mathematics

Article

Traffic Accident Detection Using Background Subtraction and CNN Encoder–Transformer Decoder in Video Frames

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

[1] Department of Autonomous Things Intelligence, Dongguk University-Seoul, Seoul 04620, Republic of Korea; 425949286@dongguk.edu
[2] Division of AI Software Convergence, Dongguk University-Seoul, Seoul 04620, Republic of Korea
* Correspondence: sung@dongguk.edu; Tel.: +82-2-2260-3338

Abstract: Artificial intelligence plays a significant role in traffic-accident detection. Traffic accidents involve a cascade of inadvertent events, making traditional detection approaches challenging. For instance, Convolutional Neural Network (CNN)-based approaches cannot analyze temporal relationships among objects, and Recurrent Neural Network (RNN)-based approaches suffer from low processing speeds and cannot detect traffic accidents simultaneously across multiple frames. Furthermore, these networks dismiss background interference in input video frames. This paper proposes a framework that begins by subtracting the background based on You Only Look Once (YOLOv5), which adaptively reduces background interference when detecting objects. Subsequently, the CNN encoder and Transformer decoder are combined into an end-to-end model to extract the spatial and temporal features between different time points, allowing for a parallel analysis between input video frames. The proposed framework was evaluated on the Car Crash Dataset through a series of comparison and ablation experiments. Our framework was benchmarked against three accident-detection models to evaluate its effectiveness, and the proposed framework demonstrated a superior accuracy of approximately 96%. The results of the ablation experiments indicate that when background subtraction was not incorporated into the proposed framework, the values of all evaluation indicators decreased by approximately 3%.

Keywords: artificial intelligence; deep learning; traffic-accident detection; background subtraction; CNN encoder; Transformer decoder

MSC: 68T99

Citation: Zhang, Y.; Sung, Y. Traffic Accident Detection Using Background Subtraction and CNN Encoder–Transformer Decoder in Video Frames. *Mathematics* **2023**, *11*, 2884. https://doi.org/10.3390/math11132884

Academic Editor: Bogdan Oancea

Received: 2 June 2023
Revised: 21 June 2023
Accepted: 25 June 2023
Published: 27 June 2023

1. Introduction

According to the Association for Safe International Road Travel, approximately 1.35 million persons die from road traffic accidents each year, with an average of 3700 fatalities occurring daily [1]. Traffic safety assessments for the period 1998–2020 found that common accidents not only cause traffic congestion and significant economic losses to society but also pose a huge threat to public safety, as nonfatal injuries frequently result in long-term disabilities [2]. Manual accident management relies on the availability of personnel at the accident site. Consequently, computer-aided traffic-accident detection has become an important research area. Accidents can be predicted and prevented by analyzing the most recent information on the present state of road conditions and object positions.

Various detection and classification models have become popular with the increase in computing power and development of neural networks [3–7]. These models have been applied in various fields, particularly image and video processing [8,9]. Traffic-accident detection is essentially a pattern-classification problem, which means that the current pattern must be detected or classified into one of two types: traffic accidents or non-traffic accidents [10]. Traditional traffic-accident-detection approaches face challenges owing to the background information in video frames, leading to difficulties in accurately detecting accidents [11]. For instance, extraneous information, such as trees, sky, and buildings in the background, may be misidentified as accident objects, resulting in false positives. Furthermore, certain approaches require manual threshold adjustments to filter the background, which can increase the experimental complexity and computational time [12].

Advancements in the field of object detection over the past decade have been strongly propelled by innovative algorithms such as You Only Look Once (YOLOv5) [13]. The application of YOLOv5 is particularly prominent; it extracts relevant features by accurately locating and drawing bounding boxes around objects in video frames. YOLOv5 adopts a one-stage anchor-free architecture that leverages a deep neural network to detect objects. The network of YOLOv5 consists of backbone, neck, and head networks. The backbone network extracts high-level features from input data. The neck network integrates multi-scale features to enhance the representational capacity of YOLOv5. The head network predicts bounding box coordinates and class probabilities for detected objects. YOLOv5 addresses the limitations of previous YOLO versions and pushes the boundaries of object-detection performance. It has significantly impacted applications such as vehicle- and pedestrian-detection within real-world traffic scenarios and assisted in analyzing and understanding complex accident scenarios.

Recently, several other significant advances have been made in deep learning, particularly in image recognition and object detection. Given the success of deep-learning techniques in solving similar challenges, they have the potential to offer more efficient traffic-accident-detection solutions [14]. Among these techniques, Convolutional Neural Networks (CNNs) have emerged as dominant. Yang et al. [15] proposed a Deep Convolutional Neural Network (DCNN) based on vehicle trajectory data for the detection and classification of six types of traffic accidents. However, the DCNN emphasizes local features within individual frames and fails to consider temporal features across video frames. Bortnikov et al. [16] proposed an accident-detection system based on a Three-Dimensional Convolutional Neural Network (3DCNN) that yielded positive results when trained on generated traffic videos. However, it was challenging to apply in practical traffic scenarios. Moreover, the spatiotemporal features extracted by the 3DCNN were independent from one another, and the system did not consider the correlations between different features. Consequently, CNN-based approaches have difficulty capturing the dynamic changes and interactions between objects over time, leading to decreased accuracy in detecting traffic accidents. To solve these problems, Recurrent Neural Networks (RNNs) were proposed for sequential data modeling. Crucial research has been conducted with traffic-accident approaches for traffic-accident detection [17]. Ijjina et al. [18] proposed a framework that utilized a Mask Regional Convolutional Neural Network (Mask R-CNN) [19] to determine whether a traffic accident has occurred based on vehicle speed and trajectory anomalies. However, the performance of this framework tended to decrease and lose parallelism as the length of the video-frame sequence increased.

The encoder–decoder and Transformer have recently garnered interest in various research fields, such as natural language processing [20], computer vision [21], and medical diagnosis [22]. The encoder–decoder is a component of machine-learning models designed to extract the most relevant information from the input data and encode and decode it into a sequence of fixed-length feature vectors [23]. The Transformer [24] is a neural network model that, unlike the RNN-based models, does not rely on sequential processing. Instead, it uses self-attention mechanisms to capture long-term dependencies in the input data, enabling the Transformer to process it in parallel. This leads to markedly faster training and inference times. These models offer solutions for overcoming the limitations of other approaches. The encoder–decoder extracts features from the input data, such as images or video frames, whereas the Transformer is designed to capture long-term dependencies and spatiotemporal relationships from sequential data. In the context of traffic-accident detection, the features of objects and relationships between objects in an image involves such relationships. Therefore, an approach that combines the encoder–decoder and Transformer is required to recognize an accident from a video or an image.

