mathematics

MDPI

Article

Traffic Accident Detection Method Using Trajectory Tracking and Influence Maps

Yihang Zhang [1] and Yunsick Sung [2,*]

[1] Department of Autonomous Things Intelligence, Dongguk University–Seoul, Seoul 04620, Republic of Korea
[2] Division of AI Software Convergence, Dongguk University–Seoul, Seoul 04620, Republic of Korea
* Correspondence: sung@dongguk.edu; Tel.: +82-2-2260-3338

Abstract: With the development of artificial intelligence, techniques such as machine learning, object detection, and trajectory tracking have been applied to various traffic fields to detect accidents and analyze their causes. However, detecting traffic accidents using closed-circuit television (CCTV) as an emerging subject in machine learning remains challenging because of complex traffic environments and limited vision. Traditional research has limitations in deducing the trajectories of accident-related objects and extracting the spatiotemporal relationships among objects. This paper proposes a traffic accident detection method that helps to determine whether each frame shows accidents by generating and considering object trajectories using influence maps and a convolutional neural network (CNN). The influence maps with spatiotemporal relationships were enhanced to improve the detection of traffic accidents. A CNN is utilized to extract latent representations from the influence maps produced by object trajectories. Car Accident Detection and Prediction (CADP) was utilized in the experiments to train our model, which achieved a traffic accident detection accuracy of approximately 95%. Thus, the proposed method attained remarkable results in terms of performance improvement compared to methods that only rely on CNN-based detection.

Keywords: trajectory tracking; traffic accident detection; machine learning; deep learning; convolutional neural network; influence map

MSC: 68T99

Citation: Zhang, Y.; Sung, Y. Traffic Accident Detection Method Using Trajectory Tracking and Influence Maps. *Mathematics* **2023**, *11*, 1743. https://doi.org/10.3390/math11071743

Academic Editors: Yiu-ming Cheung, Yuping Wang and Xin Liu

Received: 23 January 2023
Revised: 3 April 2023
Accepted: 4 April 2023
Published: 5 April 2023

1. Introduction

Traffic safety has received more attention because of the increase in traffic flow and the complexity of the road traffic infrastructure [1]. With the rapid development of intelligent surveillance traffic systems, collecting and storing traffic accident videos from advanced surveillance systems (e.g., closed-circuit television (CCTV), first-person cameras, and fisheye cameras) has become efficient and convenient [2,3]. CCTV cameras, which can obtain traffic information such as traffic flow, traffic accidents, and the number of vehicles on the road, have a wider viewing angle than other cameras. This multiscale spatiotemporal CCTV-based observation supports the understanding of traffic accidents before, during, and after their occurrence [4]. Nevertheless, to prevent traffic accidents, new methods for the detection of traffic accidents using CCTV frames must be established to detect traffic situations and provide real-time observations, analyses, and warnings.

Object detection algorithms have made substantial progress owing to the rapid development of deep-learning algorithms over the past decade [5]. Deep Simple Online Realtime Tracking (Deep SORT) [6] is a multi-object tracking (MOT) algorithm based on Intersection over Union (IoU) matching and is commonly used in numerous applications, such as autonomous driving [7] and vehicle trajectory analysis [8]. Deep SORT has emerged as a leading algorithm in object trajectory tracking owing to its speed, accuracy, and robustness.

In most research, object trajectories, typically defined as simple position coordinates or visual representations of generated object trajectories, are utilized as intermediate data

and are not directly employed in detection tasks. However, the position of vehicles or pedestrians may shift during traffic accidents, resulting in significant occlusion and affecting subsequent detection performance. Thus, significant gaps are present in the extracted object trajectory, which can be separated into two trajectories, thus affecting the detection performance of the following stage. Object trajectory tracking primarily focuses on object representation, updates of object representation, and the use of environmental change information. The analysis of the latent representation present in object trajectories is crucial for traffic accident detection.

Convolutional neural network (CNN)-based algorithms [9] are a popular approach for extracting latent representation tasks, with the goal of predicting the class of the given input sample data. A CNN is applied to the traffic accident data to improve the performance of the detection model. In particular, the object trajectories of the input data are obtained using the object trajectory-tracking method, and the latent representation of the object trajectories is extracted by the CNN. Traffic accident detection is then performed using the learned latent representation. However, the CNN cannot extract the temporal relationship between the object trajectories in complex traffic scenes when Deep SORT is combined with a CNN. Another potential issue is that the performance of Deep SORT can affect the quality of the appearance features extracted by CNNs. Inaccurate traffic accident detection may occur when Deep SORT cannot accurately extract the object trajectory. Consequently, identifying a method for improving object trajectory data and extracting the spatiotemporal relationship from the different object trajectories is a significant challenge in traffic accident detection. Thus, addressing this issue is crucial for enhancing the accuracy and performance of traffic accident detection models.

Therefore, this paper proposes a novel approach for traffic accident detection using trajectory tracking, influence maps, and a CNN. The proposed method uses YOLOv5 [10] and Deep SORT [6] to extract the trajectories of objects such as vehicles and pedestrians in CCTV frames. Subsequently, influence maps were used to analyze the spatiotemporal relationships among objects on the road. Finally, a CNN was utilized to detect traffic accidents based on the influence maps. This paper makes the following contributions:

- Unlike most previous works, this paper proposes a new method for the detection of traffic accidents, which employs YOLOv5 and Deep SORT to generate 2D object trajectories, influence maps to consider the spatiotemporal relationships between the objects, and a CNN to detect traffic accidents.
- The present paper is the first to apply influence maps to object trajectories for traffic accident detection. The influence maps deduce meaningful notations related to traffic accidents to object trajectories and can complement the deduced object trajectories, which improves the generalization and robustness of the proposed method.
- To evaluate the performance of the proposed method, experiments were conducted on a Car Accident Detection and Prediction (CADP) [11] dataset. The results demonstrate that the proposed method achieved a high detection performance for traffic accidents.

The remainder of this paper is organized as follows: In Section 2, we introduce recent research on Deep SORT, influence maps, CNNs, and traffic accident detection. Section 3 provides a brief introduction to the proposed method and the detailed implementation process. Section 4 shows the experimental results and compares the performance of the proposed method with that of the CNN-based traffic accident method. Finally, Section 5 presents the conclusions.

2. Related Works

This section introduces recent research in diverse industrial fields that mainly use Deep SORT, influence maps, and CNNs and compares these with the proposed method. Additionally, related research on traffic accident detection is reviewed.

