Comparison of deep learning models in terms of multiple object detection on satellite images

Ferdi Doğan* and IbrahimTurkoğlu**

** Head of IT Department, AdiyamanUniversity, Adiyaman, Turkey.*
*** Faculty of Technology, Software Engineering Department, Fırat University, Elazıg, Turkey.*
** Corresponding Author: fdogan@adiyaman.edu.tr*

Submitted ***:*** 08/01/2021
Revised ***:*** 26/04/2021
Accepted ***:*** 22/05/2021

ABSTRACT

The images obtained by remote sensing contain important data about ground surface. It is an important issue to detect objects on the ground surface with these images. Deep learning models are known togive better results in studies on object detection. However, the superiority of the deep learning models over each other is unknown. For this reason, it should be clarified which model is superior in terms of object detection and which model should be used in studies. In this study, it was aimed to reveal the superiorities of deep learning models by comparing their performance in detecting multiple objects. By using 11 deep learning models that are frequently encountered in the literature, the application of detecting objects of 14 classes in the DOTA dataset was made. 49,053 objects in 888 images were used for training by using AlexNet, Vgg16, Vgg19, GoogleNet, SequeezeNet, Resnet18, Resnet50, Resnet101, Inceptionresnetv2, inceptionv3, and DenseNet201 models. After the training, 13,772 objects consisting of 14 classes in 277 images were used for testing with RCNN, which is one of the object detection methods. The performance of each algorithm in the 14 classes has been demonstrated by using Average Precision (AP) and Mean Average Precision (mAP) to measure the performance of the models from their metrics. In a particular class of each deep learning model, difference in performance was observed. The model with the highest performance varies in each class. In the application, the most successful average mAP value of 14 classes was Vgg16 with 24.64, while the lowest was InceptionResnetV2 with 11.78. In this article, the success of deep learning models in detecting multiple objects has been demonstrated practically, and it is thought to be an important resource for researchers who will study on this subject.

Keywords: Deep learning; Convolutional neural networks; Remote sensing images; Deep learning algorithms; Object detection.

INTRODUCTION

The images obtained by remote sensing contain many important information such as natural resources, landforms, city and regional planning, and natural disasters. Especially, visualizing large areas of the ground surface allows access to a lot of information. Remote sensing images are obtained through drones, unmanned aerial vehicles, and satellites. Studies such as object detection, image improvement, image analysis, ground surface change, temporal differences, soil analysis, and so on are carried out on these images (Ghazali et al., 2020, and Mu et al., 2019). These images are used in fields such as agriculture, geographical information systems, defense industry, urban transformation, determination of natural and cultural resources, meteorology, and natural disaster analysis (Kadhim et al., Rabbi et al., 2020, and Said et al., 2019). Technological developments are improving the quality of the images taken from the ground surface day by day. This situation allows more detailed studies (Dogan&Turkoglu, 2019, and Liu et al., 2017). Analysis of these images and object detection is a popular topic today. Searching for a particular object on images requires a very difficult process. It is difficult to detect the object with the human eye in a low resolution image. Because the area covered by the object in the image is very small. The situation is the same in the digital environment. Since the number of pixels indicating the object in the digital image is quite low, it is almost impossible to reveal the features that will define this object (Huo et al., 2016). Remote sensing images are very high resolution because they are images covering large areas. A computer with high processing capacity is required for processing these high resolution images in digital environment. Figure 1 shows examples of high resolution images obtained by remote sensing.

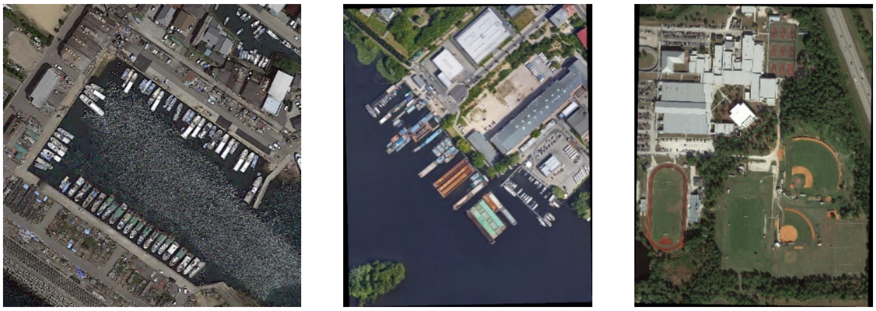


Figure 1. High definition satellite ımages.

Although manydifferent techniques are used in computer vision systems, it is seen thatmachine learning is superior to other techniques in terms of succes srates. The most successful artificial intelligence technique in machine learning systems is deep learning. It demonstrates superior performance in studies on deep learning, classification, and object detection. For this reason, studies focusing on deep learning are more common in recent years (Pereira et al., 2020). Inspired by the first convolutional neural network LeNet (LeCun et al., 1998), the first deep learning architecture AlexNet emerged (Krizhevsky et al., 2017).Object attributes are important in object detection studies with traditional machine learning. Corner numbers, circular lines, and sizes are important attributes. It is very difficult to define the attributes of objects belonging to different classes. There are many methods and algorithms for feature extraction in studies. In deep learning models, determining the properties of the object is quite different. Convolution process in the model is done automatically to determine the properties of the object(Lin et al., 2017c). Figure 2 shows the flowchart of deep learning and traditional object detection methods.

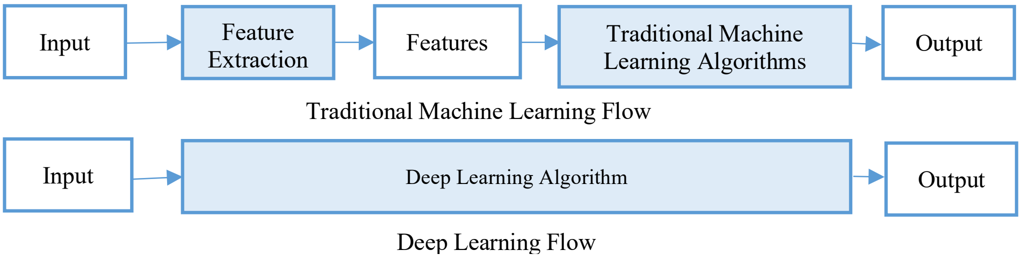


Figure 2. Traditional machine learning flow and deep learning flow.

In traditional machine learning, there are stages such as feature extraction, feature detection, and machine learning algorithms. Different methods developed for each stage and included in the literature are used. Processes that reveal the features and represent them in the best way should be selected. Afterwards, the system is set up using a machine learning algorithm suitable for the system (Ping Tian, 2013). Deep learning models, on the other hand, have few stages and act depending on the algorithm. It automatically defines the stages in machine learning methods. Feature extraction takes place automatically within the architecture (Guo et al., 2016). It has a structure that learns from data. In deep learning models, the increase in the amount of data, that is, the number of samples, is a factor that facilitates learning. For this reason, as the sample number of the model increases, the features become clearer, and the success rate increases.

There have been studies on object detection using deep learning models, but no study was found to compare different deep learning models. This article compares the applied performance of deep learning models formulti-object detection. The highlights of the article are as follows.