This paper proposes a novel framework for traffic-accident detection that involves subtracting the background by using a CNN encoder and a Transformer decoder. The proposed framework subtracts the background to generate bounding box masks and isolates the objects from the background in the video frames. Subsequently, a CNN encoder is used to extract spatial features, including the positions and dimensions of detected objects. Finally, a Transformer decoder is utilized to extract the temporal features and relationships between objects over time. The contributions of this paper are as follows:

- A framework that uses YOLOv5 to automatically subtract the road background, minimize background interference, and integrate the CNN encoder and Transformer decoder is proposed to jointly model spatiotemporal relationships between objects, thereby providing an improved understanding of traffic accidents to the model.
- Unlike with previous techniques, this method does not average or concatenate spatiotemporal features. Instead, it extracts features between different time points in the input video frames.
- The framework establishes an end-to-end model that allows parallel processing of the sequential input video frames, enabling the model to simultaneously output detection results for multiple frames. Therefore, it is feasible for application with large-scale datasets.

The remainder of this paper is organized as follows: Section 2 presents a review of recent research on traffic-accident detection, highlighting the advances and challenges in the field. In Section 3, we introduce the proposed framework, along with a detailed explanation of the implementation process, including the subtraction of the background, CNN encoder, and Transformer decoder. Section 4 presents the experimental results obtained from using the framework. Finally, Section 5 presents the conclusions of this paper and discusses potential future work.

2. Related Work

This section provides an overview of the latest research on traffic-accident detection, including a comparison of six different approaches. These studies were compared subsequently to highlight the unique advantages of the proposed framework.

Owing to rapid developments in machine learning, several approaches have been developed to detect and classify traffic accidents from input videos [25]. Early research focused on using traditional computer vision approaches, such as feature extraction [26] and object recognition [27], to detect and classify traffic accidents. Although these approaches were effective in detecting accidents, they relied on synthetic features and required a large amount of labeled training data. In recent years, there has been a shift toward the use of deep-learning approaches for traffic-accident detection. Zhou [28] proposed the Attention-Based Stack ResNet for Accident Prediction (ASRAP) framework based on feature vector extraction to detect accident distributions within a specific range. To detect the features, the ASRAP framework extracts the temporal dynamics of road features and fully utilizes the residual information for fitting. However, the training parameters are excessively large, resulting in high computational costs and reduced model interpretability. In another study, a deep-learning-based accident-detection framework was proposed [29] which used traffic data with different temporal resolutions and analyzed traffic trends, resulting in a satisfactory performance. The limitation of this framework is that it deletes a large volume of data that has a negative impact on the detection performance during data preprocessing, and the generalization ability of the model is poor. Huang et al. [30] proposed a two-stream convolutional network for the real-time detection of traffic accidents. In particular, they utilized a spatial stream network for object detection and a temporal stream network that analyzed the motion characteristics of objects for multi-object tracking. However, because it is based on 2D fisheye video data, a two-stream convolutional network cannot comprehend traffic scenarios and road objects. To strengthen this understanding, a baseline approach to accident detection was proposed [31]. This approach relies on an attention mechanism and a CNN to determine the current state of objects, such as safety or danger, while detecting them on the road. An attention R-CNN can concentrate on the local and global contexts of the traffic scenario but does not meet the need to analyze the temporal relationship between objects. Therefore, this paper proposes a framework that enhances the understanding of traffic scenarios and analyzes the temporal relationships among road objects and is designed for traffic-accident detection.

Table 1 presents a comparison of all related traffic-accident-detection studies and the proposed framework. Six studies on traffic-accident detection were compared based on factors such as the neural networks used, representation of data, and type of features. In contrast to the Gaussian Mixture Hidden Markov Model (GMHMM), Markov Random Fields (MRFs), ASRAP, the Long Short-Term-Memory-based framework considering traffic data of Different Temporal Resolutions (LSTMDTR), Two-Stream CNNs, and Attention R-CNN, the proposed framework not only considers spatial features but also temporal relationships among different objects within and across frames, providing a more comprehensive analysis of the input data and strengthening its understanding of traffic scenarios. Additionally, it enables the parallel detection of traffic accidents. This reduces the detection time significantly compared to the other approaches that process each frame individually.

Table 1. Differences between the recent traffic-accident-detection research and the proposed framework.

Recent Research	Neural Networks	Representation of Data	Type of Features	Parallel Structure	Strengthen Understanding
GMHMM [26]	-	Traffic Patterns	-	×	×
MRF [27]	-	Event Patterns	-	×	×
ASRAP [28]	ResNet, Attention	City Data	Spatiotemporal	×	√
LSTMDTR [29]	LSTM	Traffic Data	Temporal	×	×
Two-Stream CNNs [30]	CNN	Object Trajectories	Spatiotemporal	×	√
Attention R-CNN [31]	CNN, Attention	Object Bounding Boxes	Spatial	×	×
The Proposed Framework	CNN Encoder, Transformer Decoder	Bounding Box Masks	Spatiotemporal	√	√

3. Traffic-Accident-Detection Framework

In this paper, a framework is proposed to subtract the background and extract spatiotemporal features from video frames to detect traffic accidents. First, preprocessing of the input video frames is demonstrated. Subsequently, the bounding box masks obtained by subtracting the background are introduced. Finally, parallel processing for simultaneously detecting traffic accidents in multiple video frames using the CNN encoder and Transformer decoder models is explained.

3.1. Overview

The architectural design of the proposed framework is rooted in the need to efficiently process large volumes of video data and discern complex scenarios that correlate with traffic accidents. The proposed framework is primarily divided into two stages: a Bounding-Box-Masks Extractor and a Traffic-Accident Detector. In the Bounding-Box-Masks Extractor, video frames are extracted from the input video through the Video Preprocessor. YOLOv5 is utilized to detect objects in video frames represented by object bounding boxes. These object bounding boxes are then subtracted from the background to the extracted coordinates, using the Image Preprocessor, and the bounding box masks are generated using the Mask Generator. The background subtraction is performed to minimize the influence of irrelevant background or scenery on the detection results. This is an improvement over traditional traffic accident-detection approaches, which often struggle with background.

Existing models often focus on either spatial or temporal features, but not both simultaneously. Therefore, in the Traffic-Accident Detector, the CNN encoder extracts spatial features from individual video frames, including the positions and dimensions of different objects. The Transformer decoder extracts the temporal features across different frames in parallel. By combining these two models into an end-to-end model, the proposed framework outputs a traffic-accident-detection result of either "Accident" or "Non-accident". Figure 1 illustrates the process of traffic-accident detection using the Bounding-Box-Masks Extractor and the Traffic-Accident Detector.

