A Cognitive Approach To Predict the Multi-Directional Trajectory of Pedestrians

Jayachitra Virupakshipuram Panneerselvam
Department of Computer Technology,
Anna University, India
jayachitravp@annauniv.edu

Bharanidharan Subramaniam
Department of Computer Technology,
Anna University, India
bharanidharan485@gmail.com

Mathangi Meenakshisundaram
Department of Computer Technology,
Anna University, India
mathangimr@gmail.com

Abstract: *Pedestrian detection is one of the important areas in computer vision. This work is about detecting the multi-directional pedestrian's left, right, and the front movements. On recognizing the direction of movement, the system can be alerted depending on the environmental circumstances. Since multiple pedestrians moving in different directions may be present in a single image, Convolutional Neural Network (CNN) is not suitable for recognizing the multi-directional movement of the pedestrians. Moreover, the Faster R-CNN (FR-CNN) gives faster response output compared to other detection algorithms. In this work, a modified Faster Recurrent Convolutional Neural Network (MFR-CNN), a cognitive approach is proposed for detecting the direction of movement of the pedestrians and it can be deployed in real-time. A fine-tuning of the convolutional layers is performed to extract more information about the image contained in the feature map. The anchors used in the detection process are modified to focus the pedestrians present within a range, which is the major concern for such automated systems. The proposed model reduced the execution time and obtained an accuracy of 88%. The experimental evaluation indicates that the proposed novel model can outperform the other methods by tagging each pedestrian individually in the direction in which they move.*

Keywords: *Automated driving system, deep neural networks, faster recurrent-convolutional neural network, object recognition, pedestrian detection, pedestrian movement direction.*

Received January 3, 2021; accepted March 20, 2022
https://doi.org/10.34028/iajit/20/2/11

1. Introduction

In recent years, owing to the development of technology, most of the things get automated and increases productivity to a large extent. One such important automated system is driver assistance and surveillance systems. Environmental circumstances play a vital role in such automated systems. The system must sense the environment and act accordingly. Furthermore, the system should be capable of reacting to the abnormalities in the environment. This includes the interruption by other vehicles moving in the lane and the walking pedestrians. The road accidents occur especially during the nights and therefore an algorithm has been provided to detect the pedestrians precisely in [11]. The direction recognition of the pedestrians moving in the pavement plays a crucial part in assisting the automated driving system. Also, the occluded pedestrian targets need to be handled and can be detected via body parts [17]. Suppose a sudden change in directions of the pedestrian while walking on the pavement can be a life-changing threat to the driver as well as the pedestrian. In such cases, the automated driving system should be alerted to take sufficient actions and avoid misfortunes [20]. For this purpose, videos of the pedestrians walking in the pavement are captured, and the movement is continuously monitored. In regards to the easy task processing, the videos captured are split into images and then processed by the neural network.

Before the advent of deep learning, the handcrafted features were used for image recognition, where the features from edges and corners of the images were used. Also, the abrupt changes in the intensity of the image are used for the recognition task. But these handcrafted features proved to be not much efficient, especially when there is a considerable variation in the data. The high-end camera was used for capturing the videos which could detect the pedestrians exactly [18]. Over the years, with the development of deep learning and the capability of parallel processing, the tasks in Graphics Processing Unit (GPU)s stirred up the method of image recognition. Deep learning eases the job of extracting the features that constitute a significant overhead in the traditional neural network architectures, where we need experts in the field to manually design some feature extraction methods. The conventional deep neural network used for image recognition is the CNN. Despite its remarkable success, the main drawback is that it cannot correctly classify if there are multiple pedestrians moving in different directions in a single image. In general, CNN outputs a single class which has the highest probability. In this work, the MFR-CNN model is used that classifies the direction of movement of the pedestrians as left, right, or front for each pedestrian present in the image. The proposed

cognitive approach gives a multi-class prediction of pedestrians in a single image, and the process of tuning the anchor size according to the pedestrian speeds up the performance of the model. The objective and intention of the proposed work is to provide real-time assistance for autonomous vehicles in detecting multi-object and multi-directional pedestrians quickly with good accuracy.

The work of this paper is organized as follows. Section 2 describes the related works undertaken previously on pedestrian detection. Section 3 describes the proposed work, which makes use of the modified anchor sizes to detect pedestrians more efficiently. Section 4 describes the experiments, analyzes the performance of the model, and shows the results obtained on the dataset. Finally, section 5 concludes this paper.

2. Related Works

From simple to complex scenarios, various approaches have been proposed to detect pedestrians under different conditions. The previous works mostly deals about detecting whether a pedestrian is present in the given environment or not. It is an uphill task to detect pedestrians because it may be strongly impacted by appearance and shadow. The proposed work of recognizing the direction of movement is different from detecting pedestrians in an environment. The features extracted are unique to that image and different images have different feature map representations. The better the features are extracted, the better it gives insight into the image, which is the primary requirement for processing. In the early conventional approaches, handcrafted features were used for image processing, which wasn't much effective. In the later stage, the deep neural networks were used for this task, which proved to be effective with computation cost.

2.1. Conventional Approaches

In [25], Histogram of Gradients (HOG) was used to extract information from the given image. The idea behind using HOG is to capture the local object appearance and to reduce the illumination impact with the help of intensity gradients or edge directions. The image is divided into many patches, and the features for each patch are extracted by calculating the vertical gradient and the horizontal gradient. Additionally, they have used background subtraction as a step forward to improve performance. However, the assumption is that a non-uniform grid's perspective is employed, which concentrates only on important ones, and the less important regions are left behind. Yet, the less important regions may contain useful data that might help in detecting the direction of pedestrians.

