electronics

MDPI

Article

Detection of Dangerous Human Behavior by Using Optical Flow and Hybrid Deep Learning

Laith Mohammed Salim [1] and Yuksel Celik [2,*]

[1] Computer Engineering Department, Faculty of Engineering, Karabuk University, 78050 Karabuk, Turkey; 2038166021@ogrenci.karabuk.edu.tr
[2] Information Security and Digital Forensics, University at Albany—State University of New York, New York, NY 12222, USA
* Correspondence: ycelik@albany.edu

Abstract: Dangerous human behavior in the driving sense may cause traffic accidents and even cause economic losses and casualties. Accurate identification of dangerous human behavior can prevent potential risks. To solve the problem of difficulty retaining the temporal characteristics of the existing data, this paper proposes a human behavior recognition model based on utilized optical flow and hybrid deep learning model-based 3D CNN-LSTM in stacked autoencoder and uses the abnormal behavior of humans in real traffic scenes to verify the proposed model. This model was tested using HMDB51 datasets and JAAD dataset and compared with the recent related works. For a quantitative test, the HMDB51 dataset was used to train and test models for human behavior. Experimental results show that the proposed model achieved good accuracy of about 86.86%, which outperforms recent works. For qualitative analysis, we depend on the initial annotations of walking movements in the JAAD dataset to streamline the annotating process to identify transitions, where we take into consideration flow direction, if it is cross-vehicle motion (to be dangerous) or if it is parallel to vehicle motion (to be of no danger). The results show that the model can effectively identify dangerous behaviors of humans and then test on the moving vehicle scene.

Keywords: human behavior recognition; human activity recognition; optical flow; deep learning; stacked autoencoder

check for updates

Citation: Salim, L.M.; Celik, Y. Detection of Dangerous Human Behavior by Using Optical Flow and Hybrid Deep Learning. *Electronics* **2024**, *13*, 2116. https://doi.org/10.3390/electronics13112116

Academic Editors: Mustafa Ilhan Akbas, Tahir Cetin Akinci, Gokhan Erdemir and Oguzhan Topsakal

Received: 25 March 2024
Revised: 9 May 2024
Accepted: 18 May 2024
Published: 29 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Human behavior recognition is a popular research direction in the field of computer vision. With the continuous reduction in the application cost of digital cameras and video cameras and the extensive use of smartphones, it is no longer challenging to shoot and obtain video image data [1]. Studying video and image content is a core issue in the fields of computer vision and multimedia. In videos and images, human behavior is at the core of high-precision research on video image content [2]. Therefore, identifying human behavior in videos and images is a very valuable topic in computer vision problems, and its analysis results can be applied to intelligent video surveillance, human–computer interaction, and other fields [3].

The behavioral characteristics of human behavior throughout driving senses, especially when crossing the street, can be roughly divided into four types: One is the standard type, which always maintains a uniform pace and moves forward steadily [4]. The second is the mid-stop type. When pedestrians cross the road, they see many vehicles and stop or hesitate. The third is the speed-up type. This type of pedestrian crossing usually occurs after walking to the road's center line [5]. When they see a car approaching quickly, they speed up to cross the street. The fourth is the slow-down type. This type of pedestrian usually runs quickly to cross the street. When they reach the center line, they see no cars on the road, so they slowly move steadily. Among these four types of crossings, the last three types have certain dangers for safe crossing because the subsequent behavior of pedestrians

crossing the street at this time is often difficult for drivers to predict and estimate. It is easy for drivers to make judgment mistakes, leading to operational errors, thus leading to tragedy [6].

From the perspective of the human eye, it is easier for drivers to determine the intentions of pedestrians, but based on sensor technology such as cameras, this intuitive judgment is not easy. In an autonomous driving scenario, the vehicle is cautious when judging pedestrians. If it cannot judge the trajectory of subsequent movement, the vehicle must slow down when encountering a pedestrian [6]. The reason is that the machine cannot rule out the possibility of pedestrians wandering in front of the vehicle. A start-up company is now using machine learning to change and improve the vehicle perception system's ability to judge the movement trajectories of surrounding pedestrians and cyclists [7].

Judging from the industry's technological development, human behavior capabilities have made significant progress and improvement, but the real core of autonomous driving lies in decision making and control [8]. Typically, machine learning techniques use data that can be objectively measured to train algorithms. However, judging pedestrian movement intentions and trajectories is impractical under such circumstances. In contrast, trajectory awareness relies on human subjective judgment to provide the data used to train its algorithms [9].

In this work, we developed a dangerous human behavior detection using optical flow and deep learning autoencoder. However, in this work, we take into consideration the direction of a human in x direction (perpendicular direction on driving scene) as the dangerous human behavior that means the human crosses the street when the car is moving. The contribution of our work is to propose a new human activity and human behavior recognition system based on hybrid 3D bidirectional CNN-LSTM with stacked autoencoder. This model will be tested using HMDB51 datasets and JAAD dataset and compared with the recent related works. For feature extraction, we chose a stacked autoencoder neural network. Stacked autoencoders can function as logical filters to extract useful characteristics for a specific neural network learning job and can improve feature selection in deep learning models by reducing the number of features and improving detection accuracy.

2. Related Works

Several studies have been conducted in the field of human behavior. Recently, most studies have focused on utilizing deep learning and machine learning techniques to improve human behavior detection and recognition.

Mukherjee et al. (2020) [10] presented a novel ensemble that includes four classification models: CNN-Net, Encoded-Net, CNN, and LSTM. This ensemble is referred to as Ensem-ConvNet. Each classification model mentioned above is developed based on a basic 1D convolutional neural network (CNN). However, these models vary in terms of the quantity of dense layers, the kernel size used, and other significant architectural variations. Each model analyzes time series data in the form of a 2D matrix, where a window of data is analyzed at a given moment to extract relevant information and make predictions about the specific human activity. To determine its classification conclusion, the Ensem-ConvNet model employs several classifier combination methods, such as majority voting, sum rule, product rule, and an adaptive weighted strategy termed score fusion. Their proposed model was evaluated using three benchmark datasets: WISDM activity forecasting, UniMiB SHAR, and MobiAct. They conducted a comparative analysis of their EnsemConvNet model with several established deep learning models, including multiheaded CNN (a hybrid of CNN and LSTM) models. The results discussed in this study show that higher accuracy of 97.7% is reached by the Ensem-ConvNet framework compared to the other models mentioned.

