Bozok
Journal of Engineering and Architecture

e-ISSN: 3023-4298
Araştırma Makalesi/Research Article

LIDAR ve Kamera Verilerinin Birleştirilmesiyle Gerçek Zamanlı Çoklu Nesne Tanıma

Mert Can YAMAN[1], Şerafettin EREL[1]*

[1]Ankara Yildirim Beyazit University, Faculty of Engineering and Natural Sciences, Department of Electrics & Electronics Engineering, Etlik, Ankara, TURKEY

MAKALE BİLGİSİ

Makale Tarihleri:

Geliş tarihi
27.08.2023
Kabul tarihi
05.09.2023
Yayın tarihi
29.12.2023

Anahtar Kelimeler:

LIDAR,
Nesne Tanıma,
Veri Füzyonu,
Video İşleme,
iOS

ÖZET

Nesne tanıma günümüzün en önemli araştırma konularındandır. Tarımdan, savunma ve uzay sanayisine kadar her alanda kapsamlı kullanımı olan nesne tanımanın öneminin giderek artacağı beklenmektedir. Bu çalışmada, iPhone 13 Pro Max'in dahili kamerası ve LIDAR sensöründen eş zamanlı olarak alınan veriler kullanılarak kullanıcı tarafından sınırları belirlenebilen bir alan içerisinde gerçek zamanlı nesne tanıma yapılmıştır. Çalışmada Swift programlama dili, framework olarak ise SwiftUI kullanılmıştır. MS COCO veri setindeki elemanlar ile çalışma gerçekleştirilmiştir. Nesne tanıma işlemleri için YOLO V5 kullanılmış ve video işleme süreçleri Swift Metal kullanılarak gerçek zamanlı veriler üzerinde gerçekleştirilmiştir. Kamera ve LIDAR füzyonuna bağlı olarak gerçek-zamanlı olarak alınan verilerden elde edilen her bir frame üzerinde kullanıcının arayüzde belirlediği minimum-maksimum uzaklığa göre görüntü alanı daraltılarak Swift Metal ile video işleme gerçekleştirilmiştir. Alınan anlık video verilerindeki her bir frame içerisinde kullanıcının belirlediği değer aralığındaki bulunan nesnelerin konturları dışındaki alanlar karartılmıştır. Böylece karartılmış olan her bir frame içerisindeki nesnelere nesne tanıma işlemi gerçekleştirilmiştir. Sonuç olarak, arayüzde 0-15 m sınır aralığında ayarlanabilecek şekilde gerçek zamanlı LIDAR ve kamera verileriyle nesne tanıma işlemi gerçekleştirilmiştir.

Real-Time Multi-Object Recognition Using the Fusion of LIDAR and Camera Data

ARTICLE INFO

Article history:

Received
27.08.2023
Accepted
05.09.2023
Published
29.12.2023

Keywords:

LIDAR
Object Recognition
Data Fusion
Video Processing
iOS

ABSTRACT

Object recognition is currently one of the most significant research topics. Its significance is expected to steadily increase due to its extensive applications across various fields, from agriculture to defense and the space industry. In this study, real-time object recognition processes were conducted within a user-defined area using data simultaneously captured from the built-in camera of the iPhone 13 Pro Max and its integrated LIDAR sensor. The Swift programming language was employed, and SwiftUI was chosen as the framework. The study utilized elements from the MS COCO dataset, employing the YOLO V5 algorithm for object recognition. Real-time video processing was accomplished using Swift Metal. The YOLO V5 algorithm was utilized for object recognition, and video processing was carried out in real-time, narrowing down the area based on the minimum-maximum distance determined in the interface using the real-time fused data from the camera and LIDAR. Areas outside the contours of objects, defined by user-specified value ranges in each frame of the captured real-time video data, were darkened. Consequently, the object recognition process was performed on objects within each darkened frame. As a result, object recognition was successfully conducted within a user-defined range of 0-15 meters, as configured in the interface.

1. INTRODUCTION

Object recognition, one of the computer vision technologies, is used for detection, location and recognition of objects in images and videos. Object recognition is commonly used in autonomous vehicles, robotics, aircraft, weapon systems, communication systems, security, surveillance, visualization, medical imaging, autonomous machine control and inspection, and more [1]. Their use is becoming increasingly common. With the gradual development of cutting-edge object recognition algorithms, it is predicted

ORCID ID: Mert Can YAMAN: 0009-0008-5413-895X; Şerafettin EREL: 0000-0002-2437-1127
***Sorumlu yazar(lar)/Corresponding author(s):** Mert Can YAMAN, *Ankara Yildirim Beyazit University, Faculty of Engineering and Natural Sciences, Department of Electrics & Electronics Engineering, Etlik, Ankara, TURKEY*
Tel:+905053846639
Fax: -
E-mail: mertcanyaman@hotmail.com

Bu makaleye atıfta bulunmak için/To cite this article: M.C. Yaman, Ş. Erel, "Real-Time Multi-Object Recognition Using the Fusion of LIDAR and Camera Data", Bozok Journal of Engineering and Architecture, vol. 2, no. 2, pp. 1-19, 2023

that it will be used in numerous areas, considering future use cases [2]. Object recognition has witnessed rapid progress in fields like artificial intelligence, deep learning, and machine learning. Advancements in algorithms and larger datasets continuously enhance the accuracy and performance of object recognition systems. Thus, object recognition technologies represent a significant area of research and development used across various industries today [3]. In the field of object recognition, data types such as camera, LIDAR (Laser Imaging Detection and Ranging), RADAR (Radio Detection and Ranging) and ultrasonic sensor data can be used to improve object detection and recognition processes and obtain more reliable results.

It is known that camera data is used in many different applications in the object recognition field. Object recognition is accomplished by a combination of image processing and artificial intelligence techniques. Camera data is used to detect and recognize objects. LIDAR is a remote sensing technology that measures the distance and 3D structures of distant objects using laser beams. LIDAR detects the reflected signals of the emitted laser beams to generate data called 3D point clouds. This data can be used to build a precise model of objects and surfaces on Earth. LIDAR also plays a pivotal role in object recognition and detection and is used in various application areas. Nikhitha, M. et al., in their study using camera data, carried out a research that could predict diseases in fruits and vegetables causing a decrease in crop yield and determine the degree of disease [4]. Sogabe et al. pointed out that although cataract, which is one of the biggest causes of blindness, can be solved with surgical treatment, it is difficult to treat because the tools and equipment are expensive, and it takes a long time for a surgeon to acquire the necessary skills for surgery. They carried out a deep learning-based object recognition study that can detect surgical instruments placed in the eye and their positions using camera data [5]. Janakiramaiah et al. emphasized the importance of automatic target detection systems based on object recognition technologies in military and stated that the performance of existing algorithms may be low due to the lack of data to train deep learning models stemming from security restrictions. They developed a new object recognition algorithm based on camera data that could be a solution to this problem [6]. Rezaei et al. developed a deep learning-based traffic modeling system by using real-time camera data. With this system, they performed the classification and location determination of vehicles and pedestrians in traffic. In addition, this system can make future position predictions of vehicles and pedestrians [7]. Kottner, S. et al., using the LIDAR sensor of iPhone 13 Pro, have developed an application that will facilitate crime scene investigation processes in forensic incidents with object recognition technology in a short time. With this application, they successfully completed the documentation process collecting 3D evidence data at a mock-up crime scene in about 2 minutes [8]. Okochi et al. performed object recognition by extracting the 3D structure of the surrounding objects with the point cloud data obtained with the LIDAR sensor [9]. Guyot et al., in their study, performed topographic anomaly detection using deep learning algorithms based on aerial LIDAR data of the terrain in a region and presented a new perspective for archaeological mapping [10]. Tatsumi et al. developed a mobile application that estimates the root diameter and spatial coordinates of trees using LIDAR point cloud data available from iPhones with built-in LIDAR sensors [11]. The combined use of detection systems such as LIDAR and camera is called sensor fusion and requires the use of more advanced artificial intelligence algorithms. Sensor fusion is the process of combining data from multiple sensors and extracting a common meaning. In other words, it is important not only to use two sensors together, but also to effectively process and interpret these data [12]. LIDAR measures distances with high precision and generates point cloud data. This is useful for pinpointing the position and size of surrounding objects and obstacles [13]. LIDAR sends laser beams to the environment and can also work effectively at night and in low light conditions, as it works by receiving the beams reflected from objects. The combination of the two provides better environmental perception in any lighting conditions. By combining LIDAR's precise distance measurement and the camera's color and pattern detection capabilities, more exact results in object recognition and classification can be achieved [14]. In this study, real-time multi-object recognition was performed using the camera and LIDAR data fusion with the iPhone 13 Pro Max mobile phone. The details of the study are given in the rest of the article.

2. MATERIAL AND METHOD

In this study, real-time multi-object recognition is aimed in the area to be limited according to the minimum-maximum values that can be set in the user interface with sensor fusion. Since the only mobile phone brand with a built-in LIDAR sensor is Apple iPhone devices, the study aimed to be carried out in the mobile environment is planned for iOS, the operating system of these devices. The point cloud and camera data required for real-time multi-object recognition within a user-limited area were acquired using the iPhone 13 Pro Max, which has a built-in LIDAR sensor and also a built-in high-resolution camera.

2.1. Software Infrastructure

It is aimed to realize real-time object recognition within a certain area limitation by using data received simultaneously from the built-in camera and LIDAR sensor of the iPhone 13 Pro Max mobile phone. The Swift programming language was employed for the study, with SwiftUI serving as the chosen framework. The study was carried out with the elements in the dataset of MS COCO (Microsoft Common Objects in Context), which is frequently used in the literature in the field of deep learning object recognition.

