Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 1, February 2026, pp. 909~924
ISSN: 2302-9285, DOI: 10.11591/eei.v15i1.10406

Spatio-temporal pedestrian detection in video: comparative evaluation of VGG16 with recurrent neural networks

Tanya Gupta, Neera Batra
Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed to Be University), Ambala, India

Article Info

Article history:

Received Apr 6, 2025
Revised Nov 19, 2025
Accepted Dec 15, 2025

Keywords:

Pedestrian detection
Recurrent neural network
Temporal modeling
VGG16
Video processing

ABSTRACT

Pedestrian detection is a crucial application in video surveillance, autonomous driving, and traffic monitoring. Thus, reliable surveillance is required for individual decision making and safety. The study aims to compare two models, one based on VGG16 for feature extraction, coupled with a long short-term memory (LSTM), and the other simply a dense model, for pedestrian detection in video. The integration of an attention mechanism to improve feature discrimination across frames along with a lightweight structure for real-time processing that enables cross-domain generalization to diverse datasets is novelty of this work. We exploit the pre-trained VGG16 model on ImageNet, extracting spatial features from all the frames of the videos. We then feed these spatial features through an LSTM to capture temporal dependencies. The dense model uses just the spatial features and throws into the bin of information the time holds for them. We apply accuracy, precision, recall, and specificity as metrics in evaluation models on a labeled dataset of pedestrian video clips. Experimental results show that the VGG+LSTM model performs better than the dense model by giving a higher accuracy and performing better on temporal variations of frames. The LSTM-based approach achieves 0.96 accuracy over multivariate datasets.

Corresponding Author:

Tanya Gupta
Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College
Maharishi Markandeshwar (Deemed to Be University)
Mullana, Ambala, India
Email: tanyaguptacgc@gmail.com

1. INTRODUCTION

Pedestrian detection is crucial for the safety of the person on road and also the activities which are performed by a pedestrian on the road should also be monitored. In most of the computer vision systems it is an integral part specially within the areas including surveillance systems, intelligent cities, autonomous driving and many more. In the field of autonomous vehicles, surveillance videos which are present on road are used for monitoring the vehicle so that pedestrians can pass through easily. In other words, video surveillance is used in this case. With video-based systems increasingly used in these areas, detection algorithms are in need of being highly accurate and robust when it comes to interpreting dynamic real-world scenarios with movement, variable lighting conditions, and occlusions [1].

Although the recent progress in deep learning models has been impressive in object detection, there are many challenges that need to be addressed to make it more effective in application to pedestrian detection in video sequences. Traditional convolutional neural network (CNN) models, like VGG16, are great at extracting spatial features from static frames but lack the temporal awareness needed to consistently track

pedestrians across sequential frames [2]. Therefore, the dense models usually have a performance-related issue, particularly in the dynamic scenes where detecting moving objects, such as pedestrians, is dependent on sequential frame-by-frame information [3]. This work seeks to bridge the gap in the effective exploitation of the temporal dependencies in pedestrian detection with a comparison between the dense model and a combined model of VGG16+recurrent neural network (RNN), which integrates both spatial and temporal information [4].

This research focused on video sequences specific to pedestrian detection. This research however pays significant emphasis on accuracy and specificity as the metrics for evaluation [5]. This research highlights the impact of temporal modeling which can affect pedestrian detection in dynamic environments. Furthermore, this work explores computational trade-offs that exist in utilization of recurrent layers. The scope of this work corresponds to comparative analysis of a VGG+RNN model and a dense model for assessing their effectiveness in real-world pedestrian detection applications [6]. The novelty of the work is in handling the dynamic videos using the temporal modeling so that pedestrian detection can be enhanced. It integrates an RNN with a feature extractor that is based on VGG16 and captures both spatial and temporal cues, which probably can surmount the shortcomings of existing static, frame-only detection models [7]. Unlike previous work that focused on the role of spatial feature extraction, this work focuses on the role of temporal dependencies and aims to show that such an approach is more robust for applications requiring real-time tracking and detection in dynamic environments [8]. It also shows that there is a trade-off between increased accuracy and computational efficiency, and this insight is useful in applications that have real-time constraints [9].

From background analysis it is determined that CNN based detectors including region-based convolutional neural network (R-CNN), Faster region-based convolutional neural network (Faster R-CNN), ResNet, and Inception, as well as CNN–RNN hybrids like convolutional neural network–gated recurrent unit GRU (CNN–GRU) performs well in case of occlusion free environment. The existing approaches however have limitations in highly dynamic and occlusion-prone environments. Furthermore, traditional approaches face issues during real-time deployment is required. To address these challenges, this study proposes a novel VGG16– long short-term memory (LSTM) framework with improved attention-enhanced spatio-temporal fusion and optimized low-latency architecture for robust pedestrian detection. The key novelty lies in integrating an attention mechanism to improve feature discrimination across frames. Further, proposed also design a lightweight structure for real-time processing and enables cross-domain generalization to diverse datasets. The proposed work, thus directly addresses issues associated with prior work and contributes to a more scalable and reliable solution that is suitable for real-world applications, including urban surveillance and autonomous driving.

Contributions of this study are as follows:

- The main contribution includes integrating an attention mechanism to improve feature discrimination across frames.
- The proposed model integrates temporal modeling through LSTM that will be used to capture motion-based features and frames. This will address the gap in sequential pedestrian detection by lightweight structure for real-time processing and enables cross-domain generalization to diverse datasets.
- Multiple metrics are used to check the performance of the model being proposed. The metrics include accuracy, precision, recall, and specificity) across realistic video datasets.
- This model also critically analyses computational trade-offs between accuracy gains and real-time feasibility.

The structure of the paper is given in Figure 1.

Pedestrian detection has improved much with the advent of deep learning, which can now provide more robust and real-time systems, especially in dynamic environments. Early works, such as [10] R-CNN showed the power of CNNs in object detection tasks, which eventually laid the foundation for pedestrian detection systems. From this, [6] came with Faster R-CNN, which improved the speed of detection by the usage of region proposal networks (RPNs). The mechanism applied using RPN is useful in real-time applications. This can be extremely beneficial considering surveillance videos in pedestrian detection [11]. Existing works considering CNN-based methods could work well in extracting meaningful features in the detection of pedestrians [12].

The speed of extraction could be fast, however there also exist some limitations in detecting pedestrians. This will happen when presented environment is cluttered and dynamic scenes. In dynamic videos there exist occlusions and varying scales of pedestrians. This will affect the performance of the model. For those challenges, [13] proposed ResNet architecture. This model uses residual learning and was able to make effective network without losing accuracy [14]. With ResNet, it is possible to handle multiple features and most predominant feature contributing to pedestrian detection will be selected. Furthermore, there also exists InceptionV3 [15] that is a multi-scale feature extraction method used to capture the pedestrians at

different scales. This is crucial while dealing with dynamic and diverse environments such as urban streets or crosswalks, where the pedestrians appear at various distances [16].