Applied object detection was performed byusing 11 different deep learning models, which are the most used in the literature. Objects belonging to 14 different classes were detected in high resolution satellite images. The performance rates of 11 deep learning models in 14 different classes were determined and compared. Since there is no study showing the achievements of 11 deep learning models in detecting multiple objects in the literature, it is thought to be a resource for researchers. Average precision and mean average precision methods are used for the measurement of object detection performance.

RELATED WORK

Deep learning models with high classification achievements are used in the studies on object detection. The deep learning model and the object detection method are two different stages. While the deep learning model is used for training the network, the object detection method tries to find the most suitable coordinates of the object by moving according to the weights in the trained network. In many studies, detection methods have been developed to find the location of the object. In the study conducted to develop a new object detection method using the Dota data set, the Vgg16 model was used, and the result of mAP 72.43 was produced (Wang et al., 2019). In another study, it was aimed to develop an object detection method, and the performance was compared with other methods (Ying et al., 2019).

In an object detection method called HyperNet, VGG16 deep learning model was used as a back bone, and a performance above 60% mAP was obtained in different data sets (Kong et al., 2016). In another study, he examined deep learning techniques, used data sets and evaluation criteria, object detection, specific object detection, and category-specific object detection techniques. Comparative results of object detection methods using Vgg16, Resnet50, and ResNet101 models are presented (Han et al., 2018). Object detection methods like R-CNN, SPP-Net, Fast-R-CNN, Faster R-CNN, R-FCN, RPN, Mask R-CNN, Multitask Learning, Multiscale Representation, Contextual Modeling, and Thinking in Deep Learning-Based Object Detection were researched. The success of these methods in object detection has been examined. The mAP (58.5-81.6) values obtained by the methods in a 20-class data set are given. AlexNet, Vgg16, and ZF-Net deep learning models were used to train the network (Zhao et al., 2019).

In the studies on object detection on high resolution satellite images, it was aimed to detect the objects by using the deep learning model for the training of the network. In a study using VggNet, an object detection application belonging to 3 object classes has been made. Palm tree, vehicle and road were determined in the study. Images from Worldview-3 to 30 cm and Deimos-2 to 75 cm were used (Napiorkowska et al., 2018). In a study for aircraft detection, object detection methods YOLO v3, Faster R-CNN, and SSD were used (Alganci et al., 2020).

The performances of object detection methods were compared using deep learning models such as Vgg16, ResNet-101, and Inception-Resnet-v2 (Jiao et al., 2019). In a study conducted for vehicle detection, Vgg-16 and Inception-V2 were used, and the success rates were found to be over 80% (Mansour et al., 2019). There is no sample comparing the performance of deep learning models according to the application results in the studies conducted. Overall, the performance of object detectors was compared and one or more deep learning models were used.

Successful deep learning models are emerging for computer vision classification with competitions such as Microsoft COCO, ILSVRC, and PASCAL VOC (Pathak et al., 2018). The deep learning models used in these competitions show the results obtained for object classification. Models with classification performance are backsubscribed for object detection. It has been observed that there is no study to reveal the success of the selected models in detecting multiple objects in the literature. Table 1 shows the table of some methods and used deep learning models.

Table 1. Studies done with Dota data set.

Classes	Backbone (deep learning model)	Object detection methods	Performance criteria	Success rate
15	ResNet-101	R-FCN	AP	52.58 (Xia et al., 2018)
15	InceptionV2	SSD	AP	29.86 (Xia et al., 2018)
15	GoogleNet	YOLOv2	AP	39.2 (Xia et al., 2018)
15	Resnet-101	SSD	mAP	24(Liu et al., 2021)
15	GoogleNet	YOLOv2	mAP	39.2(Wang et al., 2019)
15	Inception V2	SSD	mAP	10.94 (Ying et al., 2019)
15	ResNet101	Faster R-CNN	mAP	60.46 (Ying et al., 2019)
15	ResNet101	R-FCN	mAP	47.24 (Sun et al., 2018)
15	Vgg16	YOLOv3	mAP	60 (Sun et al., 2018)
15	Vgg16	RetinaNet	mAP	50.39 (Sun et al., 2018)
15	Vgg16	SBL	mAP	64.77 (Sun et al., 2018)

Some studies have been done on object detection from high resolution satellite images with deep learning. In these studies, it is aimed to identify objects belonging to one or several classes using a specific deep learning model. In a study usingVggNet, a detection application belonging to 3 object classes has beenmade. In this study, palmtree, vehicle, and road were determined. Images from Worldview-3 to 30 cm and from Deimos-2 to 75 cm were used (Napiorkowska et al., 2018). In another study, using the Dota data set and object detection methods YOLO v3, Faster R-CNN, and SSD, aircraft detection was aimed to compare these methods (Alganci et al., 2020). In a study conducted to compare object detection detectors, object detectors were tested using deep learning models such as Vgg-16, ResNet-101, Inception-Resnet-v2, RetinaNet, and DarkNet. The performance of the detectors has been compared (Jiao et al., 2019). In a 5-class object detection study using images with 30cm and 4.8m width, it was observed that this AP value was 0.53 in images with a width of 30cm, while this performance decreased to

0.11 in images with 4.8m (Shermeyer & Van Etten, 2019). In a study for vehicle detection, Vgg-16 and Inception-V2 were used and the success rates were found to be over 80% (Mansour et al., 2019). There is no sample comparing the performance of deep learning models according to the results of the application in the studies. Generally, the performance of object detectors was compared, and one or more deep learning models were used for this.

MATERIAL AND METHOD

Material

This study was carried out with matlab 2019b version. The coordinates of the objects in the images are taken from the dota dataset. The coordinate data of the labels were rearranged by making them nonangular rectangles. Models were trained and tested by using the Matlab Deep Learning toolbox.

CNN and deep learning models

Covolutional neural networks form the basis of deep learning models. CNN models are used in the studies such as object detection and classification. Deep learning models are deep-multilayered versions of CNNs. The layers used in CNN and deep learning models are similar. Deep learning is machine learning and it is a high-layer artificial neural network architecture. Studies on layers are designed to increase the performance of the model. Hyperparemeters included in the deep learning model differ in deep learning models. The hyperparameters selected in the model also allow the network to be shaped. The dilution (dropout) in the layers within the model, activation function (relu) to be used, normalization, pooling, number of layers, number of neurons, convolution process, input data, number of inputs, number of repetitions of network are designed differently in each algorithm. These hyperparameters play a role in determining the performance of the model, the speed of training and testing periods, the number of neurons, the number of units in which the trained network is stored depending on the depth of the network and complexity. For this reason, layers providing determination of hyperparameters are very important (Young et al., 2015). The layers that are frequently used in the deep learning model are shown in figure 3.