YOLO V5 (You Look Only Once), one of the state-of-the-art artificial neural network algorithms that can detect objects in real time and classify their positions, was used for object recognition. Video processing was performed on real-time data using Swift Metal, Apple's library for graphics processing and parallel computing on the iOS platform. According to the point cloud data received in real time from the iPhone's built-in LIDAR sensor, video processing was carried out on each frame obtained from the data simultaneously received from the iPhone camera, and the area in the frames is limited within the minimum and maximum values. The images of the materials used in the study are given in Fig. 1 below.

Figure 1. Images of materials used in object recognition

Algorithms used before deep learning in the field of object recognition are considered as classical recognition methods.

2.2. Classical Recognition Methods

Classical recognition methods are based on the use of various mathematical algorithms based on the different properties of the objects in the images, mostly without the use of artificial intelligence technologies.

2.2.1. Viola Jones Sensors

Viola and Jones have developed an algorithm that detects faces in real time with high precision in video image frames and much faster than previous studies on this subject. In order to determine the facial features, the Haar features calculated according to the density difference of the pixels represented as black and white in different rectangular regions in the frame form the basis of the study. The top region of a feature may be darker while the bottom region may be brighter. This feature density is used to capture different features in different facial regions. Two-rectangle, three-rectangle and four-rectangle features that are useful for capturing feature difference along edge, line and diagonal are Haar features and are shown in Fig. 2 below.

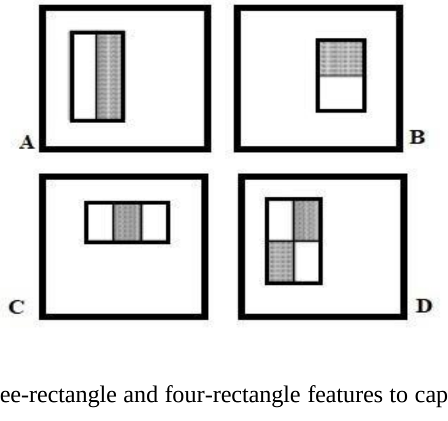

Figure 2. Pixel structures with two-rectangle, three-rectangle and four-rectangle features to capture different features in different regions in face detection used in Viola-Jones algorithm [15].

In the training phase, Haar features and classifiers are trained using the sample data set. After the weak classifiers are trained iteratively with the Boosting algorithm, they are combined to form a strong classifier [15].

2.2.2. HOG Detector

Histogram of Oriented Gradients (HOG) is a computer vision method widely used in applications such as object recognition, face recognition, and motion detection. Using HOG, a feature vector is generated that specifies the local gradient distribution of the image. After calculating the gradients on the pixels, the image is divided into small cells and the histogram of the gradients in each cell is calculated. With these histograms, the local gradient distribution is determined. Histograms of cells created by dividing the image into small cells are grouped by blocks. Blocks are regions that contain histograms of neighboring cells in an image area. Histograms within blocks are normalized to become a more efficient feature vector in difficult lighting conditions. Then, the feature vectors of all blocks are combined to obtain the HOG feature vector, which indicates the local gradient distributions of the image. These vectors are used as input to capture characteristic features for object detection. Support Vector Machine (SVM), which is a frequently used classification method in the field of machine learning, and HOG approach are widely used in object recognition applications. Dalal et al. (2005) trained the SVM classifier to detect humans with HOG features [16]. The SVM takes HOG feature vectors as input and creates a decision constraint for human detection. In the training phase, the dataset was divided into human (positive) and non-human (negative) and used for optimization of SVM classification. The Fig. 3 below shows the HOG vector distributions (b), (c-d) positive and negative SVM weighted HOG vector distributions calculated using a test image (a) and the calculated rectangular regions of this image.

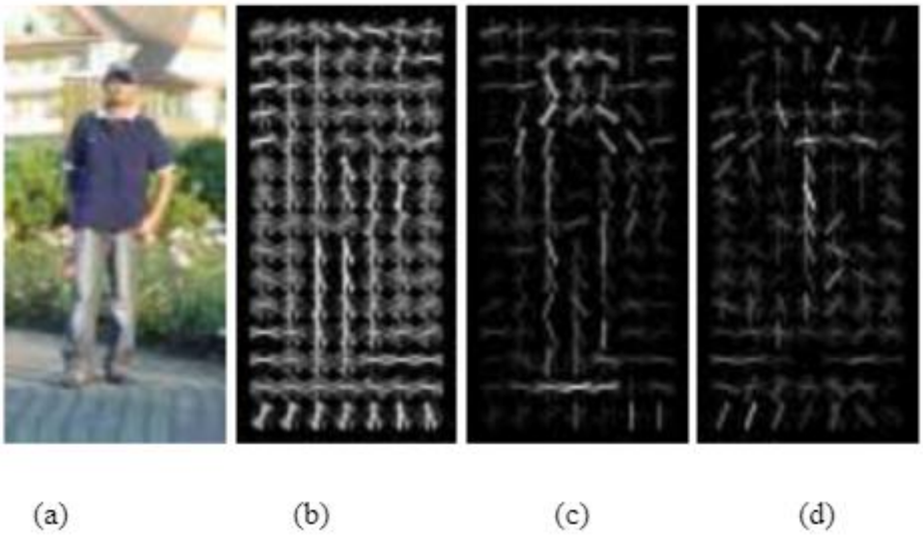

Figure 3. HOG vector distributions applied on the test image [16].

This study has greatly contributed to the HOG + SVM approach becoming a popular method in the field of human detection and object recognition. HOG has been particularly successful in object recognition. The use of HOG + SVM provides high accuracy in applications such as detection of people and vehicles. However, HOG also has some weaknesses. In particular, it can be sensitive to scale changes and rotations. Therefore, HOG may need to be used in conjunction with other feature extraction methods to ensure complete accuracy in some cases.

2.2.3. Deformable Part-Based Model (DPM)

The Deformable Part-Based Model (DPM) is a model used for object recognition and is particularly effective for the detection of people and other deformable objects. DPM is an approach that considers different parts of the object and their positions in order to detect and locate objects in an image. The basic idea of DPM is based on a model in which objects are separated into their components and the properties and locations of each component are learned. Instead of representing the object as a whole, the model treats it as a combination of parts that make up the object. Each piece represents a feature of the object and can have different sizes and shapes. Training of DPM is performed on a training set. In the training process, the properties and positions of the parts of the objects are learned. This information is combined to create a working model. The learned model is used to detect the object in a test image. The model scans the object on the image using part properties and locations and detects possible locations. One of the advantages of DPM is its flexibility to deformation and different scales. Learning parts independently allows him to better handle the different sizes and shapes of objects. In addition, the ability to represent the structure of objects and exploit the combination of object parts provides more accurate and comprehensive detection. DPM was first described by Felzenszwalb et al. as a continuation of the HOG detector. DPM uses HOG detector for object detection and uses HOG features to model and detect different parts of the object [17].

2.3. Deep Learning Based Recognition Methods

Deep learning is a sub-branch of machine learning consisting of artificial neural networks. It enables computer systems to automatically learn complex data structures and tasks. Deep learning is carried out using multi-layer artificial neural networks. Deep learning-based object detection methods are today's most advanced artificial intelligence techniques and continue to develop with studies in this field.

2.3.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) or ConvNet is an artificial neural network used in the field of deep learning, which gives successful results especially in image processing tasks. The main purpose of CNNs is to extract features from a given input image and use these features to perform tasks such as classification, detection and segmentation. The CNN architecture consists of key components such as convolutional layers, activation functions, pooling layers, and fully-connected layers [18]. An example model of the CNN architecture is shown in Fig. 4.

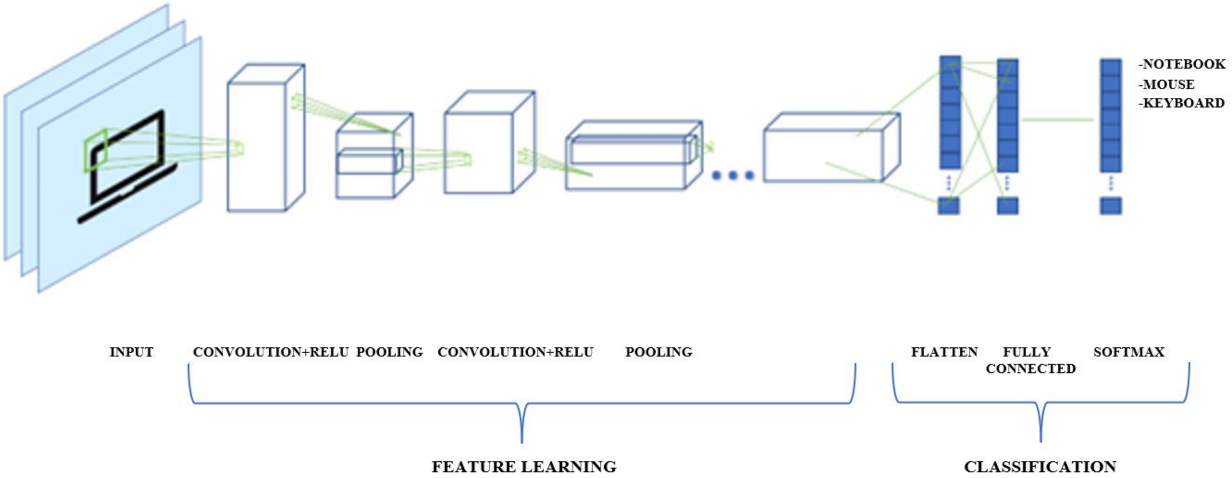

Figure 4. A convolutional neural network (CNN) architecture model

Input, image data from the dataset can be image frames obtained by separating frames from real-time or non-real-time video data with video processing applications. Input data can be from RGB, HSV, CMYK, Grayscale, YUV and similar color spaces. In addition, the resolution of the images may vary. Since it will be difficult to calculate in high resolution image data, it is very important to reduce the resolution of the images in the CNN algorithm without losing their important properties. Thus, the algorithm becomes applicable to very large datasets. Convolution layers are formed by applying filters of certain sizes on the input image by shifting. These filters are learnable parameters to capture different features of the image. For example, edges, corners, or more complex features. Each filter (kernel) is a matrix, usually 3x3 or 5x5 in size, representing a visual pattern. These filters are trained to capture patterns that will be used to detect different features of the visual data. The filter is moved over the input image step by step by convolution operation and element multiplication is done at each position [19]. As a result, a new value is obtained at each location and these values are aggregated into an output matrix. This process affects the output size based on the filter size and the convolution step size. As an example, Fig. 5 shows the convolution operation performed by shifting a filter to two-dimensional image pixels in an input. As a result of the process, a convolved feature map was obtained. Feature maps are generated by applying filters to each input image. Each filter learns its weights to detect different features, and each filter extracts a different feature map from the input data. These feature maps are then transferred to the pooling layers or other convolutional layers so that the network has the ability to learn more complex features.