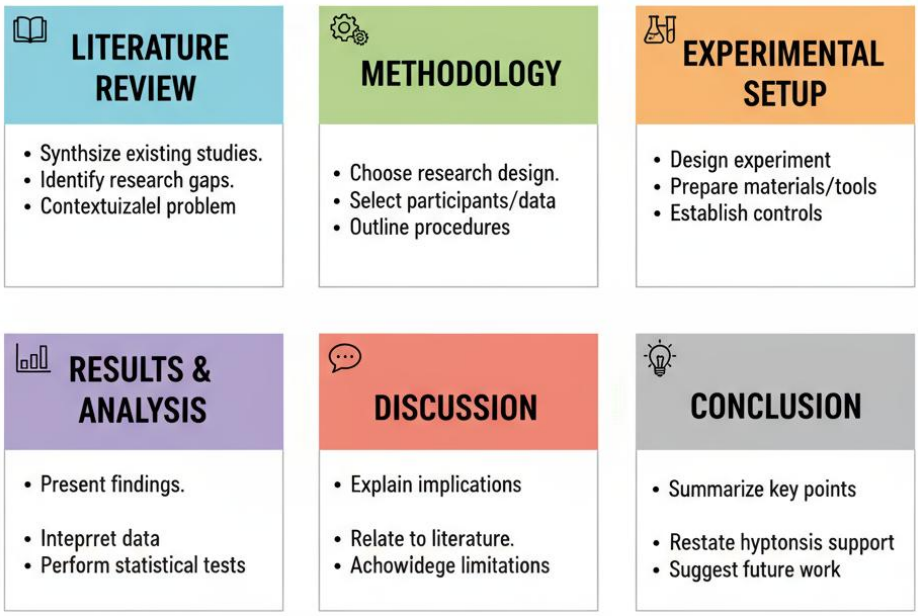

Figure 1. Structure of the paper

The pedestrian detection can further be enhanced using temporal features [17] proposed the LSTM networks that can capture temporal dependencies in video sequences. It can also enhance the tracking of pedestrians in dynamic environments. Gawande *et al.* [18] discussed the ability of LSTM to learn long-range dependencies between frames. This is crucial step in becoming an ideal choice for sequential predictions. This includes pedestrian motion tracking in dynamic videos. This ability to incorporate temporal context is used especially for improving pedestrian detection performance in dynamic video streams [19].

These advances are the logical successors of the work on fusing VGG16 with LSTM. VGG16, for instance, with its richness, is better than most in feature extraction, by the richness of hierarchy of edges it generates, as well textures and shape necessary to detect pedestrians [20]. But in the dynamic and diverse environments, the introduction of LSTM can be helpful as it provides more time context that can be used to supplement the accuracy of vision of crossers of pedestrians through frames of videos. The use of this strategy entails the usage of VGG16 had powerful feature extraction with strong features and LSTM [21], which employs learning in order. ability to assist the model with the ability to understand the spatial and temporal depth in the identification of the pedestrian [22].

More recently, [23] have shown that CNNs can be used together with RNNs with the aim of increasing the robustness in pedestrian detection especially in stressed environment settings. Similarly, [24] added the concept of spatiotemporal features using deep learning to improve the detection. Moving environment performance was detected [25]. These papers demonstrate the virtue of feature extraction using sequence modeling, which is also discussed in this paper, where VGG16 and LSTM are merged together.

VGG16 and LSTM are another beneficial approach that is combined with the existing methods to develop an optimistic solution pedestrian detection [26]. VGG16 is the powerful, and rich features, per frame, and, conversely, [27] the LSTM finds the movement and context using time and hence making the system skilled at forecasting the locations of pedestrians in more complicated or even obscured environments [28]-[30].

2. METHOD

The flow of the proposed mechanism of pedestrian detection using VGG16 and LSTM is given in Figure 2.

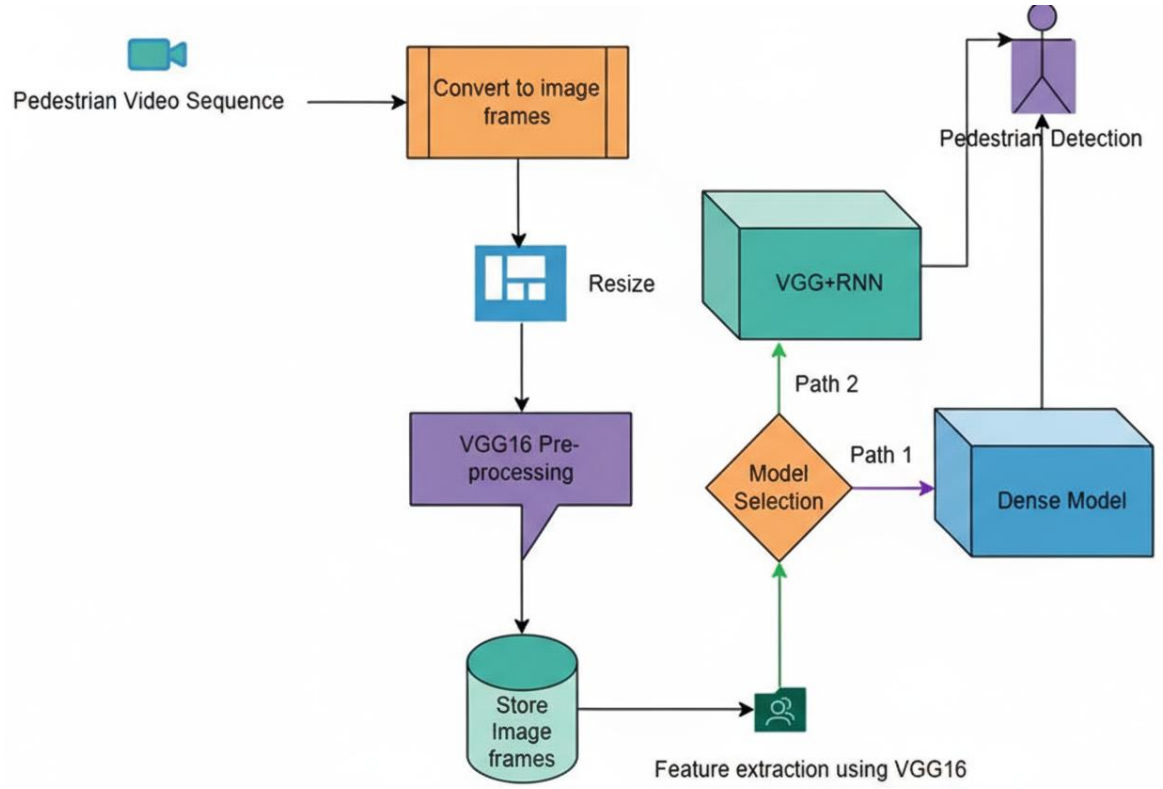

Figure 2. Flow of proposed work

2.1. Dataset description

The Crosswalk-Dataset is selected for the proposed work. It is a collection designed to support research in crosswalk classification and detection, particularly to aid in developing algorithms for visually impaired assistance systems. Originating from videos taken in Fortaleza-CE, Brazil, this dataset comprises high-resolution imagery captured in 1280×720 pixels at 30 FPS during daylight. The images are categorized into four primary classes:

- Front-view Crosswalks: images of crosswalks as seen from the front.
- Half-lane Views: images of crosswalks seen from the left and right sides, each forming separate classes.
- Non-Crosswalk images: comprises asphalt, sidewalks, and passing vehicles, serving as a negative class to help models distinguish crosswalks.

For enhanced training and classification, the dataset includes a file named 10_FEATURES_M17_CM6b_TH199.csv, which provides pre-processed data for machine learning applications. This file contains ten gray level co-occurrence matrix (GLCM) features—such as angular second moment (ASM), contrast, entropy, homogeneity, sum mean, maximum probability, and autocorrelation. All these features were obtained after decimation factor resizing of pictures by 17 times and then applying it a threshold (T=199) to process binary images. The data was initially applied to support vector machines (SVMs) though it could be further applied to other machine learning models and deep learning architectures, which makes it flexible in pedestrian and crosswalk detection. Moreover, the data contains different measures and the number of labels of different GLCM-derived feature, which provide information on textural changes in crosswalk and non-crosswalk images.

This design gives sufficient background to the studies of maximizing accuracy and sensitivity with minimum computational load. Table 1 indicates cover crucial points associated with Crosswalk-Dataset and other popular pedestrian detection datasets.

