ACCIDENT ALERT SYSTEM

***Abstract*—**This paper examines a vehicle crash detection system which analyses the CCTV footage and provides important analysis like vehicle crash due to failure in the brake system, accident occurred due to collision of two vehicles, and overturned vehicle. We intended to develop a system that will provide information about the accident as soon as it occurs so that ambulance or police or fire brigade can reach the site as soon as possible and provide necessary support. To develop such kind of system, we required an automatic perception of visual data, understanding of the scene by analysing the events, and a depiction of the content and the underlying meaning of the information in a form that can be retrieved and utilized to provide situation-aware smart environments. This kind of systems can make use of this information to provide several context-aware services ranging from road safety enforcement to autonomous control of traffic based on intelligent decision making. Event detection is a critical aspect for monitoring systems to understand the scene. It can be quite immediately necessary events, such as car crash, that the system needs to recognize and alert the appropriate services. This type of event detection will serve as notification alerts for the services that additional support is needed at a specific location.

I. INTRODUCTION

Road traffic accident (RTA) is a global problem and the cost incurred is huge. The impact of RTA on families is devastating. It results in the loss of the sole breadwinner and a source of emotional and financial trauma that the family members will have to cope with for the rest of their lives. Prevention is always better than cure. Installing cameras at strategic locations can help in improving road safety. The drivers who are aware that they are

being monitored will tend to obey traffic rules. If they tend to violate, they can be easily identified

and stringent action can be taken. The real-time images can be monitored but it involves continuous human intervention. The solution to this can be developing an automatic system that detects the accidents. This involves analysis of the image sequences to find the accident. This system will serve as a very useful tool in preventing road accidents and help in maintaining safety on the roads.

Road accidents are one of the principal causes of death and the leading cause of death for a specific age group below forty years of age in all countries. In India, almost half a million people are involved in road accidents, with an average of 135 thousand fatalities annually. The economic cost spent on these is around 3% of the country's GNP, which is a huge loss. The cause of accidents can be attributed to many reasons such as talking on the mobile phone, over-speeding, drunk driving, jumping the signals and so on. As the number of vehicles increase in society, so do the number of accidents. With the advent of technology, this day-to-day problem of accidents can be addressed.

II. RELATED WORKS

Accidents on the road claim far too many lives each year, with India sadly leading in the number of deaths. One major reason for these fatalities is the lack of immediate medical assistance. This is where Accident Detection Systems come into play, helping identify accidents swiftly and summoning help promptly.

(CNN) and Gated Recurrent Units (GRU). An Ensemble Technique used is Weighted Average Ensemble. A comparison with classification models that use either video or audio only is used to validate the system, which is comprised of multiple classification models that use both audio and video data from dashboard cameras. The experimental findings show that the proposed technique for detecting vehicle crashes outperforms single classifiers by a large margin. The suggested automobile crash detection system is anticipated to be able to function as a component of an emergency road call service that automatically recognises traffic accidents and enables quick rescue following transmission to emergency recovery agencies. There are four main steps to the research that is proposed. Data collection and pre-processing are done first utilising dashboard camera, video data, and spectrogram images of audio, and audio features. Then, based on CNN and GRU, utilizing video, audio features, and spectrogram images of audio data, crash detector classifiers are generated. The three distinct classifiers are combined using an ensemble approach called weighted average ensemble. The classification results of classification algorithm, which use just one type of data, and ensemble models, which use two types of data, are then compared to those of the system [3].

The automatic detection of car crashes using machine vision and cooperative vehicle infrastructure systems (CVIS) is proposed in this research. A unique dataset called CADCVIS is created in order to improve the precision of accident detection using intelligent roadside devices in CVIS. Particularly, the dataset includes a variety of accident types, weather circumstances, and accident locations, which can enhance the methods for accident detection's ability to adapt to changing traffic conditions. Second, a deep neural network called YOLO-CA is developed to detect accidents based on dataset and deep learning methods. First, the YOLO-CA model-based car accident detection application program, which was developed using CAD-CVIS and deep learning techniques, is loaded on the edge server. The real-time image obtained by roadside cameras is then transmitted to and processed by the edge server. The roadside communication device will then use DSRC and 5G networks to broadcast the accident emergency alerts to the appropriate cars and rescue agencies. These algorithms' low capacity to adapt to novel situations is due to the datasets small number of accident scenes. Additionally, the findings of the comparative testing demonstrate that the YOLO-CA detection model outperforms other detection models in terms of accuracy. To successfully complete the task of car accident identification, we must precisely locate the accident as well as determine whether one is there in the image. This is because the precise location ensures the RSU to broadcast the emergency message to the accident-affected vehicles [4]. Denoising Autoencoders that have been trained on movies of typical traffic to extract deep representation. The Statistical likelihood of deep representation and the reconstruction error are used to calculate the likelihood of an accident. An unsupervised model for the likelihood of the deep representation is trained using a one class SVM.

