International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 77-82, January-2025

https://doi.org/10.47001/IRJIET/2025.901010

Predicting Vehicle Collision Using Transformer Network with Multi-Modal Data

[1]Kalindu Sekarage, [2]Dr. A.L.A.R.R. Thanuja

[1]Department of Computational Mathematics, Faculty of Information Technology, University of Moratuwa, Sri Lanka

[2]University of North Carolina at Greensbro, USA

Authors E-mail: [1]kalindu.sekarage@gmail.com, [2]r_athuru@uncg.edu

***Abstract* - Road accidents pose a significant threat to human life, causing numerous injuries, fatalities, and economic damage worldwide. Recently, there has been growing interest in leveraging Artificial Intelligence (AI) to create systems that can predict vehicle crashes. This research focuses on vehicle collision prediction and aims to develop a solution combining pre-trained Convolutional Neural Networks (CNN) and transformer network to mitigate the occurrence of such accidents. By leveraging advanced deep learning techniques, this research addresses the limitations of traditional crash analysis methods. The Car Learning to Act (CARLA) simulator was used for data gathering, with an ego-vehicle attached with RGB and RGB-Depth cameras. Four pre-trained CNNs were used for feature extraction. With those extracted features, a transformer network was employed to train a model. After model training and testing, it was observed that the transformer model trained with VGG16-based feature extraction performs better than other methods.**

Keywords: CNN, Transformer Network, CARLA, Feature Extraction.

I. INTRODUCTION

Road accidents have become a major contributor to injuries, fatalities, and property damage worldwide in recent years. According to the World Health Organization (WHO), around 1.35 million individuals annually lose their lives to automobile accidents [1]. Apart from causing damage to living persons, road accidents are also responsible for damage to property and infrastructure. Motor vehicle manufacturers are currently trying to make vehicles fully autonomous. Autonomous vehicles have the potential to dramatically reduce traffic accidents by removing the human factor in traffic accidents. However, Autonomous Vehicles are still not able to prevent traffic accidents entirely. A Convolutional Neural Network (CNN) is a specific form of deep learning method that specializes in image recognition and classification tasks. A key advantage of a CNN is its ability to simultaneously learn feature extraction layers and the classification layer, resulting in a model output that is highly structured and strongly reliant on the extracted features [2]. Transformer networks have significantly transformed Natural Language Processing (NLP) domain with the introduction of the self-attention mechanism, considerably enhancing their capabilities [3]. This mechanism allows the network to obtain global interdependencies and effectively process sequences of data, making transformers suitable for analyzing sequential data like image sequences. Integrating CNN and transformer networks in vehicle crash prediction systems holds great potential for improving prediction accuracy. By leveraging CNN's ability to extract features from images without requiring manual feature engineering or domain knowledge and the sequence modeling capabilities of transformer networks, it becomes possible to detect specific patterns and features in image sequences that may indicate a potential vehicle crash. This allows for proactive measures to be taken, such as issuing warnings to drivers or triggering automatic safety mechanisms. In this thesis, we aim to explore and develop a vehicle crash prediction system utilizing pre-trained CNN and transformer networks. The primary objective is to design a system that can effectively analyze a sequence of multimodal images captured within a vehicle and provide a binary classification of whether the oncoming situation is safe or unsafe. By accurately predicting potential crashes in real-time, this system has the capacity to improve road safety and prevent accidents significantly

II. LITERATURE REVIEW

AI can be used to predict vehicle accidents using driving and vehicle data. Some of the AI techniques used for crash prediction include Neural Networks, Support Vector Machine (SVM), Decision Trees, Fuzzy Logic, and Genetic Algorithms [4]. The paper [5] addresses the issue of safety concerns surrounding autonomous vehicles and proposes a Crash Prediction Network (CPN) as a potential solution. The text aims to propose a solution for improving safety in autonomous vehicles by using an ensemble of neural networks called CPN to supervise decision-making modules. Experiments were carried out using the CARLA 0.9.6 simulator, emphasizing preventing locally avoidable catastrophes. The issue of