Zhang et al. (2020) [10] presented a new method for localizing spatiotemporal actions, which includes a combined optical flow sub-network for addressing these two problems. Our flow subnet is specifically intended to estimate optical flow with high efficiency and accuracy. Unlike traditional methods that use only two nearby frames, our subnet utilizes

many successive RGB frames in a deep network. Additionally, our network incorporates action localization, allowing for interactive flow calculation within the same network, resulting in an end-to-end solution. In order to increase the speed, we utilize a pyramid hierarchical approach to feature fusion, employing a neural-network-based method. The method combines spatial and temporal characteristics at several levels of detail using a concatenation function and point-wise convolution. This allows for the extraction of multiscale spatiotemporal activity features. The proposed method achieves state-of-the-art performance in terms of efficiency and accuracy, as demonstrated by experimental findings on three publicly available datasets: UCF101-24, JHMDB, and AVA. These results were obtained by utilizing both RGB appearance and optical flow cues. Notably, it experiences a substantial enhancement in efficiency. When compared with the currently most effective method, this method is 1.9 percent faster in terms of running speed and achieves a 1.3% higher video-mAP accuracy on the UCF101-24 dataset. Ihianle et al. (2020) [11] presented a method for recognizing the presence of HAR based on deep learning techniques. The proposed algorithms were CNN and Bi-LSTM-based. Bi-LSTM demonstrated its superiority by traveling forward and backward through a predetermined order to enhance the extracted features. The final model was proposed based on multichannel convolution bidirectional LSTM (MCBLSTM). Two primary datasets, MHEALTH and WISDM, were used to evaluate the efficiency of the proposed methods. Using both datasets, the accuracy proposed by MCBLSTM demonstrated the best performance, with a value of 97.7%. Prabono et al. (2021) [11] presented a novel methodology for domain adaptation in the context of human activity recognition. The proposed strategy utilizes a two-phase autoencoder-based framework, where their model achieved 67.9% in average accuracy. Furthermore, Garcia et al. (2021) [12] introduced a highly efficient multiclass approach comprising an ensemble of autoencoders, each specifically linked to an individual class. Utilizing a modular framework for classifiers enhances the adaptability of models in accommodating more classes. This process requires the inclusion of new autoencoders, eliminating the need to retrain the entire model. The results showed that the model achieved 71% in average accuracy.

Huang (2021) [13] first proposed a shallow CNN that analyzes cross-channel communication in an HAR scenario, in which all channels in the same layer interact comprehensively to capture more distinguishing features of sensor input. Using a graph neural network, one channel can interact with all other channels to eliminate redundant information collected among channels, which is more advantageous for implementing lightweight deep models. Extensive experiments on multibenchmark HAR datasets, including UCI-HAR, OPPORTUNITY, PAMAP2, and UniMib-SHAR, demonstrated that the proposed approach enables shallower CNNs to gather more helpful information than baseline deep networks together with competing methods. The pace of inference was measured by deploying HAR systems in an embedded system. The results showed that the model achieved 92.4% in average accuracy. Zhang et al. (2021) [14] presented a detailed assessment of the contemporary literature on deep learning approaches for 3D estimation of human poses to demonstrate the invention process of these studies' projects, track the most recent study trends, and examine the features of various types of methods. There are manty points raised by authors. One of these points is the integration of autoencoders with conventional CNNs; they achieve structured learning that maintains the efficacy of CNNs while taking dependencies into consideration.

Challa et al. (2022) [15] proposed a robust classification model for HAR using ubiquitous sensor data and a hybrid deep learning model that combined CNN and bidirectional BiLSTM. The proposed CNN-BiLSTM network with multiple layers can automatically extract features from raw sensor data with minimum preprocessing. Combining CNN and BiLSTM enables the model to learn local and long-term dependencies at sequential data. The various filter sizes utilized by the proposed model can capture different temporal local dependencies, thereby enhancing the feature extraction process. The WISDM, UCI-HAR, and PAMAP2 benchmark datasets were used to evaluate the model's efficacy.

On the WISDM, UCI-HAR, and PAMAP2 datasets, the accuracy of the proposed model was 96.05%, 96.37%, and 94.29%, respectively. The experimental results indicated that the proposed approach outperformed the competing strategies. Sun et al. (2023) [15] provided an extensive overview of current developments in deep learning techniques for HAR according to the kind of input data modalities. We examine, in particular, the fusion-based and co-learning-based frameworks, which are the current popular deep learning techniques for both single and multiple data modalities. Together with incisive remarks and motivating future research objectives, we also provide comparative results on other benchmark datasets for HAR.

3. Materials and Methods

Manually annotating image features has been well documented and can be applied to many commercial fields [16]. However, the performance and potential of deep learning in the field of pedestrian detection are far better than those of traditional methods because deep learning can learn from the original image data and extract better features through algorithms [17]. The pedestrian detection method based on deep learning has exceptionally high accuracy and robustness. This is of great significance for developing the field of autonomous driving at the heart of deep learning, which lies in neural networks, especially convolutional neural networks (CNNs)[18]. In autonomous driving, CNNs can be used for target recognition and obstacle detection to achieve safe driving. With a high-precision image sensor, the vehicle can acquire a detailed image of its surroundings and then use a CNN to classify and segment the images to identify pedestrians, vehicles, road markings, and other obstacles. This technology is essential for avoiding collisions, predicting traffic conditions, and planning driving routes [19].

3.1. Optical Flow

Optical flow is the task of predicting motion between two images, usually two consecutive video frames. Optical flow images are generally taken as input, and a "stream" is predicted: the stream represents the displacement of each pixel in the first image and maps it to its corresponding pixel in the second image. Optical flow is a (2, H, W) dimensional tensor, where the first dimension corresponds to the predicted horizontal and vertical displacement of optical flow [20,21].

Between successive frames, we can express the image intensity I as a function of space (x, y) and time t. In other words, if we take the first image I (x, y, t) and move its pixels in t time (dx, dy), we will obtain a new image I(x + dx, y + dy, t + dt). First, we assume that the pixel intensity of the object is constant between successive frames, and then we use Taylor expanding [22].

$$I(x_2, y_2, t_2) = I(x_1 + \Delta x, y_1 + y, t_1 + \Delta t) \tag{1}$$

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{2}$$

$$I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t) = I(x + y + t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{3}$$

Optimizing optical flow means making the two plots as similar as possible so we can obtain them if they are completely consistent after twisting the optical flow equation.

$$0 = \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \ldots \tag{4}$$

$$0 = \frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \tag{5}$$

$$I_x V_x + I_y V_y = -I_t \tag{6}$$

In histogram-oriented optical flow (HOF), the optical flow is used to calculate the angle between the optical flow direction and the horizontal axis. A histogram is constructed

according to the number distribution of the optical flow direction in the divided area in the circle, and, finally, the feature description is formed [23].