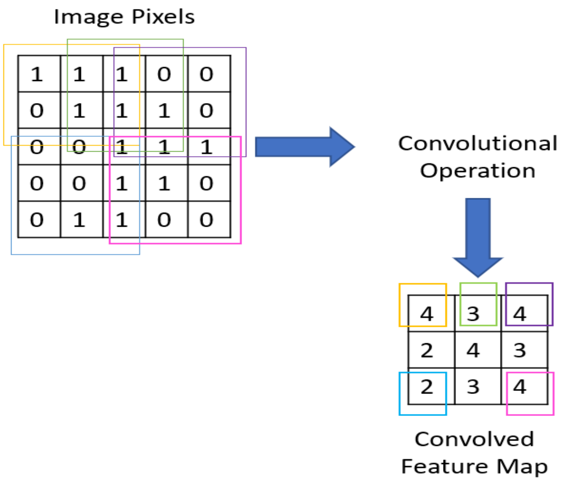

Image Pixels

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Filter

1	0	1
0	1	0
1	0	1

Convolved Feature Map

4	3	4
2	4	3
2	3	4

Figure 5. Convolution operation performed by applying a filter to a 2D image

An activation function is used after each convolution layer. This function adds a non-linearity to the output, allowing the network to learn more complex features. This can help the network learn better and reduce overfitting problems. The ReLU (Rectified Linear Unit) activation function is an example commonly used in CNNs. The non-linear activation function, ReLU, sets the negative inputs to zero, while leaving the positive inputs intact [20]. Equationally;

$$f(x) = \max(0, x) \tag{1}$$

it can be expressed with the equation (1) giving 0 as the output when $x < 0$ and giving a linear function as the output when $x \geq 0$.

Pooling layers are used to reduce the size of input data and reduce computational complexity. Common pooling operations such as max pooling, min pooling and average pooling can be used. Full link layers make the output of the CNN usable for classification or any other task. These layers flatten the output and are connected to one or more dense layers. Dense layers consist of multiple perceptrons capable of learning. CNNs are also trained with a loss function and back propagation algorithm used to update and optimize the weights of the deep learning model. Using large amounts of data usually allows the network to learn its weights appropriately to the data [21]. CNNs can be used in the areas of image classification, object recognition, face recognition, segmentation, video analytics, and more. Deep learning, and especially CNNs, has had a huge impact on computer vision and has achieved successful results in many application areas.

2.3.2. YOLO (You Look Only Once) Algorithm

The YOLO algorithm, which is a new approach in the field of deep learning-based object recognition and works much faster than classical object recognition approaches and other CNN-based object recognition algorithms, was first introduced by Redmon et al. in 2016 [22]. Classic object recognition methods, such as DPM, use a model that includes parts and their properties that are defined separately for each object class [23]. CNN algorithms provide classification and positioning information by first detecting regional objects in a video frame or an image with the region proposal step, and then extracting features in these regions. YOLO is an algorithm that performs the object detection task in a single pass. It processes the input image at once, predicting the class and position of all objects at once. It simultaneously obtains classification and positioning information as it predicts all objects in the image at the same time. Each cell is associated with the object class and bounding box information. This approach performs the object detection task in separate stages [24]. In classical approaches and algorithms such as region proposal-based CNN, the classification process is slower and more complex compared to the YOLO algorithm, especially in real-time object recognition applications, since each element must be trained separately [25]. In the YOLO algorithm presented by Redmon et al., a unified model of all stages in a convolutional neural network is constructed. After the elements in the image in the input pass through a single neural network consisting of multiple convolutional neural networks, the algorithm generates prediction vectors corresponding to each element. Unlike CNN, instead of repeating the process for elements in different regions, it performs the recognition process very quickly because it makes predictions for elements in the image at once. Also, in the YOLO algorithm, it can achieve good results even with small size datasets. CNN usually achieves good results when trained with large amounts of data. The YOLO algorithm is widely used in the field of object recognition in the literature, and the YOLO algorithm continues to be developed for faster and more precise classification. The first YOLO algorithm is called YOLO V1. The basis of YOLO algorithms is to detect an object falling in the center of each frame by dividing a video frame or image into S x S sized frames. If the algorithm detects any object in a frame, parts of that object are not taken into account in other frames.

To enable object recognition, each square is tasked with predicting B bounding boxes, along with their associated parameters and confidence scores as it can be seen in Fig. 6 below. These confidence scores indicate whether an object is present or not within a particular bounding box.

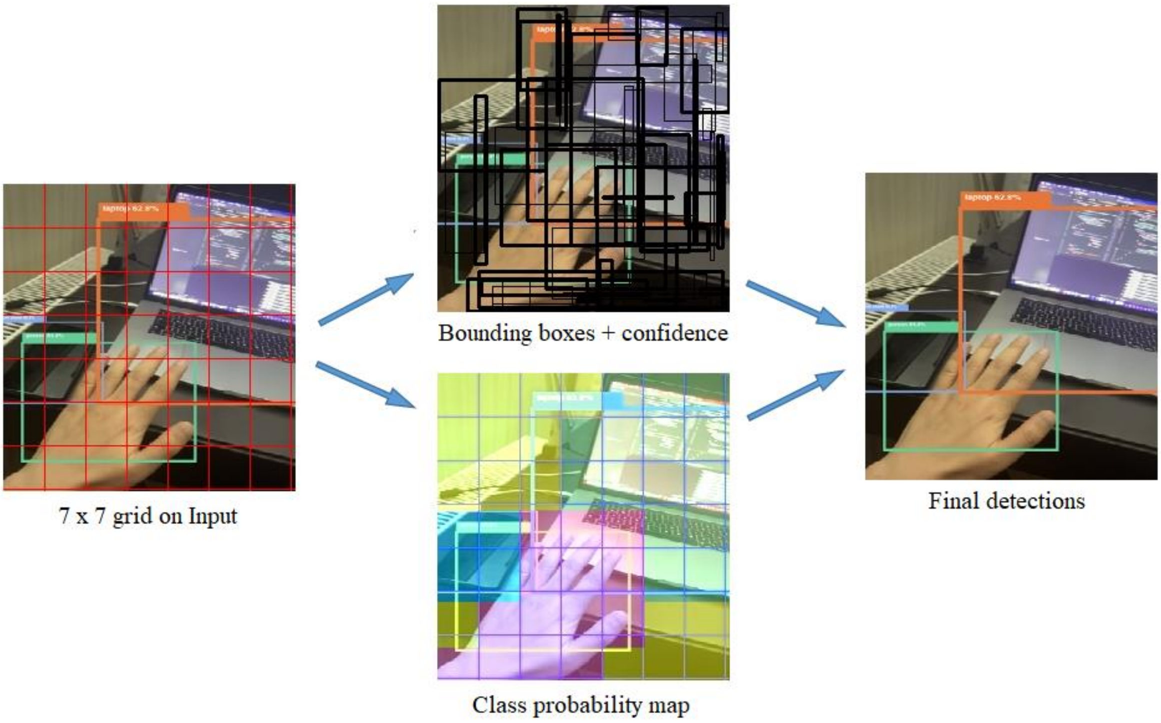

Figure 6. YOLO model with 7x7 grid cell on input image

The confidence score calculation is defined as follows in the equation (2):

$$confidence\ score = p(Object) * IoU_{pred}^{truth} \quad (2)$$

Here, $p(Object)$ represents the likelihood of an object's presence inside each square, while IoU_{pred}^{truth} corresponds to the intersection over union between the predicted bounding box and the actual ground truth box and this ratio is shown schematically in Fig. 7 below. IoU is often used to assess the accuracy of object detection algorithms. It provides a measure of how well the predicted object's location aligns with the actual object. The IoU value ranges between 0 and 1, where 0 indicates no overlap at all, and 1 indicates a perfect match between the predicted and ground truth objects. The range of $p(Object)$ lies between 0 and 1, causing the confidence score to approach 0 when no object is detected in the square. Conversely, the score becomes equal to IoU_{pred}^{truth} when an object is present [26].

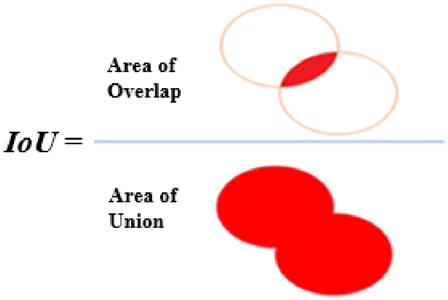

Figure 7. Intersection over union (IoU) between the predicted bounding box and the actual ground truth box

- YOLO V5 includes three key components [27]:
- Backbone: YOLO V5 uses a variety of core network architectures, typically CSPDarknet53 or CSPResNeXt50. These networks are used to learn the characteristics of the input image. The term "CSP" (Cross Stage Partial) refers to the cross-stage connections that the network contains. These links help communicate features better.

- Neck: Various neck structures can be used on the YOLO V5. For example, structures such as PANet (Path Aggregation Network) or PANet-Lite, a lightweight version of PANet, combine feature maps at different scales to detect objects more effectively. Also, various neck structures can include Convolutional, Concatenation, and other operations.

- Head: The head section of the YOLO V5 is responsible for generating object detection results. This section estimates the positions and classes of objects, usually by operating on different feature maps. Various detailing techniques can also be used in the head section.