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 77-82, January-2025

https://doi.org/10.47001/IRJIET/2025.901010

predicting accidents in dashcam videos was explored in the paper [6]. The objective of the paper is to propose a method that utilizes a Dynamic-Spatial-Attention Recurrent Neural Network (DSA-RNN) to predict road vehicle accidents in dashcam videos. In paper [7] the objective is to design and construct a vehicle collision detection system by utilizing a combined deep-learning model that utilizes various types of data from dashboard cameras. Vehicle collision detection has achieved new standards with the use of ensemble deep learning models that incorporate multimodal inputs, surpassing the performance of existing models. This study has found that audio features and spectrograms are more effective for identifying car collisions than dashboard camera visuals. Combining audio and video data improves performance.

The "Attention is All You Need" paper [8] introduces the Transformer model architecture, which discards recurrence and relies solely on an attention mechanism to capture global interdependence between input and output. This transformer model is able to address the limitation in RNN. The paper [9] introduces an attention-based hierarchical deep reinforcement learning approach for modeling lane change behaviors in autonomous driving. They incorporated temporal and spatial attention mechanisms into the deep reinforcement learning architecture, which improved the vehicle's ability to focus on nearby vehicles, resulting in smoother and more efficient lane change behavior.

III. METHODOLOGY

A. High-level Architecture

Data is collected using the CARLA simulator, where RGB and RGB-D cameras are attached to the ego-vehicle and drive around the simulated city to capture safe and unsafe situations. Then, captured images will be resized and normalized to enhance the model's generalization capabilities. Preprocessed data is subsequently divided into training and testing sets. Then, the training set is used to train the transformer network with features extracted with CNN. Pre-trained CNN is used for feature extraction and transformer network for training and classification. Classification is binary whether it is safe or unsafe. Fig. 1 shows the high-level architecture diagram of the design.

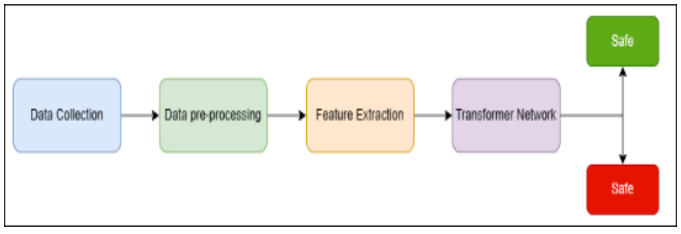

Figure 1: High-level Architecture of the Design

B. Data Collection

In order to gather a diverse and comprehensive dataset for training the transformer model, a simulation environment was meticulously created CARLA simulator. This virtual setting featured a fleet of 100 autonomous vehicles, ensuring a wide array of vehicles to be captured by the model and their driving behaviors and scenarios. To make the environment more challenging, another 20 pedestrians were integrated into the simulation. This enhances the realism and complexity of the traffic conditions that are captured by RGB and RGB-D cameras.

This simulation did not just focus on one default map, a total of four maps were selected, which are provided by CARLA, as shows in Fig. 2. The range of maps represents different urban layouts and road networks, each with its unique challenges.

Figure 2: Example of maps in CARLA simulator

C. Data Pre-processing

Collected RGB and RGB-D images cannot be directly input into a pre-trained CNN model; before that, we need to do some pre-processing. First, images must be read with the Python CV2 library and then converted into RGB format because CV2 will read images in BGR format by default. As the next step, again using the CV2 library, the image is resized into a 224 x 224 dimension. Resizing is necessary because it is a requirement of the architecture of the pre-trained CNN model. Resizing also gives some added benefits, such as training on a smaller, uniform image size, which can lead to faster computations and less memory usage during training. After resizing, we get an array of 224 x 224 x 3; 3 is the number of channels in the image. That array needs to be normalized before feeding into pre-trained CNN by dividing it 255; this will make values in the array between 0 and 1. The benefits of doing so are neural networks often perform better and cover faster when the input features are scaled to a smaller consistent range, it gives uniformity in pixel values, and during backpropagation, having smaller normalized values helps in maintaining a more stable gradient flow.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Issue 1, pp 77-82, January-2025
https://doi.org/10.47001/IRJIET/2025.901010

D. Feature Extraction

To extract features, RGB and RGB-D images were separately fed into pre-trained CNN models, and the extracted features were then concatenated. This technique is concatenated fusion, which comes under early-level fusion [10], and Fig. 3 shows its diagram. For pre-trained models, we have used VGG16, MobileNet, ResNet50, and DenseNet121.