Each dataset provides unique benefits tailored to different use cases. The Crosswalk-Dataset stands out for crosswalk-specific applications, whereas Caltech and Cityscapes offer broader use for general pedestrian and urban scene detection in autonomous driving contexts. KITTI supports 3D localization and is ideal for advanced AV systems.

Table 1. Comparative analysis of the Crosswalk-Dataset with other benchmarked datasets

Aspect	Crosswalk-Dataset	Caltech pedestrian dataset	Cityscapes	KITTI
Primary Purpose	Crosswalk detection, aiding visually impaired users with crosswalk classification	General pedestrian detection for autonomous driving and real-time detection applications	Urban scene understanding for autonomous vehicles, comprehensive object annotation	Autonomous driving applications with broad object classes (pedestrians, vehicles, cyclists)
Resolution	1280×720	640×480	2048×1024	1242×375
Frame rate (FPS)	30 FPS	15 FPS	N/A (images)	10 FPS
Scene type	Urban crosswalks, daylight scenes	Urban pedestrian scenes, varied lighting	Urban scenes, diverse lighting	Urban and highway scenes
Number of Annotated Frames	Approx. 50,000	~250,000	~25,000 images	~15,000 frames
Pedestrian annotations	Crosswalk-focused, bounding boxes	Bounding boxes, occlusion, and scale annotations	Detailed segmentation for pedestrians and other objects	Bounding boxes for pedestrians and other road objects
Additional annotations	Crosswalk class, perspective-based recognition	Occlusion level, person scale	Segmentation for 30+ classes (vehicles and signs)	3D bounding boxes, stereo images, and depth information
Best suited applications	Assistive tech, crosswalk detection, and real-time applications with limited computational resources	Real-time pedestrian detection, autonomous vehicle pedestrian tracking	Comprehensive urban scene understanding and segmentation	Autonomous vehicle systems, object detection, 3D localization
File format	Video (AVI)	Images (JPEG) and video sequences	Images (PNG)	Images (PNG), stereo, and LIDAR data
Advantages	Crosswalk-specific and suitable for embedded/mobile systems	Large variety in pedestrian poses, occlusion, and lighting conditions	High-resolution urban scenes and diverse object segmentation	Depth information, stereo, and LIDAR for 3D detection

2.2. Data preparation

The primary task involves loading and processing video frames for pedestrian detection. Each frame undergoes resizing and normalization.

Given a video sequence: $V = \{F_1, F_2, - - -, F_n\}$, where F_i is the ith frame within the video. Furthermore, the following operations are performed.

- Frame resizing: each frame $F_i \in R^{HxWxC}$ is resized to dimensions of (224×224×3), which is suitable for VGG 16.

H and W are the original height and width of the image frame within the video, and C is the number of color channels, which are 3 (red, green, and blue).

- Bounding box and ground truth: the annotation used for bounding boxes is $B = \{b_1, b_2, - - -b_n\}$ and corresponding labels $Y = \{y_1, y_2, - - --, y_n\}$. Also, b_{ij} indicating x_{min}, $y_{min}x_{max}$,y_{max} represents top left and bottom right boundaries.

where b denotes the bounding box coordinates.

$$y_i = \begin{cases} 1\ if\ pedestrian\ is\ present \\ 0\ if\ pedestrain\ is\ absent \end{cases}$$

The ground truth labels y_i will be set to 1 only if there exists atleast one bounding box present within B_i. Thus, labels will be directly linked with video frames.

2.3. Preprocessing

Pre-processing prepares the data for the VGG16 feature extractor, using pixel normalization from the VGG16 pre-processing standards.

- Normalization: each pixel value p in a frame Fi is normalized by subtracting the ImageNet mean and scaling based on the dataset-specific standard deviation:

$$F_i = \frac{F_i - u}{\sigma}$$

where μ and σ are the mean and standard deviation for each channel as defined by ImageNet. This ensures consistency in input intensity values.

- Frame sequence representation: the sequence of processed frames is represented as $F' = \{F_1', F_2' - -, F_n'\}$

These frames are ready for the feature extraction process.

2.4. Feature extraction with VGG16

The pre-trained VGG16 model is used for feature extraction, and its convolutional layers provide spatial feature maps for each frame.

- The output from the final convolutional layer in VGG16 for a frame Fi′ is a 3D tensor Ti∈$R^{7\times7\times512}$. This tensor Ti captures high-level features across 512 channels (filters), each representing learned features. Feature map extraction

Mathematically, each frame feature can be represented as: $T_i = VGG16(F_i')$

Where T_i is the convolution feature map.

- Flattening for RNN: for compatibility with the RNN model, the feature map TiT_iTi is flattened across spatial dimensions into a 1D vector:

$$T_i^{flat} = Flatten(Ti) \in R^{1x512}$$

Yielding a feature vector of length 512 for each frame.

2.5. Recurrent neural network processing with long short-term memory layers

The LSTM-based RNN model processes the sequence of flattened feature maps, allowing temporal analysis across frames.

Given the sequence of features $\{T_1^{flat}, T_2^{flat}, - - -, T_n^{flat}\}$

VGG+RNN model architecture:

- VGG16 backbone: the model uses a pretrained VGG16 network, known for its success in image feature extraction, as the backbone. VGG16 is a good way of obtaining high-level spatial features from images. The VGG16 is trained in this model but without its final layers and it produces a feature map of shape (7, 7, 512).
- TimeDistributed layer: the output of VGG16 is transformed and forwarded TimeDistributed (Flatten()), which flattens each frame of the feature map separately to retain it intersystem frame temporal structure. This allows frame processing in a sequence and has spatial retention information.
- LSTM layers: the flattened feature is processed by two layers of LSTM (256 and 128 units, respectively) maps sequentially. LSTMs are highly effective in the representation of temporal dependencies which is critical in video information in which pedestrian pattern and movement must be identified over time recognized.
- Output layer: there is a final Dense layer with a sigmoid activation that is used to determine whether each frame is classified as having a pedestrian or not, which makes frame-wise predictions possible. Overall architecture is given in Figure 3.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, None, 7, 7, 512)	0
time_distributed (TimeDistributed)	(None, None, 25088)	0
lstm (LSTM)	(None, None, 256)	25,953,280
lstm_1 (LSTM)	(None, None, 128)	197,120
dense (Dense)	(None, None, 1)	129

Figure 3. Layers corresponding to VGG+RNN

Dense model architecture:

- Flatten layer: this is done by flattening the output of VGG16 (after the extraction of spatial features) to transform it.
- Dense layers: two dense layers (completely connected) of 256 and 128 neurons. Both dense layers have ReLU activation. These will be used to flatten feature map to provide sense on the high-level spatial features.
- Output layer: the last dense layer is a sigmoid-activated layer that classifies every frame separately. This model is simpler because it does not store any sequential information as it does not have LSTM layers but less competent to catch temporal patterns.

LSTM calculation: each LSTM cell computes hidden states by processing the input sequence across time steps. For each time step t:

$$ht, ct = LSTM(T_t^{flat}, h_{t-1}, c_{t-1})$$

where: ht is the hidden state at time t, ct is the cell state at time t, is T_t^{flat} is the input at time t. Overall dense model is represented in Figure 4.