3.2. Stacked Hybrid 3D Deep Learning Autoencoder (3D SAE-LSTM-CNN)

For classification problems, the architecture called stacked autoencoder (SAE) has an output layer where the activation function (such as SoftMax) is applied. In 3D SAE-LSTM-CNN, autoencoder networks connect, where the output signal from the hidden layer of an autoencoder (features extracted from the input applied to this autoencoder) serves as an input signal to another autoencoder. Finally, there is the layer responsible for classification, the activation function, which receives the signal extracted by the hidden layer of the last stacked autoencoder as input. With the use of 3D SAE-LSTM-CNN networks, the aim is to extract characteristics from the original database in such a way that they can be used for the classification problem using the activation function in the network output.

3.3. Proposed Algorithm

We first propose a way to define a problem based on global matching and then propose a transformer-based framework to implement it. Figure 1 shows the human activity recognition proposed system.

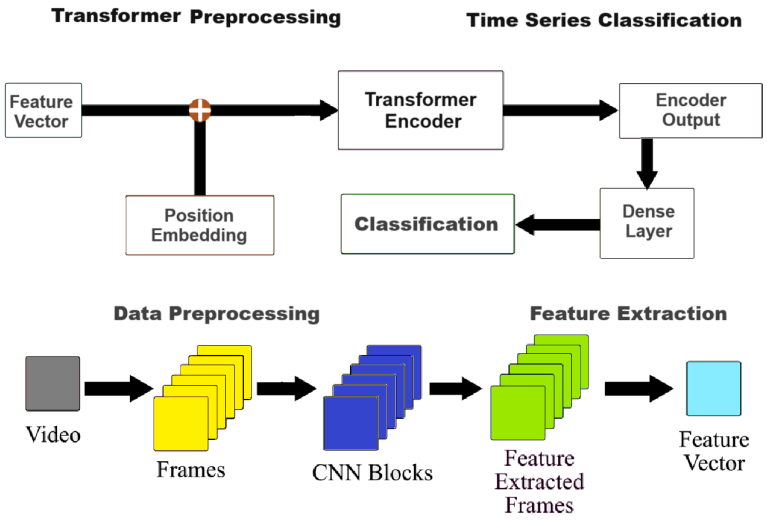


Figure 1. Proposed system structure.

The model is based on the 3dCNN model proposed by [24]; a 3D bidirectional CNN and LSTM model is fused in the proposed model pipeline. Data loading first resizes the input frames to 128 × 128 pixels. Spatial data from succeeding video frames were learned using the CNN network. The LSTM model takes in the input with several samples, time steps, with feature information. The 3D-BiCNN-LSTM generates feature maps that are transformed into a vector with nine time steps, referred to as the length of the sequence with 384 features. Under this paradigm, one sample is taken for each sequence. Hand gesture classification uses an LSTM network to extract temporal information for video frames. For a given two images, we first extract dense features using 3D–SAE-LSTM-CNN, and then compute the correlation between all points through global correlation. To obtain the corresponding points, we use a differentiable matching method; that is, we first normalize the last two dimensions of the global correlation by SoftMax to obtain the probability of matching and then use this probability to make a weighted average of the pixel grid points. Finally, the optical flow can be obtained by calculating the coordinate difference between the corresponding points. It can be seen that in the method we proposed, the core is to obtain strong features for matching.

3.3.1. Global Matching

Our initial features are derived from a 3D SAE-LSTM-CNN without considering the interdependencies between the two images. To further increase the expressiveness and discriminative nature of features, we use a transformer that includes self-attention and cross-attention. The cross-attention mechanism is an excellent way to model the interdependencies between images. Figure 2 shows the global matching strategy.

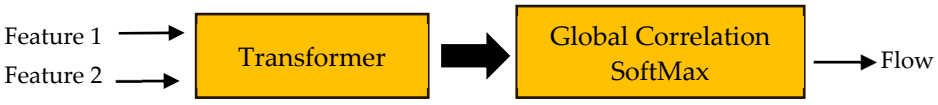


Figure 2. Global matching strategy.

From Figure 2, the dense features that are taken from the convolutional backbone structure are sent to a transformer (matching space transformer [25]) which is used for feature enhancement, along with a correlation with SoftMax layer for the global feature matching. Additionally, a self-attention layer is employed for flow propagation. Then, we compare how similar the features are by connecting each pair of features. After that, a differentiable SoftMax mapping layer is used to obtain the flow forecast. To deal with pixels that are blocked or outside the boundaries, we add an extra layer of self-attention that uses feature self-similarity to spread the high-quality flow forecast from matched pixels to unmatched ones.

3.3.2. Frameworks

The detection algorithms concentrate on detecting human behavior by extracting relevant characteristics from code datasets. We integrated a bidirectional LSTM (BiLSTM) and CNN in a stack autoencoder structure. For 3D SAE-LSTM-CNN, the encoder will comprise a series of convolution with max pooling 3D levels, where the max pooling layers are utilized for spatial downsampling. On the other hand, the decoder will be composed of a series of convolutional-LSTM with sampling layers.

The proposed BiLSTM-CNN autoencoder structure has the following description:

- Layer 1 (batch normalization): Batch normalization normalizes the input by subtracting the batch's meaning and then divides it through batch standard deviation. Input layer: the image's input size is 128 × 128, the frame length is 15, the batch size is 64, and the momentum is 0.8.
- Layer 2: Bidirectional (Conv-LSTM 2D) layer with 16 filters and (3 × 3) kernel sizes. The activation function used is (tanh). The recurrent dropout is 0.2. This layer reads the input data and outputs features to the max pooling layer.
- Layer 3: Max pooling 3D layer with pooling size of (1,2,2). It outputs a feature vector. The output of this layer is the encoded feature vector of the input data from layer 2, where it reduces the dimension.
- Layer 4: Time-distributed layer: This wrapper permits the application of a layer to each temporal input slice. Each input should have a minimum three-dimensional nature, with the temporal dimension being defined as the index, which is one dimension of the initial input. The recurrent dropout is 0.2.
- Layer 5: Bidirectional (Conv-LSTM 2D) layer with 16 filters and (3 × 3) kernel sizes. The activation function used is (tanh). The recurrent dropout is 0.2. This layer reads the input data and outputs features to the max pooling layer.
- Layer 6 (batch normalization input layer): This layer normalizes the previous activation layer's output by subtracting the batch's mean and dividing it by the batch standard deviation. The momentum is 0.8.
- Layer 7 (max pooling 3D layer): with pooling size of (1,2,2). It outputs a feature vector. The output of this layer is the encoded feature vector of the input data from layer 6, where it reduces the dimension.