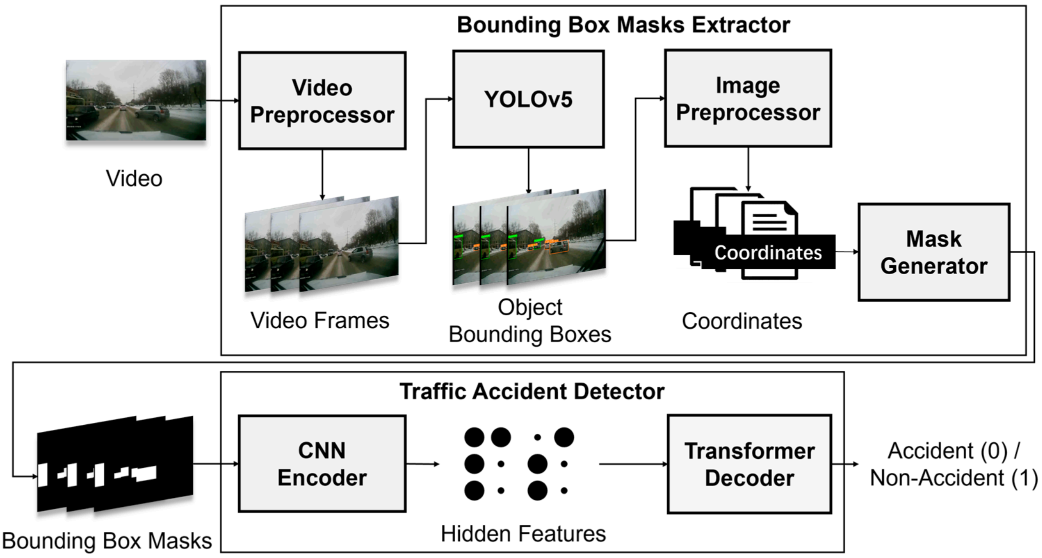


Figure 1. Stages of traffic-accident detection with Bounding-Box-Masks Extractor and Traffic-Accident Detector.

3.2. Bounding-Box-Masks Extractor

The YOLOv5 object-detection approach facilitates the processing of input video frames in real time and effectively handles complex and congested traffic scenarios. The exceptional capability of YOLOv5 enables object detection and tracking under diverse traffic scenarios. Figure 2 illustrates the structure of the Bounding-Box-Masks Extractor in the proposed framework, which is responsible for automatically extracting bounding box masks from the detected objects, using the YOLOv5 approach.

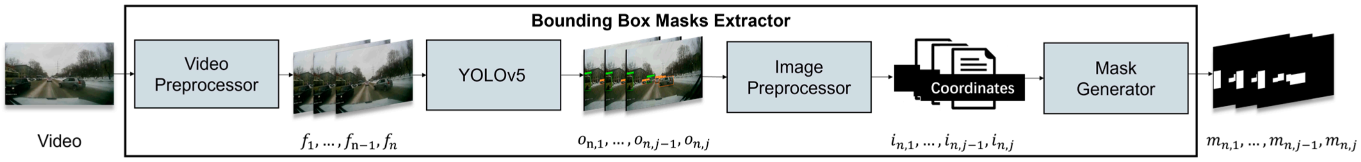

Bounding Box Masks Extractor
Video
Video Preprocessor
$f_1, \ldots, f_{n-1}, f_n$
YOLOv5
$o_{n,1}, \ldots, o_{n,j-1}, o_{n,j}$
Image Preprocessor
Coordinates
$i_{n,1}, \ldots, i_{n,j-1}, i_{n,j}$
Mask Generator
$m_{n,1}, \ldots, m_{n,j-1}, m_{n,j}$

Figure 2. Illustration of the structure of the Bounding-Box-Masks Extractor.

The input video is preprocessed to generate video frames in the Video Preprocessor. The size of the input video is 1280 × 720, and the frame rate is 25 fps. For each input video, the segment during which a traffic accident occurs is identified. This segment is then trimmed into a video clip containing 50 frames at the same resolution as the input video, which corresponds to 2 s. Subsequently, each of these 50 frames was manually annotated to denote whether the traffic accident is occurring in the frame. The input video frames, denoted as $f_1, \ldots, f_{n-1}, f_n$, are normalized and fed to the YOLOv5 in the Bounding-Box-Masks Extractor to detect all the dynamic objects in the traffic scenarios. The YOLOv5 filters bounding boxes of different scales to detect objects of different sizes and assign to them different identities. All the detected bounding boxes in each frame are denoted as $o_{n,1}, \ldots, o_{n,j-1}, o_{n,j}$, obtained by ensuring that each object in the input video frames is represented by one bounding box, where j indicates the j-th object bounding box. The coordinates of all detected bounding boxes in each frame, denoted as $i_{n,1}, \ldots, i_{n,j-1}, i_{n,j}$, are used to record the detected objects, and a Mask Generator is used to generate bounding box masks, denoted as $m_{n,1}, \ldots, m_{n,j-1}, m_{n,j}$. For each detected object, the Image Preprocessor returns a tuple consisting of five values, $(x_{n,j}, y_{n,j}, w_{n,j}, h_{n,j}, c_{n,j})$, that denote one bounding box. In this tuple, $x_{n,j}$ and $y_{n,j}$ represent the coordinates of the bounding box center, whereas $w_{n,j}$ and $h_{n,j}$ represent the width and height, respectively. The value range of $x_{n,j}$ and $w_{n,j}$ is [0, 720] pixels, whereas that of $y_{n,j}$ and $h_{n,j}$ is [0, 1280] pixels. The confidence score, $c_{n,j}$, denotes the probability of an object existing in the current bounding box, and the confidence score is in the interval [0, 1].

In this paper, the concepts of background and objects are defined to subtract the background. Specifically, the background refers to stationary or fixed elements in traffic scenarios, such as buildings, trees, and roads. The objects include elements that are in motion in a traffic scenario, such as vehicles, pedestrians, and animals. In the Image Preprocessor, the coordinates are manually defined based on the obtained object bounding boxes. First, boxes with a confidence score, $c_{n,j}$, of less than 0.6, as attributed by YOLOv5, are filtered out by the Image Preprocessor, and the rest are retained. Then, for cases in which an object is identified during the detection without a bounding box, the bounding box of nearby frames is copied as a supplement. For each bounding box mask, the confidence score is discarded, and only the center coordinates, $x_{n,j}, y_{n,j}$, and width and height, $w_{n,j}, h_{n,j}$, of the bounding boxes are retained. Figure 3 illustrates an example of a bounding-box mask. When the size of the input video frame is 1280 × 720, the detected object bounding box occupies an area of 75 pixels × 224 pixels (length × width). Therefore, the pixels within the stored object-bounding box region are shown in white, and the remaining pixels are shown in black. In the bounding box masks obtained after subtracting the background, each detected object is represented as a rectangle. The stored bounding box masks $m_{n,1}, \ldots, m_{n,j-1}, m_{n,j}$ are used as the input for the traffic-accident detector after the background is subtracted.