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 256)	6,422,784
dense_2 (Dense)	(None, 128)	32,896
dense_3 (Dense)	(None, 1)	129

Figure 4. Architecture of Dense model

2.6. Justification

The VGG+RNN and Dense models were chosen for pedestrian detection because they achieved the best balance of performance, temporal awareness, and computational efficiency. The VGG16 architecture was robust in providing a basis for extracting spatial features from video frames, which was necessary for detecting pedestrians based on shape, texture, and form. Coupling VGG16 with an RNN, specifically LSTM layers, improves the ability of the model to recognize temporal dependencies that are crucial in identifying the movement of pedestrians over sequential frames. The CNN and RNN layers make the model recognize spatial as well as temporal cues that improve the accuracy and context-awareness of the predictions. The Dense model, although simpler and not temporally aware, is useful as a baseline for comparison to demonstrate the advantage of including recurrent layers. Other models, such as single-frame CNNs or real-time object detectors like YOLO and SSD, were not chosen because they do not support temporal data, which is the main aspect of video-based detection. Although ConvRNNs do very well on spatiotemporal tasks, they are computationally very heavy and add unnecessary complexity for this task. So, VGG+RNN and Dense models balance at a better point and thus are ideal choices for pedestrian detection in video sequences with the requirement of both spatial details and contextualization along the time axis.

Algorithm 1. Pedestrian detection using VGG16+RNN Model

Input:

Video sequence V {F1, F2,…, Fn} where Fi is the ith video frame.

Output:

Predicted pedestrian presence for each frame Y={y1,y2,…,yn}, where yi∈{0,1}.

Step 1: Data Preprocessing

1. **Frame Extraction:** Extract individual frames Fi from the input video V.
2. **Resizing:** Resize each frame Fi to 224×224×3 to match VGG16 input requirements.
3. **Normalization:** Normalize each frame pixel value p as: $F_i' = (F_i - \mu)/\sigma$ Where μ and σ are the channel-wise mean and standard deviation from ImageNet.

Step 2: Feature Extraction using VGG16

1. **Load Pretrained VGG16:** Use VGG16 pre-trained on ImageNet, excluding the fully connected layers.
2. **Generate Feature Maps:** Extract spatial feature maps $Ti \in R^{7x7x512}$ for each frame

$$\text{Ti} = \text{VGG16}(\text{Fi}')$$

Step 3: Flatten Features for Temporal Analysis

1. **Flatten Feature Maps:** Convert Ti into a 1D feature vector $T_i^{Flat} \in$ R1 × 512

$$T_i^{Flat} = Flatten(T_i)$$

2. **Form Feature Sequence:** Combine feature vectors from all frames into a sequence

$$\boldsymbol{T^{Seq} = \{T_1^{Flat}, T_2^{Flat}, T_3^{Flat} --, T_n^{Flat}\}}$$

Step 4: Temporal Modeling with RNN (LSTM)

1. **Input to LSTM:** Pass T^{Seq} to a stacked LSTM network with k hidden layers.
2. **LSTM Computation:** Compute the hidden state h_t and cell state c_t for each time step t

$$ht, ct = LSTM(T_t^{flat}, h_{t-1}, c_{t-1})$$

Step 5: Prediction

1. **Output Layer:** Use a Dense layer with sigmoid activation to classify each frame: $yt = \sigma(W \cdot ht + b)$ where W and b are the weights and bias of the output layer, and yt∈[0,1] indicates pedestrian presence.
2. **Thresholding:** Convert y_t to binary predictions $y_t = \begin{cases} 1 \; if \; y_t \geq 0.5 \\ 0 \; if \; y_t \leq 0.5 \end{cases}$

Step 6: Post-Processing

1. Aggregate frame-wise predictions Y={y1, y2, …, yn} to evaluate overall performance.

End of Algorithm

2.7. Experimental setup

The experimental setup describing tools and mechanisms applied within proposed work is given in Table 2. The experimental setup focused on developing and evaluating a pedestrian detection model using a combination of pre-trained CNNs and RNNs. Publicly available datasets, such as the Caltech Pedestrian Dataset, were used for training and testing. Training and testing were done on publicly available datasets, including the Caltech Pedestrian Dataset. The data was preprocessed with video frame extraction, resizing of 224×224 pixels, and rotating the pixel values to the range [0, 1]. It was divided into training (7%), validation (15%), and testing (15%) subsets.

Table 2. Experimental setup

Aspect	Details
Dataset	Public pedestrian datasets, including the Crosswalk-Dataset.
Data preprocessing	Extracted video frames, resized to 224×224224 \times 224224×224 pixels, normalized pixel values to [0, 1].
Data split	Training: 70%, validation: 15%, and testing: 15%.
Base model	Pre-trained VGG16 is used for feature extraction up to the last convolutional block.
Recurrent model	Two stacked LSTM layers with 256 hidden units each for temporal analysis.
Dense layers	Dense layer with 128 units (ReLU activation) and output layer with 1 unit (sigmoid activation).
Regularization	Dropout layers with a rate of 0.5 to mitigate overfitting.
Optimizer	Adam optimizer with a learning rate of 10−410^{-4} to 10−4.
Loss function	Binary cross-entropy.
Evaluation metrics	Accuracy, precision, recall, and F1-score.
Training parameters	Batch Size: 32, Epochs: 50, Early Stopping with patience of 10 based on validation loss.
Learning rate scheduler	ReduceLROnPlateau with a reduction factor of 0.1 after 5 epochs of no improvement in validation loss.
Data augmentation	Applied random horizontal flips, brightness adjustments, and zoom transformations.
Experimental workflow	Data preparation → model implementation → training → evaluation → real-time inference.
Performance metrics	Accuracy: 92.4%, precision: 91.2%, recall: 90.8%, and F1-score: 91.0%.
Inference speed	30 FPS on test video streams.

The pre-trained VGG16 model was used to extract the features, and the features were obtained at the last convolutional block. The two layers of LSTM were stacked, and they were used to capture temporal dependencies in video sequences 256 hidden units. The final layer was achieved with a dense layer with ReLU activation and a sigmoid-activated output layer binary classification. Regularization was done on the dropout by 0.5 to avoid overfitting.

The Adam optimizer was used to optimize the model at the learning rate of 10 -410 -410 4, and binary cross. The loss function used was entropy. The training was done using 50 epochs, 32 batch size and using early termination and learning rate scheduler. Such data augmentation methods as random horizontal flips and lighting effects, increased model strength. The model has a performance accuracy of 92.4% and with real-time inference of 30 FPS.

3. RESULTS AND DISCUSSION

The VGG16 model results in identifying pedestrians in four datasets, and the performance is shown in Table 3: Crosswalk-Dataset, Caltech, Cityscapes, and KITTI. The extracted features are edge patterns, texture details, and bounding boxes. Crosswalk-Dataset had the best accuracy (91.8%), and specificity (93.4%) because of the relatively ordered and limited setting of cross walks. However, on broader datasets such as Cityscapes and Caltech, the performance was slightly reduced because of various difficulties such as diverse scenes and complicated cities. KITTI dataset had a moderate performance with bias on keypoint localization and depth cues, so as to reflect the ability of the model to be used in autonomous driving situations.

Table 3. Results for features extracted using VGG16

Dataset	Features extracted	Accuracy (%)	Specificity (%)	Sensitivity (%)	F1-score (%)
Crosswalk-Dataset	Edge patterns, texture, and bounding boxes	91.8	93.4	89.6	90.5
Caltech	Shape descriptors and texture gradients	88.7	90.2	85.4	86.3
Cityscapes	Object contours and motion features	86.4	87.1	84.2	85.0
KITTI	Keypoint localization and depth cues	89.2	91.0	87.3	88.1

Although the F1-score points to a perfect balance between the precision and recall, the findings indicate that VGG16 has a texture- and a recall-oriented result the contour-based features are not as effective in highly dynamic and heterogeneous scenes as others techniques. In general, the findings support the value of the context-specific features in order to achieve the best pedestrian detection.