- Layer 8 (time-distributed layer): This layer applies the same convolution operation to each input sequentially in time. The recurrent dropout is 0.2.
- Layer 9: bidirectional (Conv-LSTM 2D) layer with 16 filters and (3 × 3) kernel sizes. The activation function used is (tanh). The recurrent dropout is 0.2. This layer reads the input data and outputs features to the max pooling layer.
- Layer 10 (batch normalization input layer): This layer is used to normalize the previous activation layer's output by subtracting the batch's mean and then dividing through the batch standard deviation. The momentum is 0.8.
- Layer 11 (max pooling 3D layer): with pooling size of (1,2,2). It outputs a feature vector. The output of this layer is the encoded feature vector of the input data from layer 10, where it reduces the dimension.
- Layer 12 (time-distributed layer): This layer applies the same convolution operation to each input sequentially in time. The recurrent dropout is 0.3.
- Layer 13 (flatten layer): By flattening, the data are reduced to a one-dimensional array in preparation for their input into the dense layer (layer 14).
- Layer 14 (dense layer): This layer connects to all neurons of the flattened layer. This layer maps every neuron in layer 13 to every neuron in the output layer (layer 15).
- Layer 15 (dense layer): This layer is the output layer in which the output is the dot product of the weight matrix or kernel and tensor passed as input.

In the proposed model, the visual patterns can be learned directly from pixels by the 3D bidirectional CNN-LSTM network without requiring any preprocessing. The 3D SAE-LSTM-CNN structure includes trainable filters along with local pool operations, which are extremely beneficial for detecting hidden patterns in video frames and capturing all spatial and temporal changes. Table 1 shows the structure of the 3D SAE-LSTM-CNN network.

Table 1. The proposed 3D SAE-LSTM-CNN model.

Layer (Type)	Output Shape
Batch normalization	(None, 15, 128, 128, 3)
Bidirectional (ConvLSTM2D)	(None, 15, 126, 126, 32)
Max pooling 3D	(None, 15, 63, 63, 32)
Time-distributed	(None, 15, 63, 63, 32)
Bidirectional (ConvLSTM2D)	(None, 15, 61, 61, 32)
Batch normalization	(None, 15, 61, 61, 32)
Max pooling 3D	(None, 15, 31, 31, 32)
Time-distributed	(None, 15, 31, 31, 32)
Bidirectional (ConvLSTM2D)	(None, 15, 29, 29, 32)
Batch normalization	(None, 15, 29, 29, 32)
Max pooling 3D	(None, 15, 15, 15, 32)
Time-distributed	(None, 15, 15, 15, 32)
Flatten	(None, 108,000)
Dense	(None, 128)
Dense	(None, 51)

The kernel size for 3D pooling and convolution is d × k × k, with d representing temporal depths and k representing spatial dimension. The three-dimensional convolution is accomplished by convolving a three-dimensional kernel with the cube generated by stacking numerous contiguous frames. This architecture connects the convolution layer's feature maps to many contiguous frames from the preceding layer, capturing motion information. Intuitively, these multiple layers explain the visual material at different levels, each of which complements the other for the recognition job. The 3D SAE-LSTM-CNN network consists of three bidirectional (ConvLSTM2D) layers, three batch normalization layers, three max pooling layers (all convolution layer followed with a pooling layer), two fully connected layers, and a SoftMax loss layer. The channels (filters) for bidirectional (ConvLSTM2D) layers with 1 to 3 are 16. All 3D SAE-LSTM-CNN layers in both the temporal and spatial dimensions use 3 × 3 × 3 kernel with 1 × 1 × 1 stride. The pool1

layer, with a kernel dimension of $1 \times 2 \times 2$, aims to preserve temporal information in early phases without combining it. Each 3D SAE-LSTM-CNN layer produces a volume of feature maps as its output. All pooling layers produce the same amount of feature maps of convolution layers, but with lower spatial resolution; additionally, they introduce scale-invariant features. The model has a pair of dense layers one input and the second is output with a SoftMax layer for predicting action labels.

4. Results

In this section, we test proposed quantitative and qualitative analysis models.

4.1. Optical Flow Analysis

In this part, we test the optical flow parameters to determine the best values that can be used with the proposed model. First, we analyzed the effect of the k-filter value on image identification. Figure 3 shows the results.