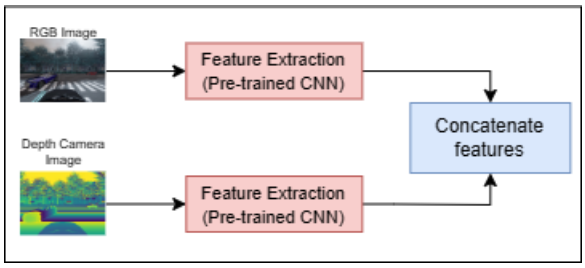

Figure 3: Extracted feature concatenation

E. Transformer Network

In our research, we will use only the encoder block of the transformer network since it is a classification task given a sequence of inputs. Our training input feature array and label array will be fed in. In modeling the temporal dependencies among the extracted features from an image sequence, a transformer architecture can be highly effective. This architecture utilizes self-attention mechanisms between different frames in the sequence. Unlike traditional models that may only capture local interactions, self-attention allows the model to weigh the importance of all frames, regardless of their position in the sequence. Additionally, positional encoding is integrated into the transformer network to encode the temporal order of the images. The TensorFlow library was used with Keras to code the transformer architecture. Fig. 4 shows the transformer encoder-only architecture used in our model training.

The training of this model was conducted with a batch size of 16. This particular size achieves a good trade-off between computational efficiency and model performance. It ensures that there is enough data for each iteration without using excessive memory. The training was scheduled to run for 100 epochs. To mitigate overfitting and enhance the model's ability to perform well on new, unknown data, early stopping was incorporated into the training process. The early stopping mechanism was set with a patience of 20 epochs. This implies that the training process would stop if there is no improvement in the validation loss over a continuous period of 20 epochs. A dropout rate of 0.5 was implemented during the training phase. Dropout is a regularization method employed to mitigate overfitting by randomly deactivating a portion of the neurons to zero during the training process. Within this configuration, a random selection of half of the units is eliminated after each training cycle.

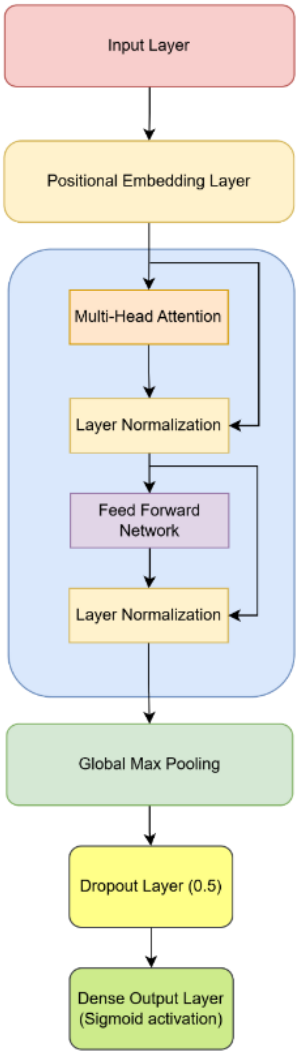

Figure 4: Transformer architecture

The Adam optimizer was chosen due to its adjustable learning rate capabilities, which enhance its efficiency in converging inside intricate landscapes commonly encountered in deep-learning models that process high-dimensional input, such as photographs. The utilized loss function was sparse categorical cross-entropy.