2.1. Recent Research Based on Deep SORT, Influence Maps, and CNNs

The wide application of MOT in traffic and CCTV surveillance has rendered it a pertinent and popular topic in the field of computer vision. In the context of surveillance systems, MOT can track multiple people or vehicles from a video feed from a surveillance camera. This can assist security in monitoring a scene for potential threats or suspicious behavior. In addition, MOT can be utilized to track vehicles and pedestrians in a traffic scene, allowing the vehicles and pedestrians to navigate safely and avoid collisions in the field of autonomous vehicles. Bewley et al. [12] proposed a SORT algorithm for tracking objects. However, SORT has difficulty tracking objects, and the detection is prone to drifting when objects occlude each other or appear identical. Another disadvantage is that SORT relies on handcrafted features, which are ineffective in complex and dynamic traffic scenarios. Deep learning has powerful feature extraction capabilities and can complete object detection by extracting object features. Wojke et al. [6] extracted object appearance information using deep learning combined with SORT to improve detection performance. However, the trajectory obtained using the Deep SORT algorithm has certain limitations. The Deep SORT algorithm cannot capture the complex or nonlinear motion of an object and can only calculate its average or approximate position. Moreover, the trajectory drawn by the bounding box does not determine the direction of travel for the vehicles and pedestrians. The accuracy and validity of the detection are reduced if the bounding box detection is inaccurate or the problem of ID switching occurs. Existing trajectory-tracking research based on the Deep SORT algorithm only utilizes the extracted object trajectories as output in the intermediate stage, and almost no additional processing is performed.

An influence map is a form of representation that can provide a clear and intuitive visual representation of the influence of objects or events in the input, also highlighting key objects and events. Influence maps are utilized to rapidly understand the relationships and patterns among objects in the input data without performing complex calculations or analyses. Influence maps evolved from analytical work in Go games and are primarily used in strategy games [13]. Each influence map provides a more detailed strategy by correlating the notations and concepts in the game. However, information in the strategy game is more complicated, making it difficult for the neural network model to comprehend the generated influence maps [14]. Applying the influence map to describe the potential motion patterns in the frame effectively considers the motion flow and interaction of moving objects. However, influence maps produce error detection when significant distortion exists in the input frames.

In recent years, several approaches have been proposed for classification and detection using standard CNNs [15]. These approaches typically involve training a CNN on a labeled dataset of input data, with each piece of input data belonging to one of several classes. The CNN architecture consists of several convolutional layers that apply filters to the input data to extract spatial and textural features. After the convolutional layers, one or more pooling layers are typically added in CNNs, which reduce the dimensionality of the features and improve the robustness of the model. CNNs also have fully connected layers that combine the extracted features and output the final classification or detection results. CNNs can be utilized for various tasks in natural language processing (NLP), including sentiment analysis, music classification, and machine translation, regardless of whether the input is text or music. For example, one can translate text from one language into another, classify various types of music, and analyze the emotions expressed in music. Qiu et al. [16] proposed a model based on unsupervised learning that could focus on music structure and used a 3D CNN to investigate the spatial relationship of multitrack MIDI files. The model performed well even with a small amount of labeled data and an imbalanced dataset. Jang et al. [17] proposed a malware classification method for cyber security and defense, in which a CNN is mainly trained to receive merged images and classify malware into different classes. A CNN is a powerful feature extraction model that has been adopted for a wide variety of detection methods. This research applied a CNN to lidar-based object detection [18]. Here, a CNN is utilized as the backbone network for

extracting point cloud features and generating high-quality 3D proposal boxes. CNNs have been applied in medical imaging techniques, such as computed tomography scans, X-rays, and magnetic resonance imaging scans, to detect and diagnose diseases or abnormalities. The image analysis performed by the CNN can provide valuable information for diagnosis and treatment. CNNs are also utilized to extract certain features from images and classify them as those containing brain tumors [19].

This paper utilized Deep SORT [6] to obtain 2D object trajectories using CCTV frames to reduce possible distortion and utilized influence maps to solve problems such as the lack of spatiotemporal relationships among objects on roads before and after accidents. Subsequently, a CNN is utilized to determine whether an accident is involved.

2.2. Traffic Accident Detection Research in Recent Times

With the development of deep-learning-based detectors, detection research has emerged as a prominent subject in the academic community [20]. In the field of traffic accidents, the authors of [21] compared six machine-learning algorithms to detect road traffic accident data and performed data mining. Another study [22] proposed a detection and recognition model for analyzing the severity of traffic accidents, which has strong detection recognition and generalization abilities. This model has a better fitting ability and prediction accuracy compared to other methods. However, different kernel functions and insensitive loss parameter choices significantly affect the model results. In another study [23], Zheng et al. proposed a traffic accident severity prediction CNN (TASP-CNN) model considering the combined relationship between the different features of traffic accidents. The model analyzes the weight of traffic accident features and applies the feature matrix to a grayscale image (FM2GI) algorithm to improve traffic accident severity prediction. However, various problems are associated with the TASP-CNN model. The model has poor generalization ability, and applying it to other public transportation datasets is challenging. Jiang et al. [24] proposed for the first time a Long Short-Term Memory (LSTM)-based framework for detecting accidents on freeways considering raw traffic data with different temporal resolutions. They conducted accident detection experiments to achieve better performance. However, the dataset used in this paper was carefully selected, leading to poor generalization of the detection model. These issues have prompted the development of improved models for traffic accident detection. Table 1 lists the differences between recent traffic accident detection research and the proposed method.

Table 1. Difference between recent traffic accident detection research and the proposed method.

Research Contents	Input Data	Detection Algorithm
Data mining [21]	Description attribute	Fuzzy-FARCHD
Rough set theory [22]	Decision table	SVM
Feature matrix to gray image [23]	Grayscale image	TASP-CNN
Different time resolutions [24]	Raw traffic data	LSTM
Proposed method	Influence map	CNN

The proposed method extracts 2D object trajectories and considers the spatiotemporal relationships among objects in two stages. Compared with traditional detection research, the proposed method utilizes a large amount of data for training, improves the accuracy of the detection results, and requires less training time.

3. Traffic Accident Detection Method

In this section, our traffic accident detection method is used to extract object trajectories in the tracking phase, and CCTV frames are analyzed in the executive phase to determine whether accidents have occurred.

3.1. Overview of Traffic Accident Detection

The proposed traffic accident detection method comprises two phases: tracking and execution. The tracking phase consists of two parts: a Bounding Box Generator and a Trajectory Extractor. In the tracking phase, the Bounding Box Generator obtains 2D object bounding boxes using You Only Look Once (YOLOv5), the fifth version of the multi-object detection algorithm [10] in CCTV frames. The 2D object bounding boxes are passed into the trajectory executor as inputs to predict the corresponding trajectory of each 2D object bounding box using the Deep SORT algorithm [6].