- There are basic structures and techniques used in the different components of YOLO V5 (Backbone, Neck and Head) [28]:

- CSP (Cross Stage Partial) Blocks: CSP blocks are blocks that are used to better communicate features by dividing the network between different stages. It reduces the loss of properties by containing cross-links.

- SPP (Spatial Pyramid Pooling): SPP is a structure that improves object detection performance by combining feature maps at different scales. This is based on the fact that objects can be of different sizes.

- Concatenation: This process aims to create a richer feature space by combining different feature maps. This can help achieve better object detection results.

- Convolution: The convolution operation allows feature maps to be processed with certain filters. This helps features learn more complex patterns.

- Upsampling: The upsampling process expands feature maps to contain higher resolution information by increasing their size. This can be used to make more detailed location estimates. In Fig. 8 below, the YOLO V5 architecture used in the study is shown.

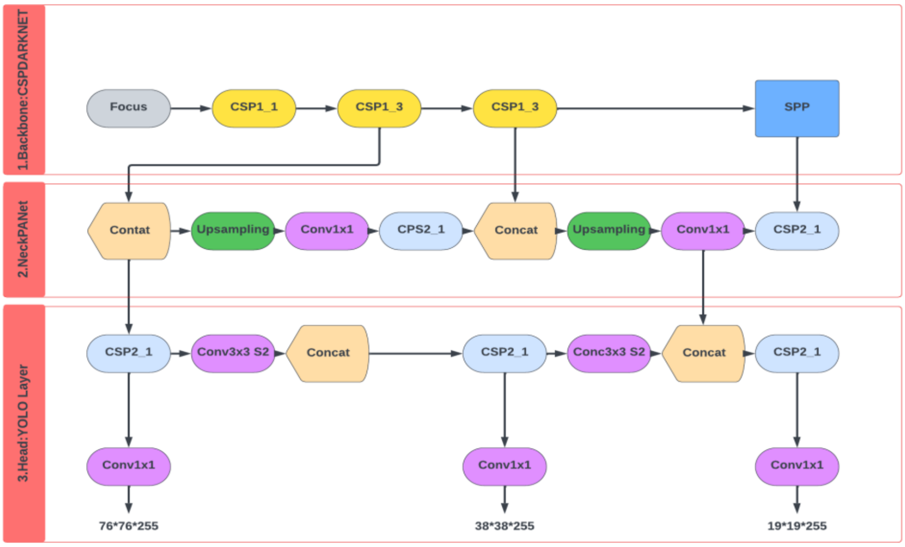

Figure 8. YOLO V5 architecture used in the study

Here, the "Focus" structure in the Backbone section is a kind of convolutional block used to learn the properties of the input image. However, unlike traditional convolution structures, the input is specifically designed to focus the data on a lower-dimensional feature map. The Focus structure is used specifically in the YOLO V5 to help detect low-dimensional objects. It is designed to capture the details of smaller objects by concentrating on feature maps of smaller size, rather than a traditional convolutional structure. The focus structure performs a special convolutional operation to shrink the feature map by reducing the number of input channels, thereby focusing on lower dimensional features. In this way, both the computational intensity is reduced, and low-dimensional objects are detected more effectively. This structural difference differs from previous versions of the YOLO V5 and is specifically designed for faster and more efficient feature extraction. The expression "CSP1" means Cross Stage Partial structure for the first stage i.e. Backbone. "1_1" refers to the CSP version. The term “CSP2” is used for neck. Here, the terms "Conv1x1" and "Conv3x3 S2" refer to the properties of the convolutional layers used. Conv1x1 refers to the convolutional layer with filters of size 1x1. Convolutions with filters of size 1x1 are also called "pointwise convolution". Such convolutions are often used to increase or decrease the number of channels of the feature map. It can also be used to reduce the dependency between features and reduce the computational load. Conv3x3 S2 denotes a convolutional layer with a stride value of 2, with filters of size

3x3. The step value determines how far the filter moves on the feature map. The expression S2 indicates that the filters move over the feature map, skipping 2 pixels. Such convolutions are used to reduce the size of the feature map by half. In the neck section, feature maps are combined or resized at different scales using convolutional layers such as Conv1x1 and Conv3x3 S2. This aims to provide detection of objects of different scales and better feature extraction. The "Head" section in YOLO V5 is where object detection results are generated. The values "76*76*255", "38*38*255" and "19*19*255" indicate the output sizes and channel numbers of Conv1x1 layers. The first two values here indicate the size of the feature map, while the third value indicates the number of channels.

1. Data Acquisition in Object Recognition

LIDAR is a technology that transforms reflected light energy into a set of (x, y, z, r) coordinates, providing a detailed representation of the 3D layout of real-world environments. These coordinates allow us to gather distance information from various points, while 'r' corresponds to reflectance data [29]. Reflectance data indicates the intensity of reflected signals, which is influenced by factors such as surface roughness, color, and material composition.

The application software was developed in XCode 14 environment on Macbook Pro with macOS 13 operating system. For the object recognition process, data from iPhone 13 Pro Max's built-in LIDAR and high-resolution camera was used on Macbook Pro.

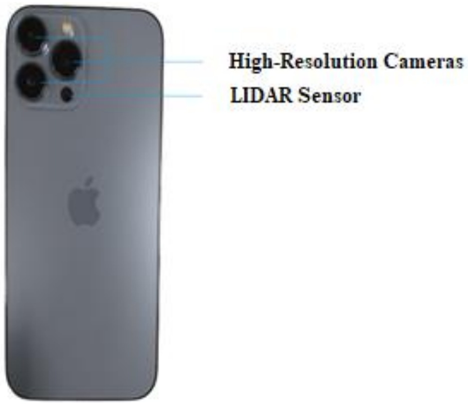

Figure 9. Photo of iPhone 13 Pro Max used to acquire data in object recognition

Data were obtained using the built-in LIDAR and high-resolution cameras of the iPhone 13 Pro Max seen in Fig. 9. With the built-in LIDAR sensor, laser beams sent on the objects enable both the determination of the object distance and the creation of a point cloud to obtain a 3D image. In addition, live video data was obtained simultaneously with the high-resolution internal camera. Sensor fusion was realized by combining LIDAR and camera data. After the live video data received from the camera was split into frames, object recognition was performed on each frame using the YOLO V5 algorithm. The object recognition algorithm classified the objects on the frame depending on the elements in the MS COCO dataset and received the location coordinates information. It is ensured that the objects detected according to the received location information are enclosed in a rectangle with their labels around them.

3.1. Swift Metal

Swift Metal is a low-level framework developed by Apple for programming the graphics processing unit (GPU). There are two types of units for making applications on iPhone devices: The one is the central processing unit (CPU) and the other is the GPU. GPUs that can perform very fast and efficient operations in parallel are used in image processing processes. Swift Metal was developed for GPU programming on iPhone devices, aiming to utilize the full functionality of the hardware and software infrastructure of iPhone devices in the field of GPU programming [30]. This development was prompted by the inadequacy of the OpenGL framework, which was used for graphics processing but became insufficient for powerful graphical applications due to the advancing capabilities of iPhone devices and OpenGL's inability to fully exploit the CPU-GPU distinction in the device's chip architecture. Thus, with Metal, more space is processed in the GPU in a more controlled way in graphical applications, while more space is reserved in the CPU for other features of the application. In addition, Swift Metal has been developed to work seamlessly with other Apple frameworks. Our work was carried out using this framework.

3.2. LIDAR

There have been significant developments in object recognition and image processing in recent years. It is seen that there will be Light Detection and Range (LIDAR) sensors that will become widespread in all areas of the industry with further developments in the future. It is predicted that LIDAR technology will make a significant contribution to humanity in terms of facilitating many works in many fields such as robotic systems, defense technologies, autonomous land vehicles, unmanned aerial vehicles, medical technologies, high-tech agricultural machinery, modeling of archaeological artifacts, reverse engineering applications in the industry.

LIDAR TOF (Time-of-Flight) is a type of LIDAR system that operates based on the principle of measuring the time it takes for a laser pulse to travel to an object and bounce back to the sensor. A LIDAR TOF system emits short pulses of laser light towards an object. The system then measures the time it takes for the laser pulse to travel to the object and return to the sensor. Since the speed of light is constant, the time it takes for the light pulse to make this round trip can be used to calculate the distance between the LIDAR sensor and the object.

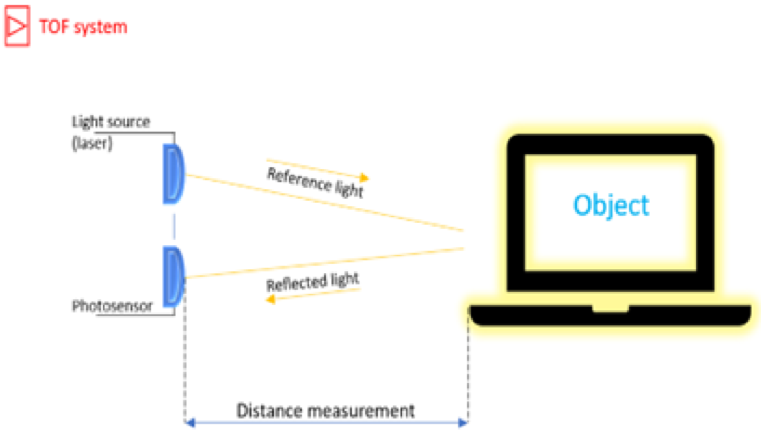

Figure 10. Basic working principle image of LIDAR sensor systems

As seen in Fig. 10, by focusing and sending the laser beam on the object via the LIDAR sensor and calculating the return process, high-accuracy instantaneous distance measurements and 3D object recognition processes can be easily performed with the created point cloud.

$$R = \frac{1}{2}ct \tag{3}$$

In Equation 3, R = object distance, c = speed of light, t = round-trip time of the laser beam sent to the object [31].