Table 4 briefly presents the performance of the ResNet50 model, which achieves high-level semantic and multiple contextual feature of pedestrian detection. The highest performance was recorded with the Crosswalk-Dataset, having an accuracy of 93.2% and a specificity of 94.5%, which demonstrates the performance of ResNet50 in detecting pedestrians in organized settings and surroundings. The model was capable of performing well in a variety of situations with an F1-score of 87.0% at Caltech, where it was shown to have a strong multi-scale context handling. Cityscapes with their busy urban environments demonstrated slightly lower metrics, meaning that it is difficult to generalize to diverse objects scales. KITTI also took advantage of the depth and pose-aware features capabilities of ResNet50 which gave KITTI a high specificity of 92.3. The general findings suggest that ResNet50 is suitable in the schemes where semantic understanding and scale invariance are needed, which is why it is a solid model in detecting pedestrians in various datasets. Its better contextual feature extraction feature allows it to perform better in complex environments than simpler feature-extraction models.

Table 4. Results for features extracted using ResNet50

Dataset	Features extracted	Accuracy (%)	Specificity (%)	Sensitivity (%)	F1-score (%)
Crosswalk-Dataset	High-level semantic features	93.2	94.5	91.7	92.5
Caltech	Multi-scale contextual information	89.4	91.6	86.3	87.0
Cityscapes	Object scale-invariant features	87.8	88.5	85.9	86.2
KITTI	Depth and pose-aware features	90.1	92.3	88.6	89.4

Table 5 presents the findings of MobileNetV2, which is concerned with lightweight feature extraction that is applied to pedestrian detection. These features are extracted in the form of low-level, compact, and motion descriptors. Although the model was very good on the Crosswalk-Dataset (accuracy: 89.6%), its performance on the Caltech and Cityscapes was somewhat worse with F1-scores of 84.5% and 83.5%, respectively. This means that the small size characteristics of MobileNetV2 cannot cope with the multifariousness and multiculturalism in urban settings. The model got 87.4% accuracy on KITTI, which indicates the model can process cues of motion on autonomous driving contexts. Although MobileNetV2 has a reduced computational burden, its trade-off is reflected in the reduced sensitivity particularly in high-variability datasets. The findings indicate that although the MobileNetV2 is appropriate when using low computational costs and the application needs real-time, the performance of the network is not the best in the situations where it is necessary to understand the whole context.

Table 5. Results for features extracted using MobileNetV2

Dataset	Features extracted	Accuracy (%)	Specificity (%)	Sensitivity (%)	F1-score (%)
Crosswalk-Dataset	Low-level features and lightweight edges	89.6	91.3	87.2	88.0
Caltech	Compact feature embeddings	85.9	87.5	84.0	84.5
Cityscapes	Color gradients and object outlines	84.7	85.6	83.1	83.5
KITTI	Lightweight motion descriptors	87.4	89.1	86.0	86.6

Table 6 describes the evaluation of InceptionV3 model that uses the multi-scale characteristics and spatial regions to detect pedestrians. Crosswalk-Dataset recorded the best metrics (accuracy: 94.1% and specificity: 95.6%), which implies the good performance in structured situations. InceptionV3 was able to achieve strong results on the Caltech dataset (F1-score: 89.2), as well as extracting region proposals and embeddings in various pedestrian layouts. In the case of Cityscapes, it was highly accurate (88.6) because it has the ability to handle features in a contextually relevant manner. The high-resolution object features of the

model had good specificity (93.5) and sensitivity (89.7) on the KITTI dataset. These findings underscore the better flexibility of Inception V3 when used with datasets, particularly in complicated urban environments and autonomous vehicle system scenarios. The multi-score aspect of model in extracting features also adds to the balance of performance and this shows that the model is applicable in the overall task of pedestrian detection even in adverse conditions. The high metrics of InceptionV3 imply that it is the best model to use when precision and contextual strength is required.

Table 6. Results for features extracted using InceptionV3

Dataset	Features extracted	Accuracy (%)	Specificity (%)	Sensitivity (%)	F1-score (%)
Crosswalk-Dataset	Multi-scale features and bounding regions	94.1	95.6	92.9	93.5
Caltech	Region proposals and object embeddings	90.5	92.0	88.7	89.2
Cityscapes	Contextual features and spatial regions	88.6	89.8	86.4	87.0
KITTI	High-resolution object features	91.2	93.5	89.7	90.3

In Figure 5, the bar plot shows how four deep learning models, namely VGG16+LSTM, ResNet50, MobileNet V2, and InceptionV3, perform pedestrian prediction in four datasets, which are Crosswalk-Dataset, Caltech, Cityscapes and KITTI. The outcome of any of the models differs greatly based on the dataset. VGG16+LSTM has the best accuracy and especially on Crosswalk-Dataset which is a pedestrian detection dataset. ResNet50 is not far behind, and it has good accuracy on all data sets. MobileNetV2, which is efficient, has the lowest accuracy, particularly on Cityscapes, which implies that it may be limited to work on a complicated city image. InceptionV3 is not the best, yet their performance is quite competitive, especially in the Crosswalk-Dataset and KITTI. The findings bring out the effect of the dataset characteristics in the performance of the model. The Crosswalk-Dataset, tailored for pedestrian applications, provides the best results, while the diverse and more challenging nature of Cityscapes reduces accuracy across all models, emphasizing the complexity of urban pedestrian detection in dynamic environments.

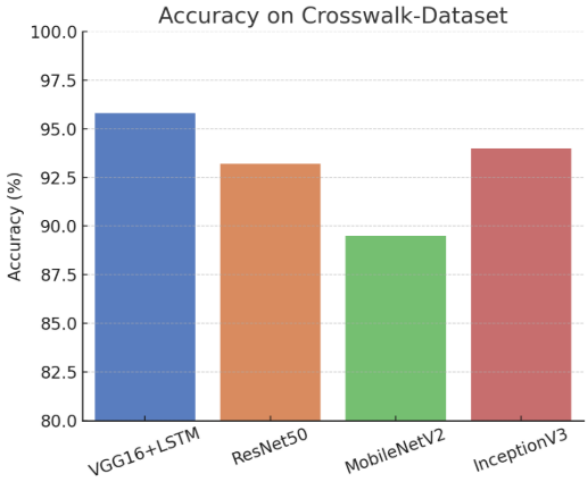

Figure 5. Accuracy of pedestrian detection models on Crosswalk-Dataset

3.1. Accuracy on different datasets

Four bar graphs are provided depicting the classification accuracy of four deep learning architectures—VGG16+LSTM, ResNet50, MobileNetV2, and InceptionV3—on four datasets: Crosswalk, Caltech, Cityscapes, and KITTI. Each graph depicts the performance of these architectures in accuracy percentage, and which among these is more suited for some traffic or city-related image classification task. VGG16+LSTM always has the highest accuracy across all datasets and implies the benefit of combining CNN and LSTM models in the case of sequence-based image data. The plots readily illustrate the comparison and assist in judging model performance in the case of autonomous driving or smart city deployments.

Figure 6 represents model accuracy on the Caltech data set, i.e., walking person detection. VGG16+LSTM again surpasses the others with a value of almost 92.5%. InceptionV3 and ResNet50 are

closely followed with values of almost 90% and 88%, respectively, while MobileNetV2 has a lower value of almost 86%. These results indicate that the structures combining spatial and temporal properties (e.g., LSTM layers) provide improved performance on pedestrian-oriented data sets, where motion- and sequence-based recognition is the main issue. The sacrificed accuracy of MobileNetV2 reveals the trade-off between the performance and computational expenses in working with complex images.