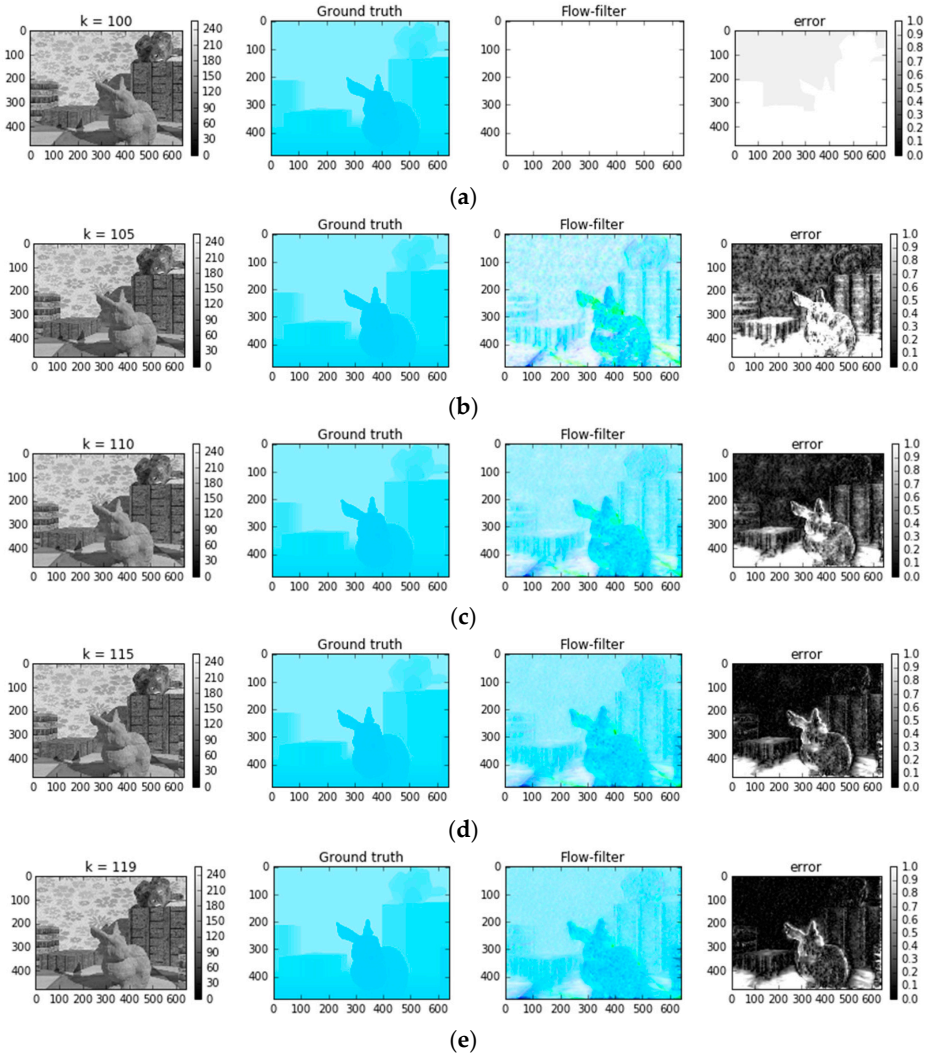


Figure 3. The effect of k filter value on image correction (**a**) k = 100, (**b**) k = 105, (**c**) k = 110, (**d**) k = 115, and (**e**) c = 115. The k filter value is a parameter used in the Lucas–Kanade pyramidal optical flow algorithm. The algorithm is used to estimate motion from a sequence of images. The prefiltering step plays a vital role in accurate optical flow computation. Here, k = 115 and k = 119 give the most accurate results.

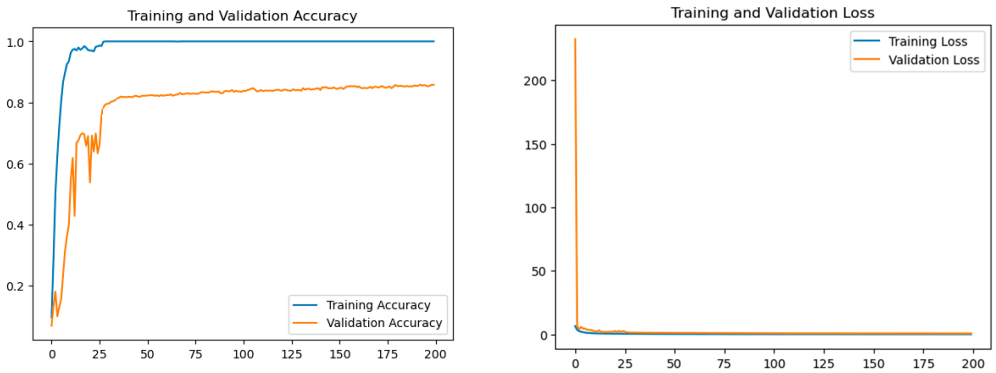
4.2. Quantitative Results

In this part, we test the model for product human activity from the HAR dataset. In this work, we intend to use the HMDB51 dataset. The HMDB51 dataset is an extensive collection of realistic videos from various sources, including movies and web videos. It contains 6766 video clips from 51 action categories, such as "jump", "run", and "walk", with each category containing at least 101 clips. Figure 4 shows the samples of the HMDB51 dataset.



Figure 4. The HMDB51 dataset samples.

To reduce computation time, we resized each frame of the videos into a fixed height and width after reading them from the dataset. Additionally, we normalized the data to the range [0–1] by dividing the pixel values by 255, accelerating convergence during network training. We generated a list comprising the normalized and resized video frames passed as an argument. While the function processes the video files frame by frame, it does not append every frame to the list because we only require a sequence length from frames that is evenly distributed. In the last step, we split our data to create training and testing sets. We also shuffled the dataset before the split to avoid any bias and obtain splits representing the overall distribution of the data. The number of epochs used for the train/test model was selected to be 200 epochs. The results are shown in Figure 5.



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

Figure 5. The training and validation results: (**a**) accuracy; (**b**) loss.

The model achieved good accuracy from the results, reaching up to 100% training accuracy and 86.86% validation accuracy.

For ablation experiments of the proposed model, we validated the performance of the model with state-of-the-art structure. Two models were evaluated; the first is 3D CNN and the second is 2 CNN + LSTM. Table 2 shows the model structure of the two models.

Table 2. 3D CNN and 2 CNN + LSTM model structure.

3D CNN		2 CNN + LSTM	
Layer (Type)	**Output Shape**	**Layer (Type)**	**Output Shape**
Conv3D	(None, 15, 128, 128, 16)	ConvLSTM2D	(None, 20, 62, 62, 4)
Max pooling 3D	(None, 15, 64, 64, 16)	Max pooling 3D	(None, 20, 31, 31, 4)
Batch normalization	(None, 15, 64, 64, 3)	Time-distributed	(None, 20, 31, 31, 4)
Conv3D	(None, 15, 64, 64, 32)	ConvLSTM2D	(None, 20, 29, 29, 8)
Max pooling 3D	(None, 15, 32, 32, 32)	Max pooling 3D	(None, 20, 15, 15, 8)
Batch normalization	(None, 15, 32, 32, 32)	Time-distributed	(None, 20, 15, 15, 8)
Conv3D	(None, 15, 32, 32, 64)	ConvLSTM2D	(None, 20, 13, 13, 14)
Max pooling 3D	(None, 15, 16, 16, 64)	Max pooling 3D	(None, 20, 7, 7, 14)
Batch normalization	(None, 15, 16, 16, 64)	Time-distributed	(None, 20, 7, 7, 14)
Conv3D	(None, 15, 16, 16, 128)	ConvLSTM2D	(None, 20, 5, 5, 16)
Max pooling 3D	(None, 15, 8, 8, 128)	Max pooling 3D	(None, 20, 3, 3, 16)
Batch normalization	(None, 15, 8, 8, 128)	Flatten	(None, 2880)
Flatten	(None, 90,112)	Dense	(None, 51)
Dense	(None, 256)		
Dense	(None, 51)		

Both models were trained for 200 epochs. The results are shown in Table 3.