IV. RESULT AND DISCUSSION

A. Evaluation of Feature Extraction Methods

This section presents the comparative analysis of different feature extraction techniques employed within our transformer-based model framework to assess their impact on performance. Specifically, we have utilized four distinct pre-trained CNNs, VGG16, MobileNet, ResNet50, and DenseNet121, as the feature extractors.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 77-82, January-2025

https://doi.org/10.47001/IRJIET/2025.901010

B. Transformer Network with VGG16

Both training and validation accuracy remained relatively high and close together through the training process, the left graph of Fig. 5. This suggests that the model generalizes well and is learning features that are relevant to unseen data.

When comparing the training loss and validation loss the right graph of Fig. 5, it is regularly observed that the training loss is generally smaller than the validation loss. The graph shows some volatility in validation loss, which could be common in smaller datasets or with certain optimization algorithms. The overall trend, however, does not indicate overfitting since validation loss does not increase as the training progresses.

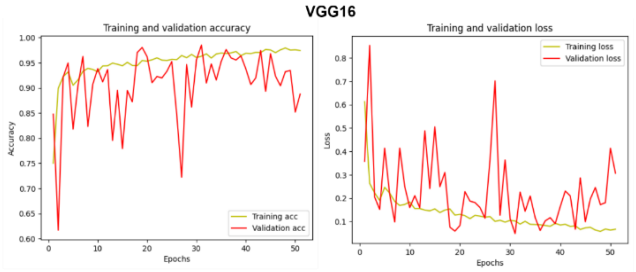

Figure 5: Accuracy and loss graphs for training vs validation for VGG16 based feature extraction

We got testing accuracy and an F1-score of 0.95, which is quite high, indicating that the model performed well on the task. The recall is 0.98 is especially good, meaning that the model is able to identify 98% of relevant instances. The precision of our model is 0.92, indicating a good level of accuracy in correctly predicting positive classes.

C. Transformer Network with MobileNet

Training and validation accuracy are quite close throughout the training process, with validation accuracy tracking slightly below training accuracy, which is expected (left graph of Fig. 6). There's a slight downward trend in validation accuracy toward the end of the epochs, which might be a sign of beginning overfitting, which was avoided due to early stopping or could be due to the variability of the validation set.

The training loss exhibits a consistent and gradual decline, indicating that the model is acquiring knowledge effectively. The validation loss exhibits variations, characterized by significant spikes, suggesting that the model has difficulties in generalizing to the validation set at certain instances. However, since the last value is lower, the model may not be overfitting (right graph of Fig. 6). The gap between training and validation loss is small towards the end, suggesting a decent generalization.

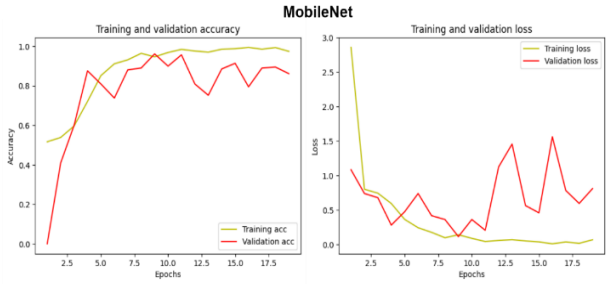

Figure 6: Accuracy and loss graphs for graining vs validation for MobileNet based feature extraction

For this method we got an accuracy of 0.92 and an F1-score of 0.92, which are strong indicators of a high-performing model. For precision, we go 0.88, which is slightly lower compared to the other metrics that indicates that when the trained model predicts a positive result, it is correct 88% of the time. The most valuable metric in our research, recall, got 0.97, which is excellent, indicating the model is identifying the relevant instances almost all the time.

D. Transformer Network with ResNet50

The training accuracy is quite stable but low, around 0.5, which is the accuracy of a random guess in binary classification tasks. The validation accuracy is extremely volatile, with perfect accuracy scores at certain epochs followed by a drop to 0. This pattern is not typical and suggests severe overfitting on a subset of the data or potential issues with the validation set or data processing (left graph of Fig. 7).

The training loss decreases, which is typical and indicates learning. However, the validation loss exhibits high variance, with very noticeable spikes. This erratic behavior in validation loss could indicate a problem with the extracted features (right graph of Fig. 7).

