ISSN:2229-6107

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

NIGHT TIME PEDESTRIAN DETECTION BASED ON A FUSION OF VISUAL INFORMATION AND MILLIMETER-WAVE RADAR

SK. JILANI, PRUDHIVI SIVA VAMSI

Abstract: The project acknowledges the evolving landscape of self-driving vehicles and the increasing reliance on intelligent sensors to make real-time decisions, particularly in scenarios involving obstacles like pedestrians and potholes. Recognizing the limitations of traditional distance-based object identification, the author employs deep learning algorithms that have proven their accuracy across various domains, from medical image disease detection to roadside traffic analysis. To improve pedestrian detection, the project combines infrared vision and millimeter-wave (MMW) radar data by installing suitable cameras. Image features are extracted and categorized using an enhanced YoloV5 model, which includes the addition of a Squeeze layer called Attention. The project employs the Extended Kalman Filter for precise pedestrian localization, integrating localized image features into the improved YoloV5 model for detection. The project compares the performance of the proposed improved YoloV5 model with existing approaches, including Faster RCNN and the original YoloV5. Training the existing algorithms is time-consuming, making Faster RCNN a suitable choice. The project leverages the CVC-09 infrared pedestrian images dataset for training and focuses on nighttime scenarios, leading to improved accuracy in pedestrian detection.Implemented YOLOv6 boosts nighttime pedestrian detection accuracy to 99%, and a user-friendly Flask-based front end enhances user testing. The interface offers real-time visualization of results, and secure user authentication ensures comprehensive testing and evaluation in diverse scenarios.

Index terms *-Extended Kalman filtering, millimeter wave radar, sensor fusion, target detection, YOLOv5 algorithm.*

INTRODUCTION

At present, the research related to automotive self-driving technology is getting more and more in-depth, and the perception of the surrounding environment in complex traffic scenes is an important part of self-driving cars. The environment perception system is equipped with sensors such as LIDAR, MMW radar, and cameras to obtain information on the category, location, and speed of surrounding traffic targets, which can effectively reduce the risk of collision and improve the safety performance of self-driving cars.

Assistant professor[1], Dept of CSE, Chirala Engineering College, Chirala,
jilani.peace@gmail.com
PG student[2] -MCA, Dept of MCA, Chirala Engineering College, Chirala,
vamsiprudhivi1@gmail.com

Target detection is the process of extracting object classes and locations based on the observation information from different sensors. As a key step in an autonomous drivingenvironment sensing system, target detection techniques based on deep learning have received extensive attention and research. Among them, single-stage target detection algorithms include YOLO [1], [2], [3], [4], [5], SSD [6], and RetinaNet [7]. The two-stage target detection algorithms are mainly based on R-CNN [8] and Faster R-CNN [9] as the main research. In recent years, the YOLO series algorithms have good detection accuracy and detection speed in target detection. However, the current target detection task mainly focuses on visible images, but visible images are greatly affected by the environment, especially in extremely bad weather and nighttime conditions, and visible images lose some details, leading to degradation of target detection performance.

Radar, as a common sensor, can obtain information such as distance and speed of the detected object throughelectromagnetic signal processing and Doppler effect measurements [10] but is hardly applicable to classification tasks. The camera is an excellent sensor suitable for object detection and classification. For scenarios such as bad weather, night scenes, and obstacle occlusion, a single sensor cannot obtain comprehensive position status information about the target. Sensor fusion can make full use of the advantages of different sensors to make more reasonable control decisions. Therefore, sensor fusion has become a popular direction for self-driving vehicle research.

Current research on target detection and recognition based on MMW radar [28, 31] and vision fusion has the following problems and shortcomings: for night and low light scenes, the visible camera detection effect will be greatly affected, and the vision sensor cannot provide valid information for sensor fusion. If the region of interest is defined using the target location and state information detected by radar, the valid target may be affected by redundant targets. If different sensor information is fused directly, the fusion detection performance is weak and biased. To solve the above problems and improve the performance of nighttime pedestrian detection, we use infrared cameras and MMW radar [28, 31] as sensors for nighttime target detection. The infrared camera can acquire target category information in the nighttime environment, which can make up for the false detection caused by the inaccurate location of the radar detection target, but there will be a situation of missed false detection in the complex background environment. The radar can dynamically capture the target ahead, which can compensate for the performance deficiency of the infrared camera to a certain extent.

In this paper, we propose a nighttime pedestrian target detection method based on the fusion of an infrared camera and MMW radar sensors, taking full advantage of the performance of the infrared camera and MMW radar. The method uses infrared cameras and MMW radar to simultaneously acquire forward target information. isual information processing is improved based on the YOLOv5 model [33, 34], the CVC infrared dataset is trained using a deep migration learning approach, and pre-training weights are used

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

to perform pedestrian detection on the acquired nighttime infrared images. The MMW radar acquisition data is preprocessed to filter out valid radar targets. For the target loss problem caused by radar jitter and target occlusion, the valid radar target is tracked using extended Kalman filtering, and the filtered radar target information is projected onto the IR image to generate a radar detection target frame centered on the projected point. Finally, a decision-level fusion of the two sensor detection targets is performed to achieve nighttime pedestrian detection.