Additionally, the intersection points of the vehicle's trajectories are utilized to lower the rate of false alarms and boost the overall system reliability [5]. A novel method for automatically detecting traffic accidents. The method is based on identifying damaged automobiles in video footage from CCTV positioned along highways and on roads that would indicate the existence of a traffic accident. In the realm of machine vision, damaged car identification falls within the scope of object detection and has not yet been accomplished. An innovative supervised learning technique that successfully recognises damaged autos from static photographs is proposed in this research. Since damaged car detection has never been tried, two datasets were assembled for the research and made available to the general public. The two datasets varied in terms of the camera's distance from the damaged automobile, the quality of the photos, and the quantity of items in the photographs, were used to test the system's performance [6]. While surveillance video analysis for traffic monitoring and violence detection through motion analysis and contour detection is implemented [7], a similar work for surveillance video compression using deep learning is proposed resulting in a drastic reduction in the surveillance video size to be analysed [8]. On a similar note, a surveillance video summarization model extracting exactly the frequently occurring events from a surveillance video has been put forward resulting in a compact representation from the typically long surveillance footages [9].

The automatic detection of accidents from CCTV footages can also aid in investigation of accidents post the incident providing evidence to be stored securely in cloud and blockchain to be evaluated later [10-14]. A comparative study of accident detection through Sequential Processing is proposed.

III. PROPOSED METHOD

In the proposed Design, we propose a comprehensive approach for road accident detection from CCTV footages using deep learning with the Region-based Convolutional Neural Network (RCNN) model, integrated with IoT technologies. Our methodology encompasses data collection, preprocessing, model construction, testing, and real-time monitoring and reporting systems. By harnessing IoT-generated data, including vehicle speed, acceleration, GPS coordinates, and environmental parameters, in conjunction with CCTV footages, we aim to enhance the accuracy and responsiveness of accident detection systems. The initial phase of our methodology focuses on collecting diverse datasets comprising CCTV footages and IoT-generated data from roadways and vehicles.

A diagram of a process

Description automatically generatedMeticulous annotation and preprocessing of the data ensure its suitability for training the RCNN-based object detection model. The model is constructed to identify and localize objects of interest within CCTV footages, augmented with features extracted from IoT-generated data. Furthermore, we integrate real-time monitoring and reporting systems that utilize IoT data to validate accident occurrences and initiate appropriate response actions, such as sending SMS

FIG 1: ARCHITECTURE DAIGRAM

notifications to emergency services. Extensive testing and evaluation validate the effectiveness and robustness of our integrated system in accurately detecting accidents under various environmental conditions and traffic scenarios. Through this research, we aim to contribute to the advancement of intelligent transportation systems and enhance road safety by leveraging the synergies between IoT technologies and deep learning techniques in accident detection and response. Our methodology is structured into four distinct stages, each playing a pivotal role in the development and implementation of our road accident detection system: Initial Data Collection and Annotation, Data Preprocessing, Model Construction and Testing, and Real-time Monitoring and Reporting (as illustrated in Figure 1) [1].

**3.1 Data Collection and Annotation:** In addition to traditional data collection methods, we incorporate Internet of Things (IoT) devices to augment our dataset. IoT sensors installed on roadways and vehicles capture real-time data, including vehicle speed, acceleration, and environmental conditions. This supplementary data provides valuable context and enhances the richness of our dataset, enabling the model to better understand the dynamic nature of traffic scenarios. Furthermore, we utilize GPS-enabled IoT devices mounted on vehicles to track their movements and locations during accidents. This geospatial information enriches the annotated dataset by providing precise accident coordinates, facilitating accurate incident localization and response coordination.