The execution phase consists of two parts: an Influence Map Generator and a Detection Executor. The Influence Map Generator focuses on analyzing the spatiotemporal relationships among the trajectories of 2D objects using influence maps. The Detection Executor extracts the features of the corresponding influence map using a CNN and then classifies whether an accident has occurred or not. Figure 1 shows a block diagram illustrating the procedures involved in the tracking and execution phases for detecting traffic accidents.

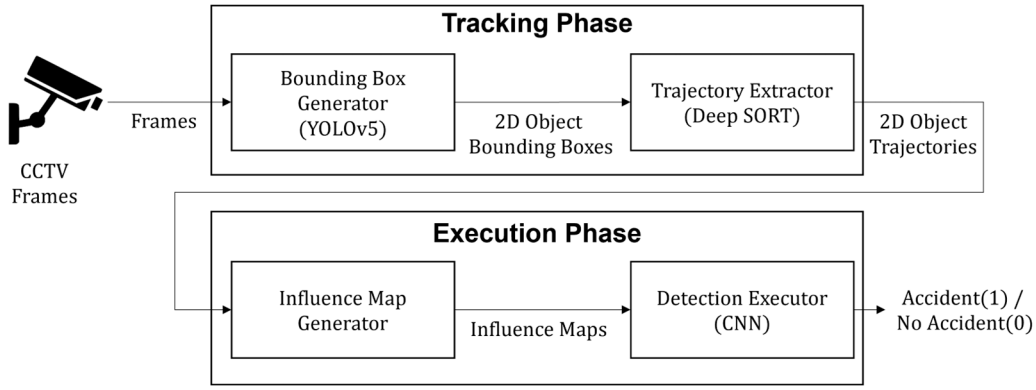


Figure 1. Processes of traffic accident detection.

3.2. Implementation of Traffic Accident Detection

In the tracking phase, the YOLOv5 network detects all dynamic objects on the road, including vehicles and pedestrians, by extracting the features of CCTV frames. This paper utilized the CADP dataset [11], a dataset for traffic accident analysis, as the input for the YOLOv5 network. In this paper, 150 CCTV segments were selected as pretrained weight models to detect objects using YOLOv5 to output 2D object bounding boxes. Subsequently, beginning from the second CCTV frame, the Deep SORT algorithm was utilized to generate one trajectory for each object, based on the 2D object bounding boxes.

In the execution phase, 2D object trajectories are given as inputs to the Influence Map Generator to output one influence map for each corresponding frame. The Influence Map Generator encodes 2D object trajectories to provide meaningful notations in the influence maps, denoting spatial and temporal information. Figure 2 shows an illustration of the influence map.

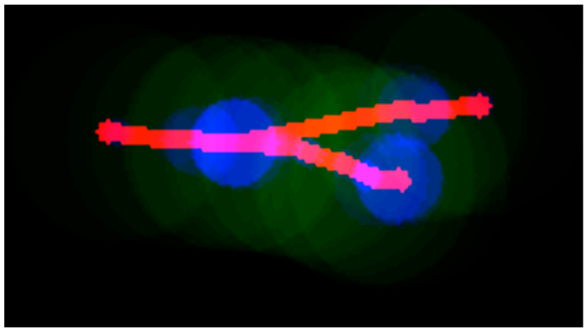


Figure 2. Illustration of an influence map.

In this paper, the influence map comprised three notations: the distances among objects, accident discrimination, and object trajectories. First, the distances between objects are represented by blue circles. The radii of the blue circles increase as the distance between the objects decrease. Furthermore, the blue circles deepen over time. The possibility of an accident is represented by green circles. When the probability of each accident increases, the radius of the corresponding green circle increases, and the intersection range of the object bounding boxes increases. Finally, the missing parts in the 2D object trajectory are supplemented and corrected to generate a smoother 2D object trajectory, which is represented as a red route in the influence map. The proposed method utilizes a CNN to extract the features of influence maps and compresses the features of influence maps into a fixed dimension, and then the results of accident detection are output. Figure 3 shows the architecture of the proposed traffic accident detection method.

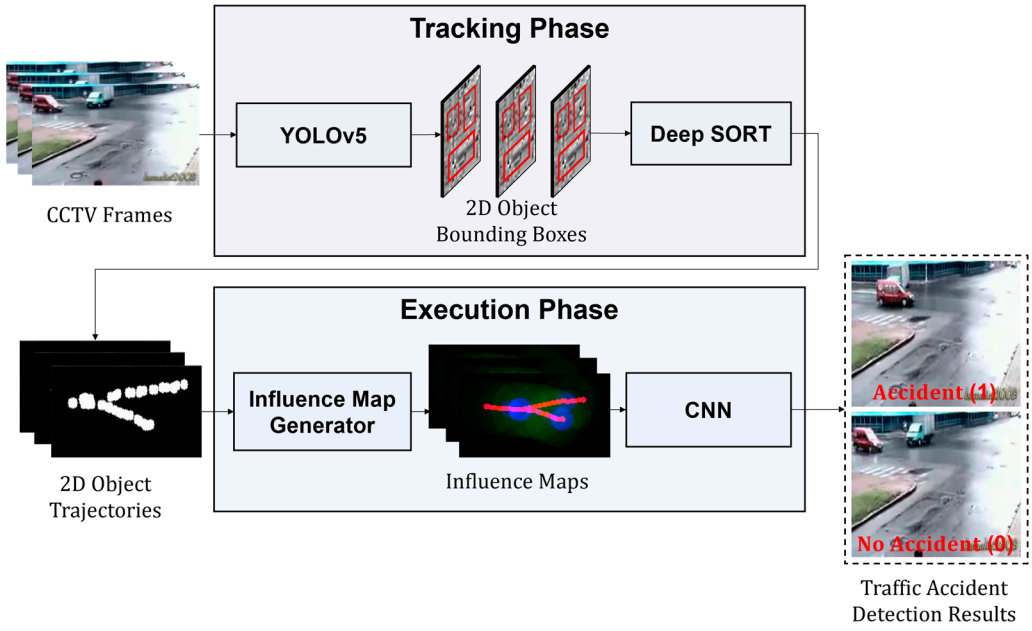


Figure 3. Architecture of traffic accident detection method.