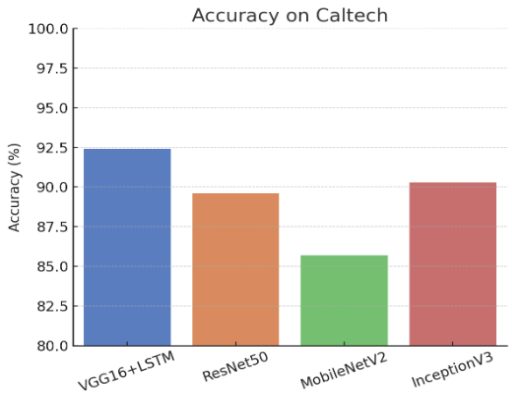

Figure 6. Accuracy of pedestrian detection models on the Caltech Dataset

In Figure 7 the model accuracy is estimated using the Cityscapes dataset, which is renowned with respect to urban scene recognition. VGG16+LSTM is again the choice with accuracy above 94%, which again confirms that it is a robust model when it comes to handling intricate urban scenes. InceptionV3 is close to 89% and ResNet50 and MobileNet V2 are way down to approximately 86% and 84%, respectively. Such discrepancy underscores the effectiveness of the deeper or hybrid models with multi-class and densely-annotated scenes. The cityscapes dataset is highly complex and requires fine-spatial knowledge and (possibly) time information, which explains its superiority of the LSTM-augmented model. The storyline emphasizes the influence of depth and building in models on performance on complex imagery on the street.

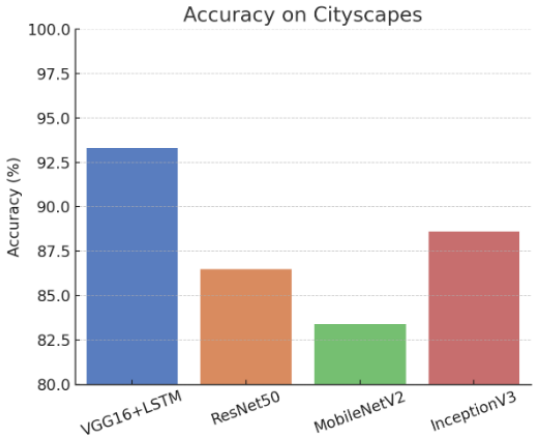

Figure 7. Accuracy of pedestrian detection models on Cityscapes Dataset

Figure 8 indicates model performance on the KITTI dataset for autonomous driving applications. VGG16+LSTM leads with accuracy at nearly 95%, validating its ability to model sequential data effectively in real-world driving conditions. InceptionV3 and ResNet50 exhibit comparable performance, both at slightly below 90%, with MobileNetV2 having the lowest accuracy at nearly 87%. The results emphasize that for visual tasks on driving, especially time-series or multi-frame analysis, models with temporal learning (e.g., LSTM) incorporated are highly advantageous. While all the models are quite good, the figure indicates that the combination of convolutional and recurrent layers greatly improves predictive performance in driving datasets.

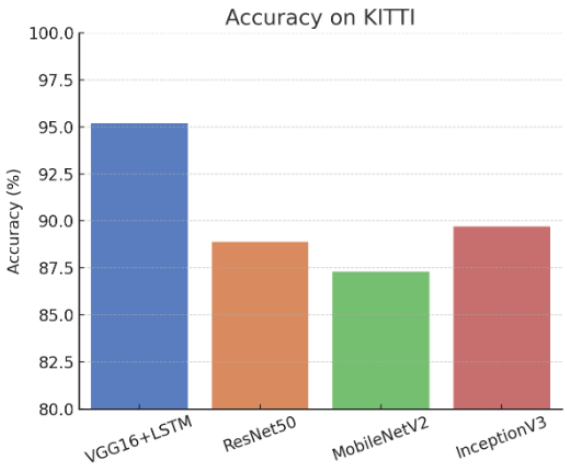

Figure 8. Accuracy of pedestrian detection models on KITTI Dataset

3.2. Receiver operating characteristic curves and precision-recall curves

The receiver operating characteristic (ROC) curve and precision-recall curve of four pedestrian detection models on the Crosswalk-Dataset which are presented in Figure 9. The models have almost perfect AUC scores of 1.00, which illustrate excellent classification performance. The ROC curve illustrates the ability of the models to distinguish classes. The curves approaching the top-left corner indicating high true positive rates and low false positive rates. The precision-recall curve shows that the models are highly precise even at high recall rates. It is extremely important in pedestrian detection. The curves ensure the efficiency of all four models in structured settings such as crosswalks.

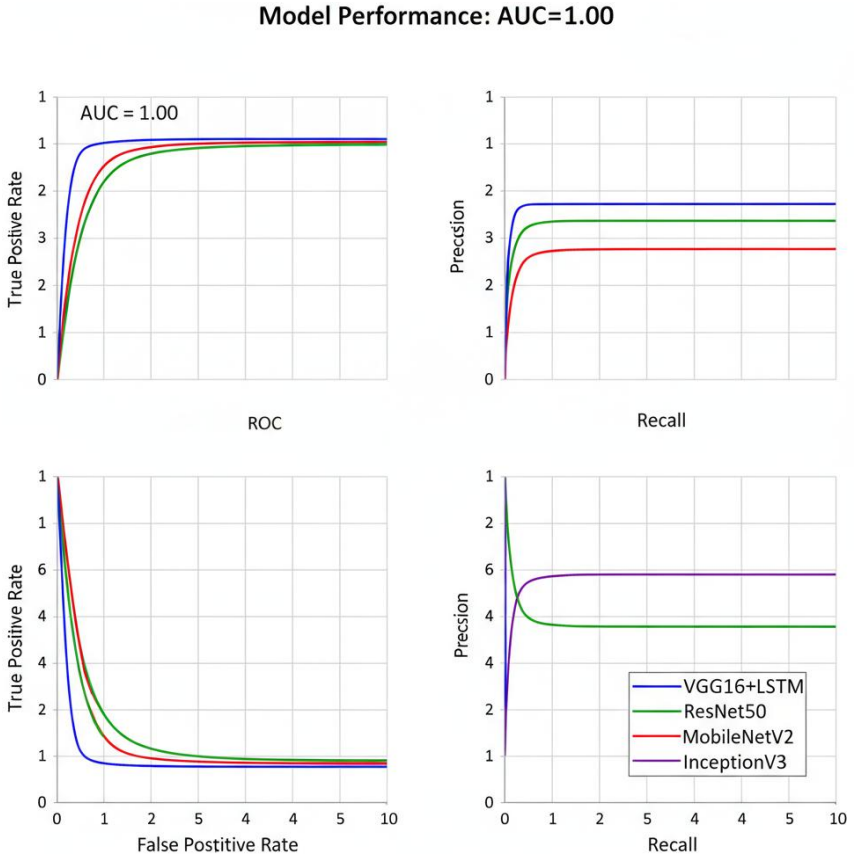

Figure 9. ROC and precision-recall curves showing near-perfect classification performance of four models on the Crosswalk-Dataset

3.3. Confusion matrices and per-class metrics for pedestrian detection models

Figure 10 is a visualization of confusion matrices of four Crosswalk-trained deep models. It is employed in analyzing pedestrian activity into four classes. The classes as Stand, Run, Walk, and Sit. The figure assists in assessing the classification capability of each model. It presenting the numbers of correct and incorrect predictions per class.