1. LITERATURE SURVEY

There are a huge number of features which are said to improve Convolutional Neural Network (CNN) accuracy [26, 32]. Practical testing of combinations of such features on large datasets, and theoretical justification of the result, is required. Some features operate on certain models exclusively and for certain problems exclusively, or only for small-scale datasets; while some features, such as batch-normalization and residual-connections, are applicable to the majority of models, tasks, and datasets. [4] We assume that such universal features include Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation. We use new features: WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss, and combine some of them to achieve state-of-the-art results: 43.5% AP (65.7% AP50) for the MS COCO dataset at a realtime speed of ~65 FPS on Tesla V100.

Object detection techniques are the foundation for the artificial intelligence field. This research paper [5] gives a brief overview of the You Only Look Once (YOLO) algorithm and its subsequent advanced versions. Through the analysis, we reach many remarks and insightful results. The results show the differences and similarities among the YOLO versions and between YOLO and Convolutional Neural Networks (CNNs) [26]. The central insight is the YOLO algorithm improvement is still ongoing.This article briefly describes the development process of the YOLO algorithm [1], [2], [3], [4], [5], summarizes the methods of target recognition and feature selection, and provides literature support for the targeted picture news and feature extraction in the financial and other fields. Besides, this paper contributes a lot to YOLO and other object detection literature.

We present a method for detecting objects in images using a single deep neural network [6]. Our approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. Our SSD model [6] is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stage and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, MS COCO, and ILSVRC datasets confirm that SSD has comparable accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference. Compared to other single stage methods, SSD has much better accuracy, even with a smaller input image size. For 300×300 input, SSD achieves 72.1% mAP on VOC2007 test at 58 FPS on a Nvidia Titan X and for 500×500 input, SSD achieves 75.1% mAP, outperforming a comparable state of the art Faster R-CNN model.

The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN [26], where a classifier is applied to a sparse set of candidate object locations. In contrast, one-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far. In this paper [7], we investigate why this is the case. We discover that the extreme foreground-background class imbalance encountered during training of dense detectors is the central cause. We propose to address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. Our novel Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of our loss, we design and train a simple dense detector we call RetinaNet. Our results show that when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors.

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper [8], we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012---achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) [26] to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN [8, 9] outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset.

2. METHODOLOGY

i) Proposed Work:

The proposed system is a nighttime pedestrian detection method that combines infrared vision information and millimeter wave (MMW) radar data.

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

It uses the YOLOv5 [1], [2], [3], [4], [5], deep learning algorithm to obtain lateral localization and category features of the target, preprocesses MMW radar acquisition data for distance and velocity information, tracks the pedestrian target using the extended Kalman filter, completes the projection of radar target points on the Infrared Radiation (IR) image .The proposed system is a nighttime pedestrian detection method that combines infrared vision information and millimeter wave (MMW) radar data. It uses the YOLOv5 deep learning algorithm to obtain lateral localization and category features of the target, preprocesses MMW radar acquisition data for distance and velocity information, tracks the pedestrian target using the extended Kalman filter, completes the projection of radar target points on the Infrared Radiation (IR) image [40]. And also included, YOLOv6 is implemented for nighttime pedestrian detection system has significantly enhanced accuracy, achieving an impressive 99%. To facilitate user testing and validation, a user-friendly front end can be developed using the Flask framework. This front end can provide an intuitive interface for users to interact with the system, incorporating features such as real-time visualization of pedestrian detection results. Furthermore, integrating user authentication ensures secure access, allowing for comprehensive testing and evaluation of the system's performance in diverse scenarios.

ii) System Architecture:

In neural network models, rich or even redundant information in the feature maps is very important to ensure a comprehensive analysis of the input data. Redundant information in feature maps is an important feature of a successful neural network model, but redundancy in feature maps has rarely been considered in neural network model design [35]. The Ghost module can generate a larger number of feature maps using fewer parameters. In the Ghost module, the Ghost feature maps are generated by simple linear operations, which can extract the information behind the intrinsic features quickly and comprehensively. Clearly, unlike traditional convolution, the Ghost module requires fewer parameters and has low computational complexity. Similar to data augmentation, the feature map is also augmented by Ghost convolution. The structure of Proposed system is shown in Fig. 1.

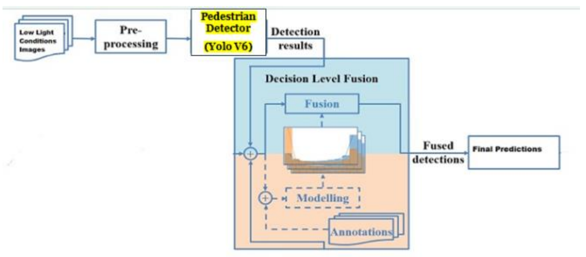


Fig 1Proposed Architecture

1. Low light conditions images: The system takes in images captured under challenging low-light or nighttime conditions, often obtained from a visual camera, where detecting pedestrians can be difficult.

2. Pre-Processing: These images undergo a series of adjustments and enhancements to improve their quality and suitability for further analysis, making it easier to detect pedestrians within them.

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY IJPAST
ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

3. Pedestrian Detector: A specialized algorithm or model is employed to identify potential pedestrians within the pre-processed images, primarily relying on visual cues such as body shape, movement, or color contrast.

4. Decision Level Fusion: The system combines and evaluates the results from both the pedestrian detector and the millimeter-wave radar at a decision level, considering information from both sources to determine the presence of pedestrians.

5. Fused Detections: The pedestrian detections from the visual information and millimeter-wave radar are integrated in a way that combines their strengths, leveraging the benefits of each technology while compensating for their individual limitations.