In [19], the pedestrian detection method based on the modified HOG features was proposed, where they tried to reduce the dimension of the HOG features extracted. Initially, they have built a set of single-channel features based on the HOG descriptor. Then, they combined several HOG Channel features to acquire a feature with a low dimension, which on further use tends to be computationally less expensive without the loss of features. Their work decreased the dimension by 44.4% compared to the original HOG feature. In [9], pedestrians were detected using the Support Vector Machine (SVM). With the help of AdaBoost and cascading methods, the pedestrian candidates were separated from the image. Then it is ensured that whether the extracted patch contains the pedestrian or not by passing it to SVM. Additionally, a method of binary conversion is employed, where the binary values were assigned to each pixel of the image. Thus, SVM tends to accomplish this by trying to find an optimal hyper plane, which has the maximum margin between the class of pedestrians and non-pedestrians only.

In [16], the Local Binary Pattern (LBP) was used to extract the features, combined with the HOG features and finally classified by a linear classifier. In [21], a multi-scale classifier used here replaced the image feature pyramid building process, which consumed more time. The Binary Pattern of Gradient (BPG) feature derived from the HOG feature maintains local, regional characters that detected pedestrians. The performance of the proposed BPG and the combination of LBP and BPG features were compared. The latter produced better results.

In [10], the local Haar-like features combined with the edge maps were used. These Haar-like features extract the edge/contour of pedestrians. The method of localized normalization is proposed to reduce the background noise. The computation time is saved, but the efficiency of the detector is reduced. Also, the detection of multi-view pedestrians has been stated as their future work to achieve better performance and accuracy. In [12], a Local Decorrelation based method for pedestrian detection was proposed that removes correlations in local neighborhoods, which boosts the performance of the classifiers such as the decision tree, random forest, or the SVM. The local decorrelation, in combination with the oblique decision trees, gives better classification. This method reduced false positives and provided a significant boost, but the decorrelation of features was not performed across the channels, which might give better performance if done.

However, for the traditional pedestrian detection methods, there is a need to design an artificially complex feature, which requires a lot of domain knowledge and possess some limitations in robustness.

2.2. Deep Learning Approaches

With the evolution of the deep neural networks and the advent of the GPU, the image processing task has improved both in terms of accuracy and speed. In [6], CNN was used to recognize the pedestrians' direction.

The concept of sum of subtracted frames is employed as a preprocessing step by which the pedestrian is separated from the background, and then the CNN classifies it. The drawback of this model is that it can't predict multiple classes present in a single image. It only gives the class with the highest probability as output. In [7], human activity recognition such 'walk', 'jump', 'side', 'skip', 'p-jump', 'run', 'wave one hand' (i.e., wave1), 'bend', 'jack', and 'wave two hands' (i.e., wave2) are the various human actions recognized with accuracy 97.8 %. However, in pedestrian direction recognition, multiple pedestrians moving in different directions may be present in a single image, and there is a need to classify their directions such as left, right and front movement of them correctly.

In [23], a deep learning method for detecting the pedestrians was proposed, which uses Faster R-CNN (FR-CNN) combined with k-means clustering to identify the region proposals, which is then used by the detector to classify accordingly. The Region Proposal Network (RPN) was combined with the clustering method. Sliding window selection was used on the feature maps of the convolutional layer and a higher accuracy was achieved. In [13], a deep CNN was used for detecting pedestrians. Non-maximal suppression technique was used to handle multiple bounding boxes over one pedestrian. The CNN classifies the pedestrian and non-pedestrian images of the patches generated by the sliding window. The fixed-size window was shifted vertically and horizontally to detect pedestrians at apparent sizes. Stochastic Gradient Method (SGM) was used for training. However, the system needs to be evaluated on a few more datasets to make it robust and it was not suitable for real-time processing.

In [22], multi-scale pedestrian detector was proposed. The detector was built using recurrent convolution and skip pooling. The model provided a consistent performance with a bit faster speed. Similarly, in [3], multilayer channel features were proposed to increase the detection speed. However, the system produced a good performance only in recognizing the pedestrians but did not predict the directions of pedestrians. In [24], a multiclass detection network was used for detecting distorted pedestrians. The distorted pedestrians may possess different shapes. The different levels of distortion were classified with the help of the multi-classification layer. This layer can explicitly define the distorted pedestrians into our target classes, where gradient descent velocity is the prime factor. The boundaries of the classes are not fixed here. The work showed better results of detection in distorted visual pictures. However, the above approaches could recognize the pedestrians but not the direction of movement of pedestrians.

In [20], an approach based on Darknet with prior information about bounding boxes was used. The model adopts a CNN structure and anchor boxes to predict the coordinates and the pedestrian scores. Further, different ConvNets had to be used in the deep network to improve the performance. In [15], another deep learning-based approach was proposed, where the feature map is extracted with the use of PVAnet, which is a lightweight neural network. The optimized PVA network reduced the computation cost. It was combined with CNN to obtain more improved results. A decision tree classifier was used for the detection and their future work revolves around improving the performance of the model and to solve the occlusion problems.

In [14], a method based on the dynamic adaptive region convolution method was proposed, which also boosts up the process and thus reduces the time of computation. The Eigen values of a particular region were extracted to determine the characteristics of the entire region, thereby reducing the dimension. Occlusion is an essential factor that needs to be addressed. The partially occluded pedestrians are identified in [2], and compared with other classifier methods. The approach could also detect not only pedestrian targets but also other occluded objects. Sudden pedestrian crossing [8] should be detected and alerted for safe driving. Their work captured the pedestrian crossing using the Far-Infrared (FIR) camera, which leverages the hotspot features. It uses cascaded random forest with low dimensional Haar-like features and oriented center-symmetric LBPs. Adaptive scaling and the segmentation algorithm for every odd horizontal line used here reduced the computational cost. The pedestrian movement and speed were computed. The path prediction of pedestrians is still a future work to be accomplished.