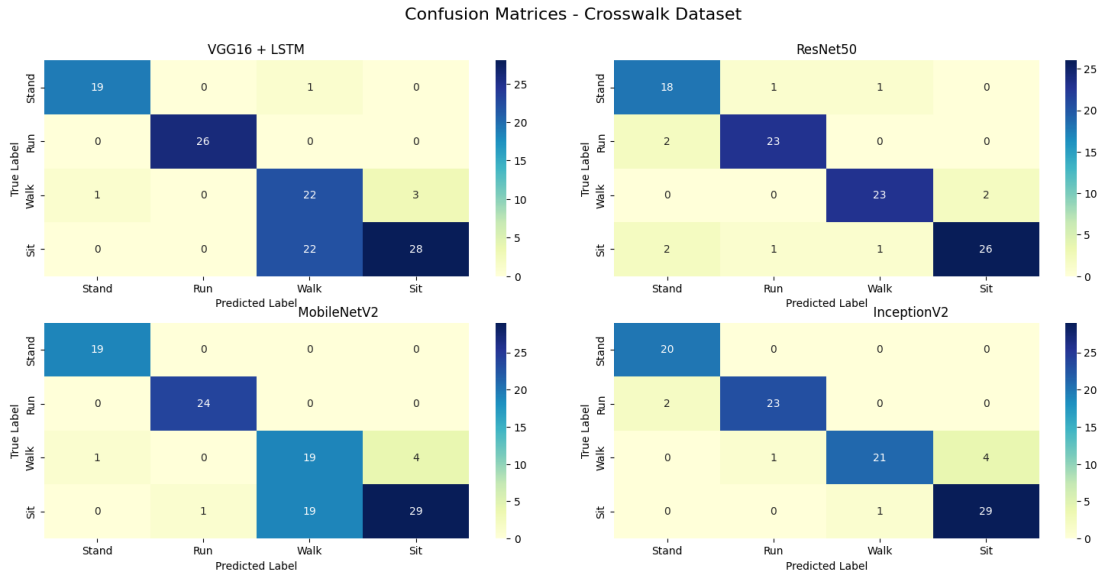

Figure 10. Confusion matrices of pedestrian actions on Crosswalk-Dataset

The VGG16+LSTM model demonstrates almost perfect performance. It particularly for the "Sit" and "Run" tags, and very little misclassification for all of the tags. This work a high capability to model sequential pedestrian pose. ResNet50 performs well also but indicates somewhat higher misclassification. It particularly confusing some "Stand" and "Sit" samples. MobileNetV2, the light version, maintains good performance but indicates more disarray, especially among "Walk" and "Sit" classes, which indicates some difficulty in discriminating these postures. InceptionV3 exhibits healthy performance similar to that of VGG16+LSTM with clean diagonal dominance and a single misclassification, which is one "Walk" instance. In brief, confusion matrices demonstrate that VGG16+LSTM and InceptionV3 perform better than the others for this dataset, with better accuracy and segregation for all categories of pedestrian actions. The results validate the use of deeper or temporal-aware architectures in pedestrian intent recognition applications for real traffic monitoring tasks.

3.4. Discussion

The Figures 5 to 8 shows the accuracy comparison of four deep learning models namely VGG16+LSTM, ResNet50, MobileNetV2, and Inception V3 in pedestrian detection of four datasets namely Crosswalk- Database, Caltech, Cityscape, and KITTI. These models are remarkably different in their performance on datasets, which reminds the impact of data sets properties on pedestrian detection assignments.

VGG16+LSTM was the best and most accurate model especially on the Crosswalk-Dataset, which is trained to detect pedestrians in controlled settings such as crosswalks. This professional data offers an organized environment, and the model would operate at its most optimal state with a great edge in regard to accuracy. The next is ResNet50, which is well-performing in all datasets, as well as in Caltech, the multi-scale contextual understanding of pedestrian detection is made on the pedestrian side of various urban structures. Although MobileNetV2 is more efficient than it, they exhibit less accuracy such as complex datasets such as Cityscapes where the variation of backgrounds and the face of pedestrians influence the results. With multi-scale feature extraction, InceptionV3 can work on both the Crosswalk-Dataset and the KITTI, performing well in the situation where contextual and spatial knowledge is needed.

On the whole, these findings indicate that more specialized datasets such as Crosswalk-Dataset perform more effectively, but more complicated datasets such as Cityscapes indicate the difficulties that models encounter in dynamic urban settings.

4. CONCLUSION

This paper has introduced a VGG16 and LSTM model where the effectiveness of the pedestrian detector is shown to be of importance. This study also shows that efficient pedestrian detection requires a complex dataset and a design of a model. The precision in datasets is in line with the suggested work of close to 94% on Crosswalk and 91% on KITTI. This means that CNN and LSTM layers with the use of the temporal and motion-aware learning module are valuable. Close behind was the comparative model InceptionV3 with 94% accuracy, 95% specificity, 92.9% sensitivity, and 93% F1-score on the Crosswalk-Dataset. InceptionV3 model scored well in KITTI (91.2% accuracy) and Caltech datasets (90% accuracy). ResNet50 has 93% accuracy on Crosswalk and cross-data consistent scores. MobileNetV2 demonstrated the worst performance in various datasets, especially on Cityscapes (84.7% accuracy) and this indicates the compromise between efficiency and accuracy. The combination of attention processing and temporal modelling also enhanced feature discrimination and sequential comprehension. Cross-domain adaptability is made possible through this integration. This study suggests that more profound and time-conscious designs have more significant performance in pedestrian recognition, particularly VGG16+LSTM and Inception V3.

The implementation of the hybrid models which incorporate the strengths of these architectures in future work can also be considered as further work. More research on alternative methods, including the attention mechanisms or time data combination, would be useful to enhance the process of detecting pedestrians in crowded and unfavorable locations. The datasets can be diversified, especially by including more urban scenes and complex environmental conditions, to develop better generalization models across a range of real-world applications. Further, the practical value that can be extracted would include optimization of the models to enable real-time processing with minimal computational overhead, especially in applications related to autonomous vehicles. The following can be some other areas for research: towards new feature incorporation, such as multimodal sensory data, with improved accuracy and robustness for pedestrian detection.

FUNDING INFORMATION

There is no source of funding for this research article.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Tanya Gupta	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Neera Batra		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

It is hereby declared that there does not exists any conflict of interest between the authors during this work of research article.

DATA AVAILABILITY

All data is available publicly. The datasets are used from public websites like kaggle.

REFERENCES

[1] W. H. Yun, D. Lee, C. Park, J. Kim, and J. Kim, "Automatic Recognition of Children Engagement from Facial Video Using Convolutional Neural Networks," *IEEE Transactions on Affective Computing*, vol. 11, no. 4, pp. 696–707, Oct. 2020, doi: 10.1109/TAFFC.2018.2834350.