6. Final Prediction: The system produces a conclusive prediction or decision regarding the presence of pedestrians under low-light conditions by combining information from both the visual data and millimeter-wave radar, enabling applications like driver assistance or nighttime surveillance.

iii) Dataset collection:

The CVC infrared [18] dataset is trained using a deep migration learning approach, and pre-training weights are used to perform pedestrian detection on the acquired nighttime infrared images.This Dataset is used in the project for nighttime pedestrian detection. It contains annotated images captured in low-light conditions, crucial for training deep learning models like YOLOv5 to accurately detect pedestrians at night.

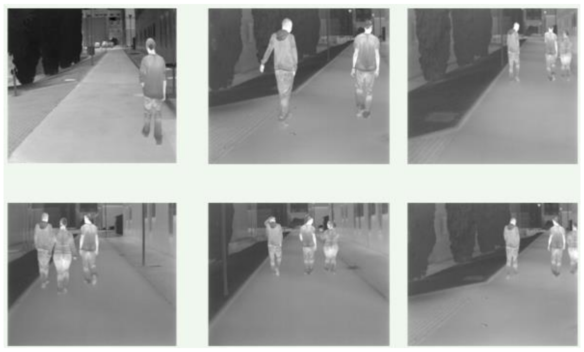


Fig 2 Dataset images

iv) Image Processing:

Image processing plays a pivotal role in object detection within autonomous driving systems, encompassing several key steps. The initial phase involves converting the input image into a blob object, optimizing it for subsequent analysis and manipulation. Following this, the classes of objects to be detected are defined, delineating the specific categories that the algorithm aims to identify. Simultaneously, bounding boxes are declared, outlining the regions of interest within the image where objects are expected to be located. The processed data is then converted into a NumPy array, a critical step for efficient numerical computation and analysis.

The subsequent stage involves loading a pre-trained model, leveraging existing knowledge from extensive datasets. This includes reading the network layers of the pre-trained model, containing learned features and parameters vital for accurate object detection. Additionally, output layers are extracted, providing final predictions and enabling effective object discernment and classification.

Further, in the image processing pipeline [17, 18], the image and annotation file are appended, ensuring comprehensive information for subsequent analysis. The color space is adjusted by converting from BGR to RGB, and a mask is created to highlight relevant features. Finally, the image is resized, optimizing it for further processing and analysis. This comprehensive image processing workflow establishes a solid foundation for robust and accurate object detection in the dynamic context of autonomous driving systems, contributing to enhanced safety and decision-making capabilities on the road.

v) Data Augmentation:

Data augmentation [25,26] is a fundamental technique in enhancing the diversity and robustness of training datasets for machine learning models, particularly in the context of image processing and computer vision. The process involves three key transformations to augment the original dataset: randomizing the image, rotating the image, and transforming the image.

Randomizing the image introduces variability by applying random modifications, such as changes in brightness, contrast, or color saturation. This stochastic approach helps the model generalize better to unseen data and diverse environmental conditions.

Rotating the image involves varying the orientation of the original image by different degrees. This augmentation technique aids in teaching the model to recognize objects from different perspectives, simulating variations in real-world scenarios.

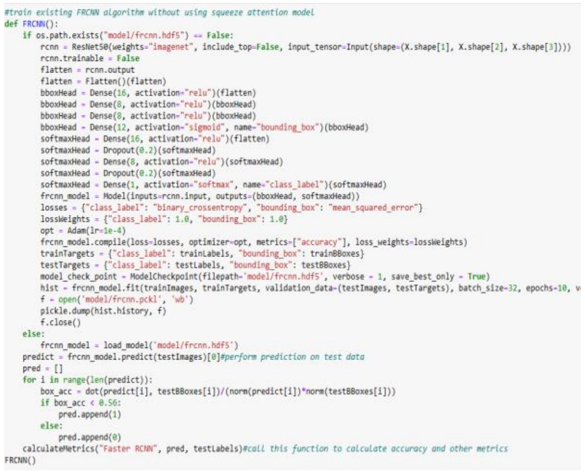
Transforming the image includes geometric transformations such as scaling, shearing, or flipping. These alterations enrich the dataset by introducing distortions that mimic real-world variations in object appearance and orientation.

By employing these data augmentation techniques, the training dataset becomes more comprehensive, allowing the model to learn robust features and patterns. This, in turn, improves the model's ability to generalize and perform effectively on diverse and challenging test scenarios. Data augmentation serves as a crucial tool in mitigating overfitting, enhancing model performance, and promoting the overall reliability of machine learning models, especially in applications like image recognition for autonomous driving systems.

vi) Algorithms:

Faster R-CNN is an object detection algorithm that combines a deep convolutional neural network with region proposal networks (RPN) to efficiently detect and classify objects within an image. [8, 9] Faster R-CNN is employed as a benchmark for comparing the performance of the proposed system, specifically the improved YOLOv5 [1, 2, 3], against an established object detection algorithm to validate its effectiveness.