3.2.1. LIDAR Point Cloud

A LIDAR system emits laser pulses and measures the time it takes for the pulses to bounce back after hitting objects. By analyzing the time delay and the speed of light, the system can accurately calculate the distance between the LIDAR sensor and the object. A LIDAR point cloud is a collection of data points in three-dimensional space that represents the surfaces and structures of objects within the scanned environment. Each point in the point cloud corresponds to a specific location in space and includes information about its XYZ coordinates (3D position) as well as additional attributes such as intensity (the strength of the laser return) and sometimes color (if the LIDAR sensor is equipped with a color camera or if additional data sources are used). Point clouds generated by LIDAR systems are used in various applications including topographic mapping, autonomous vehicles, environmental monitoring, urban planning, archaeology and cultural heritage, industrial design and manufacturing, forensics and crime scene reconstruction, and virtual reality and simulation. Processing and analyzing LIDAR point clouds can be complex due to the large amount of data involved. Specialized software is used to clean, filter, and segment point clouds to extract meaningful information. As technology advances, LIDAR systems are becoming more compact, affordable, and capable, enabling their application in an ever-expanding range of fields [32].

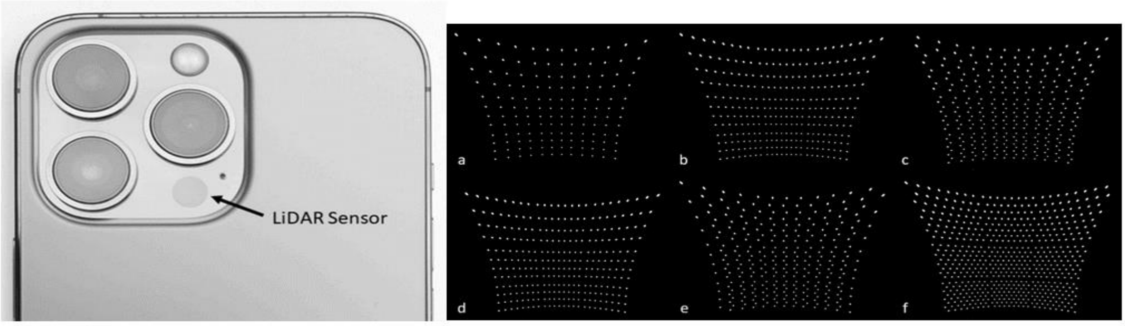

Figure 11. The 12 × 12 point cloud pattern obtained from the built-in LIDAR sensor of the iPhone 13 Pro in different positions is shown in image (a). In images from (b) to (e), combinations of point clouds from different positions before and after are illustrated. In (f), the pattern for all different positions of the point cloud is depicted [33]

Figure 11 shows the 12x12 point cloud patterns created through the iOS application developed for the iPhone 13 Pro with a built-in LIDAR sensor. With the support of the built-in LIDAR sensor added to the iPhone mobile phones released as of 2020, iOS-based software has begun to be developed that will enable quick and easy conclusion of crime scene investigations. Especially in 2022 and later, it is seen from literature research that crime scene investigations are made with iPhone with LIDAR sensor and high-resolution camera support [33]. In addition, it is seen in the literature reviews that many thesis studies on the case of crime scene investigation with the use of mobile phones with LIDAR sensors. An example of the point cloud data we obtained with the iPhone 13 Pro Max in our work is shown in Fig 12.

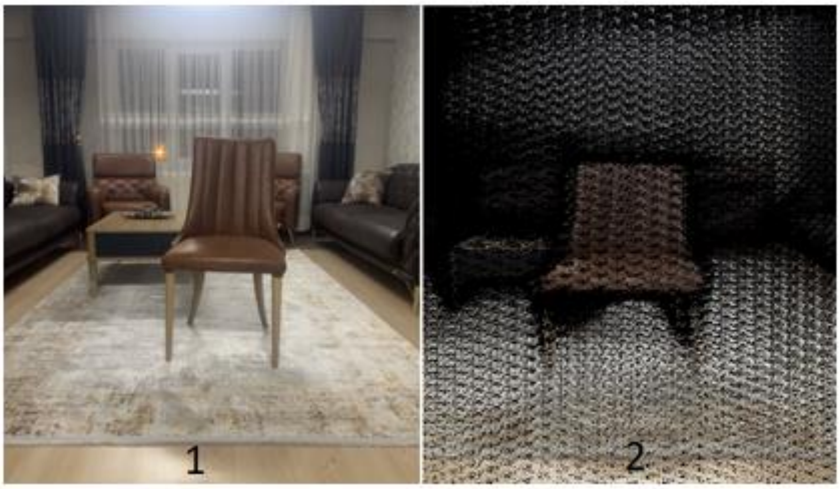

Figure 12. A sample point cloud data taken from the experimental environment in real time with the iPhone built-in LIDAR sensor

A sample frame taken from the indoor experimental environment is given in the image (1) of Fig. 12. With the software we developed, real-time point cloud data was obtained using the built-in LIDAR sensor of the iPhone 13 Pro Max. In the image number (2), the frame of a sample point cloud data obtained from the experimental environment is given.

4. Flow Chart of The Study

The integration process of lidar and camera data to enhance object recognition and visualization is carried out through a systematic sequence of operations:

The procedure initiates with the retrieval of data streams from both the lidar and camera devices, forming the basis for subsequent analysis. The lidar device is configured to output depth data, and the camera's color space is set to YCbCr format for compatibility. Following this setup, instances of data acquisition objects are created, designed to concurrently gather video frames and point cloud data. Synchronization mechanisms ensure temporal alignment between the visual and spatial information sources.Data manipulation involves extracting the PixelBuffer representation of synchronized video data. This intermediary form serves as a foundation for subsequent transformations.The conversion of the PixelBuffer data, encoded in YCbCr, to CIImage format with an ARGB color space is the pivotal next step. This transition is crucial for compatibility with downstream processing. The processing journey proceeds to the object recognition class, which receives the CIImage input. Here, the Yolo V5-based algorithm conducts multi-object recognition using deep learning techniques. The processed CIImage reverts to a UIImage, retaining the object detection results. Bounding boxes are superimposed on the UIImage, enhancing visual interpretability. The UIImage is then transformed back into a PixelBuffer with YCbCr encoding, preserving visual enhancements and color space. The depth data from lidar and camera is combined into a packaged format. Utilizing Swift Metal, this data undergoes processing according to user-defined parameters, including region-based adjustments. The output is then transmitted to the user interface, culminating in an augmented visualization enriched with object recognition outcomes.

In summary, this software-oriented process demonstrates the coherent integration of lidar and camera data through various stages, resulting in an augmented user interface experience primed for enhanced decision-making and engagement. Fig. 13. shows the flow diagram of the experimental study.

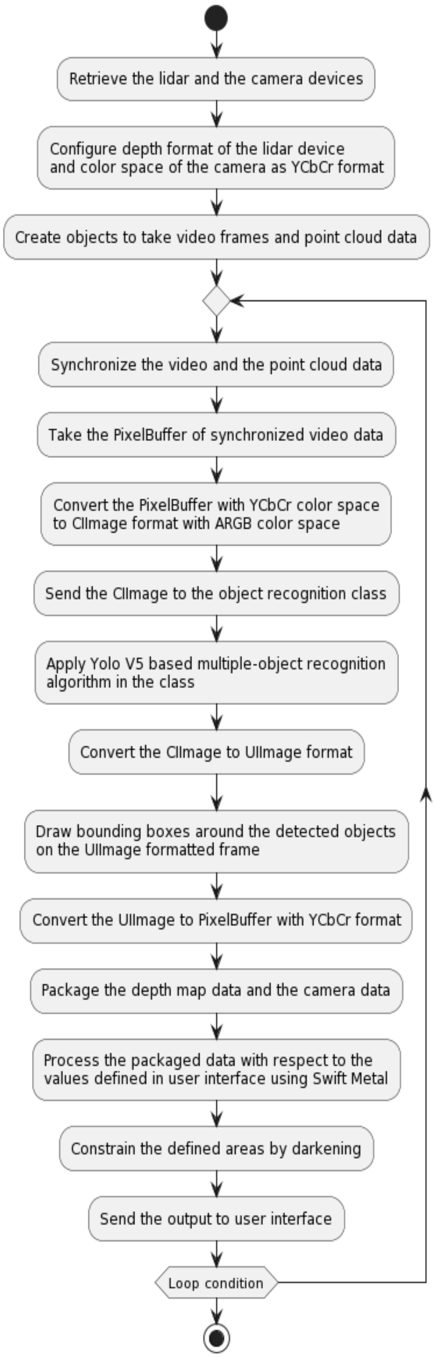
Retrieve the lidar and the camera devices
Configure depth format of the lidar device and color space of the camera as YCbCr format
Create objects to take video frames and point cloud data
Synchronize the video and the point cloud data
Take the PixelBuffer of synchronized video data
Convert the PixelBuffer with YCbCr color space to CIImage format with ARGB color space
Send the CIImage to the object recognition class
Apply Yolo V5 based multiple-object recognition algorithm in the class
Convert the CIImage to UIImage format
Draw bounding boxes around the detected objects on the UIImage formatted frame
Convert the UIImage to PixelBuffer with YCbCr format
Package the depth map data and the camera data
Process the packaged data with respect to the values defined in user interface using Swift Metal
Constrain the defined areas by darkening
Send the output to user interface
Loop condition

Figure 13. Flow chart of the experimental study

4. RESULTS AND DISCUSSION

In this study, real-time object recognition processes are carried out in an area that can be determined by the user, using real-time data from the built-in camera and LIDAR sensor of the iPhone 13 Pro Max mobile phone. With the user interface developed with SwiftUI, the minimum and maximum value ranges of the distances in the 0-15 m limit range can be easily set by the user. Based on the distance and point cloud data received in real time from the built-in LIDAR sensor, video processing was carried out with the Swift Metal framework. According to the determined value ranges, the area in the frame is darkened in real time and limited. The developed application can also extract the contours of objects within the bounded area.