A CNN was utilized to distinguish the given input influence maps and assign significant learnable weights and CNN biases to the influence maps. The CNN, which is comprised of five convolutional layers, five max-pooling layers, and three fully connected layers, received RGB three-channel influence maps with a size of 224 × 224 as inputs. In particular, each convolutional layer utilized a 3 × 3 kernel and was followed by a max-pooling layer. The second and fifth convolutional layers had 64 and 512 channels in their output, respectively. The max-pooling layer utilized a 2 × 2 kernel with a stride of 2. The max-pooling layer maximized the retention of the extracted spatiotemporal features from the influence maps and minimized the dimensionality of the feature map of the convolutional layers. Additionally, batch normalization and a rectified linear unit (ReLU) activation function were added to the convolutional layers before the corresponding max-pooling layers. Batch normalization was used to perform normalization processing on the influence maps, which prevented the network from becoming unstable for the ReLU activated on the influence maps because of its large size. The nonlinearity of the ReLU activation function helped to avoid gradient disappearance and accelerated the CNN training. The last three layers were fully connected with sizes of 4096, 1024, and 2. The last fully connected layers were utilized to obtain the results of traffic accident detection, which were either an accident or no accident. We used the SoftMax function for detection. The SoftMax function converted the output of the last fully connected layer and produced a probability distribution over

the two classes, indicating the likelihood of an accident or no accident. Figure 4 shows the architecture of a CNN in the proposed method.

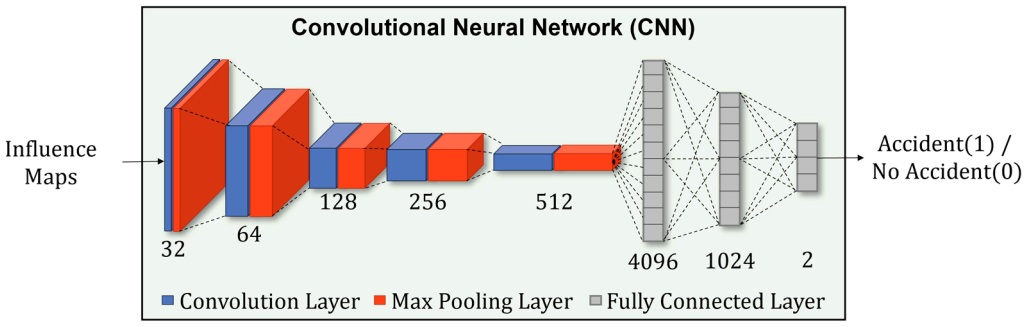


Figure 4. Architecture of CNN in the proposed method.

3.3. Influence Map Generator

The 2D object trajectories obtained from CCTV frames of traffic accidents frequently have the problem of missing trajectories because of the occurrence of traffic accidents. In particular, at the moment of a traffic accident, the vehicles and pedestrians involved in the accident will suddenly undergo a large displacement and destroy the integrity of the generated trajectories. Therefore, this paper proposes an Influence Map Generator to focus on the hidden spatiotemporal information in 2D object trajectories and complement the trajectories while weakening irrelevant information in the 2D object trajectories. The Influence Map Generator analyzes 2D object trajectories to enhance their interpretability of 2D object trajectories by adding meaningful notations based on the RGB channel. For distances among objects in the influence map notation, the equation for defining the radius of the circle is as follows:

$$r_c = \sqrt{(w_i^2 + h_i^2)}, \text{ where } r_{min} = \begin{cases} w_i & if(w_i \geq h_i) \\ h_i & if(w_i < h_i) \end{cases} \quad (1)$$

where r_c is the radius of the blue circle; r_{min} is the shorter width and height of the 2D object trajectory; i is the input 2D object trajectory; w_i is the width; and h_i is the height of the 2D object trajectory. The distance ratio of the blue circle was adjusted based on the size of the 2D object trajectory to calculate the overlapping area of the two object trajectories. A higher distance ratio indicates that the overlapping area between two objects is larger, and the blue color would be darker. In contrast, a lower distance ratio indicates that the overlapping area between the two objects is smaller, and the blue color would be lighter. Furthermore, the color of the blue circle deepened over time as follows:

$$x = \frac{d_{min}}{(r_c + r_{min})/2} \times 100 \quad (2)$$

$$c = 4 + \frac{60}{T_{max}} \times T_{now} \quad (3)$$

where x is the distance ratio, d_{min} is the distance to the nearest object around the object being measured, c is the degree of color change, T_{max} is the maximum frame in which the trajectory coordinates exist, and T_{now} is the frame in which the current trajectory coordinates are located. Accident discrimination was calculated as follows:

$$y = \begin{cases} 10 & if(\frac{w_c}{w_i} + \frac{h_c}{h_i}) \times 100 \geq 5 \\ 30 & if(\frac{w_c}{w_i} + \frac{h_c}{h_i}) \times 100 < 5 \end{cases} \quad (4)$$

where y is the area of the object bounding box, and w_c and h_c are the width and height of the object bounding box, respectively. For the object trajectories in the influence map

notation, the trajectory coordinates based on the movement of the objects in each frame were drawn with red lines, and the sizes of the red lines were maintained. First, the distance between any two objects was calculated for each 2D object trajectory in the Influence Map Generator, and a blue circle was drawn on the corresponding trajectory coordinate point. The value of accident discrimination was then determined based on each area of the object bounding box, and a green circle was drawn at the corresponding trajectory coordinate point according to the value. Finally, red lines were drawn based on the coordinates of each 2D object trajectory to represent the corrected and smoothened 2D object trajectories. Figure 5 illustrates the influence map generator processes.

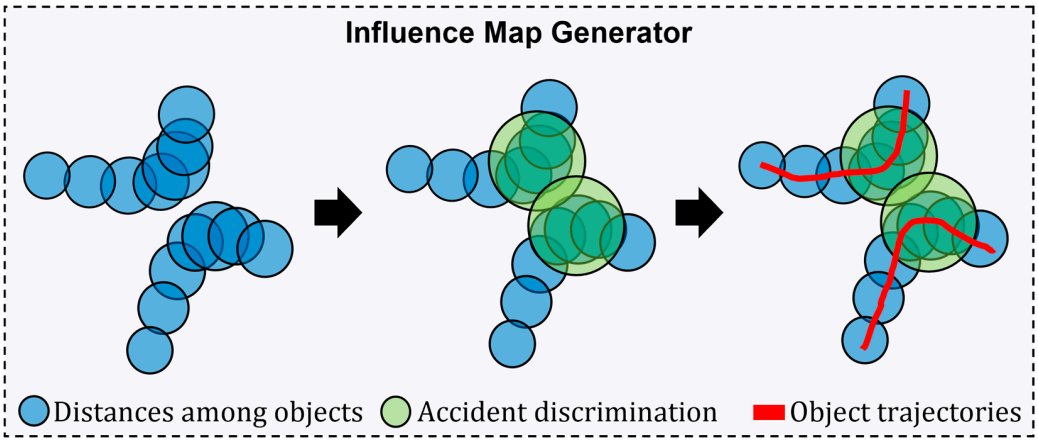


Figure 5. Illustration of the influence map generator processes.