Table 3. The comparison between the proposed model, 3D CNN, and 2 CNN + LSTM.

Structure	Training Accuracy	Test Accuracy (%)
3D CNN	99.9	52.76
2 CNN + LSTM	93.22	58
Proposed model	100	86.86

As shown in Table 3, the proposed model achieved the best result compared to the other models. Lastly, we compared our results with the most recent related works that used an HMDB51 dataset. Table 4 shows the comparison results.

Table 4. The comparison between the proposed model and some recent related works for recognizing human behavior for the HMDB51 dataset.

Ref.	Structure	Acc. Results (%)
Chen et al. (2018) [26]	Relational LSTM	70.4
Yang et al. (2019) [27]	3DCNNs + BDH-LSTM	72.2
Anvarov et al. (2020) [28]	Squeeze-and-excitation (SE) and self-attention (SA) modules with 3D CNN	74.1
Cheng et al. (2021) [29]	S3-Net	80.8
Ullah (2021) [30]	Lightweight CNN and DS-GRU	72.21
Patrick et al. (2021) [31]	GDT	72.8
Tanet al. (2022) [32]	Fusion network	70.72
Hussain (2022) [33]	VIT + LSTM	73.714
Liu et al. (2022) [34]	Spatial-Temporal Interaction Learning Two-stream network (STILT)	72.1
Ullah and Munir (2023) [35]	Dual attentional convolutional neural network (DA-CNN) and bidirectional GRU (Bi-GRU)	79.3
Proposed method	Stacked autoencoder CNN + LSTM	86.86

4.3. Qualitative Analysis

We depend on the initial annotations of walking movements in the JAAD dataset to streamline the annotating process to identify transitions. A transition from a standing to a walking condition is considered a Go candidate, whereas the reverse is regarded as a Stop candidate. The state before the transition is called the prestate, while the state after it is called the post-state. To minimize any inaccuracies in labeling and ensure more significant samples, we only deem a transition candidate legitimate if the total durations between

the prestate and post-state exceed 0.5 s. The original datasets contain distinct pedestrians that can be classified into four categories: walk, stand, stop, and go. The pedestrians in the video who walk and stand do not display any changes, while those who stop and go demonstrate the relevant transitions. The categories of stand and go are not mutually exclusive, as a pedestrian can stop and move during the same observation. The statistical data of the datasets are displayed in Figure 6. Upon examining the identified transitions, we observe that most instances of stopping and starting in JAAD are strongly associated with road intersections.

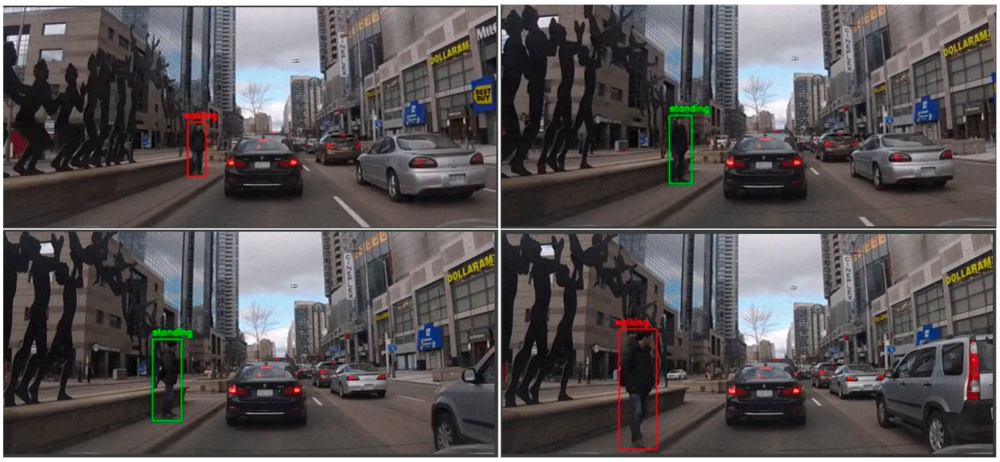
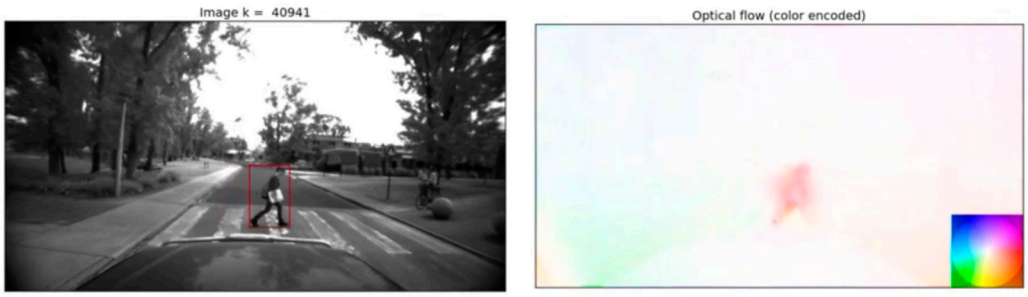


Figure 6. Qualitative results of proposed hybrid stacked autoencoder model on JAAD dataset.

Figure 6 shows qualitative example predictions from our proposed hybrid stacked autoencoder model on the JAAD dataset. We can see that pedestrian stops and goes at junctions are still challenging to predict due to the absence of accurate vehicle speed and traffic light conditions. Furthermore, rapid changes in traveling direction, atmospheric conditions (such as rainy and snowy), and irregularities, such as construction workers, can all have a detrimental impact on projections.

In Figure 7, we can observe the three cases of human behavior in a vehicle-driven sense. The first case is a person crossing the street when a vehicle passes (Figure 7a). The right side of Figure 7a shows the optical fields of sense and it is shown that the person's optical view and the direction of flow would be in the x+ direction which means that the person may be passing perpendicular to the vehicle path, which means that the vehicle should stop (red box). In the second case (Figure 7b), the bicycle driver crosses the street. The optical flow bicycle driver motion is also in the x+ direction, so the vehicle should also need to stop (red box). In the last case (Figure 7c), there are some people that are walking parallel to the vehicle motion (y+ direction), so this is not dangerous as the humans do not try to cross the street and the system identifies that there is no danger (green box).