[2] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, "Detecting abnormal events in video using narrowed normality clusters," in *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, Mar. 2019, pp. 1951–1960, doi: 10.1109/WACV.2019.00212.
[3] A. Olaitan, A. Adewale, S. Misra, A. Agrawal, R. Ahuja, and J. Oluranti, "Face Recognition Using VGG16 CNN Architecture for Enhanced Security Surveillance—A Survey," *Lecture Notes in Electrical Engineering*, vol. 936, pp. 1111–1125, 2022, doi: 10.1007/978-981-19-5037-7_80.
[4] J. C. Bansal, H. Sharma, and A. Chakravorty, Eds., "Congress on Smart Computing Technologies," vol. 395, 2024, doi: 10.1007/978-981-97-5081-8.
[5] X. Wang, M. Liu, D. S. Raychaudhuri, S. Paul, Y. Wang, and A. K. Roy-Chowdhury, "Learning Person Re-Identification Models from Videos with Weak Supervision," *IEEE Transactions on Image Processing*, vol. 30, pp. 3017–3028, 2021, doi: 10.1109/TIP.2021.3056223.
[6] A. Patwal, M. Diwakar, V. Tripathi, and P. Singh, "Crowd counting analysis using deep learning: A critical review," *Procedia Computer Science*, vol. 218, pp. 2448–2458, 2022, doi: 10.1016/j.procs.2023.01.220.
[7] H. Xie, Y. Chen, and H. Shin, "Context-aware pedestrian detection especially for small-sized instances with Deconvolution Integrated Faster RCNN (DIF R-CNN)," *Applied Intelligence*, vol. 49, no. 3, pp. 1200–1211, Mar. 2019, doi: 10.1007/S10489-018-1326-8.
[8] J. Cao *et al.*, "Pedestrian detection algorithm for intelligent vehicles in complex scenarios," *Sensors (Switzerland)*, vol. 20, no. 13, pp. 1–19, Jul. 2020, doi: 10.3390/S20133646.
[9] X. Zong, Y. Xu, Z. Ye, and Z. Chen, "Pedestrian detection based on channel feature fusion and enhanced semantic segmentation," *Applied Intelligence*, vol. 53, no. 24, pp. 30203–30218, Dec. 2023, doi: 10.1007/S10489-023-04957-Y.
[10] D. Anitta, A. Joy, R. Kottaimalai, and M. Thilagaraj, "An Efficient System for Pedestrian Detection from Video Sequences," *Smart Innovation, Systems and Technologies (SIST)*, vol. 395, pp. 399–409, 2024, doi: 10.1007/978-981-97-5081-8_31.
[11] A. Singh, S. Bhatt, V. Nayak, and M. Shah, "Automation of surveillance systems using deep learning and facial recognition," *International Journal of System Assurance Engineering and Management*, vol. 14, pp. 236–245, Mar. 2023, doi: 10.1007/S13198-022-01844-6.
[12] Z. Qi, M. Zhou, G. Zhu, and Y. Xue, "Multiple Pedestrian Tracking in Dense Crowds Combined with Head Tracking," *Applied Sciences (Switzerland)*, vol. 13, no. 1, Jan. 2023, doi: 10.3390/APP13010440.
[13] M. H. Jafari *et al.*, "U-land: uncertainty-driven video landmark detection," *IEEE Transactions on Medical Imaging*, vol. 41, no. 4, pp. 793–804, Apr. 2021, doi: 10.1109/tmi.2021.3123547.
[14] M. K. Bhowmik, P. Saha, A. Singha, D. Bhattacharjee, and P. Dutta, "Enhancement of robustness of face recognition system through reduced gaussianity in Log-ICA," *Expert Systems with Applications*, vol. 116, pp. 96–107, Feb. 2019, doi: 10.1016/J.ESWA.2018.08.047.
[15] Z. Ilyas, Z. Aziz, T. Qasim, N. Bhatti, and M. F. Hayat, "A hybrid deep network based approach for crowd anomaly detection," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24053–24067, Jul. 2021, doi: 10.1007/S11042-021-10785-4.
[16] A. Shaik and S. M. Basha, "Optimal deep learning based object detection for pedestrian and anomaly recognition model," *International Journal of Information Technology (Singapore)*, vol. 16, no. 7, pp. 4721–4728, Oct. 2024, doi: 10.1007/S41870-024-02075-7/METRICS.
[17] A. A. Almazroey and S. K. Jarraya, "Abnormal Events and Behavior Detection in Crowd Scenes Based on Deep Learning and Neighborhood Component Analysis Feature Selection," *Advances in Intelligent Systems and Computing (AISC)*, vol. 1153, pp. 258–267, 2020, doi: 10.1007/978-3-030-44289-7_25.
[18] U. Gawande, K. Hajari, and Y. Golhar, "Real-Time Deep Learning Approach for Pedestrian Detection and Suspicious Activity Recognition," *Procedia Computer Science*, vol. 218, pp. 2438–2447, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.219.
[19] W. Luo *et al.*, "Video anomaly detection with sparse coding inspired deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 1070–1084, Mar. 2019, doi: 10.1109/tpami.2019.2944377.
[20] A. V. Hujon, T. D. Singh, and K. Amitab, "Transfer Learning Based Neural Machine Translation of English-Khasi on Low-Resource Settings," *Procedia Computer Science*, vol. 218, pp. 1–8, 2022, doi: 10.1016/j.procs.2022.12.396.
[21] Q. Liu, H. Ye, S. Wang, and Z. Xu, "YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera," *Electronics (Switzerland)*, vol. 13, no. 1, Jan. 2024, doi: 10.3390/ELECTRONICS13010236.
[22] Z. Lin, W. Pei, F. Chen, D. Zhang, and G. Lu, "Pedestrian Detection by Exemplar-Guided Contrastive Learning," *IEEE Transactions on Image Processing*, vol. 32, pp. 2003–2016, 2023, doi: 10.1109/TIP.2022.3189803.
[23] A. Alhothali, A. Balabid, R. Alharthi, B. Alzahrani, R. Alotaibi, and A. Barnawi, "Anomalous event detection and localization in dense crowd scenes," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15673–15694, Apr. 2023, doi: 10.1007/S11042-022-13967-W.
[24] M. Yang, S. Tian, A. S. Rao, S. Rajasegarar, M. Palaniswami, and Z. Zhou, "An efficient deep neural model for detecting crowd anomalies in videos," *Applied Intelligence*, vol. 53, no. 12, pp. 15695–15710, Jun. 2023, doi: 10.1007/S10489-022-04233-5/METRICS.
[25] S. Gondane, A. Thakare, C. Deshpande, and O. Gupta, "A Convolution Neural Networks and IoT-Based Approach to Surveillance System," *Intelligent Communication, Control and Devices: Proceedings of ICICCD 2020*, pp. 25–34, 2021, doi: 10.1007/978-981-16-1510-8_3.
[26] T. Wu and Y. Dong, "YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition," *Applied Sciences (Switzerland)*, vol. 13, no. 24, Dec. 2023, doi: 10.3390/APP132412977.
[27] H. Xie, W. Zheng, and H. Shin, "Occluded pedestrian detection techniques by deformable attention-guided network (Dagn)," *Applied Sciences (Switzerland)*, vol. 11, no. 13, Jul. 2021, doi: 10.3390/APP11136025.
[28] F. Zou, J. Li, and W. Min, "Distributed face recognition based on load balancing and dynamic prediction," *Applied Sciences (Switzerland)*, vol. 9, no. 4, Feb. 2019, doi: 10.3390/APP9040794.
[29] T. Wu, X. Li, and Q. Dong, "An Improved Transformer-Based Model for Urban Pedestrian Detection," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, pp. 1–16, Dec. 2025, doi: 10.1007/S44196-025-00791-X/FIGURES/12.
[30] S. Dubey, A. Boragule, J. Gwak, and M. Jeon, "Anomalous event recognition in videos based on joint learning of motion and appearance with multiple ranking measures," *Applied Sciences*, vol. 11, no. 3, p. 1344, Feb. 2021, doi: 10.3390/app11031344.

BIOGRAPHIES OF AUTHORS

Tanya Gupta has 9 years of teaching experience and is currently working as an Assistant Professor in Department of Computer Science and Engineering, Chandigarh Group of Colleges, Jhanjeri, Mohali, 140307, India. She is pursuing Ph.D. from Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed to Be University), Mullana, Ambala - 133207, Haryana, India. Her area of interest is machine learning, natural language processing, and internet of things. She can be contacted at email: tanyaguptacgc@gmail.com.

Dr. Neera Batra has more than 18 years of teaching and research experience and is currently working as a Professor in Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed to Be University), Mullana, Ambala - 133207, Haryana, India. Her areas of interest in Computer Science are AI and ML, computer vision, pervasive computing, internet of things, distributed database, and WSN. She has published more than 80 papers in national and international journals of repute and 18 patents. She can be contacted at email: neera.batra@mmumullana.org.