tuning	BatchNormalization
Layer	AveragePooling
dropout rate	30%
Padding	Same
output layer	1 using sigmoid activation function
input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	128
optimizer	SGD (learning rate = 0.001)
Batch size	128
Epochs	20

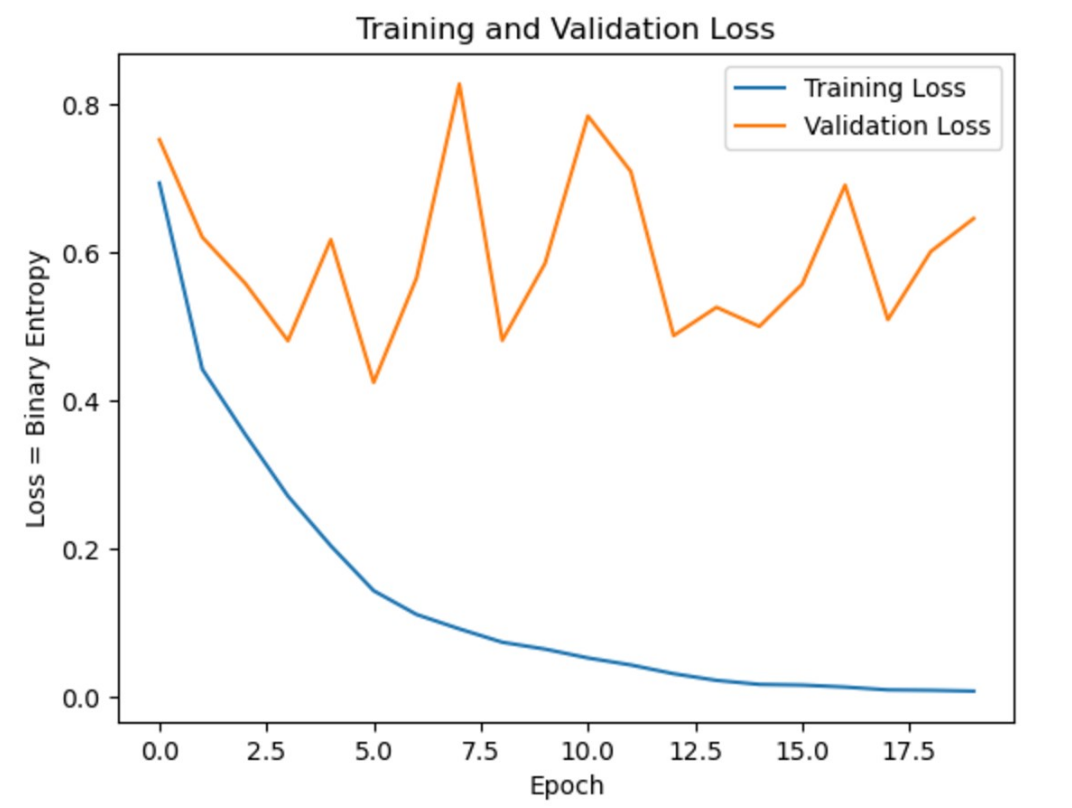

Figure 18: Training and Validation Loss of Model 5

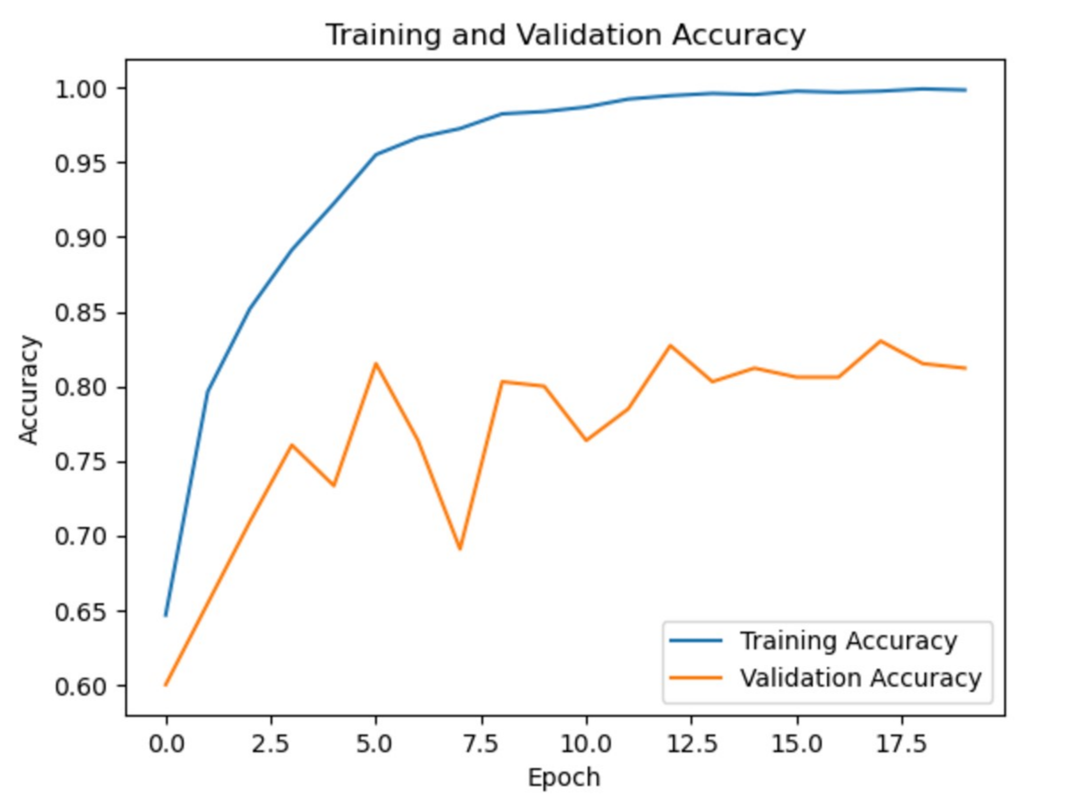

Figure 19: Training and Validation Accuracy of Model 5

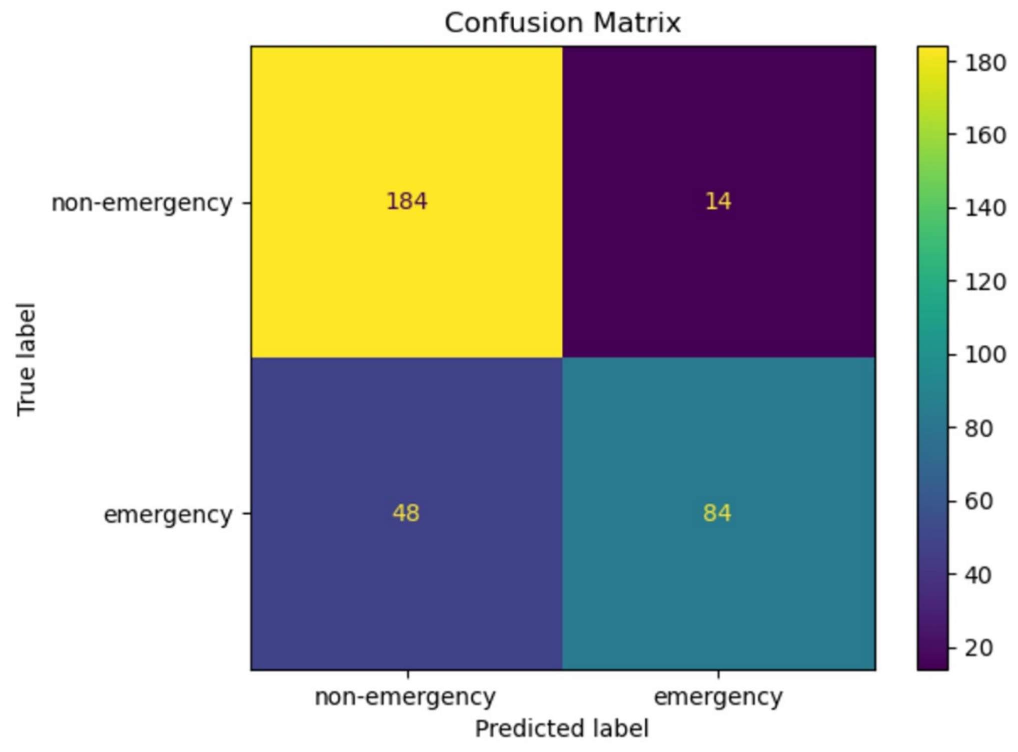

Figure 20: Confusion Matrix of Model 5 Table 9: Model 5 Evaluation Metrics

	precision	recall	f1-score	support
0	**0.79**	**0.93**	**0.86**	**198**
1	0.86	0.64	0.73	132
accuracy			0.81	330
macro avg	0.83	0.78	0.79	330
weighted avg	0.82	0.81	0.81	330

To resolve the under sampling of the emergency class batch normalization was introduced to this model the model performs and identifies the non-emergency class (Class 0) better than the emergency class (Class 1) still.