After gaining knowledge of the previous works done, a conclusion can be drawn that no action has been done regarding detecting multiple pedestrians moving in different directions in a single frame. The proposed work can effectively detect each pedestrian present and the class (left, right, and front) to which they belong.

Therefore, the contributions of our work can be summarized as follows.

- A modified Faster R-CNN (MFR-CNN) architecture is proposed, which can uniquely identify multiple pedestrians moving in different directions in a single image with ease.
- The anchor sizes are fixed in such a way that the pedestrians within only a specific range are concentrated.
- A fine-tuning of the convolutional layers is done to extract a better feature map representation.
- Training of the model in the Graphics Processing Unit thereby reducing the execution time and increasing the accuracy of the model.

3. Pedestrian Detection Methodologies

This section explains the proposed work of this paper. The MFR-CNN model is built for detecting the direction

of motion of the pedestrians, i.e., left, right, and the front on the pavement. The proposed methodology of recognizing the direction of pedestrians is to reduce the processing time of mapping individual pedestrians with their corresponding class. As an initial step, the dataset is preprocessed and then given as an input to the neural network in a way that neurons could handle the images.

3.1. Data Collection and Preprocessing

3.1.1. Data Collection

The dataset includes videos of the pedestrians walking in the pavement in multiple directions. A video is simply a sequence of frames depicting the pedestrians walking, with different parameters, and the frame rate contributing the higher in detecting the movement. Frame rate usually determines the quality of videos. Typically, the frame rate in the videos is in the range of twenty frames per second to sixty frames per second. On average, there will be around thirty images generated per second on a standard video. An image may contain a single pedestrian or multiple pedestrians present in it.

3.1.2. Data Preprocessing

The images containing detailed information can be removed without losing the information present in the video with a factor known as the distance effect. Instead of eliminating images as a whole, as it may lead to loss of information, removing only three or four frames in between the frames reduces the computational task without losing the vital information present in the videos. This way of extracting images from the videos under different environmental conditions helps in detecting the motion of pedestrians. In addition to the removal of unnecessary frames, the dataset is manually annotated and the frames containing only the plain background is removed, as there is no use in training such images. The performance of the model decreases if those images were used for training.

a) Moving left. b) Moving right c) Moving front.

Figure 1. Sample images in the dataset.

The sample images of the pedestrians moving left, right, and the front is depicted in Figure 1. Principally, the time consumed for processing the videos is much higher due to the computational process involved. Owing to the processing task, the videos captured are split into images. The ground truth values are generated for each image, which includes finding the corresponding coordinates for each pedestrian to be fed into the neural network.

For the classification task, the image and the ground truth values are given as the input to the neural network. A rectangular box is recognized as an outline of the pedestrian present inside the image. The coordinates contain the exact position of the top left corner and the bottom right corner through which the coordinates of the entire rectangular box are obtained. So, a total of 4 points are obtained for each pedestrian present in the image. The class label indicating the direction of movement of the pedestrian present inside the box is also obtained. The coordinates obtained along with the class label are fed into the neural network. Figure 2 shows an image with the corresponding ground truth values. Suppose if there are as many pedestrians present in a single image, the coordinates for each of them are found separately and then fed into the network as a whole, along with the image.

Figure 2. Ground truth values for an image.

3.1.3. Noise Reduction

Another important factor that is to be dealt with the digital image processing is the noise associated with the images, which dent the clarity of feature extraction.

Image noise is the random variation of color, brightness, shades, and other attributes present in the image. Moreover, the image noise can span from almost indistinguishable flecks on a digital snapshot captured in good light, to radio-astronomical images. They are practically entirely noise, from which a small amount of data can be acquired by refined acceptable processing. A noise level would be impermissible in an image because it is not feasible to determine the subject. It critically impacts the success rate of classification. Therefore, it is removed to make the image intact.

A Gaussian function is applied to the image to remove the noise. The maximum value of the Gaussian function decrease with the increasing value of sigma which controls the degree of smoothness based on which the other pixel values vary. It computes each pixel value by calculating the weighted average of the neighbouring pixels. For higher accuracy, the nearby pixels will have the higher weights compared to the pixels which are far away. This helps in preserving the image smoothness preventing the loss of information

from the image. The Gaussian equation is stated in Equation (1), where x is the distance from the center, i.e., the origin, in the horizontal axis. Similarly, y is in the vertical axis, μ is the mean and σ is the standard deviation of the Gaussian distribution. The Gaussian function computes each pixel's new value by taking the weighted average of the neighboring pixels. As a result, the original pixel receives the highest weight and the weight decreases as the pixel moves away from the original pixel. Depending on the value of the σ, the influence of other pixels varies controlling smoothness.

Figure 3. Image with noise factor removed.

$$G(x, y) = Ae^{\frac{-(x-\mu_x)^2}{\sigma_x^2} + \frac{-(y-\mu_y)^2}{\sigma_y^2}} \quad (1)$$

This noise reduction technique helps in viewing the original image through a translucent screen, which resembles the original image, as shown in Figure 3. The exact purpose of removing noise from the image gives additional features which assist in classification besides the existing features of the normal image. This provides the classifier with the sophisticated set of features to be considered during classification.

3.2. Proposed Work

3.2.1. Workflow

The workflow starts by extracting the frames from video as an initial step, followed by manual annotation of the images. Then the set of images is separated into a set of training images and testing images. The testing set consists of images that the model has never seen during the training process. The ground truth values are generated for the training set images followed by feeding them into the proposed neural network model.