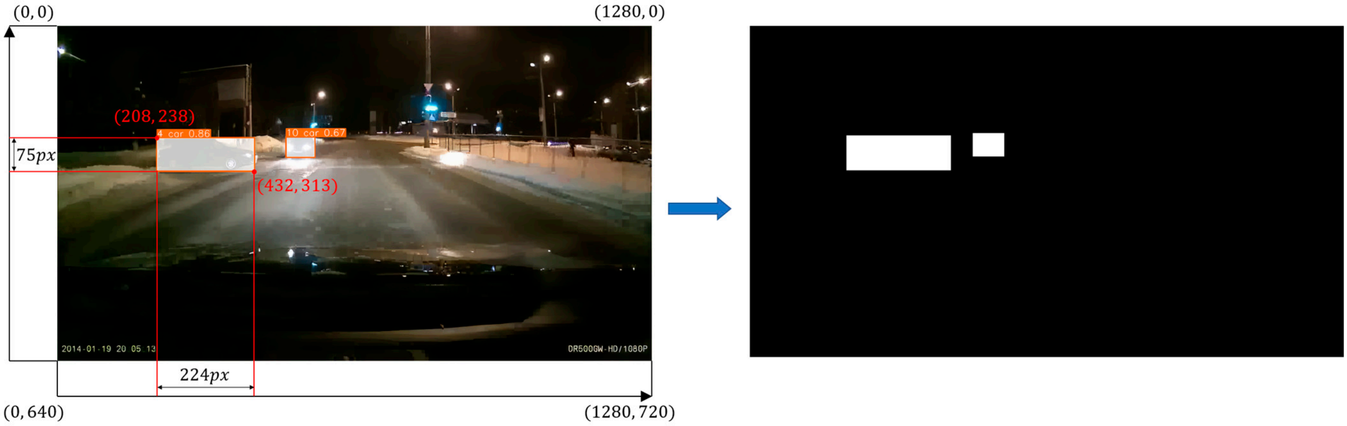


Figure 3. Example of the bounding box mask.

3.3. *Traffic-Accident Detector*

The CNN encoder in the proposed framework comprises several convolutional and max-pooling layers that are iteratively applied to process the input bounding box masks, denoted as $m_{n,j}$. Specifically, the input bounding box masks have a size of 224 × 224 pixels. The convolutional layers have filter sizes of 3 × 3 pixels, and the padding is set to 1 pixel to preserve the spatial dimensions of the feature maps. The max-pooling layers have a filter size of 2 × 2 pixels and a stride of 2 pixels, resulting in a reduction in the spatial dimensions of the feature maps and the extraction of the most important features. Figure 4 depicts the architecture of the CNN encoder. Every two convolutional layers are followed by a max-pooling layer used to down-sample the feature maps. Generally, a basic CNN can consist of alternating individual convolutional and pooling layers without the depth provided by stacking convolutional layers. Therefore, a CNN encoder is more suitable for spatial feature extraction because of the stacking of convolutional and max-pooling layers. The CNN encoder extracts the hidden features by successively applying these convolutional and max-pooling layers. The hidden features, denoted as $h^{m_{n,j}}$, are the intermediate data generated by the CNN encoder that extracts the spatial features of each input bounding box mask, including the positions and dimensions of the objects. The size changes of the hidden features in the CNN encoder are as follows: 112 × 112, 56 × 56, and 28 ×28. After extracting all bounding box masks, these hidden features are used as input to the Transformer decoder, where they are processed to extract the temporal features among the bounding box masks and execute the final traffic-accident detection. The continuous convolution and max-pooling processes within the CNN encoder can be defined mathematically as follows:

$$h^{m_{n,j}} = M(F(a_k \times (M(F(a_k \times (M(F(a_k \times m_{n,j} + b_k)) + b_k)) + b_k)))), \tag{1}$$

where F denotes the activation function, which is a Rectified Linear Activation Unit (ReLU), M is the max-pooling operation, a is the convolution-layer weight matrix, and b is the bias vector. Initially, k-th convolution operation $(a_k \times m_{n,j} + b_k)$ is applied to the input $m_{n,j}$, followed by the activation function F, and finally the max-pooling operation M. This procedure is repeated until the final convolution and max-pooling operations have been executed.

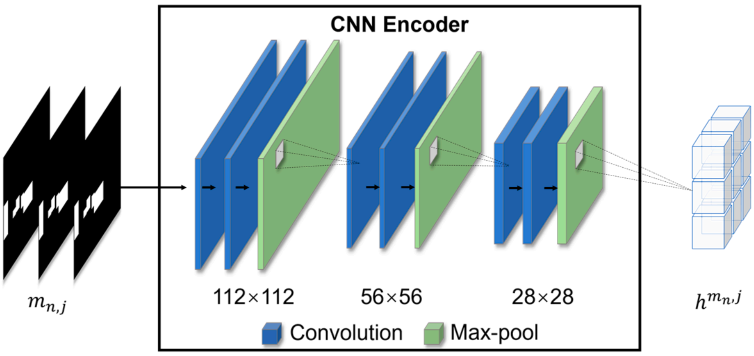


Figure 4. The architecture of the CNN encoder.

Figure 5 illustrates the architecture of the Transformer decoder. The hidden features denoted as $h^{m_{n,j}}$ of $m_{n,j}$ generated by the CNN encoder are used as inputs to the Transformer decoder, which consists of a linear layer, multi-head attention mechanism, add-and-norm layers, and a feed-forward network. The design of the Transformer decoder in this paper includes several modifications with respect to the original Transformer. The original Transformer is a dual structure, consisting of both an encoder and a decoder, whereas the proposed Transformer decoder employs only the decoder. Further adaptations include the integration of two additional linear layers within the Transformer decoder. In the proposed framework, linear layers are included before and after the Transformer decoder. In the first linear layer, the hidden input features are mapped to a high-dimensional space and resized to 256 neurons to fit the subsequent layers after processing. The second and final linear layer consists of two neurons and acts as the output layer. It maps the hidden features to the probability distribution of the target classes and outputs the classification results of multiple video frames of traffic accidents simultaneously, denoted as $r_1, \ldots, r_{n-1}, r_n$. For each frame, the traffic-accident-detection results are passed through a softmax layer to ensure that the value lies between 0 and 1. A value closer to 1 indicates a higher probability that an accident is occurring in that frame, while a value closer to 0 indicates a lower probability. The multi-head attention mechanism extracts temporal features from the input hidden features. Specifically, it can process the input hidden features in parallel and weigh and combine information from different positions to obtain global temporal features. The Transformer decoder permits a configurable number of attention heads in the multi-head attention mechanism. It consists of eight heads with a hidden size of 64. The multi-head attention mechanism can be expressed as follows:

$$h^{m_{n,j}} = softmax(\frac{(h_q^{m_{n,j}} \times h^{m_{n,j}}) \times (h_k^{m_{n,j}} \times h^{m_{n,j}})^T}{\sqrt{d_{h_k^{m_{n,j}}}}}) \times (h_v^{m_{n,j}} \times h^{m_{n,j}}), \tag{2}$$

where $h_q^{m_{n,j}}$, $h_k^{m_{n,j}}$, and $h_v^{m_{n,j}}$ denote the query, key, and value matrices, respectively, embedded in the hidden input features. The softmax function, $softmax(\cdot)$, is applied to the scaled dot product between the query, $h_q^{m_{n,j}}$, and key matrices, $h_k^{m_{n,j}}$, and it is then multiplied by the value matrix $h_v^{m_{n,j}}$. In addition, $d_{h_k^{m_{n,j}}}$ denotes the dimensionality of the key, and the square root is used to scale the dot product. T is the transpose symbol.