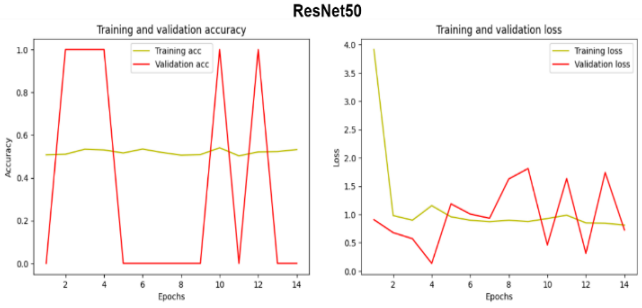

Figure 7: Accuracy and loss graphs for training vs validation for ResNet50 based feature extraction

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 1, pp 77-82, January-2025

https://doi.org/10.47001/IRJIET/2025.901010

When it comes to performance metrics, we got an accuracy and precision of 0.5, which indicates that the model is not better than random guessing. This is not a good sign for model performance, and the model is only able to predict one class. A recall of 1 suggests that the model is classifying all positive instances correctly, but given the low precision, it is likely also classifying many negative instances as positives, which is typical for models that predict only one class. An F1-score of 0.66 is not particularly high and suggests an imbalance between precision and recall in this case skewed towards recall.

E. Transformer Network with DenseNet121

The training accuracy consistently exhibits a high and steady performance, indicating that the model has effectively acquired knowledge from the training data. The validation accuracy has a consistent rising trajectory, remaining in close proximity to the training accuracy. However, there is a noticeable decline that aligns with the sudden increase in validation loss, as seen in the left graph of Fig. 8.

The training loss shows a general downward trend, which is good. The validation loss decreases overall but has a significant spike around epoch 14. This might suggest a batch of particularly challenging validation data or possible overfitting at that point (right graph of Fig. 8).

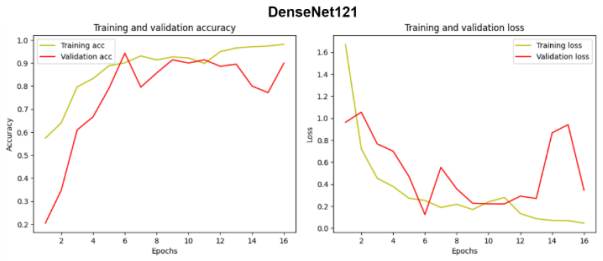

Figure 8: Accuracy and loss graphs for training vs validation for DenseNet121 based feature extraction

After evaluating the model, an accuracy of 0.93 and an F1-score of 0.93 were obtained, indicating that the model's predictions are generally reliable. For precision and recall, we got 0.91 and 0.95, respectively, which are both considerably high.

F. Consideration Beyond Raw Metrics

- **Computational Efficiency:** MobileNet is designed to be more efficient than VGG16 and DesnseNet121, which might be beneficial in real-time applications where resources are constrained.
- **Model Complexity:** Although VGG16 performs well, it is quite heavy in terms of parameters and computation. DenseNet121 also shares this trait but is more efficient in parameter usage due to its dense connectivity patterns.
- **Application-Specific Needs:** Depending on the application, the trade-off between recall and precision can be critical. In our research, missing a collision scenario is more problematic than false positives, so a model with higher recall is preferred.

G. Summary of Result

Out of all pre-trained CNN models used for feature extraction, VGG16 has the highest accuracy (0.95) and very high precision (0.92), and recall (0.98). This suggests that it is effective in both identifying collisions (high recall) and ensuring that detected collisions are likely real (high precision).

MobileNet shows a good balance but slightly lower performance metrics compared to VGG16 in all aspects. While still effective, it is a more lightweight model, which could be beneficial if computational efficiency is a priority.

ResNet50 shows significantly lower performance in accuracy and precision but a perfect recall. The high recall indicates it detects all collisions, but the low precision suggests many false positives. The very low accuracy hints at a potential feature extracted from ResNet50 that is not good enough for this task.