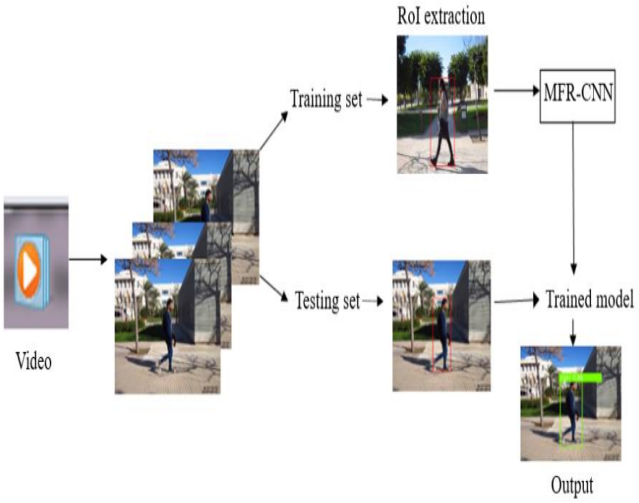


Figure 4. Flow diagram of the proposed work.

The model gets trained on the images with the specified hyper parameter values. Then, the testing set images are used to evaluate the performance of the model. Figure 4 depicts the workflow of the proposed work.

3.2.2. Feature Extraction

Initially, padding is applied to the input image, which appends zeros to the input so as to match the dimension of the input mentioned. The reason is that different images may have different sizes. As the network layers have a static input dimension, the zero padding is done to bring all the input to the same dimension.

The CNN's are the biologically inspired variants of a multi-layer perceptron. It consists of many layers which produce an output by applying a differentiable function on the input. The convolutional layer is the most important layer, which does the major part of the feature extraction task. During the forward computational pass, the filter slide across the width and height of the input, which in turn produce the activation map as the output. This gives the response of the filter at each spatial position of the input. To reduce the computational process, our proposed work restricts the connections between the input units and the hidden units of the network, thereby allowing each hidden unit to connect to only a small contiguous region of pixels in the input. The convolutional layer calculates the output of neurons that are connected to the localized regions in the input, with each of it computing the dot product between the weights of the filters and a small region of the input. The filter weights are initialized with some random values initially, which on course, gets assigned with the appropriate weights depending on the network. In general, the convolutional layer applies filters across pixels, computes the values, and then passes the output of it to the next layer.

The Rectified Linear Unit activation (ReLU) function is used, an element-wise activation function that converts the negative values to zero and thus

improves the computational process. The Rectified Linear Unit (ReLU) equation is stated as,

$$f(x) = \begin{cases} x & ,\text{if } x > 0 \\ 0 & ,\text{otherwise} \end{cases} \quad (2)$$

Here, *x* is the input to which the activation function is applied. That is, the output activation function values are all greater than zero. The pooling layer is used to reduce the size of the model by reducing the number of parameters, which combines the output of neuron clusters at a particular layer into a single neuron in the following layer. The pooling operation performs a down sampling along the spatial dimension resulting in a reduced volume. Here, the maximum pooling is used, which takes the maximum value from each cluster of neurons at the previous layer. Each layer of the network is configured with different parameters with the convolutional layer taking some special parameters, whereas the ReLU layer and the pooling layer don't take such parameters. The initial layers tend to extract the layman features while the layers deep through the network tend to extract more precise features. Figure 5 shows the convolutional layers used for feature extraction. The multispectral pedestrian detection used two convolution layers to detect under adverse illumination [5]. However, our work used four convolutional layers and provided a better accuracy. Therefore, it should be noted that using very higher number of layers tend to diminish the features.

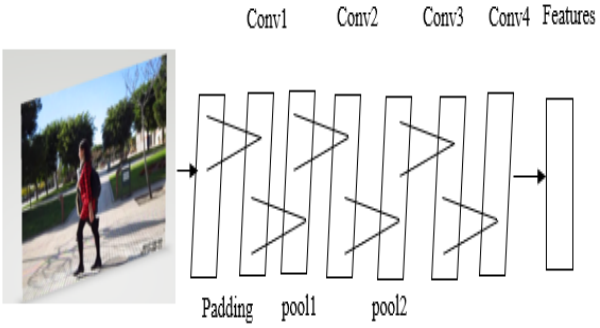


Figure 5. Convolutional layers for feature extraction.

3.2.3. Base Network

The idea of initializing the weights of the model is to give better performance since random weights will be difficult to approximate, thereby leading to performance degradation. However, the random initialization of weights needs much more epochs to specialize, resulting in much amount of training time being consumed. The ResNet model is a more generalized one which is used for multi-class classification problem. The base weights of the network are initialized with the weights of the ResNet model. The ResNet architecture overcomes the huge problem of vanishing gradient in the deep neural networks. This is because of the reasons that as the neural networks become deeper; it will be unable for the gradient to back propagate the errors through the earlier layers of the network. The ResNet architecture incorporated identity shortcut connections as a solution to the vanishing gradient problem. Once the model starts training on the training set, the network weights get updated over the training process and they get tuned in such a way that it becomes more specialized for classifying the pedestrians.

3.2.4. Modified Faster R-CNN (MFR-CNN) Network

The pedestrian detection is a way of detecting objects on a particular image. This kind of recognition takes place in real-time. Hence, the need to process the input faster and generating the response has been a major challenge. The CNN with multiple layers is usually needed to detect the presence of multiple pedestrians in a single frame. Therefore, as a similar network, we go for Recurrent Convolutional Neural Network (R-CNN). R-CNN uses selective search, which tends to produce bad region proposals. Besides, R-CNN has an extensive computational task, so it cannot be used for processing real-time applications.