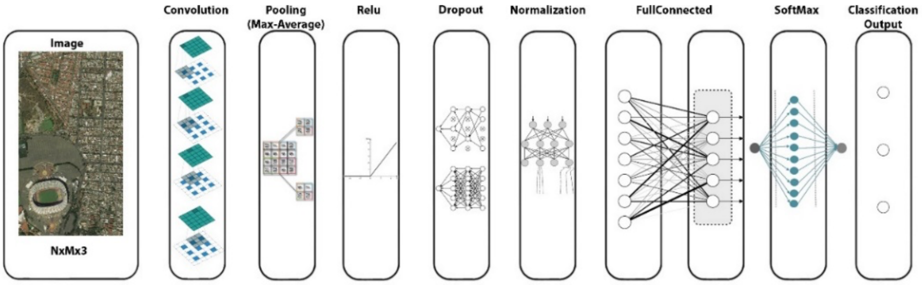


Figure 3. Some layers in deep learning algorithms.

The dimensions of the samples entering the model are determined at the input layer. Increasing the input data can be evaluated as a network performance hyperparameter. For this reason, the diversity of samples and having lots of the samples that enter the network are valuable for the training of the network to be better (Tishby&Zaslavsky, 2015). The basic layer in determining the attribute is the convolution layer. Here, a kind of filtering process is performed and the edges of the shapes in the image are extracted. The odd-number of filters used are preferred to be 3x3, 5x5, 7x7 (Yi et al., 2014). There are too many hidden layers in deep learning algorithms. This increases the number of parameters and calculation time. Pooling layer reduces the number of parameters in the network, reducing the computation time. In the most used maximum pooling process, 2x2 matrix is used and the number of parameters is reduced by half (Lee et al., 2009). One of the most important layers used in the network is the layer containing the activation function. A fast activation function is preferred due to the fastness of the network calculation and the continuity of the backpropagation algorithm within the deep layers. Although there are different types of activation (Gu et al., 2018,Xu et al., 2015), there should be a rapid response activation during network training (Yang et al., 2015). Dropout Layer is a layer used to reduce the number of neurons in the network. It performs the task of dropping out neurons in the network randomly or according to a threshold value (Xiao et al., 2016).

AlexNet, which has the least layers among the models used for multiple object detection, consists of 25 layers. Alex Krizhevsky and colleagues created this model inspired by LeNet (LeCun et al., 1989). It consists of about 60 million parameters. It works fast due to the low number of layers. It consists of 5 convolution layers and 8 hidden layers. Classification error rate is 15.4%. The Vgg16 and Vgg19 models, which are similar to the AlexNet model, have a structure consisting of 41 and 47 layers. Vgg16, developed by Oxford University Visual Geometry Group, has 16 convolution layers, and Vgg19 has 19 convolution layers. There are 138 million parameters in the network (Kong et al., 2016). A model consisting of 144 layersand 6.79 million parameters was created in the study conducted under the name of GoogleNet in 2014. In this model, a module called inception is used (Han et al., 2018). In the module shown in figure 4, it was aimed to reduce the number of parameters in the network by using a 1x1 convolution layer.

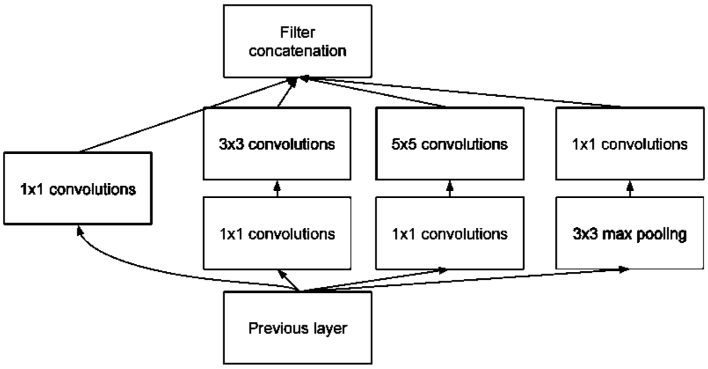


Figure 4. Inception module.

Inception module created with GoogleNet is also used in new models created. It is a highly successful algorithm with 6.6% error rate. Another application that brings a different perspective to deep learning models is the ResNet model. Residual blocks are created in this model. The blocks in figure 5 are located along the architecture. There are residual block structures used in ResNet18, ResNet50, andResNet101 models (Han et al., 2018).

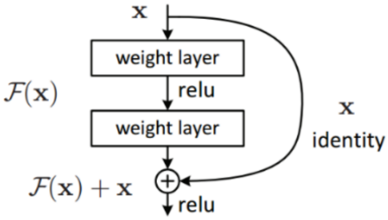

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 5. Residual Blok in ResNet.

ResNet18 model has 18 convolution layers out of 77 layers, ResNet50 has 177 layers and 50 convolution layers, and ResNet101 has 347 layers and 101 convolution layers. ResNet also has models consisting of different layers. In the model called SqueezeNet, a structure called fire module is used. 1x1 and 3x3 convolution layers are used in this module in figure 6. The SequeezeNet model, which consists of 68 layers, consists of 8 fire modules. It is fast although its performance rate is low. The storage space of the model is quite small (Zhao et al., 2019).

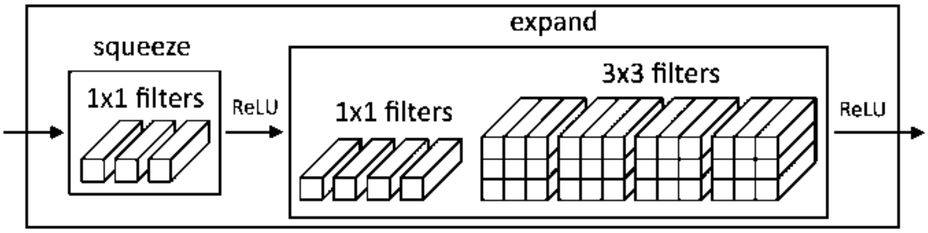


Figure 6. Fire module in SqueezeNet architecture.

The other model used in this study is Inception ResNetv2. It is a very deep model with 825 layers. It works with 55.9 million parameters and over 95% performance. Inception and residual blocks are used together in this model. Network has been created within the network. Although the training and testing process is long, it takes up more space as a storage area compared to other models due to the high number of layers (Szegedy et al., 2017). The model called Inceptionv3 is a network consisting of 316 layers, in which inception modules are used intensely (Xia et al., 2018). A different point of view has been introduced in the model created with dense-sparse-dense logic called DenseNet 201. Although it consists of 709 layers, the number of parameters is 23.8 million. Dense-sparse-dense logic is used to reduce the number of parameters. There are dense blocks similar to the ResNet model (Liu et al., 2021). These blocks are shown in figure 7.

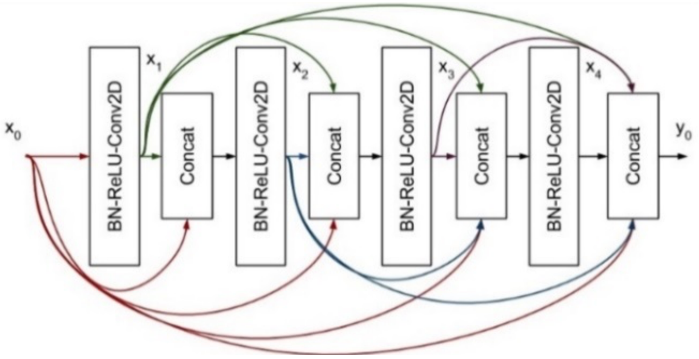


Figure 7. Densenetblocks.