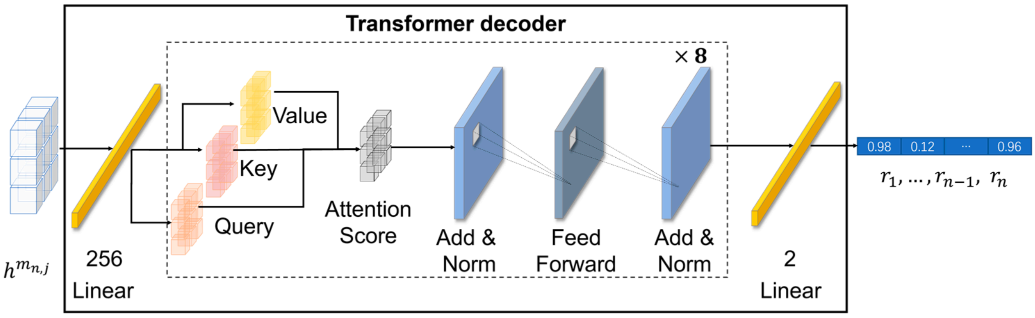


Figure 5. The architecture of the Transformer decoder.

The add-and-norm layer contains a residual connection and normalization function, which are applied after the multi-head attention mechanism and feedforward network to stabilize the learning process. The feedforward network is composed of two linear transformations with a ReLU activation function between and is responsible for transforming the attention-weighted hidden features.

4. Experiment

This section describes the experiments conducted to evaluate the proposed framework for traffic-accident detection. An environment is presented to demonstrate the effectiveness of the proposed framework. The experimental dataset was analyzed to ensure a balanced distribution of accident and non-accident samples. Finally, the experimental results were compared with three other models for traffic-accident detection, and an ablation experiment was conducted.

4.1. Experimental Environment

The experiment was conducted using a model based on the proposed framework as a traffic-accident detector and three additional models for comparison. The models included a DCNN [15], an LSTMDTR [29], the Vision Transformer for Traffic Accidents (ViT-TA) [32], and the proposed framework. Table 2 lists the hyperparameters used for the training and validation of each model. These models were trained using bounding box masks as inputs, which were uniformly cropped to 224 × 224 pixels. The models had several shared hyperparameters, including the maximum sequence length, learning decay rate, total training epochs, and objective function. The maximum sequence length was set to 256, with a dropout of 0.3 and a batch size of 128. In addition, the dropout of LSTMDTR was 0.4, and the batch size of ViT-TA was set to 64. The initial learning rate of the proposed framework was set to a relatively small value of 2×10^{-6}. To facilitate convergence to the other models, the learning rates were fine-tuned specifically for each model as follows: 1×10^{-5} for DNN, 3×10^{-4} for LSTMDTR, and 3×10^{-6} for ViT-TA. The learning rate decayed according to the cosine method during the training process, with a decay rate of 1×10^{-4}. The total number of training epochs was 1000, and the optimizer of the proposed framework and DCNN was consistently set to Stochastic Gradient Descent (SGD). The optimizer of LSTMDTR was Adaptive Moment Estimation (Adam), and that of the ViT-TA was Rectified Adaptive Moment Estimation (RAdam). All models used softmax as the objective function. The attention heads were set to eight, but this was only applicable to the proposed framework and ViT-TA model. The training speeds for each model were also provided, indicating the number of iterations per second achieved during training. The training speeds of the proposed framework, DCNN, LSTMDTR, and ViT-TA were 4.06, 1.87, 2.09, and 3.56 iterations per second (it/s), respectively. The proposed framework exhibited a faster training speed and more efficient detection performance than the other models.

Table 2. Parameters for training the proposed framework and other traffic-accident-detection models.

Hyperparameter	Proposed Framework	DCNN [15]	LSTMDTR [29]	ViT-TA [32]
Bounding Box Masks Dimension	(224, 224)	(224, 224)	(224, 224)	(224, 224)
Max Sequence Length	256	256	256	256
Attention Heads	8	-	-	8
Dropout	0.3	0.3	0.4	0.3
Batch Size	128	128	128	64
Learning Rate	2×10^{-6}	1×10^{-5}	3×10^{-4}	1×10^{-6}
Decay learning rate	2×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Total Training Epochs	1000	1000	1000	1000
Optimizer	SGD	SGD	Adam	Radam
Objective Function	Softmax	Softmax	Softmax	Softmax
Training Speed	4.06 it/s	1.87 it/s	2.09 it/s	3.56 it/s

The experiments were conducted on an Ubuntu 20.04.1 LTS operating system powered by a six-core Intel i7-6850K processor and an Nvidia Titan RTX (48 GB) graphics card. The source code was written in Python 3.6, utilizing various libraries: PyTorch 1.10.0 for deep-learning model building, NumPy 1.19.5 for numerical computations, OpenCV 4.5.5.64, Pillow 6.0.0 for input video frame preprocessing, and CUDA version 11.6. Each library served a specific function that contributed to the execution of the experiments.

4.2. Experimental Data

The Car Crash Dataset (CCD) [33] is a collection of annotated videos specifically designed for the study and development of traffic-accident-detection algorithms. Each video in the CCD is annotated with relevant information, such as the location of vehicles and pedestrians, type of accident, and time of the accident. In these experiments, the CCD underwent a data-cleaning process that involved removing instances with erroneous labels and low-resolution videos. Table 3 lists the components of the CCD.

Table 3. Contents of the Car Crash Dataset.

Car Crash Dataset	Value
Preprocessed Videos	179
Frames per Videos	50
Total Coordinates Files	179
Total Frames	8950
Frames Labeled as Accident	2496
Frames Labeled as Non-Accident	6454
Total Bounding Box Masks	8950
Training Data	7160 (80%)
Validation Data	1790 (20%)

A subset of the refined CCD was then created to train and validate the proposed traffic-accident-detection framework. This approach ensured the reliability and accuracy of the experimental results derived from the proposed framework. The dataset was preprocessed to include 179 videos, each containing 50 frames, resulting in 8950 frames. Of these, 2496 were labeled "accident frames", and 6454 were labeled "non-accident frames." Each frame in the dataset was accompanied by a corresponding bounding box mask generated by subtracting the background, thereby providing 8950 bounding box masks. The dataset was divided into training and validation datasets. The training set consisted of 80% of data, amounting to 7160 frames, whereas the remaining 20% was reserved for the validation set, amounting to 1790 frames.