Algorithm 1: MFR-CNN

Input: Set of Images I
Parameters: Gaussian function G, loop variable r, old feature map F, new feature mapF', Region Proposals R, Fully Connected Layer FC
Output: Class Label L, Offset P
1: Procedure CLASSIFIER(I)
2: For image in I do
3: Perform convolution operation on the image
4: Apply the activation function f(x),

$$f(x) = \begin{cases} x & ,if\ x > 0 \\ 0 & ,otherwise \end{cases}$$

5: Perform a maximum pooling operation
6: Flatten the extracted F
7: Apply G on the image,

$$G(x, y) = Ae^{\frac{-(x-\mu_x)^2}{\sigma_x^2} + \frac{-(y-\mu_y)^2}{\sigma_y^2}}$$

8: Extract F' for the new image
9: Concatenate F and F'
10: For r in anchor_ratio do
*11: width[r] = scale * anchor_ratio[r.width]*
*12: height[r] = scale * anchor_ratio[r.height]*
13: End for
14: Generate R
15: Reshape R and feed them to the FC layer
16: Apply the softmax equation and obtain L and P

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}}$$

17: End For
18: End Procedure

The FR-CNN generally has quicker response time compared to its other counterparts in image processing. The FR-CNN uses a separate network passing the entire image to CNN instead of applying a selective search algorithm. The network is known to generate good features from the images. The response from the FR-CNN network doesn't change the originality of the image. The shape and the structure of the image remain the same. Initially, the feature map for the image is extracted by

passing it to the convolutional layers. With the help of the feature map, the RPN is used for generating region proposals [4]. It tends to be fast and can be deployed in real-time. Usually, anchors of different sizes are used to detect the objects. The different anchor values are used such as 128, 256, and 512, and, in the end, the object proposals are obtained. Since the pedestrian will be of some fixed size and it doesn't vary to a large extent, we use fixed size anchors. The standard ratios of anchor box are 1:1, 1:2, and 2:1. However, the proposed work fixes the anchor size as 256, which boosts up the computational speed. Then pooling is applied to bring down all the proposals to the same size.

Finally, a fully connected layer is used to classify the direction of the pedestrian movement and find its offset.

Figure 6 represents the architecture of the pedestrian direction recognition model. The proposed algorithm for MFR-CNN is explained in Algorithm (1). For training the model, the input image, along with the coordinates for the particular region of interest, is fed into the model. As we need the final output probability, a softmax function is used. The softmax function converts the output values of any range to be between 0 and 1. The output of the softmax function is the probabilities of the pedestrian moving to the left, right, or to the front.

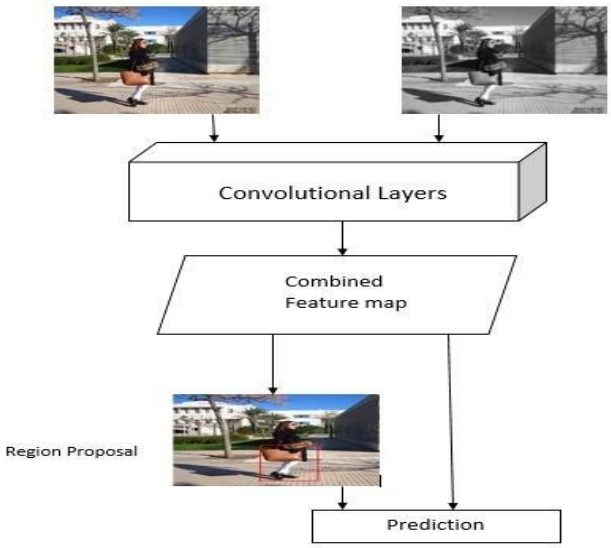


Figure 6. Architecture of the pedestrian detection model.

3.2.5. Hard Negative Mining

To improve the performance of the model, the idea of hard negative mining is used. If there is a bunch of images that contain one or more pedestrians and the ground truth values for each pedestrian is passed along with the image as input to the model. The samples may be either positive or negative samples. The positive samples are to be classified by the model by giving the corresponding output label, and the negative samples are the ones that do not belong to any of the classes. In this case, the pedestrians belong to the positive class, and all other objects belong to the negative class. So, the model will be aware of which proposals are pedestrians, so that helps in giving out the correct predictions.

For each pedestrian, the positive training example is created by looking inside the bounding box. The negative samples are generated by picking up a bunch of random bounding boxes. For each randomly generated box that doesn't overlap with the positive samples are named as negative. Once the positive and negative samples are generated, the classifier is trained with those samples, and in order to test the performance of the model, it is run on the set of training images again with a sliding window. But it turns out that the classifier isn't very good, because it throws a bunch of false positives. The false positives are the ones that are classified as pedestrians by the model, but they are not actually pedestrians. The concept of hard negative mining is that the set of falsely detected patches are collected and explicitly generating a negative example out of that patch, and add that negative sample to the training set. When the classifier is retrained, it performs better because it had an additional knowledge of negative samples and does not classify as false positives. Hence, the MFR-CNN model provides better accuracy as hard negative mining is incorporated.

4. Experimental Evaluation

4.1. Dataset

There are many datasets for detecting pedestrian movement in different directions. Caltech benchmark dataset consists of videos recorded for 10 hours captured from a moving vehicle driven through the usual traffic. The videos were split, composed of the images of numerous pedestrians with their corresponding ground truth values. The Pascalvoc dataset also accounts for detecting the motion of persons. The INRIA person dataset consists of images from various sources, including the images collected from personal digital image collections captured over a longer period consisting of only upright people. The persons above a certain minimum height are taken into consideration for easy detection of objects. Few other datasets are available for the detection of pedestrians with the normal view and from the aerial point of view. The main focus of the above datasets was only on detecting pedestrians surrounded by the bounding boxes. The limitations and the challenges in recognizing the pedestrians were not counted.