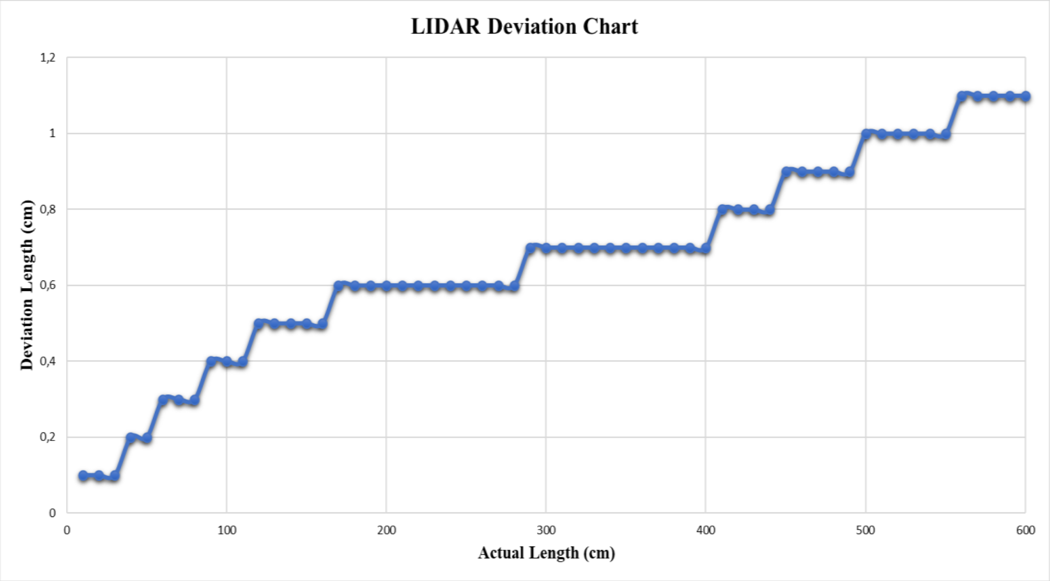

Figure 14. Actual length vs. deviation length graph of distance measurement with LIDAR sensor

In Fig. 14, the actual length - deviation graph obtained by comparing the distance measurement values made with the built-in LIDAR sensor in a 6 meter long room with the actual length values made with the measuring tape is given. As shown in the graph, it is seen that the deviation rate is minimal in the measurements made at a length of 6 meters and the distance is measured accurately with a success rate of over 99%.

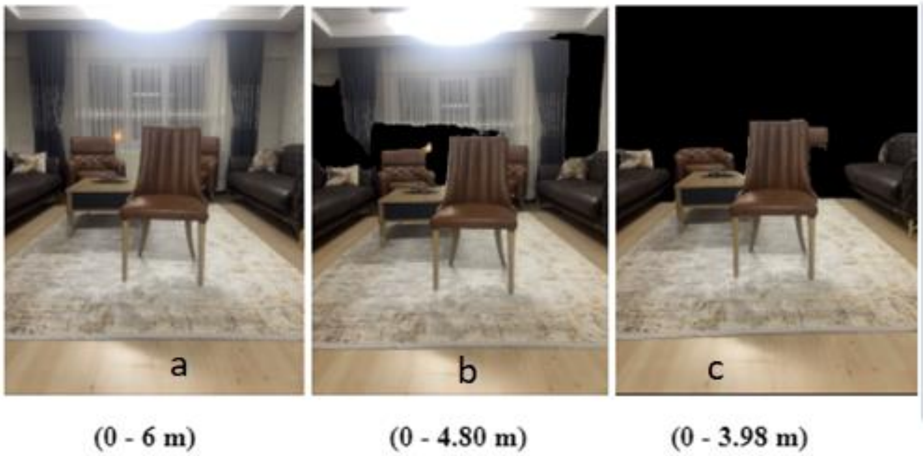

Figure 15. Area limitation at different distances by applying video processing on frames in real time

In Fig. 15, the images taken from the indoor experimental environment of the iPhone 13 Pro Max, which is positioned to be 5 m from the starting point to the farthest point, are presented. Limitation process has been carried out according to the minimum and maximum distance values determined by the user. In (a), 0 m is selected as the minimum value and 6 m is selected as the maximum value. Since the farthest distance from the point where the phone is positioned is 5 m, the developed application showed the entire environment on the phone screen without any limitations within this limit. If the minimum value is 0 and the maximum value is greater than 6 m, the farthest distance of 5 m would also show the entire environment without any limitations, since it is still within these limits. (b) shows the situation where the minimum value is 0 m and the maximum value is 4.80 m by the user. These selected values mean that the areas from 4.80 m ahead of the starting point of the image will be limited. Areas from 4.80 m in the image are limited by darkening. The inwardly recessed part of the window, the curtains and the parts within 4.80 m of the seats are not darkened by the application developed as they are within the specified limit. In (c), the user selected 0 m as the minimum value and 3.98 m as the maximum value. It is seen that the image is limited to the range of 0 - 3.98 m by darkening the sections beyond 3.98 m.

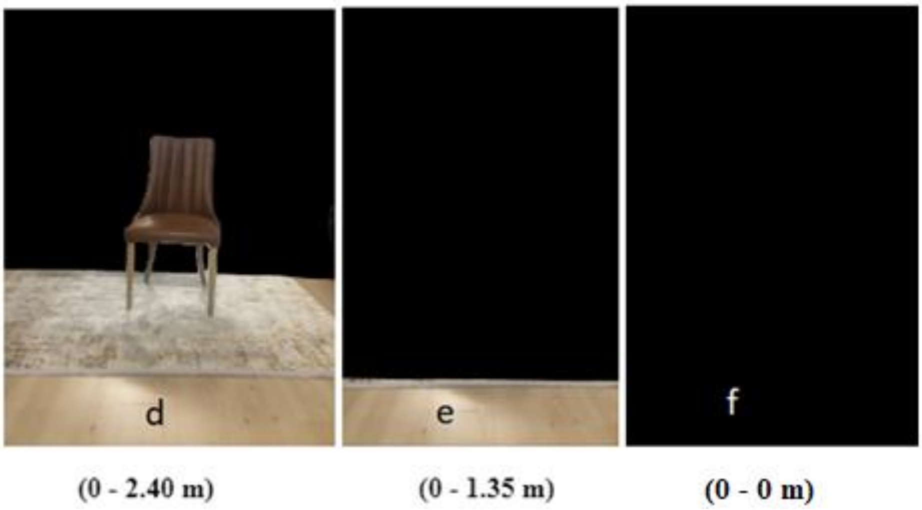

Figure 16. Area limitation at different distances by applying video processing on frames in real time

In Fig. 16, the visuals obtained in the experimental study by keeping the minimum values constant and reducing the maximum values further are shown. Limitation has been made in the range selected as 0 - 2.40 m in (d). Objects 2.40 m ahead appear to have disappeared. In (e), the image is limited to the range of 0 - 1.35 m. It was observed that the chair, which was located more than 1.35 m away, was also outside the border and disappeared. In (f), it is shown that when the ranges are selected as 0 - 0 m, the screen is completely blacked out.

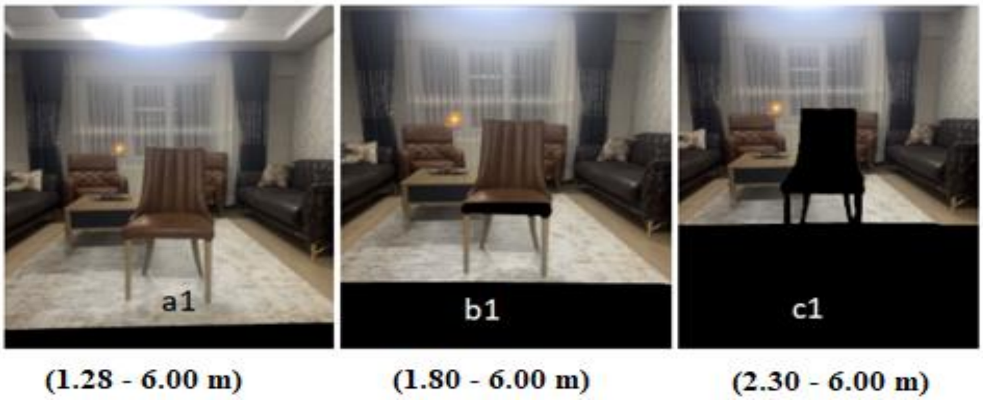

Figure 17. Area limitation at different distances by applying video processing on frames in real time

The Fig. 17 shows the images of the field limitations obtained by keeping the maximum limit value constant at 6 m and increasing the minimum value in the experimental study. Increasing the minimum value means that the area limit will narrow from the starting point where the phone is placed. The limitation area obtained for 1.28 - 6.00 m is shown in (a1). In the range of 1.80 - 6.00 m selected in (b1), it is seen that the blackout filter moves a little further and includes some of the carpet and chair. If (c1)'d is at 2.30 - 6.00 m, it is seen that the entire contour of the chair in the foreground is darkened.