4. Experiment

The experimental objectives, detailed training parameters of the proposed method, and traffic accident detection results are described in this section. Moreover, the detection results of the proposed method are compared with a CNN-based traffic accident method.

4.1. Experimental Objectives

Experiments were performed to validate the accuracy of the proposed method in accurately classifying traffic accidents. The performance of the proposed method was evaluated by comparing it with a CNN-based traffic accident detector.

4.2. Experimental Environment

The proposed method utilized various training parameters (Table 2). The proposed method used CCTV frames as inputs, which were then processed in the tracking phase to generate 2D object trajectories. The data format of the generated 2D object trajectories was an image, and the size of the 2D object trajectories was the same as that of the CCTV frames. Subsequently, the 2D object trajectories were directly fed into the execution phase without any preprocessing. After generating the influence maps, their results were uniformly cropped to 224 × 224 pixels and fed to the CNN model. A batch size of 64 and a learning rate of 1×10^{-5} were used. The total number of training epochs was 15, and each epoch had 94 steps. A stochastic gradient descent (SGD) was the optimizer, and the SoftMax function was selected as the objective function of the proposed method.

The experiments were conducted on Windows 10, a six-core Intel i7-6850K, and an Nvidia Titan RTX (48 GB). The proposed method was implemented using Python 3.6. To improve computing efficiency, the YOLOv5, Deep SORT, and CNN models were implemented with a deep-learning library and PyTorch, which utilized the computing functions of a graphics-processing unit (GPU) in the Nvidia Titan RTX. Based on the hardware configuration used, the average total inference time for YOLOv5 on each piece of video data was found to be approximately 43.3 milliseconds (ms), and the average total tracking time for Deep SORT was approximately 14.7 ms. The proposed method

utilizing trajectory tracking, and influence maps demonstrated that the average training time was 10.70 iterations per second (it/s), and the average validation time was 29.65 it/s. Furthermore, the average training time of a CNN-based detector was 10.49 it/s, and the average validation time was approximately 28.52 it/s.

Table 2. Parameters for training the proposed method.

Hyperparameter	Value
CCTV frame dim	(Weight, Height)
2D object trajectories dim	(Weight, Height)
Influence map dim	(224, 224, 3)
Batch size	64
Learning rate	1×10^{-5}
Total training epochs	15
Steps per epoch	94
Optimizer	SGD
Objective function	softmax

In the proposed traffic accident detection method, the unmodified YOLOv5 algorithm was used as the backbone network and incorporated into the Bounding Box Generator, ensuring efficient and accurate object detection results. Concurrently, to track object trajectories, the original Deep SORT algorithm without any modifications was employed as the backbone network within the Trajectory Extractor. In addition, the YOLOv5 algorithm was extensively trained on the comprehensive COCO dataset for image annotation and displayed outstanding performance by achieving an average precision of 48.2% while maintaining a processing speed of 13.7 ms. The YOLOv5 algorithm was well suited for accurate object detection within the CADP dataset, making it highly compatible with the proposed method.

4.3. Experimental Data

The CADP dataset [11] was used in the experiments, and Table 3 lists its contents. The dataset utilized in this paper was obtained from YouTube and comprised 1416 CCTV segments, each of which is fully annotated in terms of spatial and temporal information. In particular, 150 CCTV segments were preprocessed, and each CCTV segment was segmented into 50 CCTV frames to generate 7500 CCTV frames. The preprocessed 7500 CCTV frames were used for YOLOv5 detection and Deep SORT tracking to generate 2D object trajectories. Similarly, 7500 2D object trajectories were passed through an Influence Map Generator to generate 7500 influence maps. The 90% influence maps were used as training data (6750 in total). The 10% influence maps were utilized as validation data (750 in total).

Table 3. Contents of the CADP dataset.

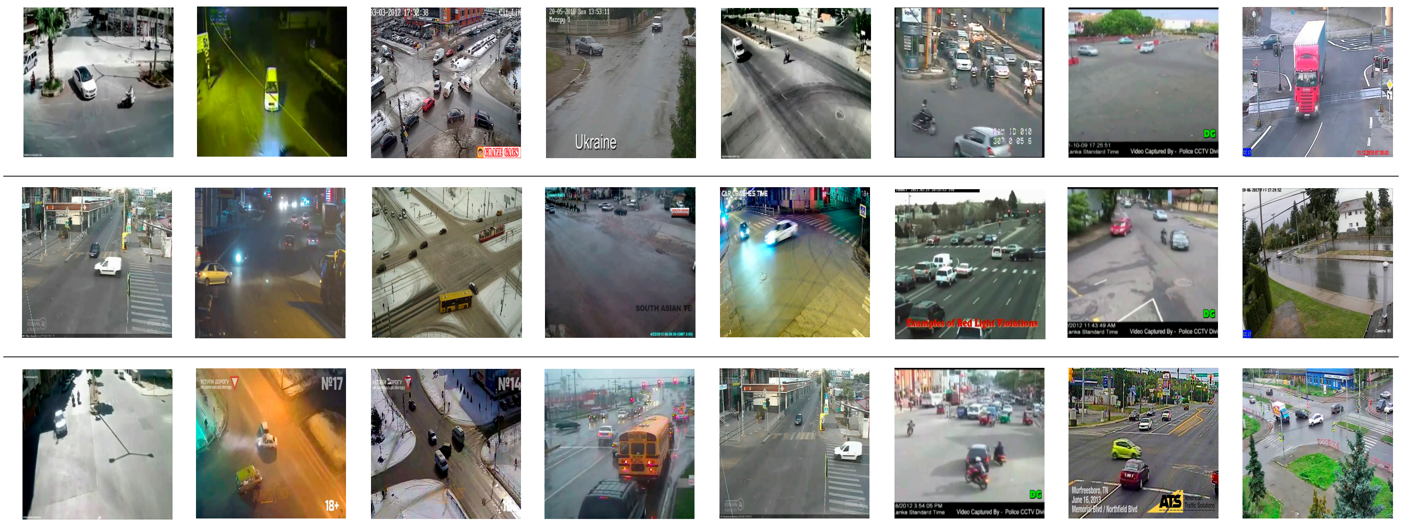
CADP Dataset	Value
CCTV segments	1416
Preprocessed CCTV segments	150
Frames per CCTV segments	50
Total input data	7500
Training data	6750 (90%)
Validation data	750 (10%)

The CADP dataset provides comprehensive training and testing traffic accident data for various road scenarios by providing different viewpoints, varying illumination levels, and diverse weather conditions, which cater to precise traffic accident detection challenges. Table 4 shows the various traffic scenarios in the CADP dataset, including different times of day and weather conditions, such as daytime/nighttime and snowy/rainy, to provide a diverse set of challenging scenarios for tracking objects and detecting traffic accidents.