Architecture of the CNN Model 6 Table 10

Parameters	**Details**
Hidden Layers	3
neurons for each layer respectively	32,64,128
activation function	relu
Kernel size	5,5
Hyperparameter tuning	BatchNormalization
Layer	AveragePooling
dropout rate	30%
Padding	Same

output layer	1 using sigmoid activation function
input layer shape	(32, 32, 3)
Fully Connected (Dense) Layer	64
optimizer	SGD (learning rate = 0.001)
Batch size	128
Epochs	20

Figure 19: Training and Validation Loss of Model 6

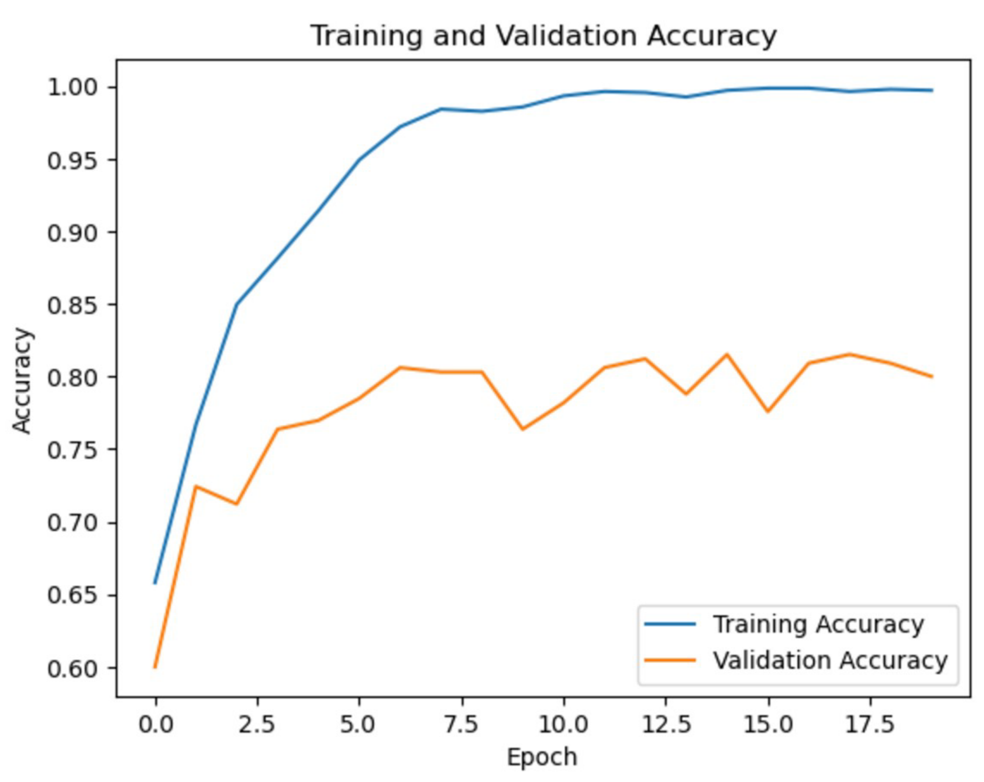

Figure 19: Training and Validation Accuracy of Model 6

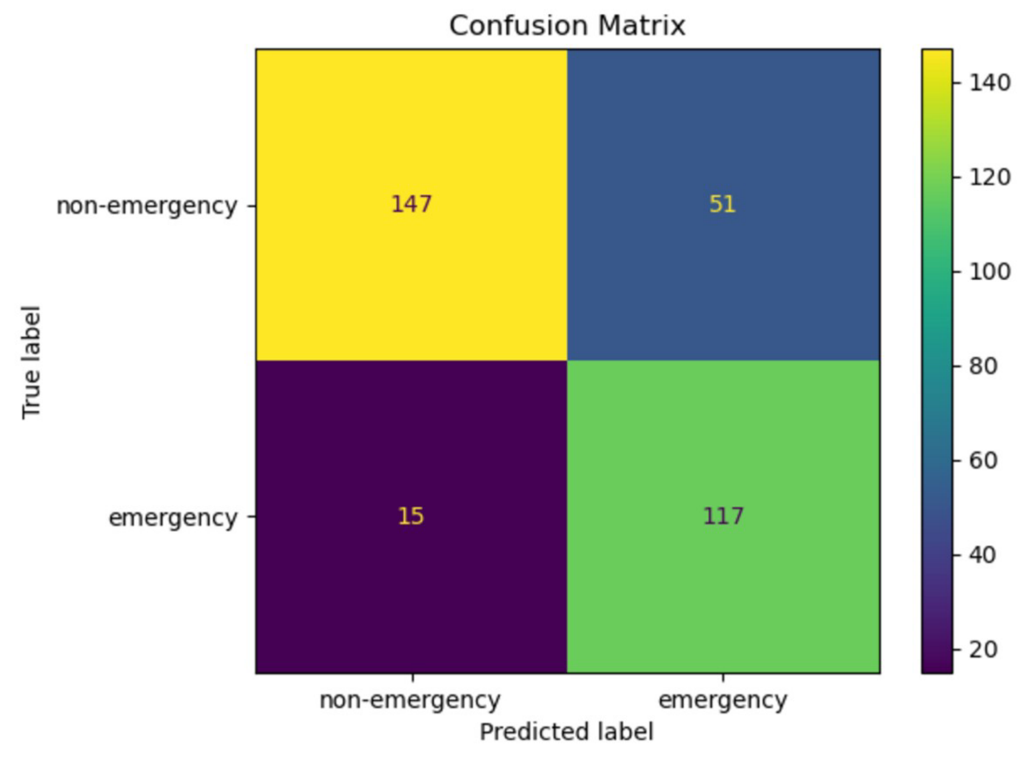

Figure 19: Confusion Matrix of Model 6 Table
11: Model 3 Evaluation Metrics

	precision	recall	f1-score	support
0	**0.91**	**0.74**	**0.82**	**198**
1	0.70	0.89	0.78	132
accuracy			0.80	330
macro avg	0.80	0.81	0.80	330
weighted avg	0.82	0.80	0.80	330

Reducing the fully connected layer to 64 improved the performance of the model, the model correctly classified 80% of cases, indicating good overall performance, however our model can predict the non- emergency class better and the emergency class is also more balanced with our result.

DISCUSSION

The application of Convolutional Neural Networks (CNNs) for binary image classification produced notable results across different models. Through modification of the architecture and hyperparameter tuning, the models reached a peak accuracy of 80%, with Model 6 exhibiting the best performance. The analysis reveals the struggles associated with class imbalance and under sampling, as the non-emergency class (Class 0) consistently outperformed the emergency class (Class 1) in terms of precision, recall, and F1-scores. Introducing batch normalization and modifying fully connected layers enhanced model performance and helped in identifying emergency vehicles better. However, the emergency class continues to face a higher risk of misclassification, as indicated by lower recall values across various models, which points to the model's failure to identify certain genuine emergencies (false negatives). This highlights the necessity for further investigation into methods such as oversampling, refined data augmentation, or ensemble techniques to address this challenge.

CONCLUSION

The analysis successfully developed CNN models for the classification of emergency and non-emergency vehicles. The final model exhibited strong overall performance, especially in distinguishing non-emergency vehicles, while also enhancing balance in identifying emergency vehicles. These findings illustrate the useability of CNNs for realtime detection of emergency vehicles but also indicate areas for additional refinement to achieve greater sensitivity in emergency classification.

REFERENCES

Krizhevsky, A., Sutskever, I., & Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, [online] 60(6), pp.84–90. Available at: https://doi.org/10.1145/3065386

Sharma, N., Jain, V. & Mishra, A., 2018. An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, 132, pp.377-384. Available at: https://doi.org/10.1016/j.procs.2018.05.198.