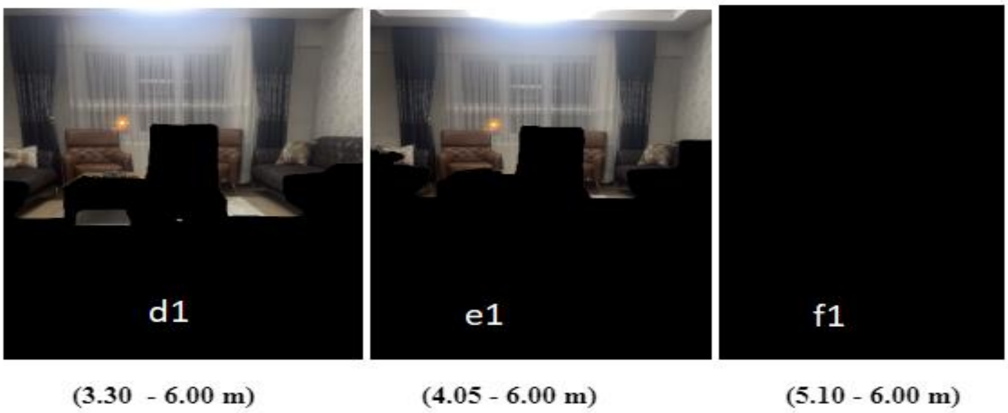

Figure 18. Area limitation at different distances by applying video processing on frames in real time

In Fig. 18, the images obtained in the experimental study by keeping the maximum values constant and increasing the minimum values are shown. It is seen that with the minimum limit increased in (d1) and (e1), the objects gradually start to disappear by staying outside the limit. In (f1), it is seen that the entire screen goes black when 5.10 m exceeds the 5-meter boundary length in the 5.10 - 6.00 m value range.

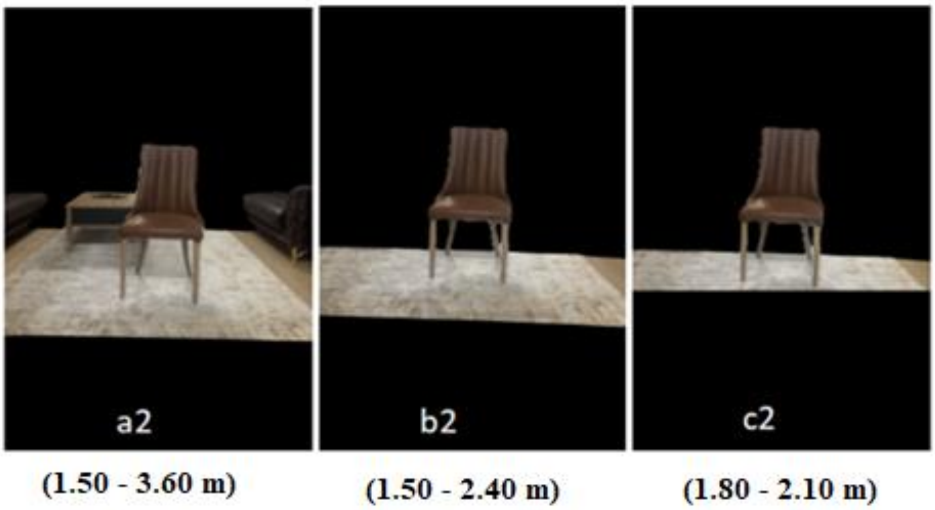

Figure 19. Area limitation at different distances by applying video processing on frames in real time

The Fig. 19 shows the imagess of the experimental results obtained by changing both the minimum and maximum values. In (a2), the output obtained by choosing a minimum area of 1.50 m and a maximum area of 3.60 m is shown. In (b2) and (c2), it is seen that only the chair is visible by narrowing the range of minimum and maximum values.

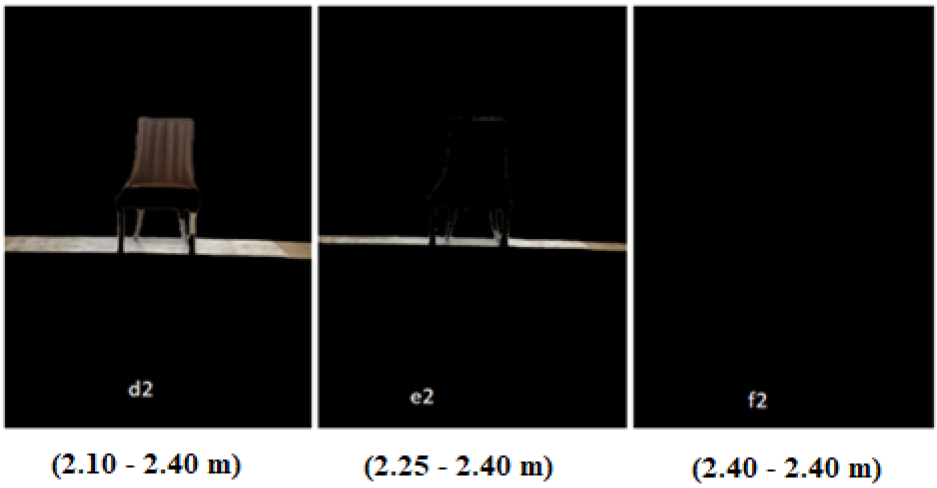

Figure 20. Area limitation at different distances by applying video processing on frames in real time

In Fig. 20, (d2), (e2) and (f2) show the limitation areas obtained by bringing the minimum and maximum values closer to each other. In (d2) and (e2), it is seen that the chair starts to disappear with the narrowing of the value range. In (f2), it is seen that the screen completely darkens at 2.40 - 2.40 m.

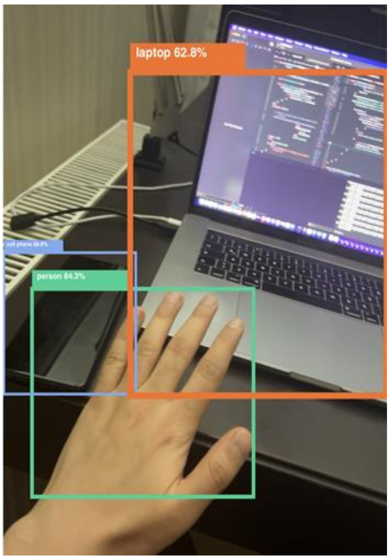

Figure 21. Real-time multiple object recognition output obtained as a result of the work done with the application of YOLO V5 algorithm

In Fig. 21, the output of the real-time multi-object recognition process we performed using the YOLO V5 algorithm on the iPhone 13 Pro Max is shown. In the image of the study performed in real time, it is shown that 3 different objects are detected at the same time and taken into bounding boxes. The information of which class the detected objects included in the bounding box belong to and their accuracy percentages are presented in the top label of the bounding box. The ability of the YOLO algorithm to classify for each object at the same time allows each object in the real-time received frames to be matched with the class with the highest accuracy in the data set, that is, with the highest similarity. In the image, it is seen that laptop, cell phone and person class labels are detected in the bounding box in real time, along with the accuracy percentages.

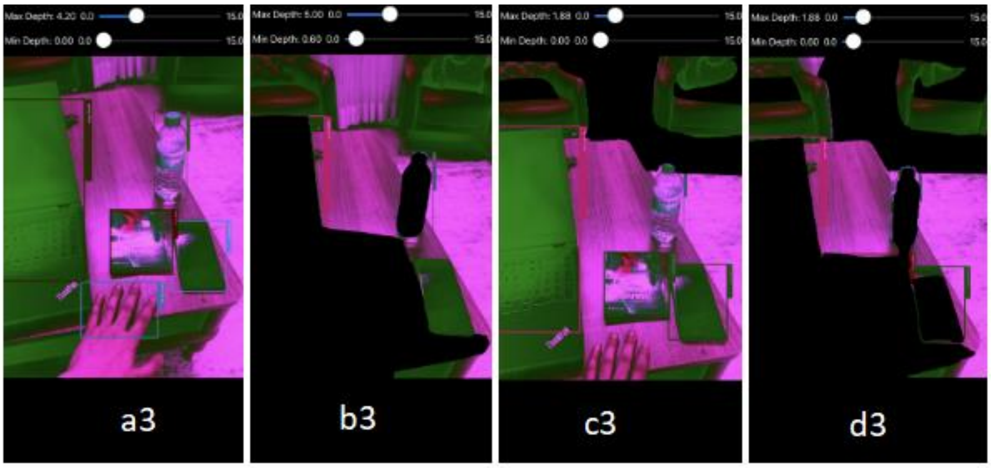

Figure 22. Real-time multiple object recognition application outputs in the area whose boundaries can be adjusted with LIDAR and camera sensor fusion

In Fig. 22, the images of the real-time multi-object recognition application with the fusion of the built-in camera and LIDAR sensor of the iPhone 13 Pro Max performed in the study are shown. The slider added to the user interface appears at the top of the images. The minimum and maximum limit values can be adjusted easily with the slider. The slider can be adjusted in the range of 0 - 15 m. As seen on the slider in (a3), the limit values are set as minimum 0 m and maximum 4.20 m. Since the elements in the frame are within these limits, there is no limitation on the screen, only object recognition is performed. In (b3), the minimum value is set to 0.60 m and the maximum value to 5 m. Increasing the minimum value means that the area limit will narrow from the starting point where the phone is placed. The application limited it to include the contours of the objects and at the same time recognized the objects it could detect. In (c3), the image of the situation where the minimum value is set as 0 m and the maximum value as 1.88 m is given. In (c4), the situation where the minimum value is 0.60 m and the maximum value is 1.88 m is given. Objects that can be detected within these limits appear to be recognized.