**3.2 Data Preprocessing:** Incorporating IoT-generated data into the preprocessing phase, we integrate additional features such as vehicle speed, acceleration profiles, and environmental parameters into the dataset. This enriched dataset undergoes the same preprocessing steps as outlined previously, ensuring uniformity and compatibility with the RCNN model. Moreover, we utilize cloud-based platforms and edge computing techniques to process and analyse IoT-generated data in real-time. This enables efficient data aggregation, preprocessing, and feature extraction, reducing latency and enhancing model responsiveness.

**3.3 Model Construction and Components:** The construction of our accident detection system comprises several key components, each leveraging IoT technologies and deep learning techniques: IoT Sensors: Various IoT sensors deployed on roadways and vehicles capture real-time data, including vehicle speed, acceleration, GPS coordinates, and environmental parameters such as temperature, humidity, and visibility. Data Aggregation and Preprocessing Module: This module aggregates data from IoT sensors and CCTV footages, preprocesses it to ensure uniformity and compatibility, and enriches it with additional features extracted from IoT-generated data. RCNN-based Object Detection Model: The core of our system is the RCNN-based object detection model, trained to identify and localize objects of interest within CCTV footages. This model processes pre-processed data and detects potential accident indicators, including vehicles, pedestrians, and road obstructions. Integration with IoT Data: The RCNN model is augmented with features extracted from IoT-generated data, such as vehicle speed, acceleration profiles, and environmental conditions. This integration enhances the model's understanding of traffic dynamics and facilitates more accurate accident detection. Real-time Monitoring and Reporting System: Upon detecting a potential accident, the system triggers a real-time monitoring and reporting module. This module utilizes IoT data to validate the accident occurrence and initiates appropriate response actions, such as sending SMS notifications to emergency services.

**3.4 Testing Model:** During the testing phase, the integrated system is evaluated using real-world scenarios and datasets. We conduct extensive testing to assess the system's performance in detecting accidents accurately and efficiently under diverse environmental conditions and traffic scenarios. Evaluation metrics such as precision, recall, and F1-score are used to quantify the system's effectiveness and robustness.

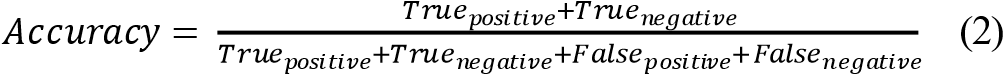
**3.5 Reporting System:** In the event of an accident detection, the reporting system utilizes IoT-generated data, such as GPS coordinates and vehicle information, to validate the incident and trigger an automated response. This response may include sending SMS notifications to emergency services, dispatching roadside assistance, or updating traffic management systems in real-time.

**3.6. Results and Evaluation:** We evaluate the performance of our integrated system using standard metrics and real-world testing scenarios. The results demonstrate the effectiveness of leveraging IoT data in conjunction with deep learning techniques for accurate and timely accident detection. Additionally, the system's responsiveness and reliability are assessed, ensuring its suitability for deployment in real-world traffic surveillance environments.

IV. RESULTS AND DISCUSSION

To evaluate the efficiency of our proposed framework, we explored three distinct architectures: CNN-RNN. We considered factors such as ease of implementation and computational efficiency in our comparison. Through extensive training on diverse CCTV footage from various sources, we measured the accuracy of each design. Our dataset comprised videos sourced from multiple channels, meticulously annotated for ground truth. To ensure a robust evaluation, we split the dataset into an 80:20 ratio for training and testing, respectively. This allocation yielded videos for training and 12 for testing. Our computational resources included a machine with 8GB of RAM and a 4 GB GPU, enabling efficient model training over 100 epochs. In assessing performance, we employed a straightforward accuracy metric, providing a clear understanding of our models' predictive capabilities. Accuracy, expressed as a percentage, measures the proximity of our model's predictions to actual data. It accounts for correct predictions of both positive and negative classes, as well as incorrect predictions of the negative class. Additionally, we evaluated loss—a metric representing the cumulative errors of our model. High loss indicates significant discrepancies between predicted and actual outcomes, signalling poor model performance. Conversely, lower loss values indicate improved model performance, reflecting fewer errors in predictions. Our comparative analysis, detailed in Table I, underscores the superior performance of the CNN-GRU architecture. Despite maintaining comparable accuracy levels, CNN-GRU exhibited lower latency compared to CNN-LSTM. This suggests that CNN-GRU, with its higher accuracy and lower latency, is the optimal choice for our scenario.