(**a**)

Figure 7. *Cont.*

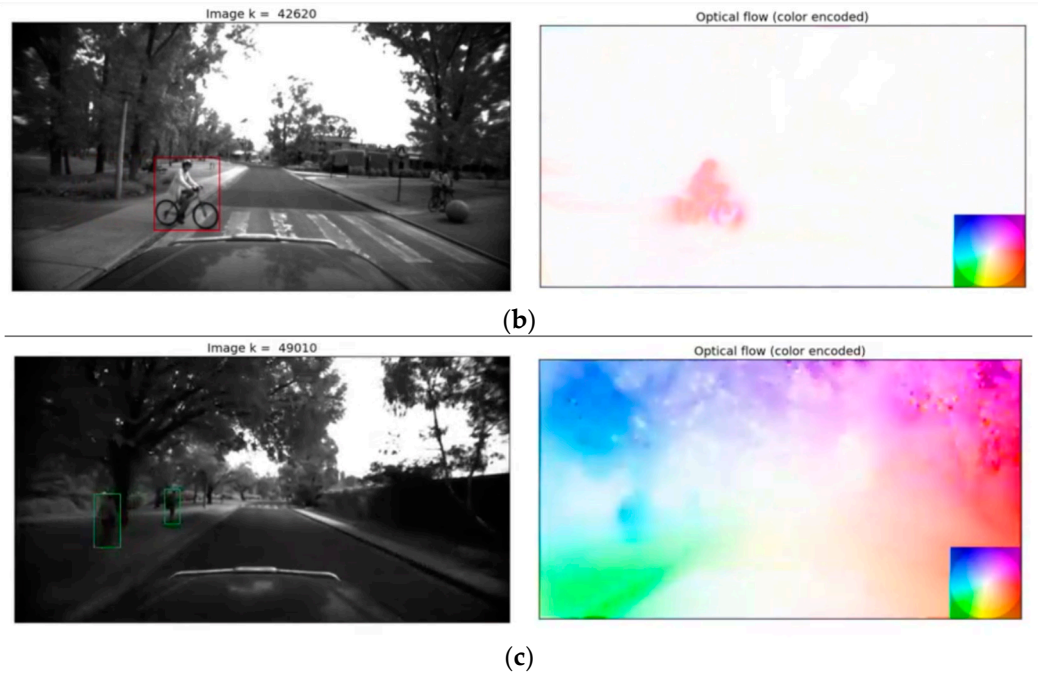


Figure 7. Qualitative results with optical flow results for the proposed hybrid stacked autoencoder model for several human behaviors on the street: (**a**) human on a cross-walk walking in a direction toward the street; (**b**) bicycle rider trying to cross the street; (**c**) people walking parallel to the street.

5. Conclusions

This study presents the issue of predicting pedestrian stopping and walking behavior. It is essential to accurately forecast these complex and nonlinear changes to comprehend the paths of pedestrians and ensure their safety. This model was tested using HMDB51 datasets and JAAD dataset and compared with recent related works. For a quantitative test, the HMDB51 dataset was used to train and test models for human behavior. For qualitative analysis, we utilized the JAAD dataset, which is the initial comprehensive dataset for predicting pedestrian stopping and walking behavior from the viewpoint of a vehicle. The system incorporates multiple preexisting datasets to encompass a wide range of scenarios and environments. Subsequently, we developed a novel deep learning model that utilizes video clips and high-level features pertaining to pedestrians and ambient situations. This model incorporates the use of a hybrid feature combination technique. We then assessed the performance of this model, as well as several other baseline models, on JAAD. The quantitative results show that the proposed model achieved good accuracy of about 86.86%, which outperforms recent works. The qualitative results show that the model can effectively identify dangerous behaviors of humans and then test them on moving vehicle scene.

Author Contributions: Conceptualization, L.M.S. and Y.C.; methodology, L.M.S. and Y.C.; software, L.M.S. and Y.C.; validation, L.M.S. and Y.C.; formal analysis, L.M.S. and Y.C.; investigation, L.M.S. and Y.C.; resources, L.M.S. and Y.C.; data curation, L.M.S. and Y.C.; writing—original draft preparation, L.M.S. and Y.C.; writing—review and editing, L.M.S. and Y.C.; visualization, L.M.S. and Y.C.; supervision, L.M.S. and Y.C.; project administration, L.M.S. and Y.C.; funding acquisition, L.M.S. and Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available in a publicly accessible repository. The original data presented in the study are openly available at [https://www.kaggle.com/datasets/easonlll/hmdb51] and [https://data.nvision2.eecs.yorku.ca/JAAD_dataset/] (accessed on 24 March 2024).