The dataset used in our work is obtained from the members of the University of Alicante [6]. The dataset consists of videos of pedestrians walking in the pavement in different directions, i.e., considering the movement of pedestrians to the left direction, in the right direction, and the front. It includes videos taken in different environments under different climatic conditions. The video is preprocessed, and images are extracted from it, which is manually annotated, producing around 1100 images. In each image, a single

pedestrian or more than one pedestrian is present, each of them moving towards some direction. So, as a result, the images consist of around 1800 pedestrians present, moving in different directions.

4.2. Result Analysis

The proposed model is trained using the GPU in the Kaggle kernel. The use of GPUs boosts up the training of the deep neural network, which generally takes much time to train due to a large number of layers in it. The GPU performs parallel processing by dividing the tasks into sub-tasks and assigns them to multiple threads. The class probabilities and the offset values generated by the proposed model are compared with the ground truth information to generate the accuracy of the model. The normal model is trained for 50 epochs with each epoch possessing 500 steps. The accuracy tends to be low for the initial epochs, and it gradually increases and remains constant after executing a certain number of epochs. The normal model provides an accuracy of around 85%. The time taken for the different tasks is listed out in Table 1.

Initially, the anchor scales are in the size of 128, 256, and 512 with the ratios of 1:1,1:2, and 2:1. These ratios are used to generate the region proposals. Generally, this is the standard configuration of the anchor box that the FR-CNN model tends to use to detect objects. The anchor box comprises of varying sizes from small to large, so that it can detect objects of different dimensions. Also, the use of the noise-reduced image as an additional segment tends to improve the accuracy of the model to a certain extent. Figure 7 gives the comparison of accuracy between the normal general model (FR-CNN) and proposed model (MFR-CNN).

Table 1. Time taken for testing an image.

Task	Time taken (sec)
Noise reduction	0.06
Prediction using CPU	1.67
Prediction using GPU	0.38

For the purpose of detecting an object, the MFR-CNN model generates region proposals with the help of anchor boxes. As in the case of object detection data sets, there may be different sizes of objects available. Therefore, the anchor boxes of the small, medium and larger sizes are used with different ratios of height and width. This accounts for a total of 9 different anchor boxes.

In this work, as the concern is only about the pedestrians, the anchor boxes are modified and made to be fixed with the size of 256 along with the ratios of 1:1, 1:2 and 2:1 with the assumptions that a person won't be too near the camera, and a person who is far away from the camera tend to be smaller in size which is not a concern in the automated driving systems. With this process of fixing the anchor size, the number of region

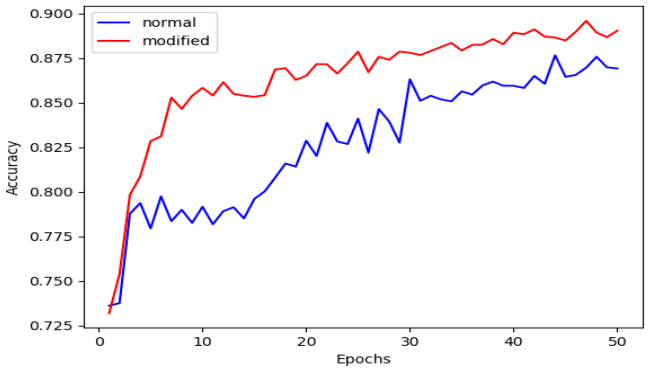


Figure 7. Comparison of accuracy between the models.

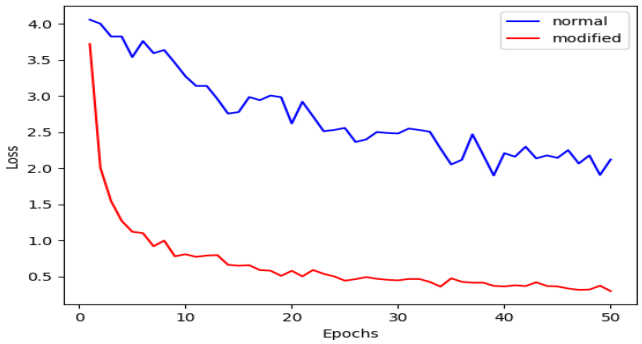


Figure 8. Comparison of loss value between the models.

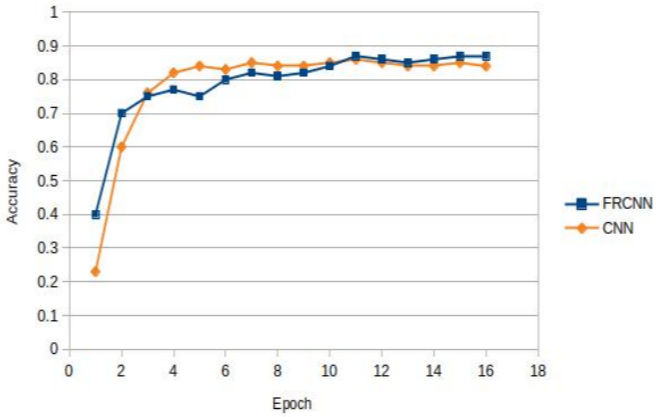


Figure 9. Comparison of proposed model vs. CNN model [6].

Proposals generated is considerably reduced and, in turn, boosts the computational speed of the model to the extent of thrice of the original speed of the model. This proves that the use of anchor sizes helps in boosting up the speed as that of the cascaded implementation of additive Kernel SVM [1] which reduces computing time in detecting postures of pedestrians.