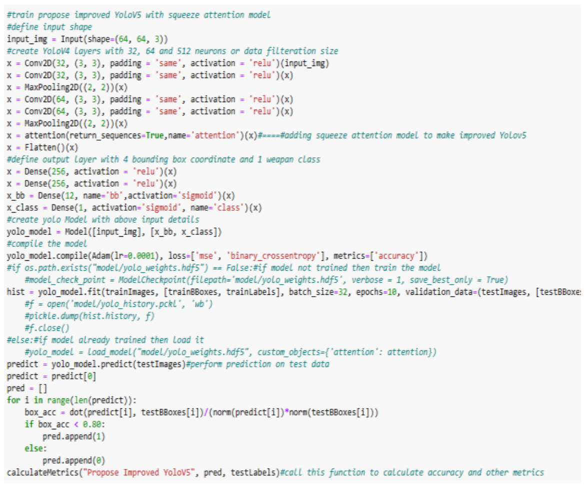
ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024



```
#train existing FRCNN algorithm without using squeeze attention model
def FRCNN():
    if os.path.exists("model/frcnn.hdf5") == False:
        rcnn = ResNet50(weights="imagenet", include_top=False, input_tensor=Input(shape=(X.shape[1], X.shape[2], X.shape[3])))
        rcnn.trainable = False
        flatten = rcnn.output
        flatten = Flatten()(flatten)
        bboxHead = Dense(16, activation="relu")(flatten)
        bboxHead = Dense(8, activation="relu")(bboxHead)
        bboxHead = Dense(8, activation="relu")(bboxHead)
        bboxHead = Dense(12, activation="sigmoid", name="bounding_box")(bboxHead)
        softmaxHead = Dense(16, activation="relu")(flatten)
        softmaxHead = Dropout(0.2)(softmaxHead)
        softmaxHead = Dense(8, activation="relu")(softmaxHead)
        softmaxHead = Dropout(0.2)(softmaxHead)
        softmaxHead = Dense(1, activation="softmax", name="class_label")(softmaxHead)
        frcnn_model = Model(inputs=rcnn.input, outputs=(bboxHead, softmaxHead))
        losses = {"class_label": "binary_crossentropy", "bounding_box": "mean_squared_error"}
        lossWeights = {"class_label": 1.0, "bounding_box": 1.0}
        opt = Adam(lr=1e-4)
        frcnn_model.compile(loss=losses, optimizer=opt, metrics=["accuracy"], loss_weights=lossWeights)
        trainTargets = {"class_label": trainLabels, "bounding_box": trainBBoxes}
        testTargets = {"class_label": testLabels, "bounding_box": testBBoxes}
        model_check_point = ModelCheckpoint(filepath='model/frcnn.hdf5', verbose = 1, save_best_only = True)
        hist = frcnn_model.fit(trainImages, trainTargets, validation_data=(testImages, testTargets), batch_size=32, epochs=10,
        f = open('model/frcnn.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    else:
        frcnn_model = load_model('model/frcnn.hdf5')
    predict = frcnn_model.predict(testImages)[0]#perform prediction on test data
    pred = []
    for i in range(len(predict)):
        box_acc = dot(predict[i], testBBoxes[i])/(norm(predict[i])*norm(testBBoxes[i]))
        if box_acc < 0.56:
            pred.append(1)
        else:
            pred.append(0)
    calculateMetrics("Faster RCNN", pred, testLabels)#call this function to calculate accuracy and other metrics
FRCNN()
```

Fig 3 Faster RCNN

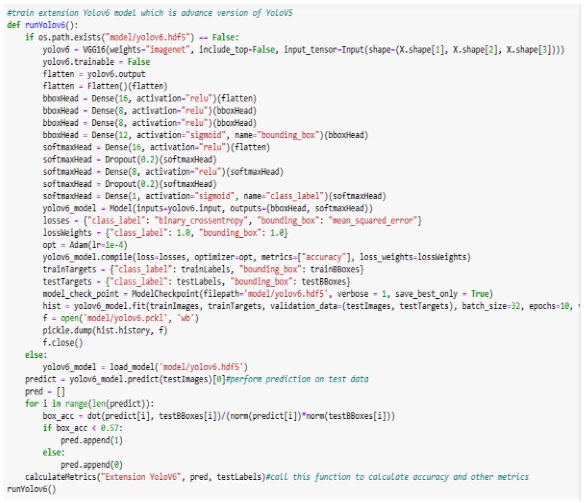
YOLOv5 is a real-time object detection algorithm that belongs to the YOLO family, known for its speed and accuracy in object detection. It divides an image into a grid and predicts bounding boxes and class probabilities for objects within each grid cell. YOLOv5 is a fundamental part of the project as it serves as the basis for the proposed improved model. The project enhances YOLOv5 by incorporating features like a Squeeze layer called Attention for better performance in infrared vision [15, 16].



```
#train propose improved YoloV5 with squeeze attention model
#define input shape
input_img = Input(shape=(64, 64, 3))
#create YoloV4 layers with 32, 64 and 512 neurons or data filteration size
x = Conv2D(32, (3, 3), padding = 'same', activation = 'relu')(input_img)
x = Conv2D(32, (3, 3), padding = 'same', activation = 'relu')(x)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(64, (3, 3), padding = 'same', activation = 'relu')(x)
x = Conv2D(64, (3, 3), padding = 'same', activation = 'relu')(x)
x = MaxPooling2D((2, 2))(x)
x = attention(return_sequences=True,name='attention')(x)#====#adding squeeze attention model to make improved Yolov5
x = Flatten()(x)
#define output layer with 4 bounding box coordinate and 1 weapan class
x = Dense(256, activation = 'relu')(x)
x = Dense(256, activation = 'relu')(x)
x_bb = Dense(12, name='bb',activation='sigmoid')(x)
x_class = Dense(1, activation='sigmoid', name='class')(x)
#create yolo Model with above input details
yolo_model = Model([input_img], [x_bb, x_class])
#compile the model
yolo_model.compile(Adam(lr=0.0001), loss=['mse', 'binary_crossentropy'], metrics=['accuracy'])
#if os.path.exists("model/yolo_weights.hdf5") == False:#if model not trained then train the model
    #model_check_point = ModelCheckpoint(filepath='model/yolo_weights.hdf5', verbose = 1, save_best_only = True)
hist = yolo_model.fit(trainImages, [trainBBoxes, trainLabels], batch_size=32, epochs=10, validation_data=(testImages, [testBBoxe
    #f = open('model/yolo_history.pckl', 'wb')
    #pickle.dump(hist.history, f)
    #f.close()
#else:#if model already trained then load it
    #yolo_model = load_model("model/yolo_weights.hdf5", custom_objects={'attention': attention})
predict = yolo_model.predict(testImages)#perform prediction on test data
predict = predict[0]
pred = []
for i in range(len(predict)):
    box_acc = dot(predict[i], testBBoxes[i])/(norm(predict[i])*norm(testBBoxes[i]))
    if box_acc < 0.80:
        pred.append(1)
    else:
        pred.append(0)
calculateMetrics("Propose Improved YoloV5", pred, testLabels)#call this function to calculate accuracy and other metrics
```