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

References

1. Diraco, G.; Rescio, G.; Siciliano, P.; Leone, A. Review on Human Action Recognition in Smart Living: Sensing Technology, Multimodality, Real-Time Processing, Interoperability, and Resource-Constrained Processing. *Sensors* **2023**, *23*, 5281. [CrossRef]
2. Yang, J.; Zhang, Z.; Xiao, S.; Ma, S.; Li, Y.; Lu, W.; Gao, X. Efficient data-driven behavior identification based on vision transformers for human activity understanding. *Neurocomputing* **2023**, *530*, 104–115. [CrossRef]
3. Ko, K.E.; Sim, K.B. Deep convolutional framework for abnormal behavior detection in a smart surveillance system. *Eng. Appl. Artif. Intell.* **2018**, *67*, 226–234. [CrossRef]
4. Wang, F.; Zhang, J.; Wang, S.; Li, S.; Hou, W. Analysis of Driving Behavior Based on Dynamic Changes of Personality States. *Int. J. Environ. Res. Public Health* **2020**, *17*, 430. [CrossRef]
5. Mohammed, H.A. Assessment of distracted pedestrian crossing behavior at midblock crosswalks. *IATSS Res.* **2021**, *45*, 584–593. [CrossRef]
6. Zhou, X.; Ren, H.; Zhang, T.; Mou, X.; He, Y.; Chan, C.Y. Prediction of Pedestrian Crossing Behavior Based on Surveillance Video. *Sensors* **2022**, *22*, 1467. [CrossRef]
7. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 146–177. [CrossRef]
8. Wang, J.; Huang, H.; Li, K.; Li, J. Towards the Unified Principles for Level 5 Autonomous Vehicles. *Engineering* **2021**, *7*, 1313–1325. [CrossRef]
9. Gesnouin, J. Analysis of Pedestrian Movements and Gestures Using an On-Board Camera to Predict Their Intentions. September 2022. Available online: https://pastel.hal.science/tel-03813520 (accessed on 20 January 2024).
10. Zhang, D.; He, L.; Tu, Z.; Zhang, S.; Han, F.; Yang, B. Learning motion representation for real-time spatio-temporal action localization. *Pattern Recognit* **2020**, *103*, 107312. [CrossRef]
11. Prabono, A.G.; Yahya, B.N.; Lee, S.L. Atypical Sample Regularizer Autoencoder for Cross-Domain Human Activity Recognition. *Inf. Syst. Front.* **2021**, *23*, 71–80.
12. Garcia, K.D.; de Sá, C.R.; Poel, M.; Carvalho, T.; Mendes-Moreira, J.; Cardoso, J.M.; de Carvalho, A.C.; Kok, J.N. An ensemble of autonomous auto-encoders for human activity recognition. *Neurocomputing* **2021**, *439*, 271–280. [CrossRef]
13. Huang, W.; Zhang, L.; Gao, W.; Min, F.; He, J. Shallow Convolutional Neural Networks for Human Activity Recognition Using Wearable Sensors. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 2510811. [CrossRef]
14. Zhang, D.; Wu, Y.; Guo, M.; Chen, Y. Deep Learning Methods for 3D Human Pose Estimation under Different Supervision Paradigms: A Survey. *Electronics* **2021**, *10*, 2267. [CrossRef]
15. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human Action Recognition From Various Data Modalities: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3200–3225. [CrossRef]
16. Aljabri, M.; AlAmir, M.; AlGhamdi, M.; Abdel-Mottaleb, M.; Collado-Mesa, F. Towards a better understanding of annotation tools for medical imaging: A survey. *Multimed. Tools Appl.* **2022**, *81*, 25877–25911. [CrossRef]
17. Su, J.; An, Y.; Wu, J.; Zhang, K. Pedestrian Detection Based on Feature Enhancement in Complex Scenes. *Algorithms* **2024**, *17*, 39. [CrossRef]
18. Karadeniz, A.T.; Çelik, Y.; Başaran, E. Classification of Walnut Varieties Obtained from Walnut Leaf Images by the Recommended Residual Block Based CNN Model. *Eur. Food Res. Technol.* **2023**, *249*, 727–738.
19. Iftikhar, S.; Zhang, Z.; Asim, M.; Muthanna, A.; Koucheryavy, A.; El-Latif, A.A.A. Deep Learning-Based Pedestrian Detection in Autonomous Vehicles: Substantial Issues and Challenges. *Electronics* **2022**, *11*, 3551. [CrossRef]
20. Lo, K.M. Optical Flow Based Motion Detection for Autonomous Driving. *arXiv* **2022**, arXiv:2203.11693.
21. Ladjailia, A.; Bouchrika, I.; Merouani, H.F.; Harrati, N.; Mahfouf, Z. Human activity recognition via optical flow: Decomposing activities into basic actions. *Neural Comput. Appl.* **2019**, *32*, 16387–16400. [CrossRef]
22. OpenCV: Optical Flow. Available online: https://docs.opencv.org/3.4/db/d7f/tutorial_js_lucas_kanade.html (accessed on 20 January 2024).
23. Wang, T.; Snoussi, H. Detection of abnormal events via optical flow feature analysis. *Sensors* **2015**, *15*, 7156–7171. [CrossRef] [PubMed]
24. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. Human Activity Classification Using the 3DCNN Architecture. *Appl. Sci.* **2022**, *12*, 931. [CrossRef]
25. Hu, V.T.; Zhang, D.W.; Mettes, P.; Tang, M.; Zhao, D.; Snoek, C.G.M. Latent Space Editing in Transformer-Based Flow Matching. *arXiv* **2023**, arXiv:2312.10825. [CrossRef]
26. Chen, Z.; Ramachandra, B.; Wu, T.; Vatsavai, R.R. Relational Long Short-Term Memory for Video Action Recognition. *arXiv* **2018**, arXiv:1811.07059.
27. Yang, H.; Zhang, J.; Li, S.; Luo, T. Bi-direction hierarchical LSTM with spatial-temporal attention for action recognition. *J. Intell. Fuzzy Syst.* **2019**, *36*, 775–786. [CrossRef]
28. Anvarov, F.; Kim, D.H.; Song, B.C. Action recognition using deep 3D CNNs with sequential feature aggregation and attention. *Electronics* **2020**, *9*, 147. [CrossRef]
29. Cheng, Y.; Yang, Y.; Chen, H.B.; Wong, N.; Yu, H. S3-Net: A Fast Scene Understanding Network by Single-Shot Segmentation for Autonomous Driving. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–19. [CrossRef]

30. Ullah, A.; Muhammad, K.; Ding, W.; Palade, V.; Haq, I.U.; Baik, S.W. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Appl. Soft. Comput.* **2021**, *103*, 107102. [CrossRef]
31. Patrick, M.; Asano, Y.M.; Kuznetsova, P.; Fong, R.; Henriques, J.F.; Zweig, G.; Vedaldi, A. On Compositions of Transformations in Contrastive Self-Supervised Learning. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9557–9567. [CrossRef]
32. Tan, K.S.; Lim, K.M.; Lee, C.P.; Kwek, L.C. Bidirectional Long Short-Term Memory with Temporal Dense Sampling for human action recognition. *Expert. Syst. Appl.* **2022**, *210*, 118484. [CrossRef]
33. Hussain, A.; Hussain, T.; Ullah, W.; Baik, S.W. Vision Transformer and Deep Sequence Learning for Human Activity Recognition in Surveillance Videos. *Comput. Intell. Neurosci.* **2022**, *2022*, 3454167. [CrossRef]
34. Liu, T.; Ma, Y.; Yang, W.; Ji, W.; Wang, R.; Jiang, P. Spatial-temporal interaction learning based two-stream network for action recognition. *Inf. Sci.* **2022**, *606*, 864–876. [CrossRef]
35. Ullah, H.; Munir, A. Human Activity Recognition Using Cascaded Dual Attention CNN and Bi-Directional GRU Framework. *J. Imaging* **2023**, *9*, 130. [CrossRef] [PubMed]

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.