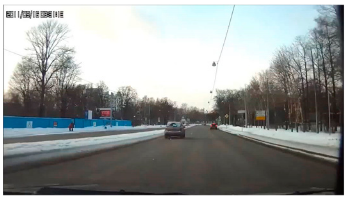
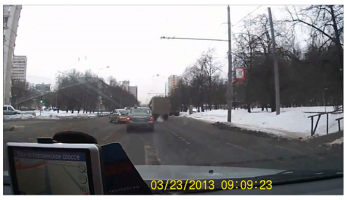
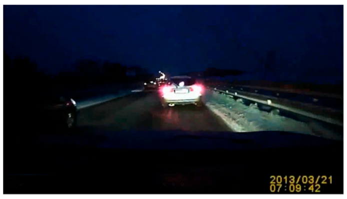
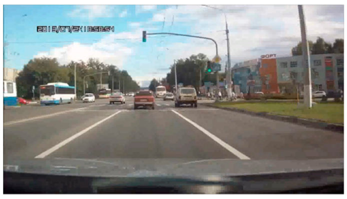
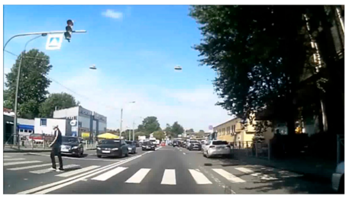
In this paper, the videos were categorized based on various traffic scenarios. Table 4 lists the video-frame data for the various traffic scenarios. The dataset encompassed a diverse array of real-world traffic-accident scenarios. Specifically, the dataset comprised six distinct types of traffic situations: daytime and nighttime weather conditions, snowy and rainy lighting conditions, and low and high traffic volumes. With regard to the latter, the high traffic volumes represent congested road traffic conditions. Scenes with high traffic volumes present unique challenges for accident detection due to the increased number of vehicles and their interactions. However, the efficiency of YOLOv5, which is capable of detecting any number of vehicles within a video frame, provides a robust detection process that is essential for background subtraction. Additionally, the multi-head attention mechanism within the Transformer decoder ensures optimized performance even in situations involving multiple object interactions. This robustness of the proposed framework affirms its ability to detect traffic accidents even under congested road conditions.

The variety in the dataset provided a comprehensive and challenging set of testing data for evaluating traffic-accident detection. Video frames from these diverse scenarios were analyzed, allowing us to detect traffic accidents under various complex environmental conditions. The inclusion of multiple scenarios improved the ability of the proposed framework to detect traffic accidents and prevent overfitting to specific scenarios.

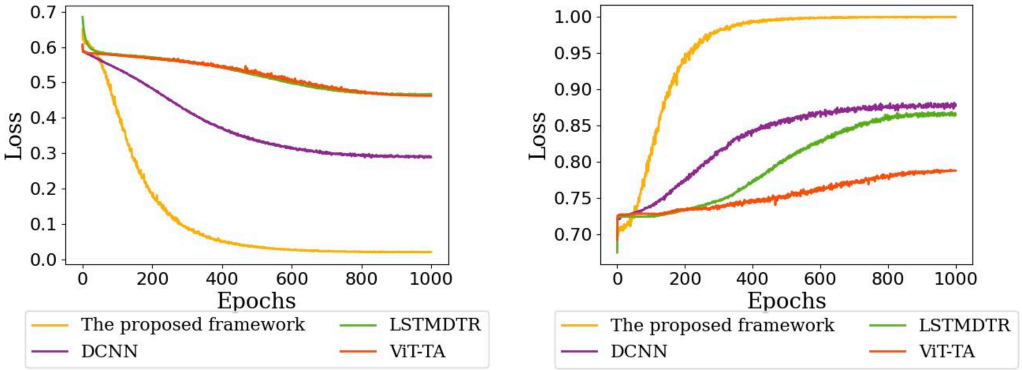
4.3. Experimental Results

Figure 6 depicts the loss and accuracy convergence plots, demonstrating the training results of the proposed framework and traffic-accident-detection models DNN, LSTMDTR, and ViT-TA. The experiment required 1000 epochs to complete the training process. This allowed the proposed framework and other models to adapt effectively to the diverse traffic scenarios presented in the training datasets. For the proposed framework, as indicated in Figure 6a, the training loss begins at 0.65 and converges to a low value after approximately 400 epochs. After 1000 training epochs, the loss value decreases to 0.02. The training accuracy, as shown in Figure 6b, begins at 0.68, increases to a high value after approximately 400 epochs and continues to gradually increase to 0.99 by the end of the 1000th epoch. In contrast, DNN, LSTMDTR, and ViT-TA exhibited slower convergence rates and lower final accuracy values. For instance, the DNN exhibited a final training loss of 0.29 and an accuracy of 0.88, whereas the LSTMDTR reached a loss of 0.46 and an accuracy of 0.86. The ViT-TA model had a final loss of 0.46 and an accuracy of 0.78. The results illustrated in Figure 6 demonstrate the superior performance of the proposed framework when compared with the other models. The proposed method not only converged faster but also achieved higher accuracy in detecting traffic accidents. The performance of the proposed framework in detecting traffic accidents demonstrates its potential for practical applications in various traffic scenarios.

Table 4. The various traffic scenarios in Car Crash Dataset.

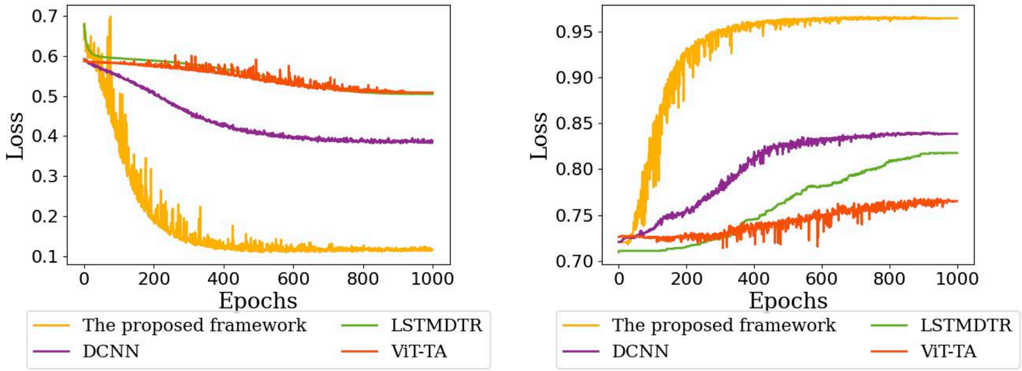
Traffic Scenarios	Frame	
Daytime		
Nighttime		
Snowy		
Rainy		
Low Traffic Volumes		
High Traffic Volumes		