The model is fed with an image from the test set, which consists of images never seen by the model before; it can be taken as a good measure to assess the performance of the model. When such image is passed, the feature map for the entire image is generated by passing it to the CNN. Then the different region proposals are generated by using the anchor boxes and with the help of the feature map extracted from the image. If the center of the pedestrian coincides with the center of the region proposal generated, then the coordinates for the bounding box along with the probability that it belongs to a particular direction is

returned as output. The highest probability of the pedestrian moving to a particular direction is returned.

From the graph, it is observed that the modified model tends to give a better accuracy of about88%. Figure 8 compares the loss value between the normal model (FR-CNN) and the proposed model (MFR-CNN). The plotting and the visualization of the loss always provide a better insight to show the performance of the model. Figure 9 compares the model accuracy between the proposed model and the CNN model which is obtained in [6]. Generally, CNN model is used for image classification. Though it is possible to regress bounding boxes using CNN even for multiple objects but performs poorly due to interference. Whereas, the proposed FRCNN model detects multiple objects and predicts the multi-directional movement of the pedestrian in the single scene image much speedier than CNN and hence more suitable to deploy in real-time.

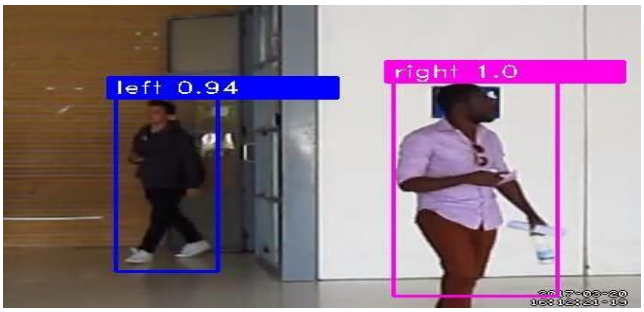
Table 2. Time comparison between the models.

Model used	Time taken per step (sec)
Normal	1.380
MFR-CNN	0.365

Generally, time and space are the two factors that are considered while executing the code. We can observe the considerable improvement in the run time of the model, as shown in Table 2, which is one of the critical factors in such real-time systems.

5. Discussions

The sample images are shown in Figure 1 and are fed as an input to the model. The model prediction for the test images are depicted in Figure 10. In Figure 10-a), there are two pedestrians present in the image, one moving to the left and the other moving to the right. The model gives the bounding box and also the probabilities. These probabilities are the outputs of the Softmax function, which converts the scores of the neural network into probabilities.



a) Pedestrian moving left and right.

b) Pedestrian moving right and left.

c) Pedestrian moving front.

d) Pedestrian moving left.

Figure 10. [a-d] Prediction yielded output for the test images.

The model gives a probability of 1.0 for the person moving to the right, indicating that it is sure that the person must move to the right. For the person moving to the left, the probability of 0.94 suggests that the model is not cent percent sure, but more or less, the person is moving to the left. A minimum threshold is fixed such that the probability less than 0.6 is ignored. Also, the probability depends on whether the exact bounding box is produced. This is because if a large bounding box is generated for a pedestrian, where only half the area represents the pedestrian and the other half consisting of unwanted information, then the features of the area without pedestrian also contribute to the probability, thereby decreasing the performance of the model.

Hence, the exact bounding box is generated to avoid the above problem. The output probabilities and the direction of a few more images are depicted in Fig. 10.

6. Conclusions

In this work, the cognitive MFR-CNN model was proposed to detect the direction of movement of the pedestrians along with the impact of using noise removed image as an additional input to the model. Also, the tuning of anchor boxes is done to deduct the number of region proposals generated, thereby reducing the time complexity by 73.5%. The faster response time of the model makes it easier to deploy in real-time. Also, the previous approaches ignored considering the direction with the lowest probability. The proposed approach overcomes this problem by giving a multi-class output, thereby considering every pedestrian present in the image. The model can be employed in autonomous driving to alert the system when pedestrians come across the way. A camera can be deployed in a vehicle to take continuous videos of pedestrians walking in the pavement. The videos taken can be fed to the model to make classifications. Suppose a pedestrian moving on the left side of the road suddenly turns right, there is a probability that he may end up crossing the road, which will be captured by the model and can be used to alert the system that a pedestrian is coming up on its way. Generally, the dataset for pedestrians available is used to detect pedestrians, not their directions. The experimental results on the dataset used; show that the model works well under different conditions giving a multi-class output.

Occlusion occurs when an object hides another object. This is the case with the pedestrians as a pedestrian walking in the pavement may conceal the presence of other pedestrians walking in the pavement. As a part of future work, the idea is to solve the occlusion problem between pedestrians, especially when multiple pedestrians are walking on the pavement.

Conflict of Interest

The authors declared that there is no conflict of interest.

References

[1] Baek J., Kim J., and Kim E., "Fast and Efficient Pedestrian Detection via the Cascade Implementation of an Additive Kernel Support Vector Machine," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 902-916, 2017.

[2] Bin W. and Shiru Q., "A study on Occluded Pedestrian Detection Based on Block-Based Features And Ensemble Classifier," *in Proceedings of 34th Chinese Control Conference*, Hangzhou, pp. 4710-4715, 2015.

[3] Cao J., Pang Y., and Li X., "Learning Multilayer Channel Features for Pedestrian Detection," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3210-3220, 2017.

[4] Chen E., Tang X., and Fu B., "A Modified Pedestrian Retrieval Method Based on Faster R-CNN with Integration of Pedestrian Detection and Re-Identification," *in Proceedings of International Conference on Audio, Language and Image Processing*, Shanghai, pp. 63-66, 2018.

[5] Chen Y., Xie H., and Shin H., "Multi-layer fusion Techniques Using A cnn for Multispectral Pedestrian Detection," *IET Computer Vision*, vol. 12, no. 8, pp.1179-1187, 2018.