The information about the classification success of the deep learning models used in this study is given in table 2. This information includes the number of layers, the number of connections in the network, the number of convolution layers, the number of parameters in the network, and the best and worst error rates.

Table 2. Classification achievements in deep learning architectures

Architecture	Layers	Connections	ConvolutionLayers	Parameters	Top-1 Error	Top-5 Error
AlexNet (Krizhevsky et al., 2017)	25	-	8	62m	36.7	15.4
Vgg16 (Kong et al., 2016)	41	-	16	138m	25.6	8.1
Vgg19 (Kong et al., 2016)	47	-	19	144m	25.5	8
GoogleNet (Han et al., 2018)	144	170	22	5m	-	6.67
Resnet18 (He et al., 2016)	72	79	18	11.7m	30.43	10.76
ResNet50 (He et al., 2016)	177	192	50	25.6m	22.8	6.71
Resnet101 (He et al., 2016)	347	379	101	40m	21.75	6.05
SqueezeNet (Zhao et al., 2019)	68	75	18	1.2m	41.90	19.58
InceptionResnetv2 (Szegedy et al., 2017)	825	922	-	55.9m	19.9	4.9
Inceptionv3 (Xia et al., 2018)	316	350	-	23.8m	21.2	5.6
DenseNet201 (Liu et al., 2021)	709	806	-	20	21.46	5.54

Algorithms For Object Detection

Object detection, which is an important application area, locates the target object and its location in images or videos (Ying et al.). Traditional method studies for object detection are performed by matching simple templates. In such methods, object attributes are created on the basis of the target object, then sliding on the image and searching according to the result of matching object property vectors. Since the objects can be of different sizes in the image, they were scanned using different window sizes. Apart from traditional methods, there are also deep learning based object detection methods. These are RCNN (Regions with CNN) (Wang et al., 2019), Fast RCNN (Girshick, 2015), Faster RCNN (Sun et al., 2018), YOLO (You Only Look Once) (Redmon et al., 2016), SPP (Spatial Pyramid Pooling) (He et al., 2015), Feature Pryramid Networks (Lin et al., 2017a), RetinaNet (Lin et al., 2017b), and SSD (Single Shot MultiBox Detector) (Liu et al., 2016).

RCNN is one of the first studies on deep learning based object detection. Training of the network was carried out using VggNet and ResNet algorithms, and the classifier layer number of classes + background was determined during the training of the network. The similarity ratio is determined in the range of 0-1. This value represents the similarity ratio of the object to be searched on the image. In the RCNN test process, regardless of the number of classes, it offers 2000 different regions. As shown in figure 8, 4 closest regions are recommended for each class from these 2000 different regions.

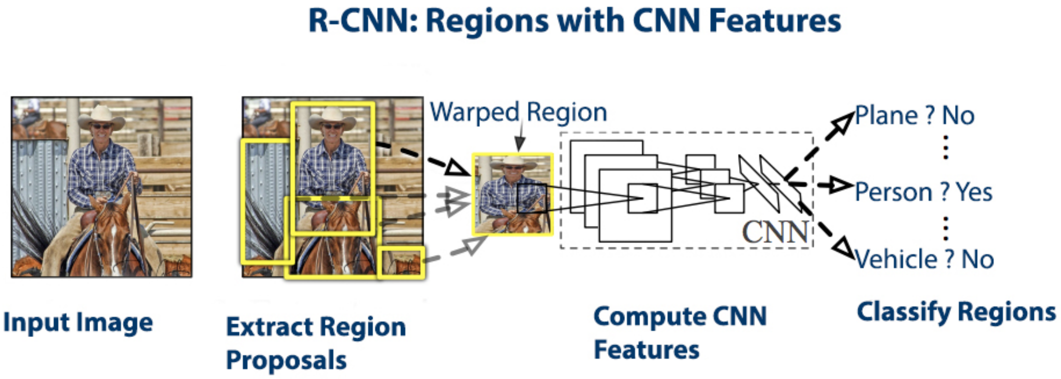


Figure 8. RCNN region inference scheme

The region with the highest similarity among the specified regions represents the object. The training time of the RCNN object detection method is shorter than the Fast RCNN and Faster RCNN methods. Fast RCNN and Faster RCNN studies have been carried out in order to eliminate the problems that occur with RCNN and to reduce the test time. RCNN test time takes longer than Fast RCNN and Faster RCNN. These methods are not preferable for every device due to their large area. With YOLO, it tries to find a single bounding box with a network that specifies class probabilities. It is a more suitable method to be used in real time systems due to its plainness and simplicity. SSD and YOLO are similar. However, filters of different sizes are applied in the model. It is an object detection detector with a high rate of performance. The performance rates of these detectors are included in table 3.

Table 3. Success rates of object detection algorithms.

Algorithm	Dataset Used	Result
YOLO	2012 PASCAL VOC	57,9
R-CNN	2012 PASCAL VOC	62.4
Fast R-CNN	2012 PASCAL VOC	68.4
Faster R-CNN	2012 PASCAL VOC	75.9
SSD	2012 PASCAL VOC	82.2

Object Detection Detector

Ground truth data is used to test the performance of deep learning models in object detection. Ground truth consists of target point data of objects in the image .The similarity between the ground truth value created in the test data and the results of the model estimation is found by the Jaccard index (Vorontsov et al., 2013). For this, the Intersection Over Union account is checked. Intersection is the intersection of the location of the object and the resulting position found by the model, and Union is the combination of these two positions. The rectangular position information of the object in the ground truth data is compared with the rectangular position information detected by the algorithm. Figure 9 shows the intersection and combination results of the ground truth and object location found by the model.

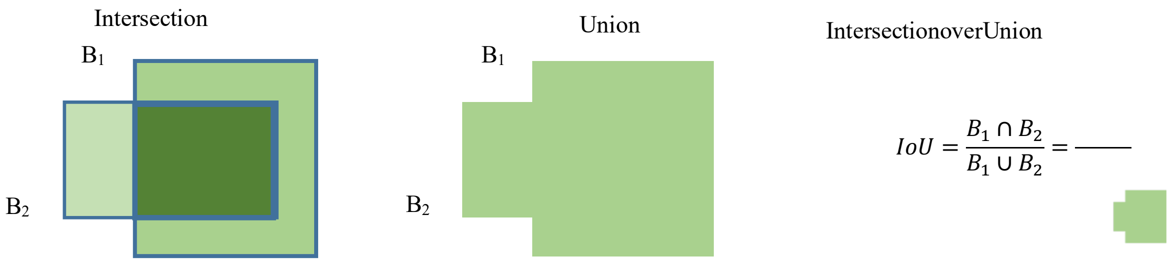


Figure 9. Image showing ground truth and the results detected by the model.

B_1 refers to the target and B_2 represents the forecast. Some examples detected by the Ground Truth and Deep learning model are shown in figure 10.