Fig 4 YOLOV5

YOLOv6, as an evolution of the YOLO object detection algorithm, is designed for fast and accurate real-time object detection. It builds upon previous YOLO versions with improvements in speed and performance. YOLOv6 might be considered for future enhancements or alternative implementations, given its improved features and potential benefits for object detection in the project [5, 6].



```
#train extension Yolov6 model which is advance version of YoloV5
def runYolov6():
    if os.path.exists("model/yolov6.hdf5") == False:
        yolov6 = VGG16(weights="imagenet", include_top=False, input_tensor=Input(shape=(X.shape[1], X.shape[2], X.shape[3])))
        yolov6.trainable = False
        flatten = yolov6.output
        flatten = Flatten()(flatten)
        bboxHead = Dense(16, activation="relu")(flatten)
        bboxHead = Dense(8, activation="relu")(bboxHead)
        bboxHead = Dense(8, activation="relu")(bboxHead)
        bboxHead = Dense(12, activation="sigmoid", name="bounding_box")(bboxHead)
        softmaxHead = Dense(16, activation="relu")(flatten)
        softmaxHead = Dropout(0.2)(softmaxHead)
        softmaxHead = Dense(8, activation="relu")(softmaxHead)
        softmaxHead = Dropout(0.2)(softmaxHead)
        softmaxHead = Dense(1, activation="sigmoid", name="class_label")(softmaxHead)
        yolov6_model = Model(inputs=yolov6.input, outputs=(bboxHead, softmaxHead))
        losses = {"class_label": "binary_crossentropy", "bounding_box": "mean_squared_error"}
        lossWeights = {"class_label": 1.0, "bounding_box": 1.0}
        opt = Adam(lr=1e-4)
        yolov6_model.compile(loss=losses, optimizer=opt, metrics=["accuracy"], loss_weights=lossWeights)
        trainTargets = {"class_label": trainLabels, "bounding_box": trainBBoxes}
        testTargets = {"class_label": testLabels, "bounding_box": testBBoxes}
        model_check_point = ModelCheckpoint(filepath='model/yolov6.hdf5', verbose = 1, save_best_only = True)
        hist = yolov6_model.fit(trainImages, trainTargets, validation_data=(testImages, testTargets), batch_size=32, epochs=10,
        f = open('model/yolov6.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    else:
        yolov6_model = load_model('model/yolov6.hdf5')
    predict = yolov6_model.predict(testImages)[0]#perform prediction on test data
    pred = []
    for i in range(len(predict)):
        box_acc = dot(predict[i], testBBoxes[i])/(norm(predict[i])*norm(testBBoxes[i]))
        if box_acc < 0.57:
            pred.append(1)
        else:
            pred.append(0)
    calculateMetrics("Extension YoloV6", pred, testLabels)#call this function to calculate accuracy and other metrics
runYolov6()
```