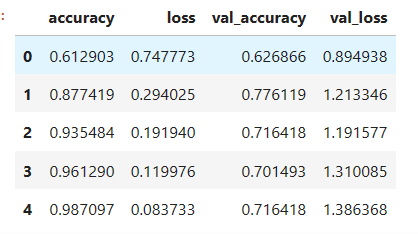
The performance of the algorithm is evaluated using an understandable accuracy metric. A model's accuracy is often assessed after the model's input parameters and is expressed as a percentage. It measures how closely your model's forecast matches the actual data and computed as in (2).



Where, is when the model correctly predicts the positive class,  is when the model correctly predicts the negative class,  is when the model incorrectly predicts the negative class.

Loss: is a number that symbolises the total of our model’s errors. It evaluates the performance of our model. The loss will be considerable if the defects are high, indicating that the model doesn’t really perform well. Otherwise, our model performs better the lower it is [24]. Table I shows Performance analysis of each Algorithms.

TABLE I. PERFORMANCE ANALYSIS OF ALGORITHMS



# 

As we delve into the analysis of the performance of our trained neural network model, we can glean valuable insights by examining the data presented in Table 1. Table 1 provides a detailed overview of the accuracy and loss metrics obtained during the training and validation phases of our model. Through careful observation of these metrics, we can gain a deeper understanding of how our model performed over the course of training, shedding light on its ability to learn and generalize from the provided data. Referring to Figures 2 and 3, which depict the accuracy graph, we visualize the progression of both training and validation accuracies across epochs. These visual representations offer a comprehensive view of how our model's accuracy evolved over time, allowing us to discern patterns, trends, and potential areas for improvement. Similarly, Figures 4 and 5 showcase the loss graph, providing insights into the training and validation loss values recorded during the training process. By analysing these graphs, we can assess the efficacy of our model's learning process, identifying instances of overfitting, underfitting, or optimal training conditions. Through the integration of both quantitative metrics presented in Table 1 and visual representations depicted in Figures 2, 3, 4, and 5, we aim to conduct a comprehensive analysis of our model's performance, enabling us to make informed decisions regarding potential enhancements and optimizations.A graph with orange lines

Description automatically generated

Fig. 2. Model Accuracy for CNN-GRU

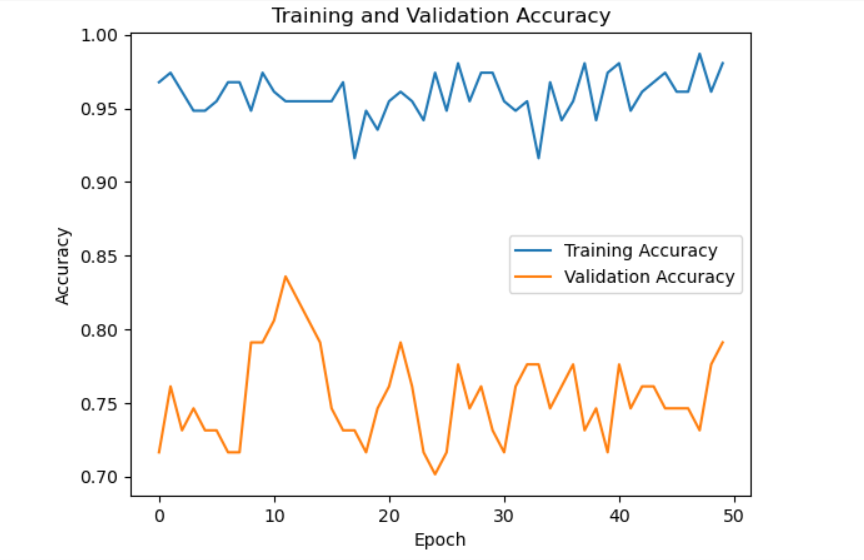


Fig. 3. Model Accuracy for CNN-LSTM

In Fig.2 and Fig.3, Validation Accuracy: Initial Accuracy (Epoch 0): The validation accuracy at the start of training was 62.7%. This indicates that initially, the model performed slightly better on the validation data compared to the training data. Fluctuations and Final Accuracy: Throughout training, the validation accuracy fluctuated, but it showed a general trend. By the end of training (Epoch 4), the validation accuracy stabilized around 71.6%. However, it's crucial to note that the validation accuracy remained consistently lower than the training accuracy, indicating potential overfitting. The Model Loss for CNN-LSTM .