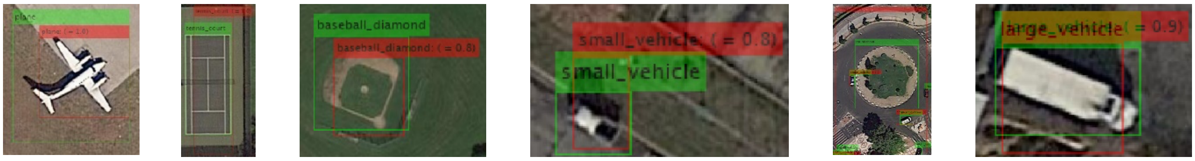


Figure 10. Examples of Ground Truth and Deep Learning Model Results.

In order to express the performance of deep learning models numerically, recall and precision and average precision values are used (Padilla et al., 2020). Here, True, False and Positive, Negative values are given depending on the matching status of the target position and the model position. This metric, known as a confusion matrix, is shown as follows.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

TP: The number of objects that the deep learning model has found and indicating that the object is in reality,
FP: The number of objects that the deep learning model found but indicating that the object is not in reality,
FN: The number of objects that the deep learning model has not found but the object is in reality,
TN: The number of objects that the deep learning model has not found and indicating that it is not in reality.

Precision is the rate of finding all objects accurately. Recall determines how many objects predicted as correct are detected. Looking at these values, Recall and Precision values are calculated using the following equation (Szegedy et al.).

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP}$$

Precision and recall take values between 0 and 1. One of the object detection evaluation methods is the average precision calculated according to precision and recall values. The most commonly used method in the literature is mean average precision (mAP). This value, which reveals the average sensitivity, defines the average performance rate of each object. 11-point interpolated method is used to calculate the mAP value. Precision and recall values are used for calculating mAP. In figure 11, an example of average precision is shown. For the recall value that takes a value between 0 and 1 (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1) is obtained by the sum of the precision values corresponding to 11 values and then divided by 11. The highest precision value in consecutive intervals of 11 values is obtained (Henderson & Ferrari, 2016). Here, if the highest precision value that corresponds to the range is lower than the highest precision value in the next range, the next precision value replaces the precision value of the previous range. Recall and precision values are inversely proportional to each other (Srinivas, 2020).

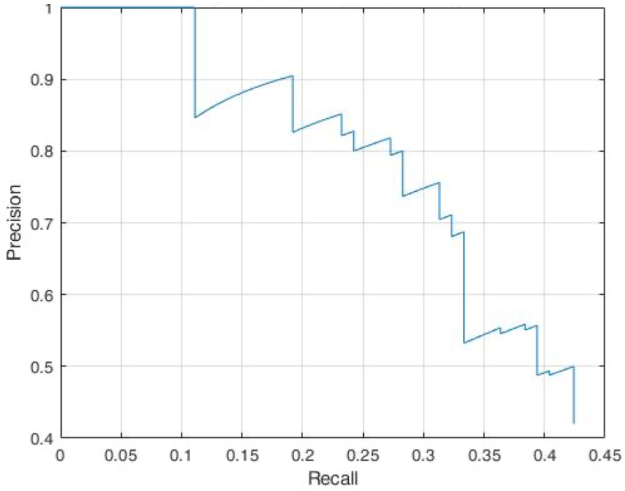


Figure 11. An example of average precision chart.

Dataset

The Dota (Xia et al., 2018) dataset was used for this study. In the dataset, 14 classes of objects were determined. 888 images in different sizes were used for training. Training images are high resolution images in the size range of 432x559 to 5193x6054. There are 49053 objects in 14 classes in 888 images. 277 images were used for the test. The sizes of these images range from 448x511 to 6313x6400. The images include black-white visuals taken at night and colored visuals. The coordinates of the objects in the Dota dataset are rearranged in rectangular dimensions. The different sizes of the high resolution samples in the dataset is seen as an important criterion to reveal the average performance of deep learning models. Therefore, it wanted to show what kind of situation it will exhibit in different dimensions. The images used for the test contain a total of 13772 objects. Table 4 shows the class used for training and testing and the number of objects in each classroom.

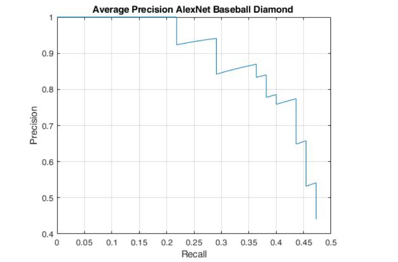
Table 4. Numerical data of objects used.

	Baseball Diamond	**Basketball Court**	**Bridge**	**GroundTrackField**	**Harbor**	**LargeVehicle**	**Plane**	**roundabout**	**Ship**	**Small Vehicle**	**SoccererBallField**	**Storage Tank**	**SwimmingPool**	**Tennis Court**
Training	185	345	238	148	2495	9865	2014	161	5999	21862	201	2944	486	2110
Test	55	105	95	89	929	2903	1030	38	1408	5502	105	775	99	639
	Training : 49053 Test:13772 Total: 62825 images													

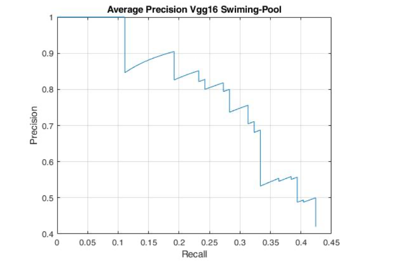
Result (Multiple Object Detection)

49053 objects in a total of 888 images belonging to 14 classes in the images were trained with 11 different deep learning algorithms. The number of epochs was kept low due to the very long training period. AlexNet, Vgg16, Vgg19, SqueezeNet, Resnet18, Resnet50, Resnet101, InceptionV3, InceptionResnetV2, GoogleNet, DenseNet201 deep learning models were used for training. For the test, RCNN object detection method was used.

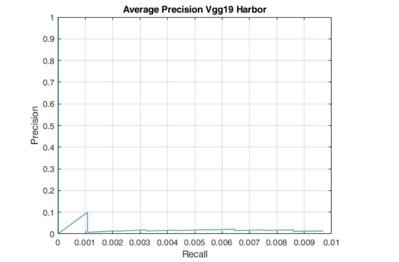
After the training processes of each algorithm were completed, the testing phase was started. Precision and recall values were calculated according to IoU> 0.5. In order to evaluate the results obtained during the test phase, mAP values accepted in the literature were used. Accordingly, the success rates of each algorithm on detecting objects belonging to 14 classes were revealed. Some graphics of the precision and recall values obtained for this are shown in figure 12.