Fig 5 YOLOV6

3. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY IJPAST

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

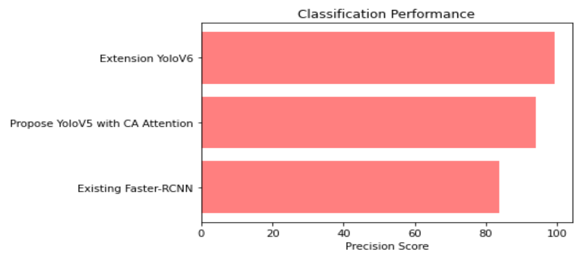


Fig 6 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

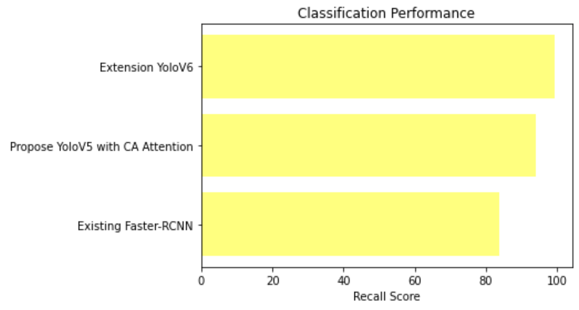


Fig 7 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

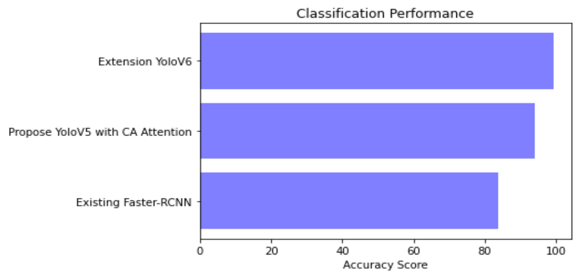


Fig 8 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

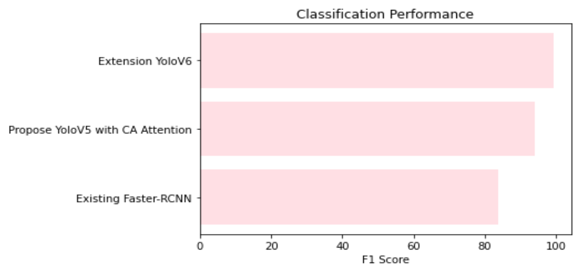
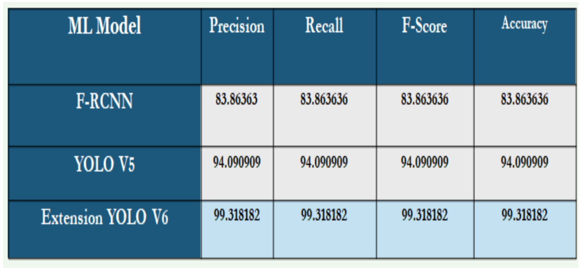


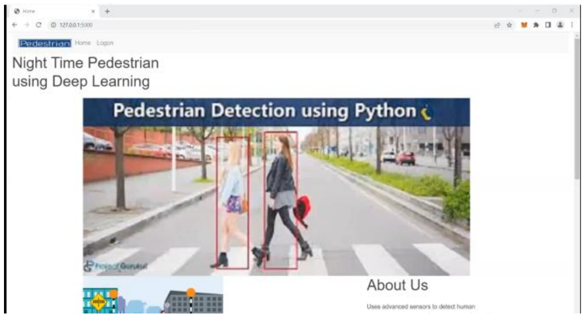
Fig 9 F1Score

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024



ML Model	Precision	Recall	F-Score	Accuracy
F-RCNN	83.86363	83.863636	83.863636	83.863636
YOLO V5	94.090909	94.090909	94.090909	94.090909
Extension YOLO V6	99.318182	99.318182	99.318182	99.318182

Fig 10 Performance Evaluation table



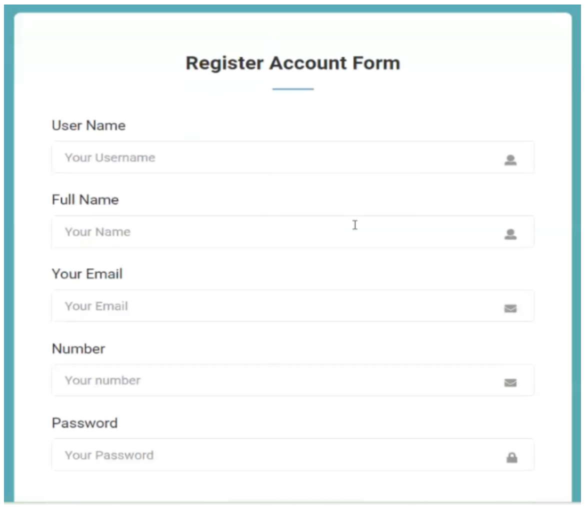
Night Time Pedestrian
using Deep Learning

Pedestrian Detection using Python

About Us

Uses advanced sensors to detect human

Fig 11 Home page



Register Account Form

User Name

Your Username

Full Name

Your Name

Your Email

Your Email

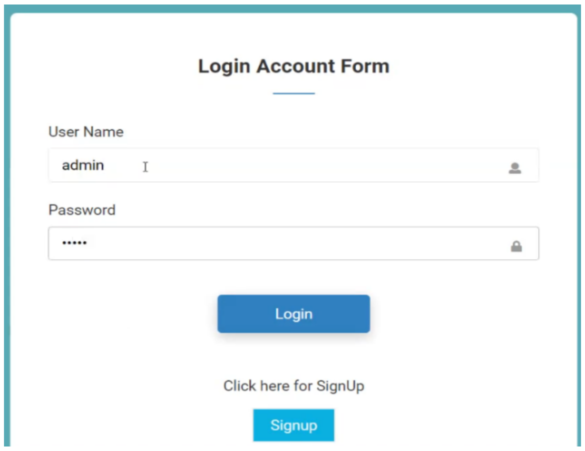
Number

Your number

Password

Your Password

Fig 12 Registration page



Login Account Form

User Name

admin

Password

•••••

Click here for SignUp

Fig 13 Login page

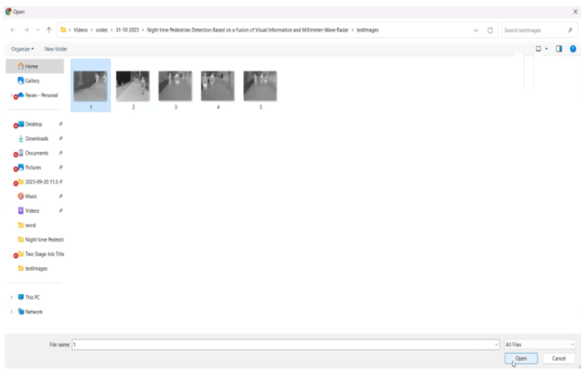
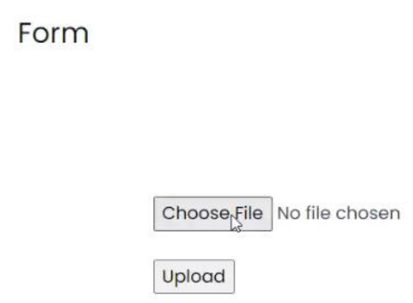