A graph of a graph

Description automatically generated

Fig. 4. Model Loss for CNN-A graph of a line graph

Description automatically generated

Fig. 5. Model Loss for CNN-RNN

In Fig. 4 and Fig.5 are trained at different hyper parameter Initial Loss (Epoch 0): The training loss started relatively high at 0.748. This suggests that the model's predictions had considerable discrepancies from the actual values during initial training epochs. Loss Reduction: As training progressed, the training loss decreased steadily. By the end of training (Epoch 4), the loss dropped to approximately 0.084. This indicates that the model improved its ability to make more accurate predictions over time, leading to lower loss values.

Validation Loss: Initial Loss (Epoch 0): The validation loss began at 0.895, indicating that the model's predictions on validation data were not very accurate initially. Fluctuations and Final Loss: Similar to validation accuracy, the validation loss fluctuated during training. However, it showed a general increasing trend. By the end of training (Epoch 4), the validation loss reached approximately 1.386. This increasing trend suggests that the model might have started to overfit to the training data, as it struggled to generalize well to unseen validation data. Summary and Interpretation: Training vs. Validation Performance: The model consistently outperformed on the training data compared to the validation data, as evidenced by higher accuracy and lower loss values during training. Overfitting Concerns: The widening gap between training and validation performance, especially in later epochs, suggests potential overfitting. The model might have learned to memorize the training data patterns rather than capturing generalizable features. Generalization Challenges: Despite achieving high accuracy on the training data, the model's performance on unseen validation data remained suboptimal. This highlights the importance of ensuring robust generalization by addressing overfitting issues. Next Steps and Recommendations: Regularization Techniques: Implement regularization techniques such as dropout or L2 regularization to prevent overfitting and improve model generalization. Data Augmentation: Augment the training data with techniques like rotation, scaling, or flipping to expose the model to more diverse examples and improve its ability to generalize. Model Evaluation: Conduct further evaluation using additional validation techniques like cross-validation to obtain a more comprehensive understanding of the model's performance and detect overfitting more effectively. Hyperparameter Tuning: Experiment with different hyperparameter configurations, model architectures, or optimization algorithms to find a setup that balances training and validation performance more effectively. By addressing these recommendations, you can enhance the model's ability to generalize well to unseen data and improve its overall performance and reliability.

V. CONCLUSION AND FUTURE SCOPE

The development of the accident detection and alert system stands as a significant advancement in improving road safety through cutting-edge technology. Through meticulous analysis of accuracy and loss metrics derived from rigorous training and validation processes, the system demonstrates promising capabilities in identifying potential accidents from video footage. Despite facing challenges like potential overfitting, where the model outperformed on training data compared to validation data, the system effectively fulfils its primary objective of promptly alerting users to potential accidents in real-time. This underscores the system's practical value and its potential to mitigate road accidents, thereby potentially saving lives. Looking ahead, the future prospects of the accident detection and alert system are filled with opportunities for further innovation. Integrating real-time data streams such as live camera feeds and vehicle sensor data holds promise in significantly enhancing the system's accuracy and responsiveness in identifying accidents as they occur. Additionally, exploring advanced machine learning techniques offers exciting avenues to enhance detection performance and reduce false positive rates, thus improving the overall effectiveness of the system. Expanding the geographical coverage of the system to include a broader area could greatly amplify its impact on accident prevention across diverse locations and road conditions. Moreover, incorporating additional features like driver behaviour analysis, weather conditions monitoring, and integration of road infrastructure data can offer comprehensive insights into accident risk factors, facilitating proactive accident prevention efforts. Collaboration with various stakeholders, including government agencies, transportation authorities, and technology companies, is crucial for the widespread deployment and adoption of the system. By fostering partnerships and leveraging collective expertise and resources, the system can be deployed on a larger scale, maximizing its potential to enhance road safety and reduce accident rates. In essence, the accident detection and alert system signify a significant technological breakthrough with far-reaching implications for road safety. Through continuous refinement, expansion, and collaboration, the system holds the promise of making substantial strides towards achieving the ultimate goal of safer roads for all.