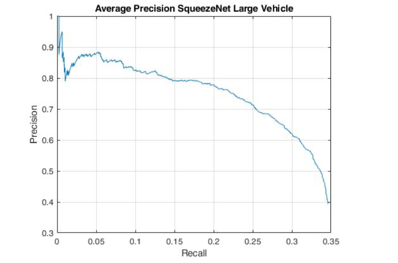
(1)



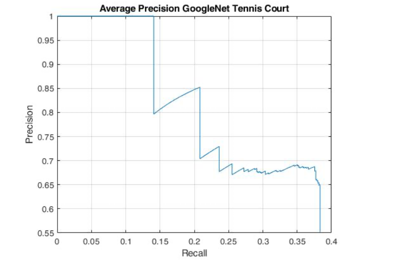
(2)



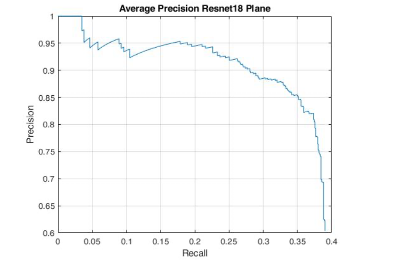
(3)



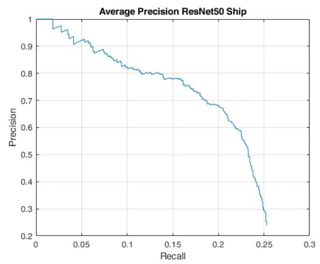
(4)



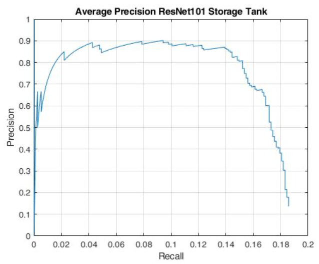
(5)



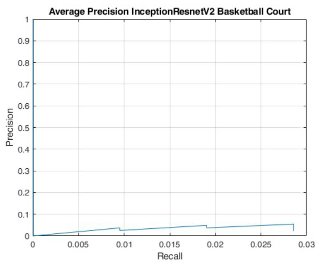
(6)



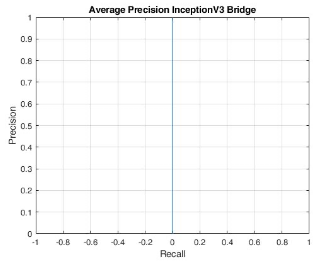
(7)



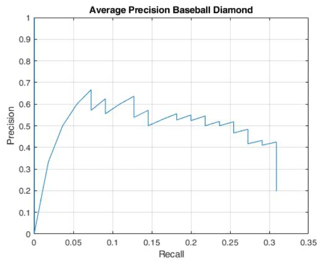
(8)



(9)



(10)

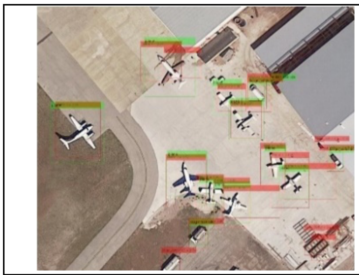


(11)

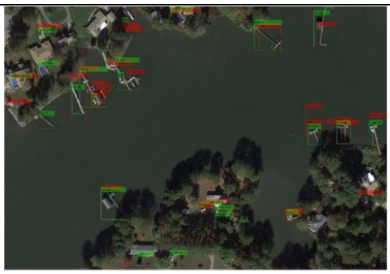
Figure 12. Some AP graphics obtained according to test results.

In the graphs, there are AP graphs containing precison and recall values of different models and different object classes. In determination with a high performance rate, the graphic precision value starts from 1 and moves towards 0, while the recall value startsfrom 0 and moves towards 1. As it can be understood from this situation, 1, 2, 5, and 6 numbers are the graphics with high performance rate. The object and model graph with the lowest achievement is seen in the pictures 9 and 10.

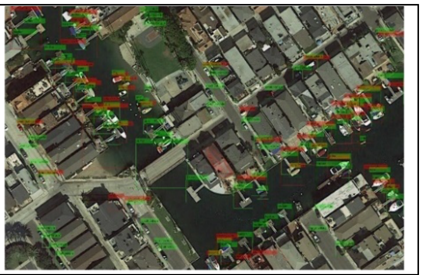
The results obtained in the images in figure 13 are given. Objects, which are in the images in figure 13, detected by the model, are labeled. While most of the objects were detected in some images, it was observed that the number of detected objects was quite low in some images.



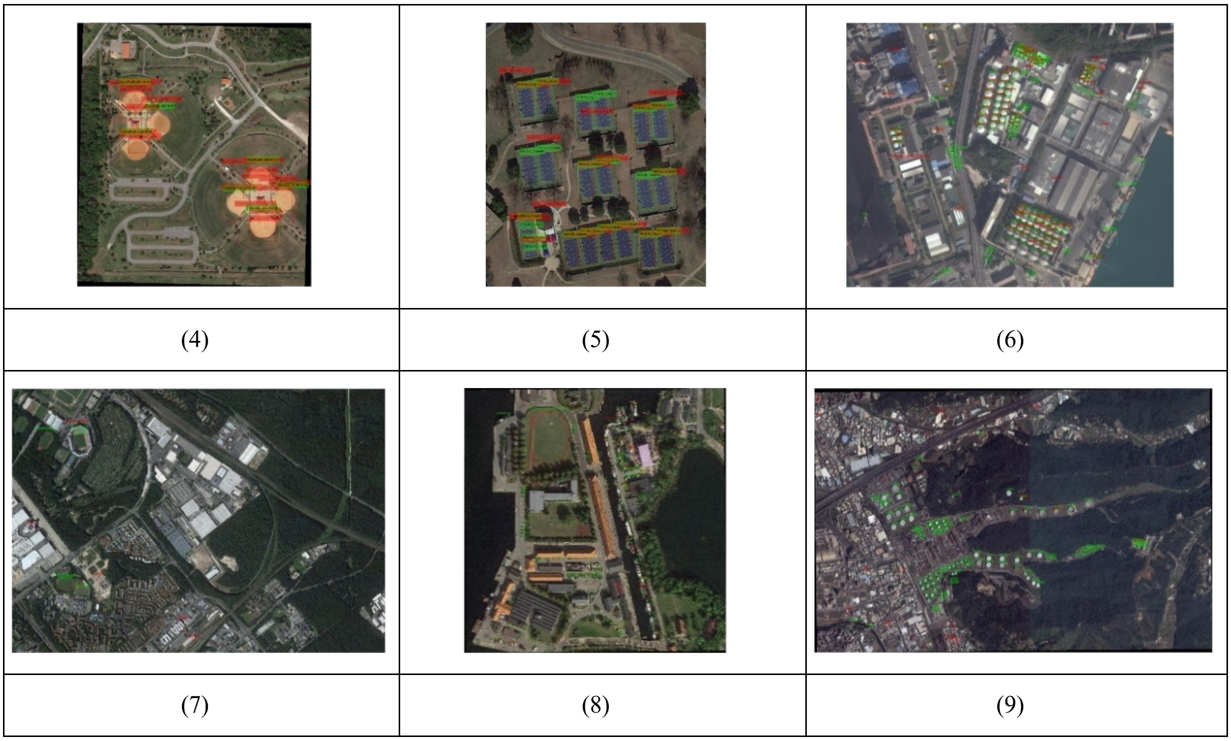
(1)



(2)



(3)



(4) (5) (6)

(7) (8) (9)

Figure 13. Some results ımages gained from deep learning architectures.