The CADP dataset contains scenarios with different traffic densities from low to high traffic volumes, reflecting varying traffic situations. The dataset also includes low- and high-resolution scenarios, which allows for the impact of different levels of frame details on the accuracy of traffic accident detection to be studied. Notably, each accident in the CADP dataset was captured from different camera angles, which increased the variety of data and provided a more comprehensive view of the traffic accidents.

Table 4. Various traffic scenarios in the CADP dataset.

Daytime	Nighttime	Snowy	Rainy	Low Traffic Volumes	High Traffic Volumes	Low Resolution	High Resolution



The input data for the traffic accident detection method included CCTV frames, 2D object trajectories, and influence maps. Figure 6 shows the input data for each part of the proposed method. CCTV frames are video images captured by CCTV cameras. There were 25 accident frames and 25 non-accident frames for each of the 50 CCTV frames. "No accident" is represented by the number 0, and an "Accident" is represented by the number 1. In addition, 2D object trajectories and influence maps were processed in the same manner as CCTV frames.

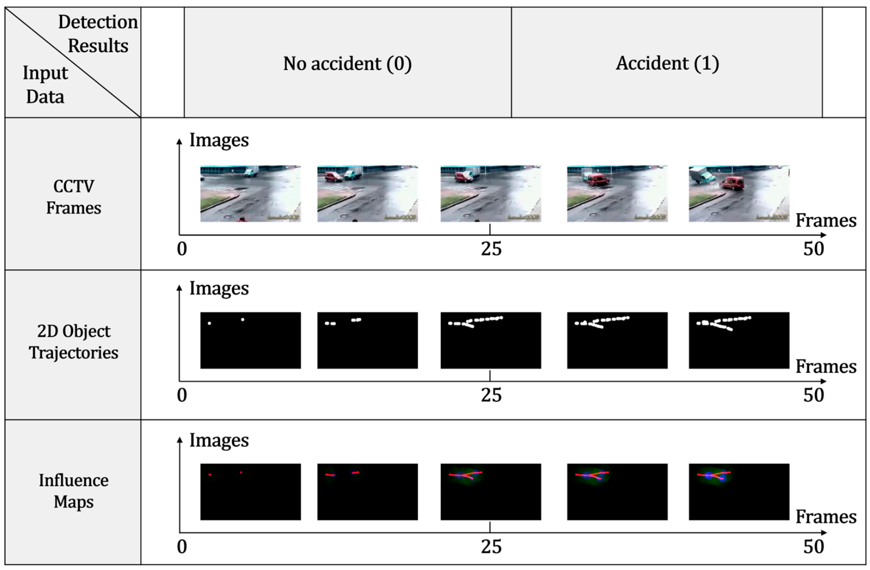
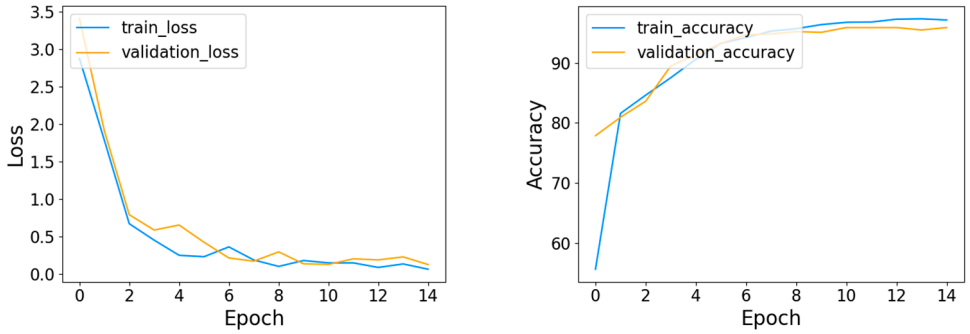


Figure 6. Input data for each part of the proposed method.

4.4. Experimental Results

Figure 7 shows the training and validation results of the proposed method. Figure 7a shows that the initial value was approximately 2.88 as the training loss of the proposed method. The training loss started to converge after four epochs and finally converged to approximately 0.06. The initial value of the validation loss for the proposed method was approximately 3.41. After four epochs, the validation loss began to converge and finally converged at approximately 0.12.

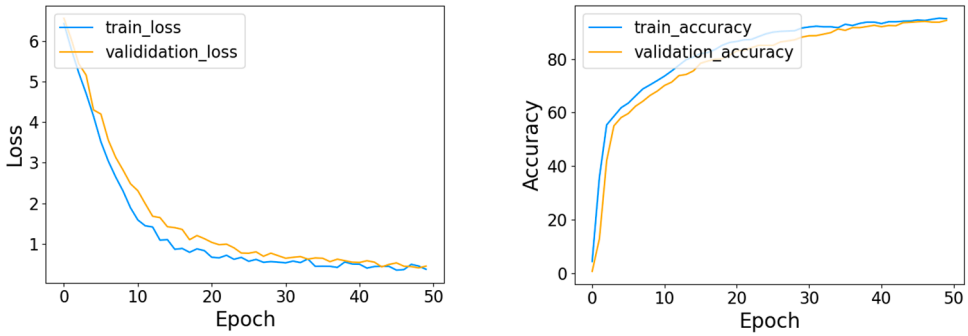


(**a**) (**b**)

Figure 7. Experimental training and validation results of the proposed method. (**a**) Loss of the proposed method; (**b**) accuracy of the proposed method.

Figure 7b shows that the initial value of the training accuracy of the proposed method was approximately 55.63. After two epochs, the training accuracy converged and finally increased to 97.13. The validation accuracy of the proposed method started with a value of approximately 77.87 during the initial training and slowly increased to 95.87 during further training. Figure 7 shows that the proposed method extracted the features of the influence maps and completed the traffic accident detection.