5. CONCLUSION

Object recognition is one of the priority research topics in computer science. Studies carried out in the fields of video processing and artificial intelligence have affected many aspects of technology and have been used in various application areas. The importance of this technology is increasing and it points to great potential for future studies. It is widely used in a wide range of areas such as security, medicine, retail, industry, agriculture, automotive and augmented reality. LIDAR sensor is a technology that determines the distance of objects by providing distance measurement and also creates three-dimensional models of objects by creating point cloud as a method used to obtain environmental information. These point cloud data are widely used in object recognition. This data can be used to describe the shapes, sizes and positions of objects, providing benefits in a variety of applications. The LIDAR sensor creates 3D models of objects using point cloud data, increasing its capacity to determine the precise dimensions and positions of objects. In this aspect, it is superior to the camera. The camera, on the other hand, is effective in recognizing objects by using visual features such as color, pattern and texture. In this way, they can capture detailed features of objects and provide higher resolution data. Thanks to the fusion of both sensors, both 3D models and visual properties of objects are analyzed more comprehensively. In this study, it is aimed to perform real-time multi-object recognition in an area whose boundaries can be adjusted using camera and LIDAR data fusion. Using the built-in LIDAR and built-in camera of the iPhone 13 Pro Max mobile phone, multi-object recognition is performed in real-time in the area limited by the minimum-maximum distance ranges that can be adjusted with the slider on the user interface. In the study carried out with the MS COCO dataset, Swift, which is widely used in iOS programming, was used as the programming language, and SwiftUI was used as the framework. Video processing processes were performed with Swift Metal. When the distance measurements made according to the data obtained from the built-in LIDAR sensor from the starting point where the mobile phone is placed in the indoor experimental environment are compared with the measurement results made with the measuring tape, it has been determined that the sensor measures over 99% successfully. By processing the point cloud data received from the LIDAR sensor with Swift Metal, the visual field is limited according to the minimum-maximum distance values determined by the user using the slider in the interface. The limitation process was carried out by darkening the screen in the determined distance range. The application also extracts the contours of objects within the boundary range with the blackout filter. Multiple object recognition was successfully performed with the structure developed by using the YOLO V5 algorithm on objects in the region within the distance ranges limited in the experimental environment of the objects within the boundary range. As a result, multi-object recognition has been successfully accomplished in real-time with the built-in LIDAR sensor by fusion of camera data within an area whose distances can be limited. It is predicted that a sensor fusion study, which can be performed by narrowing the field of view at the desired range with the point cloud and object recognition in this narrowed area, can be used in a wide variety of applications depending on the clear distance information determined by the LIDAR sensor in iPhone mobile phones. For example, since such a mobile application can prevent unnecessary recognition of distant objects, it is predicted that it will make the lives of visually impaired people easier as it will enable them to know more successfully what obstacles and objects are in front of them by detecting objects located near them. Home security can be increased by having the camera monitor a specific area. This approach can be used to monitor and recognize movements in critical areas such as a doorway, garden or hallway. A particular production line or workstation can be monitored. It can be used to spy on the movement of products and production processes. This approach can be used to monitor a particular road section or intersection, analyze traffic flow, and detect congestion. Again, such systems can be used to monitor a particular border area, detect potentially dangerous situations early, and prevent border intrusions. It can be used for monitoring and accurate placement of instruments or supplies during surgical operations. It can be used to detect and track targets in a specific area, detect and track enemy movements in military operations. As a result, the sensor fusion realized in this approach has potential in many areas where applications can increase efficiency, improve safety and achieve better results in a wide range of areas. This technology can provide a wide range of benefits, from defense industry to healthcare, agriculture to security applications. Sensor fusion will continue to play an important role in the future, combining the advantages of different sensors to provide more comprehensive and precise results.

ACKNOWLEDGEMENTS

The authors would like to thank the contributions of Turkey Halkbank Inc. General Directorate for their support to this article study.

AUTHOR CONTRIBUTIONS

The authors contributed equally at every stage of the article.

CONFLICT OF INTEREST

There is no conflict of interest.

ETHIC

There are no ethical problems in publishing this article.

REFERENCES

[1] R. Solovyev, W. Wang and T. Gabruseva "Weighted boxes fusion: Ensembling boxes from different object detection models", Image and Vision Computing, vol. 107, 104117 pp. 1-6, 2021. doi: 10.1016/j.imavis.2021.104117

[2] S. Qi, X. Ning, G. Yang, L. Zhang, P. Long, W. Cai, and W. Li "Review of multi-view 3D object recognition methods based on deep learning" Displays, vol. 69, 102053, pp. 1-12, 2021. doi: 10.1016/j.displa.2021.102053

[3] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, 2023. doi: 10.1109/JPROC.2023.3238524.

[4] M. Nikhitha, S. Roopa Sri and B. Uma Maheswari, "Fruit Recognition and Grade of Disease Detection using Inception V3 Model," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 1040-1043, 2019. doi: 10.1109/ICECA.2019.8822095.

[5] M. Sogabe, N. Ito, T. Miyazaki, T. Kawase, T. Kanno and K. Kawashima "Detection of Instruments Inserted into Eye in Cataract Surgery Using Single-shot Multibox Detector," Sensors & Materials, vol. 34, no. 1, pp. 47–54, 2022. doi: 10.18494/SAM3762

[6] B. Janakiramaiah, G. Kalyani, Karuna, A. et al. "Military object detection in defense using multi-level capsule networks," Soft Comput 27, pp. 1045–1059, 2023. doi: 10.1007/s00500-021-05912-0

[7] M. Rezaei, M. Azarmi and F. M. P. Mir, "3d-net: Monocular 3d object recognition for traffic monitoring," Expert Systems with Applications, vol. 227, 120253, pp.1-17, 2023. doi: 10.1016/j.eswa.2023.120253

[8] S. Kottner, M. J. Thali and D. Gascho, "Using the iPhone's LiDAR technology to capture 3D forensic data at crime and crash scenes," Forensic Imaging, vol. 32, 200535. pp. 1-7, 2023. doi: 10.1016/j.fri.2023.200535

[9] Y. Okochi, H. Rizk, T. Amano and H. Yamaguchi, "Object Recognition from 3D Point Cloud on Resource-Constrained Edge Device," 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 2022, pp. 369-374, 2022. doi: 10.1109/WiMob55322.2022.9941552.

[10] A. Guyot, M. Lennon, T. Lorho and L. Hubert-Moy, "Combined detection and segmentation of archeological structures from LiDAR data using a deep learning approach," Journal of Computer Applications in Archaeology, 4(1), pp.1-19, 2021. doi : 10.5334/jcaa.64

[11] S. Tatsumi, K. Yamaguchi and N. Furuya, "Forest Scanner: A mobile application for measuring and mapping trees with LiDAR-equipped iPhone and iPad," Methods in Ecology and Evolution, 14(7), pp. 1603-1609, 2023. doi: 10.1111/2041-210X.13900

[12] V. Partel, L. Costa and Y. Ampatzidis, "Smart tree crop sprayer utilizing sensor fusion and artificial intelligence," Computers and Electronics in Agriculture, vol. 191, 106556.pp. 1-13, 2021. doi: 10.1016/j.compag.2021.106556

[13] Y. Ji, S. Li, C. Peng, H. Xu, R. Cao and M. Zhang, "Obstacle detection and recognition in farmland based on fusion point cloud data," Computers and Electronics in Agriculture, vol. 189, 106409, pp. 1-6, 2021. doi: 10.1016/j.compag.2021.106409

[14] X. Zuo, P. Geneva, W. Lee, Y. Liu and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, pp. 5848-5854, 2019. doi: 10.1109/IROS40897.2019.8967746

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001. doi: 10.1109/CVPR.2001.990517

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, vol. 1, pp. 886-893, 2005. doi: 10.1109/CVPR.2005.177

[17] P. Felzenszwalb, D. McAllester and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, pp. 1-8, 2008. doi: 10.1109/CVPR.2008.4587597

[18] T. W. Wu, H. Zhang, W. Peng, F. Lü and P. J. He, “Applications of convolutional neural networks for intelligent waste identification and recycling: A review. Resources,” Conservation and Recycling, vol. 190, 106813, pp. 1-16, 2023. doi: 10.1016/j.resconrec.2022.106813

[19] M. M. Taye, “Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions,” Computation, vol.11(3), 52, pp. 1-23, 2023. doi: 10.3390/computation11030052

[20] A. F. Agarap, “Deep learning using rectified linear units (relu),” arXiv preprint arXiv:1803.08375, pp. 1-7, 2018. doi: 10.48550/arXiv.1803.08375

[21] J. Bharadiya, “Convolutional Neural Networks for Image Classification,” International Journal of Innovative Science and Research Technology, 8(5), pp. 673-677, 2023. doi: 10.5281/zenodo.8020781

[22] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016. doi: 10.1109/CVPR.2016.91.

[23] R. Girshick, F. Iandola, T. Darrell and J. Malik, “Deformable part models are convolutional neural networks,” In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 437-446, 2015. doi: 10.48550/arXiv.1409.5403.

[24] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014. doi: 10.1109/CVPR.2014.81.

[25] Y. H. Lee and Y. Kim, “Comparison of CNN and YOLO for Object Detection,” Journal of semiconductor and display technology, vol. 19(1), pp. 85-92, 2020.

[26] P. Jiang, D. Ergu, F. Liu, Y. Cai and B. Ma, “A Review of Yolo algorithm developments,” Procedia Computer Science, vol. 199, pp. 1066-1073, 2022. https://doi.org/10.1016/j.procs.2022.01.135.

[27] 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 Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 2778-2788. doi: 10.1109/ICCVW54120.2021.00312

[28] L. Ting, Z. Baijun, Z. Yongsheng and Y. Shun, “Ship detection algorithm based on improved YOLO V5," In 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), IEEE, Dalian, China, pp. 483-487, 2021. doi: 10.1109/CACRE52464.2021.9501331.

[29] J. Kim, J. Kim and J. Cho, "An advanced object classification strategy using YOLO through camera and LiDAR sensor fusion," 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, QLD, Australia, December 16-18, 2019 pp. 1-5. doi: 10.1109/ICSPCS47537.2019.9008742

[30] J. Clayton, “I: Metal Basics,” Metal Programming Guide: Tutorial and Reference Via Swift, Addison-Wesley, USA 2017.

[31] B. Behroozpour, P. A. Sandborn, M. C. Wu and B. E. Boser, “Lidar system architectures and circuits,” IEEE Communications Magazine vol. 55(10), pp. 135-142, 2017. doi: 10.1109/MCOM.2017.1700030.

[32] Y. Li et al., "Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3412-3432, Aug. 2021. doi: 10.1109/TNNLS.2020.3015992.

[33] S. Kottner, M. J. Thali and D. Gascho, D. “Using the iPhone's LiDAR technology to capture 3D forensic data at crime and crash scenes,” Forensic Imaging, vol. 32, 200535, pp. 1-7, 2023. doi.org/10.1016/j.fri.2023.200535.