Previously labeled objects and objects found by the algorithm are shown in figure 13. The detection rate of objects is very low due to the large area of images in images numbered 6, 7, 8, and 9. Images numbered 1, 2, 3, 4, and 5 are images in which objects on the ground appear more clearly and are more successful by deep learning models. According to the seresults, the results of the detection of 14 objects of 11 deep learning models are given in table 5 Main Average Precision values. In the table, the best 2 algorithm values for each object are shown as bold.

Table 5. The mAP success rates obtained in object detection of deep learning models.

Mean Average Precision	AlexNet	Vgg16	Vgg19Net	GoogleNet	SquezeNet	Resnet18	Resnet50	Resnet101	InceptionResnetV2	InceptionV3	DenseNet201
Baseball Diamond	42,32	42,64	22,77	15,15	16,51	26,67	9,09	16,84	12,12	16,23	23,74
Basketball Court	14,65	17,48	9,09	9,09	9,09	12,66	9,09	9,09	9,09	9,09	9,09
Bridge	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09
GroundTrackField	16,53	17,42	9,09	9,09	9,09	17,42	9,09	14,55	9,09	15,38	15,38
Harbor	16,95	19,13	17,59	20,89	15,59	16,62	15,03	20,25	13,36	16,88	17,00

LargeVehicle	16,18	28,99	12,45	12,11	29,34	29,91	28,54	36,64	16,87	31,56	27,39
Plane	32,91	34,68	12,16	13,12	28,13	34,43	23,24	31,42	11,21	27,49	28,22
Roundabout	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09	9,09
Ship	22,50	25,02	23,55	22,12	22,33	24,22	22,78	25,41	10,44	24,07	27,37
Small Vehicle	14,07	16,21	11,81	12,07	14,31	16,08	14,86	15,47	12,32	15,45	13,91
SoccererBallFielde	18,18	18,18	9,09	9,09	11,53	12,03	9,09	12,73	9,09	15,51	10,54
Storage Tank	17,15	17,70	9,09	9,09	17,53	17,58	17,55	17,15	10,40	14,63	17,79
SwimmingPool	26,33	37,34	24,85	26,69	11,36	23,72	17,44	25,56	17,37	24,09	28,60
Tennis Court	53,82	52,03	29,87	32,22	27,20	54,27	52,34	48,37	15,32	49,01	45,11
AverageScore Rate	22,13	24,64	14,97	14,92	16,44	21,70	17,59	20,83	11,78	19,83	20,17

In this article, object detection performance of different deep learning models is discussed. The results of the achievements of detecting objects have been presented with the deep learning models, which are the most used in the literature. The study is thought to be a pioneering study since no similar study has been found in the literature. It is thought that it can help researchers working on object detection in choosing the deep learning model they will use and on the way they will follow. The weakest aspect of the study is that it was not tested with different object detectors. As only RCNN object detector is used, it has not been revealed how they will perform in other object detectors.

In the study, the object detection performance results of 11 deep learning algorithms in 14 classes are given in Table 5. Average performance rates of each model are also in the last row of the table. VggNet model has the highest average performance rate with 24.64 in the findings obtained. The lowest average performance was found to be InceptionResnetV2 with 11.78. The object with the highest availability rate is the tenniscourt with 54,27 and it was obtained with ResNet18. The lowest finding rate was roundabout and bridge with a rate of 9.09. It was found with a very low rate in all models. It is thought that one of the reasons for the low availability rate of roundabout and bridge classes is the few number of samples in training and testing. However, it is thought that the models cannot produce a correct result in defining objects belonging to the roundabout and bridge object class. The set two classes are considered not to be fully trained. When the tenniscourt image samples, the object class with the highest availability rate, are examined, it is concluded that the samples in the dataset are close to the groundsurface and are easier to define by the model in terms of shape.

The average performance of AlexNet on multiple object detection was 22.13 and ResNet 18 was 21.70. It has been observed that models with few number of layers are more successful than models with higher number of layers. This is seen as a finding that shows that those with a high number of layers should not be preferred in the deep learning model when detecting multiple objects. In the light of the results obtained from the models, the two best successful results in object classes are seen in Table 5. Accordingly, the models that show the best performance in object classes in the result table are shown in Table 6.

Table 6. Class-classeswherearchitecture is mostsuccessful.

	AlexNet	Vgg16	Vgg19Net	GoogleNet	SquezeNet	Resnet18	Resnet50	Resnet101	InceptionResnetV2	InceptionV3	DenseNet201
Best top tworesults	4	8	0	1	0	4	0	3	0	1	3
Best resultsproducingnumbers	1	6	0	1	0	2	0	1	0	0	2

Models	**Top Two Object Classes**	**Mostsuccessfulobjectclass**
AlexNet	Baseball Diamond, Basketball Court, SoccorerBallFielde, Tennis Court	SocorrerBallFielde
Vgg16	Baseball Diamond, Basketball Court, GroundTrackField, Plane, Small Vehicle, SoccorerBallFielde, Storage Tank, SwimmingPool	Baseball Diamond, Basketball Court, GroundTrackField, Plane, Small Vehicle, SoccorerBallFielde, SwimmingPool
Vgg19Net	-	-
GoogleNet	Harbor	Harbor
SquezeNet	-	-
Resnet18	GroundTrackField, Plane, Small Vehicle, Tennis Court	GroundTrackField, Tennis Court
Resnet50	-	-
Resnet101	Harbor, LargeVehicle, Ship,	LargeVehicle,
InceptionResnetV2	-	-
InceptionV3	LargeVehicle	-
DenseNet201	Ship, Storage Tank, SwimmingPool	Ship, Storage Tank

CONCLUSION

The values obtained from the results can be seen that the performance of each algorithm in object detection reveal different results. It is thought that success rates are high in single object detection in the literature searches, but as the number of classes increases in detecting multiple objects, the rate of performance decreases.

It has been observed that as the size of the images obtained by high-resolution remote sensing increases, there is a difficulty in detecting objects. It has been observed that object detection in images taken higher than the ground surface and covering much larger areas is quite difficult for all algorithms. It has been observed that as the field of view decreases, the number of detected objects and their performance increase. The images preferred for training and testing in the dataset were randomly selected depending on the classes. Choosing the right samples for studies to be conducted on this subject is important for testing architectures. The number of samples to be used in the training being above a certain number will be important in terms of a better evaluation of the results. While carrying out training and testing processes eliminating such problems will reveal better results. Researchers who will study on this subject should especially be careful on such points. We may suggest that these models can be tested with data of multispectrallidar.

REFERENCES

Alganci, U., Soydas, M. & Sertel, E., 2020. Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. Remote Sensing, **12(3)**: 458.

Dogan, F. & Turkoglu, İ., 2019. Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, **10(2)**: 409-445.

Ghazali, M.F., Wikantika, K., Harto, A.B. & Kondoh, A., 2020. Generating soil salinity, soil moisture, soil pH from satellite imagery and its analysis. Information Processing in Agriculture, **7(2)**: 294-306.