Figure 8 shows the training and validation results for the CNN-based detector. As shown in Figure 8a, during the CNN-based detector training, the training loss dropped sharply from 6.43 to 0.86 before the fifteenth epoch and then dropped slowly to 0.37 until convergence. The validation loss was similar to the training loss, with a value of 6.56 at the beginning of training. After 50 epochs, the validation loss converged to 0.45.



(**a**) (**b**)

Figure 8. Experimental training and validation results of the CNN-based detector. (**a**) Loss of the CNN-based detector; (**b**) accuracy of the CNN-based detector.

Figure 8b shows the accuracy rate of the CNN-based detector. The CNN-based detector training accuracy sharply increased from 4.67 to 55.4 before the fourth epoch and then slowly increased to 95.2 until convergence. Similarly, after 50 training epochs, validation accuracy increased from 0.75 to 94.38.

As shown in Figure 8, the proposed method exhibited lower loss and faster convergence than the CNN-based detector. In addition, the accuracy of the proposed method was 1.93 higher than that of the CNN-based detector trained for 50 epochs, the proposed method only trained for 15 epochs, and the loss of the proposed method converged to 0.31 lower than that of the CNN-based detector. This suggests that the proposed method performed better and was superior at detecting traffic accidents.

As shown in Figure 9, notable differences in traffic accident detection performance across various approaches were observed in the histogram contrast experimental results. When an influence map was employed, the highest training and validation accuracy of 97.3 and 95.9 were achieved, respectively. In contrast, using an object bounding box [8] resulted in a training accuracy of 96.4 and a validation accuracy of 91.7. In the experiments with an object trajectory approach, the training accuracy reached 93.8, and the validation accuracy was at 93.0. The optical flow approach [25] exhibited substantially lower traffic accident detection performance, with a training accuracy of 67.0 and a validation accuracy of 65.9. Finally, the attention map approach [26] achieved a training accuracy of 86.6 and validation accuracy of 84.3. Within the domain of traffic accident detection tasks, the influence map approach exhibited a strong superiority in performance with respect to the other approaches. The influence map approach accounts for the spatiotemporal relationships among objects and provides more meaningful information. The disparities in performance display the significance of the influence map in enhancing the performance of the model in traffic accident detection tasks. Consequently, the effectiveness of influence maps is highlighted, as they exceeded the alternative input data types in both the training and validation stages.

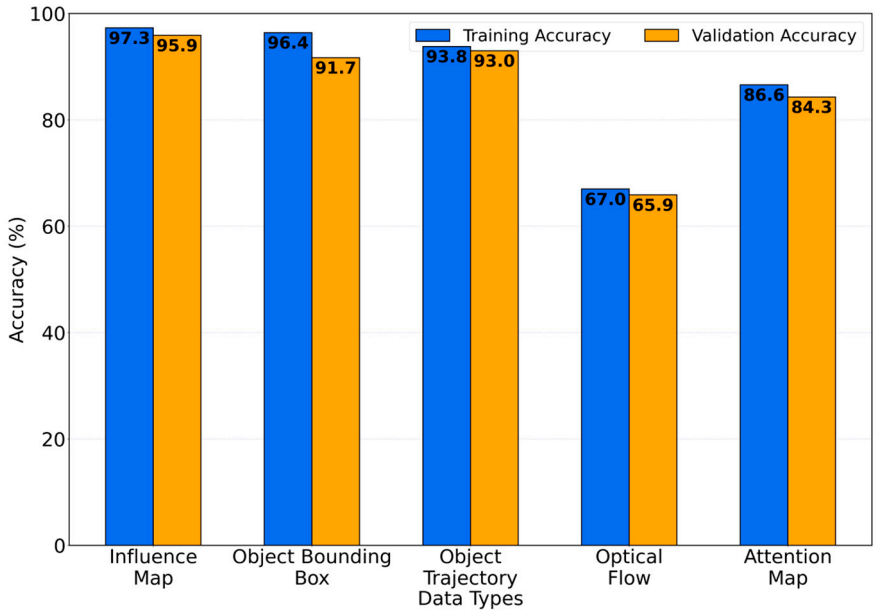


Figure 9. Accuracy of traffic accident detection with different approaches.

5. Conclusions

This paper has proposed a traffic accident detection method that utilizes trajectory tracking, influence maps, and a CNN. First, in the tracking phase, all dynamic objects on roads in the CCTV frames were detected by the YOLOv5 network and expressed by 2D object bounding boxes. The corresponding trajectories of the 2D object bounding boxes were deduced using a Deep SORT algorithm [6]. In the execution phase, influence maps were generated from the deducted trajectories by the Influence Map Generator. The CNN then extracted the features of the influence maps and determined whether any accidents occurred. The experimental results showed that the traffic accident detection method using trajectory tracking and influence maps had a high detection performance compared to

other detection methods. The issue of distortion that may arise in traffic accident detection based on CCTV frames was resolved using trajectory tracking, influence maps, and a CNN. Compared to standard CNN-based detection methods, the proposed method had a faster detection speed and higher detection accuracy.

The following four aspects could be investigated in the future. First, the proposed method can be combined with transfer learning or incremental learning to further improve its performance. Next, fine-tuning with state-of-the-art pre-trained detection models such as EfficientDet [27] or Faster R-CNN [28] can be achieved on the CADP traffic accident dataset. This transfer learning approach aims to capitalize on the existing knowledge acquired by these models during their training on large-scale datasets, thus enhancing their performance in traffic accident detection. The proposed method could train with incremental learning on multiple related tasks simultaneously, such as object detection, scene segmentation, and accident severity estimation. By sharing the learned features across multiple tasks, the proposed method would benefit from the additional supervision signals, resulting in enhanced generalization and improved traffic accident detection performance. Next, the severity of traffic accidents should be assessed while detecting accidents, given that the proposed method only determines whether a traffic accident occurred or not from the CCTV frames. Developing a framework to simultaneously detect the occurrence and severity of traffic accidents or incorporating additional features and context information such as the vehicle speed, collision angle, and road conditions to enhance the detection accuracy are some future directions to carry this research forward. In the case of the first-person camera, the proposed method can be adapted to address the challenges associated with this viewpoint, including not only first-person cameras and fisheye lenses but also ego-motion, dynamic backgrounds, and occlusions. Next, the recording capabilities of cameras should be handled more as a core factor, considering the resolution, frame rate, compression artifacts, and lens distortion and the impact of these elements on the quality of CCTV frames. Additionally, addressing distortion can be accomplished by implementing lens correction algorithms and developing deep-learning algorithms. These improvements can lead to better object detection, tracking, and traffic accident detection.