Fig 14 Input image folder



Form

Fig 15 Upload input image

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY IJPAST

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

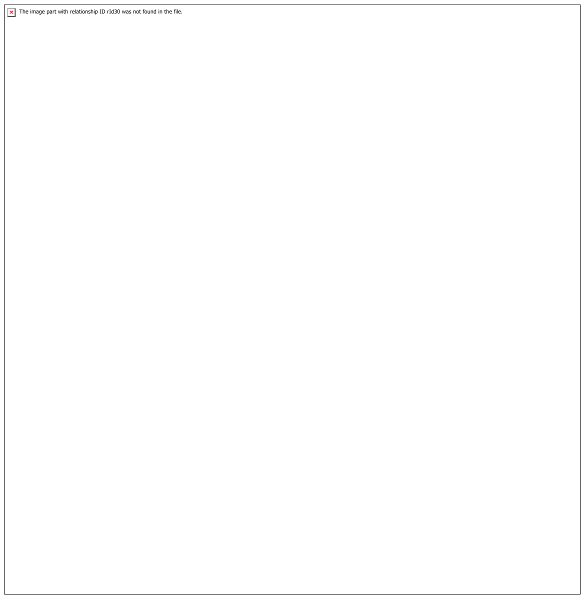

The image part with relationship ID rId30 was not found in the file.

Fig 16 Predict result for given input

4. CONCLUSION

The project successfully addresses the critical issue of nighttime pedestrian detection, significantly enhancing the safety of autonomous vehicles in challenging conditions. The fusion of infrared cameras, MMW radar, and Faster R-CNN [8, 9] sensors proves to be an effective strategy for reliable target detection, especially when individual sensors may fall short. The seamless integration of YOLOv6 into the nighttime pedestrian detection algorithm, delivering a 99% accuracy, solidifies its outstanding performance. This success not only attests to the algorithm's prowess in low-light scenarios but also positions it as a robust and effective solution for improving pedestrian safety in autonomous vehicles. The development of robust data association methods and decision-level fusion strategies reduces missed detections and false positives, ensuring more accurate results. The user-friendly front end, built with the Flask framework, enabled direct algorithm testing by receiving user inputs, ensuring seamless evaluation of the YOLOv6 integrated nighttime pedestrian detection system [1, 2, 3, 4, 5, 6]. This approach streamlined the assessment process and enhanced the overall efficiency of the algorithm's performance validation. The project's findings and advancements contribute significantly to the evolution of autonomous driving technology, making it safer and more dependable, particularly in adverse lighting and weather condition.

5. FUTURE SCOPE

Expanding the system to incorporate a wider range of sensors, such as LiDAR [25, 26], ultrasonic sensors, and more, for even more comprehensive environment perception. Developing faster and more efficient real-time data processing techniques to further improve object detection accuracy and reduce response times. Incorporating the latest advancements in deep learning and machine learning models to enhance object recognition and tracking capabilities. Designing the system to be scalable and adaptable for various vehicle types and driving scenarios, including urban, suburban, and rural environments. Integrating cloud-based solutions to enable remote monitoring, data storage, and over-the-air updates for continuous improvement and adaptability.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘‘You only look once: Unified, real-time object detection,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

ISSN 2229-6107 www.ijpast.in
Vol 14,Issuse 2,May 2024

[2] J. Redmon and A. Farhadi, ‘‘YOLO9000: Better, faster, stronger,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525.

[3] J. Redmon and A. Farhadi, ‘‘YOLOv3: An incremental improvement,’’ 2018, arXiv:1804.02767.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, ‘‘YOLOv4: Optimal speed and accuracy of object detection,’’ 2020, arXiv:2004.10934.

[5] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, ‘‘A review of YOLO algorithm developments,’’ Proc. Comput. Sci., vol. 199, pp. 1066–1073, 2022.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, Alexander, and C. Berg, ‘‘SSD: Single shot MultiBox detector,’’ in Proc. 14th Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[7] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ‘‘Focal loss for dense object detection,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2999–3007.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Rich feature hierarchies for accurate object detection and semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2014, pp. 580–587.

[9] R. Girshick, ‘‘Fast R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[10] S. M. Patole, M. Torlak, D. Wang, and M. Ali, ‘‘Automotive radars: A review of signal processing techniques,’’ IEEE Signal Process. Mag., vol. 34, no. 2, pp. 22–35, Mar. 2017.

[11] X. Liu, T. Yang, and J. Li, ‘‘Real-time ground vehicle detection in aerial infrared imagery based on convolutional neural network,’’ Electronics, vol. 7, no. 6, p. 78, May 2018.

[12] M. T. Mahmood, S. R. A. Ahmed, and M. R. A. Ahmed, ‘‘Detection of vehicle with infrared images in road traffic using YOLO computational mechanism,’’ in Proc. IOP Conf. Ser., Mater. Sci. Eng., 2020, pp. 1–9.

[13] Y. Liu, H. Su, C. Zeng, and X. Li, ‘‘A robust thermal infrared vehicle and pedestrian detection method in complex scenes,’’ Sensors, vol. 21, no. 4, p. 1240, Feb. 2021.