Girshick, R., 2015. Fast r-cnn, Proceedings of the IEEE international conference on computer vision, pp. 1440-1448.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G. & Cai, J., 2018. Recent advances in convolutional neural networks. Pattern Recognition, **77**: 354-377.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. & Lew, M.S., 2016. Deep learning for visual understanding: A review. Neurocomputing, **187**: 27-48.

Han, J., Zhang, D., Cheng, G., Liu, N. & Xu, D., 2018. Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Processing Magazine, **35(1)**: 84-100.

He, K., Zhang, X., Ren, S. & Sun, J., 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans Pattern Anal Mach Intell, **37(9)**: 1904-1916.

He, K., Zhang, X., Ren, S. & Sun, J., 2016. Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

Henderson, P. & Ferrari, V., 2016. End-to-end training of object class detectors for mean average precision, Asian Conference on Computer Vision. Springer, pp. 198-213.

Huo, C., Chen, K., Ding, K., Zhou, Z. & Pan, C., 2016. Learning relationship for very high resolution image change detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, **9(8)**: 3384-3394.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z. & Qu, R., 2019. A survey of deep learning-based object detection. IEEE Access, **7**: 128837-128868.

Kadhim, N.A., Hussein, A.K., Jaber, A.S. & Abojassim, A.A., Land Use And Land Cover Change Detection Using Satellite Images For The Kufa District, Najaf, Iraq.

Kong, T., Yao, A., Chen, Y. & Sun, F., 2016. Hypernet: Towards accurate region proposal generation and joint object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 845-853.

Krizhevsky, A., Sutskever, I. & Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM, **60(6)**: 84-90.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. & Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, **1(4)**: 541-551.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, **86(11)**: 2278-2324.

Lee, H., Grosse, R., Ranganath, R. & Ng, A.Y., 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, Proceedings of the 26th annual international conference on machine learning, pp. 609-616.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S., 2017a. Feature pyramid networks for object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P., 2017b. Focal loss for dense object detection, Proceedings of the IEEE international conference on computer vision, pp. 2980-2988.

Lin, Y.z., Nie, Z.h. & Ma, H.w., 2017c. Structural damage detection with automatic feature-extraction through deep learning. Computer-Aided Civil and Infrastructure Engineering, **32(12)**: 1025-1046.

Liu, Q., Hang, R., Song, H. & Li, Z., 2017. Learning multiscale deep features for high-resolution satellite image scene classification. IEEE Transactions on Geoscience and Remote Sensing, **56(1)**: 117-126.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A.C., 2016. Ssd: Single shot multibox detector, European conference on computer vision. Springer, pp. 21-37.

Liu, Y., Sun, P., Wergeles, N. & Shang, Y., 2021. A survey and performance evaluation of deep learning methods for small object detection. Expert Systems with Applications: 114602.

Mansour, A., Hassan, A., Hussein, W.M. & Said, E., 2019. Automated vehicle detection in satellite images using deep learning, IOP Conference Series: Materials Science and Engineering. IOP Publishing, pp. 012027.

Mu, M., Wu, C., Li, Y., Lyu, H., Fang, S., Yan, X., Liu, G., Zheng, Z., Du, C. & Bi, S., 2019. Long-term observation of cyanobacteria blooms using multi-source satellite images: a case study on a cloudy and rainy lake. Environmental Science and Pollution Research, **26(11)**: 11012-11028.

Napiorkowska, M., Petit, D. & Marti, P., 2018. Three applications of deep learning algorithms for object detection in satellite imagery, IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 4839-4842.

Padilla, R., Netto, S.L. & da Silva, E.A., 2020. A survey on performance metrics for object-detection algorithms, 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE, pp. 237-242.

Pathak, A.R., Pandey, M. & Rautaray, S., 2018. Application of deep learning for object detection. Procedia computer science, **132**: 1706-1717.

Pereira, R.F., da Silva Filho, V.E., Moura, L.B., Kumar, N.A., Auzuir, R. & de Albuquerque, V.H.C., 2020. Automatic quantification of spheroidal graphite nodules using computer vision techniques. The Journal of Supercomputing, **76(2)**: 1212-1225.

Ping Tian, D., 2013. A review on image feature extraction and representation techniques. International Journal of Multimedia and Ubiquitous Engineering, **8(4)**: 385-396.

Rabbi, J., Ray, N., Schubert, M., Chowdhury, S. & Chao, D., 2020. Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network. Remote Sensing, **12(9)**: 1432.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016. You only look once: Unified, real-time object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788.

Said, N., Ahmad, K., Riegler, M., Pogorelov, K., Hassan, L., Ahmad, N. & Conci, N., 2019. Natural disasters detection in social media and satellite imagery: a survey. Multimedia Tools and Applications, **78(22)**: 31267-31302.

Shermeyer, J. & Van Etten, A., 2019. The effects of super-resolution on object detection performance in satellite imagery, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 0-0.

Srinivas, V., 2020. LFBNN: Robust and Hybrid Training Algorithm to Neural Network for Hybrid Features-Enabled Speaker Recognition System. Journal of Engineering Research, **8(2)**.

Sun, P., Chen, G., Luke, G. & Shang, Y., 2018. Salience biased loss for object detection in aerial images. arXiv preprint arXiv:1810.08103.

Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning, Proceedings of the AAAI Conference on Artificial Intelligence.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D. & Ioffe, S., 2014. Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441.

Tishby, N. & Zaslavsky, N., 2015. Deep learning and the information bottleneck principle, 2015 IEEE Information Theory Workshop (ITW). IEEE, pp. 1-5.

Vorontsov, I.E., Kulakovskiy, I.V. & Makeev, V.J., 2013. Jaccard index based similarity measure to compare transcription factor binding site models. Algorithms for molecular biology : AMB, **8(1)**: 23.

Wang, P., Sun, X., Diao, W. & Fu, K., 2019. FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing, **58(5)**: 3377-3390.

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M. & Zhang, L., 2018. DOTA: A large-scale dataset for object detection in aerial images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3974-3983.

Xiao, T., Li, H., Ouyang, W. & Wang, X., 2016. Learning deep feature representations with domain guided dropout for person re-identification, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1249-1258.

Xu, B., Wang, N., Chen, T. & Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

Yang, J., Nguyen, M.N., San, P.P., Li, X. & Krishnaswamy, S., 2015. Deep convolutional neural networks on multichannel time series for human activity recognition, Ijcai. Buenos Aires, Argentina, pp. 3995-4001.

Yi, D., Lei, Z., Liao, S. & Li, S.Z., 2014. Deep metric learning for person re-identification, 2014 22nd International Conference on Pattern Recognition. IEEE, pp. 34-39.

Ying, X., Wang, Q., Li, X., Yu, M., Jiang, H., Gao, J., Liu, Z. & Yu, R., 2019. Multi-attention object detection model in remote sensing images based on multi-scale. IEEE Access, **7**: 94508-94519.

Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.-H. & Patton, R.M., 2015. Optimizing deep learning hyper-parameters through an evolutionary algorithm, Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, pp. 1-5.

Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, **30(11)**: 3212-3232.