DensNet121 shows good accuracy and precision, slightly lower than VGG16 but better than MobileNet. It has a relatively high recall, indicating it also effectively identifies collision scenarios. Table 1 shows the summary of performance metrics for each feature extraction method.

Table 1: Table Summary of Performance Metrics in Percentage for Each Feature Extraction Method

Model	Test Accuracy	Precision	Recall	F1-Score
VGG16	95%	92%	98%	95%
MobileNet	92%	88%	97%	92%
RestNet50	50%	50%	100%	66%
DenseNet121	93%	91%	95%	93%

V. CONCLUSION

This research focuses on enhancing vehicular safety through the integration of advanced warning systems capable of predicting imminent collision with static or dynamic objects in the environment. Utilizing the capabilities of the transformer model, originally renowned for its exceptional performance in NLP, we extended their application to the domain of images for real-time crash prediction. The performance of the transformer model depends on the pre-trained CNN model used for feature extraction. One of the

ISSN (online): 2581-3048
Volume 9, Issue 1, pp 77-82, January-2025
https://doi.org/10.47001/IRJIET/2025.901010

primary objectives of this project is to augment conventional vehicles with an intelligence warning system. This system leverages both RGB and RGB-D cameras as input devices, enabling the perception of the vehicle's surroundings through color imagery and depth data. By processing these inputs through the transformer model, the system can accurately detect potential collision scenarios and alert the driver, thereby significantly enhancing road safety.

In the current implementation, features extracted separately from RGB and RGB-D images are concatenated to form a comprehensive feature set for collision prediction. While effective, this method primarily relies on simple concatenation, which may not fully capitalize on the potential synergies between the two types of data. For further work, a more sophisticated fusion method could be explored to enhance the integration of features derived from RGB and RGB-D inputs.

REFERENCES

[1] World Health Organization, Global status report on road safety 2018. *Geneva: World Health Organization*, 2018. Accessed: Jun. 03, 2023. [Online]. Available: https://apps.who.int/iris/handle/10665/276462

[2] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[3] A.Cheok and E. Zhang, From Turing to Transformers: A Comprehensive Review and Tutorial on the Evolution and Applications of Generative Transformer Models. 2023. doi: 10.32388/3NTOLQ.2.

[4] Z. Halim, R. Kalsoom, S. Bashir, and G. Abbas, "Artificial intelligence techniques for driving safety and vehicle crash prediction," *Artif. Intell. Rev.*, vol. 46, no. 3, pp. 351–387, Oct. 2016, doi: 10.1007/s10462-016-9467-9.

[5] A.P. Staff et al., "An Evaluation of ``Crash Prediction Networks'' (CPN) for Autonomous Driving Scenarios in CARLA Simulator".

[6] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating Accidents in Dashcam Videos," in Computer Vision – ACCV 2016, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., in Lecture Notes in Computer Science. *Cham: Springer International Publishing*, 2017, pp. 136–153. doi: 10.1007/978-3-319-54190-7_9.

[7] J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Expert Systems with Applications*, vol. 183, p. 115400, Nov. 2021, doi: 10.1016/j.eswa.2021.115400.

[8] A.Vaswani et al., "Attention is All you Need".

[9] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, "Attention-Based Hierarchical Deep Reinforcement Learning for Lane Change Behaviors in Autonomous Driving," *in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA: IEEE*, Jun. 2019, pp. 1326–1334. doi: 10.1109/CVPRW.2019.00172.

[10] M. Gao, J. Jiang, G. Zou, V. John, and Z. Liu, "RGB-D-Based Object Recognition Using Multimodal Convolutional Neural Networks: A Survey," *IEEE Access*, vol. 7, pp. 43110–43136, 2019, doi: 10.1109/ACCESS.2019.2907071.

Citation of this Article:

Kalindu Sekarage, & Dr. A.L.A.R.R. Thanuja. (2025). Predicting Vehicle Collision Using Transformer Network with Multi-Modal Data. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(1), 77-82. Article DOI https://doi.org/10.47001/IRJIET/2025.901010

****