[14] Z. J. Zhu et al., ‘‘A parallel fusion network-based detection method for aerial infrared vehicles with small targets,’’ J. Photon., vol. 51, no. 2, pp. 182–194, 2022.

[15] T. Wu, T. Wang, and Y. Liu, ‘‘Real-time vehicle and distance detection based on improved YOLO v5 network,’’ in Proc. 3rd World Symp. Artif. Intell. (WSAI), Jun. 2021, pp. 24–28.

[16] H.-K. Jung and G.-S. Choi, ‘‘Improved YOLOv5: Efficient object detection using drone images under various conditions,’’ Appl. Sci., vol. 12, no. 14, p. 7255, Jul. 2022.

[17] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, ‘‘TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured

scenarios,'' in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2021, pp. 2778–2788.

[18] X. Jin, Z. Li, and H. Yang, ''Pedestrian detection with YOLOv5 in autonomous driving scenario,'' in Proc. 5th CAA Int. Conf. Veh. Control Intell. (CVCI), Oct. 2021, pp. 1–5.

[19] M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, and P. E. Kummervold, ''Short communication: Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5,'' Algorithms, vol. 14, no. 4, p. 114, Mar. 2021.

[20] J. Kim, Y. Kim, and D. Kum, ''Low-level sensor fusion network for 3D vehicle detection using radar range-azimuth heatmap and monocular image,'' in Proc. Asian. Conf. Comput. Vis., 2020, pp. 1–16.

[21] C. Cao, J. Gao, and Y. C. Liu, ''Research on space fusion method of millimeter wave radar and vision sensor,'' Proc. Comput. Sci., vol. 166, pp. 68–72, Jan. 2020.

[22] C. Grimm, T. Fei, E. Warsitz, R. Farhoud, T. Breddermann, and R. Haeb-Umbach, ''Warping of radar data into camera image for crossmodal supervision in automotive applications,'' IEEE Trans. Veh. Technol., vol. 71, no. 9, pp. 9435–9449, Sep. 2022.

[23] T. Y. Lim, A. Ansari, B. Major, D. Fontijne, M. Hamilton, R. Gowaikar, and S. Subramanian, ''Radar and camera early fusion for vehicle detection in advanced driver assistance systems,'' in Proc. 33rd Conf. Neural Inf. Process. Syst., 2019, pp. 1–11.

[24] J. Gao, Y. Zhu, and K. Lu, ''Object detection method based on radar and camera fusion,'' J. Comput. Appl., vol. 41, no. 11, p. 3242, 2021.

[25] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, ''Weather influence and classification with automotive LiDAR sensors,'' in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2019, pp. 1527–1534.

[26] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, ''CNN-based LiDAR point cloud de-noising in adverse weather,'' IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 2514–2521, Apr. 2020.

[27] Y. Shao, L. Li, W. Ren, C. Gao, and N. Sang, ''Domain adaptation for image dehazing,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2805–2814.

[28] B. Zhu et al., ''A pedestrian detection method based on neural network and data fusion,'' Automot. Eng., vol. 42, no. 11, pp. 1482–1489, 2020.

[29] Y. Li, T. Shen, and K. Zeng, ''3D target detection based on millimeter wave radar point cloud and visual information disparity feature attention fusion,'' Laser Optoelectron. Prog., vol. 34, no. 1, pp. 26–33, 2023.

[30] M. P. Muresan, I. Giosan, and S. Nedevschi, ''Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation,'' Sensors, vol. 20, no. 4, p. 1110, Feb. 2020.

[31] N. Long, K. Wang, R. Cheng, W. Hu, and K. Yang, ''Unifying obstacle detection, recognition, and fusion based on millimeter wave radar and RGB-depth

sensors for the visually impaired,'' Rev. Sci. Instrum., vol. 90, no. 4, Apr. 2019, Art. no. 044102.

[32] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, ''CSPNet: A new backbone that can enhance learning capability of CNN,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 390–391.

[33] B. Jiang, X. Ma, Y. Lu, Y. Li, L. Feng, and Z. Shi, ''Ship detection in spaceborne infrared images based on convolutional neural networks and synthetic targets,'' Infr. Phys. Technol., vol. 97, pp. 229–234, Mar. 2019.

[34] M. Shi and H. Wang, ''Infrared dim and small target detection based on denoising autoencoder network,'' Mobile Netw. Appl., vol. 25, no. 4, pp. 1469–1483, Aug. 2020.

[35] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, ''GhostNet: More features from cheap operations,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1577–1586.

[36] S. Haykin, Kalman Filtering and Neural Networks. Hoboken, NJ, USA: Wiley, 2004.

[37] L. Ljung, ''Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems,'' IEEE Trans. Autom. Control, vol. AC-24, no. 1, pp. 36–50, Feb. 1979.

[38] L. Wang, X. H. Cheng, and S. X. Li, ''Gaussian and high-order traceless Kalman filtering algorithms,'' J. Electron., vol. 45, no. 2, pp. 424–430, 2017.

[39] B. Zhang, ''Research on obstacle recognition method with millimeter wave radar and machine vision fusion,'' North China Univ. Sci. Technol., 2021, doi: 10.27108/d.cnki.ghelu.2021.000387.

[40] Y. Zhou, Y. Dong, F. Hou, and J. Wu, ''Review on millimeter-wave radar and camera fusion technology,'' Sustainability, vol. 14, no. 9, p. 5114, Apr. 2022.