Author Contributions: Conceptualization, Y.Z. and Y.S.; methodology, Y.Z. and Y.S.; software, Y.Z. and Y.S.; validation, Y.Z. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a Korea Institute of Police Technology (KIPoT) grant funded by the Korea government (KNPA) (No. 092021D75000000, AI driving ability test standardization and evaluation process development).

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this paper. This data can be found here: https://ankitshah009.github.io/accident_forecasting_traffic_camera, accessed on 10 August 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tan, K.; Bremner, D.; Le Kernec, J.; Zhang, L.; Imran, M. Machine Learning in Vehicular Networking: An Overview. *Digit. Commun. Netw.* **2022**, *8*, 18–24. [CrossRef]
2. Khan, S.W.; Hafeez, Q.; Khalid, M.I.; Alroobaea, R.; Hussain, S.; Iqbal, J.; Almotiri, J.; Ullah, S.S. Anomaly Detection in Traffic Surveillance Videos using Deep Learning. *Sensors* **2022**, *22*, 6563. [CrossRef] [PubMed]
3. Xu, Y.; Huang, C.; Nan, Y.; Lian, S. TAD: A Large-Scale Benchmark for Traffic Accidents Detection from Video Surveillance. *arXiv* **2022**, arXiv:2209.12386.
4. Ahmed, M.I.B.; Zaghdoud, R.; Ahmed, M.S.; Sendi, R.; Alsharif, S.; Alabdulkarim, J.; Saad, B.A.A.; Alsabt, R.; Rahman, A.; Krishnasamy, G. A Real-Time Computer Vision based Approach to Detection and Classification of Traffic Incidents. *Big Data Cogn. Comput.* **2023**, *7*, 22. [CrossRef]
5. Zhou, Y. Vehicle Image Recognition using Deep Convolution Neural Network and Compressed Dictionary Learning. *J. Inf. Process. Syst.* **2021**, *17*, 411–425.
6. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Real-Time Tracking with A Deep Association Metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.

7. Shen, Y.; Hou, Q.; Wang, J.; Zhu, X. Research on the Simulation Test Method of Autonomous Driving Vehicle Based on Cloud Platform. In Proceedings of the 2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tianjin, China, 1–3 July 2022; pp. 149–153.
8. Huang, X.; He, P.; Rangarajan, A.; Ranka, S. Intelligent Intersection: Two-Stream Convolutional Networks for Real-time Near-Accident Detection in Traffic Video. *ACM Trans. Spat. Algorithms Syst. (TSAS)* **2020**, *6*, 1–28. [CrossRef]
9. Elngar, A.A.; Arafa, M.; Fathy, A.; Moustafa, B.; Mahmoudm, O.; Shaban, M.; Fawzy, N. Image Classification based on CNN: A Survey. *J. Cybersecur. Inf. Manag.* **2021**, *6*, 18–50. [CrossRef]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
11. Shah, A.P.; Lamare, J.B.; Nguyen-Anh, T.; Hauptmann, A. CADP: A Novel Dataset for CCTV Traffic Camera-based Accident Analysis. In Proceedings of the 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–9.
12. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple Online and Realtime Tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
13. Jang, S.H.; Cho, S.B. Evolving Neural NPCs with Layered Influence Map in the Real-time Simulation Game 'Conqueror'. In Proceedings of the 2008 IEEE Symposium on Computational Intelligence and Games (CIG), Perth, WA, Australia, 15–18 December 2008; pp. 385–388.
14. Lee, D.G.; Suk, H.I.; Park, S.K.; Lee, S.W. Motion Influence Map for Unusual Human Activity Detection and Localization in Crowded Scenes. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1612–1623. [CrossRef]
15. Park, K.; Baek, S.; Jeon, J.; Jeong, Y.S. Music Plagiarism Detection Based on Siamese CNN. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*, 12–38.
16. Qiu, L.; Li, S.; Sung, Y. 3D-DCDAE: Unsupervised Music Latent Representations Learning Method based on a Deep 3D Convolutional Denoising Autoencoder for Music Genre Classification. *Mathematics* **2021**, *9*, 2274. [CrossRef]
17. Jang, S.; Li, S.; Sung, Y. Fasttext-based Local Feature Visualization Algorithm for Merged Image-based Malware Classification Framework for Cyber Security and Cyber Defense. *Mathematics* **2020**, *8*, 460. [CrossRef]
18. Zhang, X.; Bai, L.; Zhang, Z.; Li, Y. Multi-Scale Keypoints Feature Fusion Network for 3D Object Detection from Point Clouds. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*. [CrossRef]
19. Swarup, C.; Kumar, A.; Singh, K.U.; Singh, T.; Raja, L.; Kumar, A.; Dubey, R. Biologically Inspired CNN Network for Brain Tumor Abnormalities Detection and Features Extraction from MRI Images. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*. [CrossRef]
20. Qiu, L.; Li, S.; Sung, Y. DBTMPE: Deep Bidirectional Transformers-based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics* **2021**, *9*, 530. [CrossRef]
21. Kumeda, B.; Zhang, F.; Zhou, F.; Hussain, S.; Almasri, A.; Assefa, M. Classification of Road Traffic Accident Data using Machine Learning Algorithms. In Proceedings of the 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 12–15 June 2019; pp. 682–687.
22. Jianfeng, X.; Hongyu, G.; Jian, T.; Liu, L.; Haizhu, L. A Classification and Recognition Model for the Severity of Road Traffic Accident. *Adv. Mech. Eng.* **2019**, *11*, 1687814019851893. [CrossRef]
23. Zheng, M.; Li, T.; Zhu, R.; Chen, J.; Ma, Z.; Tang, M.; Wang, Z. Traffic Accident's Severity Prediction: A Deep-learning Approach-based CNN Network. *IEEE Access* **2019**, *7*, 39897–39910. [CrossRef]
24. Jiang, F.; Yuen, K.K.R.; Lee, E.W.M. A Long Short-term Memory-based Framework for Crash Detection on Freeways with Traffic Data of Different Temporal Resolutions. *Accid. Anal. Prev.* **2020**, *141*, 105520. [CrossRef] [PubMed]
25. Le, T.N.; Ono, S.; Sugimoto, A.; Kawasaki, H. Attention R-CNN for Accident Detection. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 313–320.
26. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, HI, USA, 21–26 July 2017; pp. 2462–2470.
27. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020; pp. 10781–10790.
28. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.