In a comparative validation analysis, the proposed framework was assessed against other traffic-accident-detection models. The performances of these models were evaluated based on their respective validation losses and accuracy curves, as shown in Figure 7. As indicated in Figure 7a, for the proposed framework during the validation phase, the loss begins at 0.64 and reaches a low value after approximately 300 epochs. After 1000 epochs, the validation loss decreased to 0.11. In Figure 7b, the validation accuracy begins at 0.72, achieves a high value at approximately 300 epochs, and then steadily rises to 0.96 by the end of the 1000th epoch. However, the three other detection models demonstrated a less acceptable validation performance. For instance, the DNN model attained a final validation loss of 0.39 and an accuracy of 0.84, whereas the LSTMDTR reported a loss of 0.50 and an accuracy of 0.81. The ViT-TA model displayed a final validation loss of 0.50 and an accuracy of 0.76. As demonstrated in Figure 7, the proposed framework outperformed the other models in terms of validation loss and accuracy. This not only highlights the superior performance of the proposed framework but also indicates its robustness in detecting traffic accidents.



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

Figure 6. Comparative training results for the proposed framework and other traffic-accident-detection models. (**a**) Loss curve. (**b**) Accuracy curve.



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

Figure 7. Comparative validation results for the proposed framework and other traffic-accident-detection models. (**a**) Loss curve. (**b**) Accuracy curve.

4.4. Ablation Experimental Results

In this section, the outcomes of an ablation experiment are presented to evaluate the influence of background subtraction on the performance of the proposed traffic-accident-detection framework. The ablation experiment was used to compare the performance of the model with and without background subtraction. All hyperparameters in the experiment were consistent with those of the proposed framework. The training and validation results of the ablation experiments are highlighted in Figure 8. The purpose of this experiment was to assess the performance of the proposed framework without background subtraction. Figure 8a depicts the training and validation loss curves for the proposed framework when the background is not subtracted. In this case, the training loss is approximately 0.03, whereas the validation loss is approximately 0.23. Although the training loss was nearly identical to that of the proposed method, the validation loss was higher by 0.12. Higher loss values signify a less effective learning process. Figure 8b reveals the accuracy curves for the training and validation stages. The lack of background subtraction resulted in a decline in the overall accuracy, as the model could not accurately distinguish objects from the background. After 1000 epochs, the training and validation accuracies reached approximately 0.98 and 0.91, respectively. Compared with the experimental results of the proposed method, the training accuracy decreased by only 0.01, whereas the validation accuracy dropped by 0.05. These results suggest that integrating background subtraction is essential for reducing losses during the training phase and improving the traffic-accident-detection performance of the proposed framework.

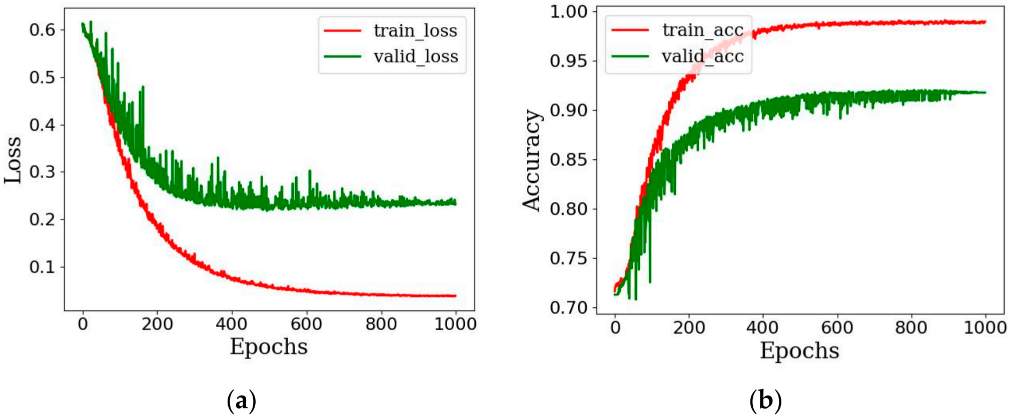


Figure 8. The training and validation results of the ablation experimental. (**a**) Loss of the proposed framework without background subtraction. (**b**) Accuracy of the proposed framework without background subtraction.

The evaluation indicators of the ablation experiments are summarized in Table 5. The proposed framework with background subtraction achieved a precision of 0.98, a recall of 0.98, an F1-score of 0.97, and an overall accuracy of 0.96. In contrast, when the background subtraction was not applied, these indicators decreased to 0.95, 0.95, 0.96, and 0.91, respectively. The results of the ablation experiment clearly demonstrate that the integration of background subtraction into the proposed framework significantly improved its performance in detecting traffic accidents, leading to more accurate and reliable traffic-accident detection.

Table 5. Evaluation indicator results for ablation experiment.

Method	Precision	Recall	F1-Score	Accuracy
Subtracting Background	0.98	0.98	0.97	0.96
Without Subtracting Background	0.95	0.95	0.96	0.91

5. Discussion

The efficacy of the proposed framework in detecting traffic accidents has two implications. First, it shows that the framework generates pertinent data that can be utilized to improve an automatic accident-detection system by providing an analysis of the objects in various traffic-accident scenarios. Second, the insights derived from the spatiotemporal analysis of traffic accidents can enable traffic safety measures to be modified and updated, potentially minimizing accident occurrences. These insights can be derived using the detection capabilities of the proposed framework, which was tested and proven effective in a variety of complex traffic scenarios. In addition to aiding in the improvement of road safety measures, the proposed framework could be applied in real-driving simulators, in which detected accidents can be simulated, which would also contribute to the development of autonomous driving systems. By providing traffic detection results for different traffic scenarios, the proposed framework would enable autonomous driving systems to better recognize and handle complex traffic situations in order to improve its performance.