[6] Dominguez-Sanchez A., Cazorla M., and Orts-Escolano S., "Pedestrian Movement Direction Recognition Using Convolutional Neural Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3540-3548, 2017.

[7] Elmezain M. and Al-Hamadi A., "Vision-Based Human Activity Recognition Using LDCRFs," *The International Arab Journal of Information Technology*, vol. 15, no. 3, pp. 389-395, 2018.

[8] Jeong M., Ko B., and Nam J., "Early Detection of Sudden Pedestrian Crossing for Safe Driving During Summer Nights," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 6, pp.1368-1380, 2017.

[9] Kharjul R., Tungar V., Kulkarni Y., Upadhyay S., and Shirsath R., "Real-Time Pedestrian Detection Using SVM and AdaBoost," *in Proceedings of International Conference on Energy Systems and Applications*, Pune, pp.740-743, 2015.

[10] Li Y., Lu W., Wang S., and Ding X., "Local Haar-Like Features in Edge Maps for Pedestrian Detection," *in Proceedings of 4th International Congress on Image and Signal Processing*, Shanghai, pp. 1424-1427, 2011.

[11] Lin C., Lin S., Hwang C., and Chen Y., "Real-Time Pedestrian Detection System with Novel Thermal Features At Night," *in Proceedings of IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Montevideo, pp. 1329-1333, 2014.

[12] Nam W., Dollar P., and Han J., "Local Decorrelation for Improved Detection," *arxiv.org, abs/1406.1134*, 2014.

[13] Orozco C., Buemi M., and Berlles J., "New Deep Convolutional Neural Network Architecture for Pedestrian Detection," *in Proceedings of 8th International Conference of Pattern Recognition Systems*, Madrid, pp. 1-6, 2017.

[14] Qiu D. and Liu D., "The Optimal Pedestrian Detection Algorithm Based on Dynamic Adaptive Region Convolution Model," *Chinese Automation Congress (CAC)*, Jinan, pp. 7808-7910, 2017.

[15] Sun W., Zhu S., Ju X., and Wang D., "Deep Learning Based Pedestrian Detection," *Chinese*

Control and Decision Conference (CCDC), Shenyang, pp. 1007-1011, 2018.

[16] Wang A., Dai S., Yang M., and Iwahori Y., "A Novel Human Detection Algorithm Combining HOG with LBP Histogram Fourier," *in Proceedings of 10th International Conference on Communications and Networking in China*, Shanghai, pp. 793-797, 2015.

[17] Wang S., Cheng J., Liu H., Wang F., and Zhou H., "Pedestrian Detection via Body Part Semantic and Contextual Information with DNN," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3148-3159, 2018.

[18] Wang Y. and Liu F., "A New Pedestrian Detection Algorithm Used for Advanced Driver-Assistance System with One Cheap Camera," *in Proceedings of International Conference on Mechatronic Sciences, Electric Engineering and Computer*, Shengyang, pp.1315-1318, 2013.

[19] Weixing L., Haijun S., Feng P., Qi G., and Bin Q., "A fast Pedestrian Detection Via Modified HOG Feature," *in Proceedings of 34th Chinese Control Conference*, Hangzhou, pp. 3870- 3873, 2015.

[20] Yang Z., Li J., and Li H., "Real-Time Pedestrian Detection for Autonomous Driving," *in Proceedings of International Conference on Intelligent Autonomous Systems*, Singapore, pp. 9-13, 2018.

[21] Yu B., Ma Y., and Li J., "Fast Pedestrian Detection with Multi-Scale Classifiers," *in Proceedings of International Conference on Computing Intelligence and Information System*, Nanjing, pp. 225-230, 2017.

[22] Zhang C. and Kim J., "Multi-Scale Pedestrian Detection Using Skip Pooling and Recurrent Convolution," *Multimedia Tools and Applications*, vol. 78, pp. 1719-1736, 2018.

[23] Zhang H., Du Y., Ning S., Zhang Y., Yang S., and Du C., "Pedestrian Detection Method Based on Faster R-CNN," *in Proceedings of 13th International Conference on Computational Intelligence and Security*, Hong Kong, pp. 427-430, 2017.

[24] Zhang J., Xiao J., Zhou C., and Peng C., "A Multi-Class Pedestrian Detection Network for Distorted Pedestrians," *in Proceedings of 13th IEEE Conference on Industrial Electronics and Applications*, Wuhan, pp. 1079-1083, 2018.

[25] Zhang S. and Wang X., "Human Detection and Object Tracking based on Histograms of Oriented Gradients," *in Proceedings of 9th International Conference on Natural Computation*, Shenyang, pp. 1349-1353, 2013.

Jayachitra Virupakshipuram Panneerselvam received the B.E. degree in computer science and engineering from university of Madras, Chennai, India, in 2000 and the M.E degree in Computer Science and Engineering and Ph.D. degree in Information and communication engineering from Anna University, Chennai, India, in 2008 and 2017 respectively. Currently, she is an assistant professor in Department of Computer Technology, MIT campus, Anna University, Chennai, India. Her research interest includes Wireless Sensor Networks, Machine learning and Internet of Things.

Bharanidharan Subramaniam received the Bachelors of Engineering degree in Computer Science and Engineering from Anna University, India in 2019. He is currently pursuing a career in the software industry as lead software engineer at Samsung. His areas of interest include Data structures and Machine Learning.

Mathangi Meenakshisundaram received the Bachelors of Engineering degree in Computer Science and Engineering at Department of Computer Technology from Anna University, India, graduated in the year 2020. She is currently pursuing a career in the software industry as software engineer at Fidelity Investments. Her areas of interests include Machine Learning, Big Data, Database Management and Internet of Things.