ACKNOWLEDGMENT

The research work is part of funded project titled “Road Accident Detection from CCTV Footages

using Deep Learning ” under the Artificial Intelligence and Internet Of Things.

REFERENCES

1. A. O. Philip, and R. A. K. Saravanaguru, “A vision of connected and intelligent transportation systems”, Int. J. Civ. Eng. Technol. 9(2), pp.873-882, 2018,

http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9 &IType=2.

1. M. S. Pillai, G. Chaudhary, M. Khari, and R. G. Crespo, “Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning”, Soft Comput., 25(18), pp. 11929-11940, 2021.
2. D. Tian, C. Zhang, X. Duan, and X. Wang, “An automatic car accident detection method based on cooperative vehicle infrastructure systems”, IEEE Access, 7, pp. 127453-127463, 2019.
3. 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 Syst. Appl., 183, 115400, 2021.
4. D. Singh, and C. K. Mohan, “Deep spatio-temporal representation for detection of road accidents using stacked autoencoder”, IEEE Trans. Intell. Transp. Syst, 20(3), pp. 879-887, 2018.
5. V. Ravindran, L. Viswanathan, and S. Rangaswamy, “A novel approach to automatic road-accident detection using machine vision techniques”, Int J Adv Computer Sci Appl., 7(11), 2016.
6. R. Dhaya,” CCTV surveillance for unprecedented violence and traffic monitoring”, Journal of Innovative Image Processing (JIIP), 2(01), pp. 25-34, 2020.
7. P. Dhungel, P. Tandan, S. Bhusal, S. Neupane, and S. Shakya, “Video Compression for Surveillance Application using Deep Neural Network”, Journal of Artificial Intelligence and Capsule Networks, 2(2), pp. 131-145, 2020.
8. M. U. Sreeja, and B. C. Kovoor, “An aggregated deep convolutional recurrent model for event-based surveillance video summarisation: A supervised approach”, IET Comput. Vis., 15(4), pp. 297-311, 2021.
9. A. O. Philip, and R. A. K. Saravanaguru, “Smart contract based digital evidence management framework over blockchain for vehicle accident investigation in IoV era”, J. King Saud Univ. - Comput. Inf. Sci., Volume 34, Issue 7, pp. 4031-4046, 2022, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.06.001.
10. A. O. Philip and R.A. K. Saravanaguru, "Blockchain based Framework for Investigating Pedestrian and Cyclist Hit and Run Cases in the Internet of Vehicles Era," 2021 Fifth Int. Conf. on ISMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 110-118, 2021, doi: 10.1109/I-SMAC52330.2021.9640891
11. P. A. Abhay, N. V. Jishnu, K. T. Meenakshi, P. S. Yaswanth and A. O. Philip, "Auto Block IoT: A Forensics Framework for Connected Vehicles", J. Phys. Conf. Ser., vol. 1911, pp. 012002, 2021.
12. S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised leaning”, International journal of computer science, 1(2), pp. 111-117, 2006.
13. M. Naveenkumar, and A. Vadivel, “OpenCV for computer vision applications”, In Proc. of national conf. on big data and cloud computing (NCBDC’15), pp. 52-56, 2015 March.
14. T. W. Ke, A. S. Brewster, S. X. Yu, D. Ushizima, C. Yang, and N. K. Sauter, “A convolutional neural network-based screening tool for Xray serial crystallography”, J. Synchrotron Radiat., 25(3), pp. 655670, 2018.
15. Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, and D. H. P. Chau, “CNN explainer: learning convolutional neural networks with interactive visualization”, IEEE Trans Vis Comput Graph, 27(2), pp. 1396-1406, 2020.
16. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting”, J Mach Learn Res., 15(1), pp. 1929-1958, 2014.
17. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications” arXiv preprint arXiv:1704.04861, 2017.
18. A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”, Physica D: Nonlinear Phenomena, 404, 132306, 2020.
19. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation”, arXiv preprint arXiv:1406.1078, 2014.
20. R. C. Staudemeyer, and E. R. Morris, E. R., “Understanding LSTM--a tutorial into long short-term memory recurrent neural networks”, arXiv preprint arXiv:1909.09586, 2019.
21. T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection”, In Proceedings of the IEEE Intl. conf. on computer vision (ICCV), pp. 2980-2988, 2017.