6. Conclusions

This paper proposed a new traffic-accident-detection framework that combines a CNN encoder with a Transformer decoder to subtract the background from a video in order to detect the object. The proposed framework comprises two stages: the Bounding-Box-Masks-Extractor and Traffic-Accident-Detector stages. During the Bounding Box-Masks-Extractor stage, YOLOv5 is used to automatically detect bounding boxes in the traffic scenario extracted from an input video frame. The coordinates are then defined for all bounding boxes. Subsequently, these coordinates are analyzed to subtract the background and generate the corresponding bounding box masks. In the Traffic-Accident-Detector stage, the CNN encoder conducts convolutional and max-pooling operations on the bounding box masks to extract spatial features. In addition, the Transformer decoder is responsible for extracting the spatiotemporal features. Finally, the Transformer decoder simultaneously analyzes multiple frames of traffic-accident-detection results to determine whether an accident has occurred in each frame. The performance of the proposed framework was evaluated by comparing its effectiveness with and without background subtraction to assess its impact on accuracy and robustness. The ablation experiments reveal that the proposed framework demonstrated a 5% increase in accuracy for traffic-accident detection compared to the approach without background subtraction. Furthermore, a comparison of the accuracy of the proposed framework with other traffic-accident-detection models revealed that it offered a higher accuracy rate, reaching 96%. Future work on this paper will include the following: (1) the proposed framework will be tested on larger, more diverse datasets to enhance the accuracy of traffic-accident detection; (2) the robustness of the proposed framework will be assessed under various traffic conditions and scenarios; and (3) the potential for integrating the proposed framework with other traffic monitoring or safety systems will be investigated.

Author Contributions: Methodology, Y.Z. and Y.S.; conceptualization, Y.Z. and Y.S.; validation, Y.Z. and Y.S.; formal analysis, Y.Z. and Y.S.; resources, Y.Z. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the 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 do not issue DOIs. Publicly available datasets were analyzed in this paper. This data can be found here: https://github.com/Cogito2012/CarCrashDataset (accessed on 31 March 2023).

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

References

1. Road Safety Facts. Available online: http://www.asirt.org/safe-travel/road-safety-facts/ (accessed on 6 April 2023).
2. Damjanović, M.; Stević, Ž.; Stanimirović, D.; Tanackov, I.; Marinković, D. Impact of the Number of Vehicles on Traffic Safety: Multiphase Modeling. *Facta Univ. Ser. Mech. Eng.* **2022**, *20*, 177–197. [CrossRef]
3. 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*, 2274–2290. [CrossRef]
4. 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]
5. Qiu, L.; Li, S.; Sung, Y. DBTMPE: Deep Bidirectional Transformers-based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics* **2021**, *9*, 530. [CrossRef]
6. Zhaoyou, M.; Changjun, W.; Shouen, F.; Shuo, L. Comparative Analysis and Control Strategy for Traffic Accidents in Different Types of Tunnels. In Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS), Liverpool, UK, 14–17 July 2019; pp. 1132–1136.
7. Chen, X.; Wu, S.; Shi, C.; Huang, Y.; Yang, Y.; Ke, R.; Zhao, J. Sensing Data Supported Traffic Flow Prediction via Denoising Schemes and ANN: A comparison. *IEEE Sens. J.* **2020**, *20*, 14317–14328. [CrossRef]
8. Ji, S.; Xu, W.; Yang, M.; Kai, Y. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef] [PubMed]
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
10. Jin, X.; Srinivasan, D.; Cheu, R.L. Classification of Freeway Traffic Patterns for Incident Detection using Constructive Probabilistic Neural Networks. *IEEE Trans. Neural Netw.* **2001**, *12*, 1173–1187. [CrossRef] [PubMed]
11. Liu, G.; Jin, H.; Li, J.; Hu, X.; Li, J. A Bayesian Deep Learning Method for Freeway Incident Detection with Uncertainty Quantification. *Accid. Anal. Prev.* **2022**, *176*, 106796. [CrossRef] [PubMed]

12. Hadi, R.A.; George, L.E.; Mohammed, M.J. A Computationally Economic Novel Approach for Real-Time Moving Multi-Vehicle Detection and Tracking Toward Efficient Traffic Surveillance. *Arab. J. Sci. Eng.* **2017**, *42*, 817–831. [CrossRef]
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
14. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent Advances in Convolutional Neural Networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
15. Yang, D.; Wu, Y.; Sun, F.; Chen, J.; Zhai, D.; Fu, C. Freeway Accident Detection and Classification based on the Multi-Vehicle Trajectory Data and Deep Learning Model. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103303. [CrossRef]
16. Bortnikov, M.; Khan, A.; Khattak, A.M.; Ahmad, M. Accident Recognition via 3D CNNs for Automated Traffic Monitoring in Smart Cities. In Proceedings of the 2019 Computer Vision Conference (CVC), Las Vegas, NV, USA, 25–26 April 2019; pp. 256–264.
17. Tian, D.; Zhang, C.; Duan, X.; Wang, X. An Automatic Car Accident Detection Method based on Cooperative Vehicle Infrastructure Systems. *IEEE Access* **2019**, *7*, 127453–127463. [CrossRef]
18. Ijjina, E.P.; Chand, D.; Gupta, S.; Goutham, K. Computer Vision-Based Accident Detection in Traffic Surveillance. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
19. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
20. Humeau, S.; Shuster, K.; Lachaux, M.A.; Weston, J. Poly-Encoders: Transformer Architectures and Pre-Training Strategies for Fast and Accurate Multi-Sentence Scoring. *arXiv* **2019**, arXiv:1905.01969.
21. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in Transformer. In Proceedings of the 2021 35th Advances in Neural Information Processing Systems (NIPS), Virtual, 7–10 December 2021; Volume 34, pp. 15908–15919.
22. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Alan, Y.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306.
23. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder–Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, T.; Polosukhin, I. Attention is All You Need. In Proceedings of the 2017 31st Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
25. Chan, F.H.; Chen, Y.T.; Xiang, Y.; Sun, M. Anticipating Accidents in Dashcam Videos. In *Revised Selected Papers, Part IV 13, Proceedings of the 13th Asian Conference on Computer Vision (ACCV), Taipei, Taiwan, 20–24 November 2016*; Springer International Publishing: Berlin/Heidelberg, Germany; pp. 136–153.
26. Li, X.; Porikli, F.M. A Hidden Markov Model Framework for Traffic Event Detection Using Video Features. In Proceedings of the IEEE 11th International Conference on Image Processing (ICIP), Singapore, 24–27 October 2004; pp. 2901–2904.
27. Kamijo, S.; Matsushita, Y.; Ikeuchi, K.; Sakauchi, M. Traffic Monitoring and Accident Detection at Intersections. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 108–118. [CrossRef]
28. Zhou, Z. Attention based Stack ResNet for Citywide Traffic Accident Prediction. In Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM), Hong Kong, China, 10–13 June 2019.
29. 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]
30. 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]
31. 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), Melbourne, Australia, 7–11 September 2020; pp. 313–320.
32. Kang, M.; Lee, W.; Hwang, K.; Yoon, Y. Vision Transformer for Detecting Critical Situations and Extracting Functional Scenario for Automated Vehicle Safety Assessment. *Sustainability* **2022**, *14*, 9680. [CrossRef]
33. Bao, W.; Yu, Q.; Kong, Y. Uncertainty-Based Traffic Accident Anticipation with Spatio–Temporal Relational Learning. In Proceedings of the 28th ACM International Conference on Multimedia, New York, NY, USA, 12–16 October 2020; pp. 2682–2690.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.