Editor

Prof. Dr. Mehmet Kamanlı

EĞİTİM yayınevi

International Research in

Engineering Sciences III

2022

Nov. - Dec.

Editor:

Prof. Dr. Mehmet Kamanlı

International Research in

Engineering Sciences III

EĞİTİM
yayınevi

INTERNATIONAL RESEARCH IN ENGINEERING SCIENCES III
Editor: Prof. Dr. Mehmet Kamanlı

Executive Editor: Yusuf Ziya Aydoğan (yza@egitimyayinevi.com)
Publishing Coordinator: Yusuf Yavuz (yusufyavuz@egitimyayinevi.com)
Interior Designer: Eğitim Yayınevi Graphics Unit
Cover Designer: Eğitim Yayınevi Graphics Unit

Republic of Türkiye Ministry of Tourizm and Culture
Publisher Certificate No: 47830

ISBN: 978-625-6382-25-1
1. Edition, November-December 2022

Printing and Binding
Bulut Dijital Matbaa Sanayi Ticaret Limited Şirketi
Musalla Bağları Mah. İnciköy Sok. 1/A Selçuklu / KONYA
Printing House Certificate No: 48120

Library Information Card
INTERNATIONAL RESEARCH IN ENGINEERING SCIENCES III
Editor: Prof. Dr. Mehmet Kamanlı
p.302, 165x240 mm
Includes references, no index.
ISBN: 978-625-6382-25-1

EĞİTİM
yayınevi

Publisher Turkey Office: İstanbul: Eğitim Yayınevi Tic. Ltd. Şti., Atakent mah. Yasemen sok. No: 4/B, Ümraniye, İstanbul, Türkiye

Konya: Eğitim Yayınevi Tic. Ltd. Şti., Fevzi Çakmak Mah. 10721 Sok. B Blok, No: 16/B, Safakent, Karatay, Konya, Türkiye
+90 332 351 92 85, +90 533 151 50 42, 0 332 502 50 42
bilgi@egitimyayinevi.com

Publisher USA Office: New York: Egitim Publishing Group, Inc.
P.O. Box 768/Armonk, New York, 10504-0768, United States of America
americaoffice@egitimyayinevi.com

Logistics and Shipping Center: Kitapmatik Lojistik ve Sevkiyat Merkezi, Fevzi Çakmak Mah. 10721 Sok. B Blok, No: 16/B, Safakent, Karatay, Konya, Türkiye
sevkiyat@egitimyayinevi.com

Bookstore Branch: Eğitim Kitabevi, Şükran mah. Rampalı 121, Meram, Konya, Türkiye
+90 332 499 90 00
bilgi@egitimkitabevi.com

Internet Sales: www.kitapmatik.com.tr
+90 537 512 43 00
bilgi@kitapmatik.com.tr

ilk okuyan siz olun
kitapmatik
internetteki kitapçınız

İÇİNDEKİLER

DEVELOPING A PREDICTION MODEL WITH THE BATTLE ROYALE OPTIMIZATION ALGORITHM 5
Safa DÖRTERLER

HYBRID DEVICE BASED ON NANOSTRUCTURED ZnO FILM DOPED WITH GRAPHENE OXIDE 21
Arife GENCER IMER, Abdulkadir KORKUT

INVESTİGATİON OF ENERGY RECOVERY FROM REFUSE DERİVED FUELS (RDF) BY HEAT TREATMENTS 37
Gizem AYAS, İbrahim BOZ

MONKEYPOXHYBRIDNET: A HYBRID DEEP CONVOLUTIONAL NEURAL NETWORK MODEL FOR MONKEYPOX DISEASE DETECTION 49
Burak GÜLMEZ

SEISMIC RISK ASSESSMENT OF RC STRUCTURES USING A NEW RAPID EVALUATION METHOD 65
Didem BÜTÜN, Taha Yasin ALTIOK, Ali DEMİR

USABILITY OF WASTE MATERIAL REINFORCED SOILS IN RAILWAY STRUCTURE 93
Nurdan BAYKUŞ

COMPARATIVE STUDIES ON CONVENTIONAL & BACTERIAL SELF COMPACTING CONCRETE 109
L.V. Prasad MEESARAGANDA, Bhavani SIRANGI, Giri P. S. N. RATNA

DETERMINATION of SENSORY ACCEPTABILITY of FUNCTIONAL ICE CREAM PRODUCED USING WHEY PROTEIN GEL and BLUEBERRY 119
Zeynep SUNAL, Halime PEHLİVANOĞLU, Aslı AKSOY, Mustafa YAMAN

EVALUATION OF WIND ENERGY POTENTIAL IN MERSIN PROVINCE OF TURKEY 133
Mehmet KARAYEL, Çağatay ERSİN

EXAMINATION OF IMPORTANT FACTORS AFFECTING ASPHALT PAVEMENT PERFORMANCE 143
Sedat OZCANAN

INVESTIGATION OF THE ECOTOXICOLOGICAL EFFECTS OF COMMONLY USED PHARMACEUTICALS 155
Süheyla TONGUR, Sevil YILDIZ

INVESTIGATION OF THE EFFECT OF NANO-GRAPHENE ADDITIVES ON THE IMPACT BEHAVIOUR OF CARBON FIBER SANDWICH COMPOSITES BY IMAGE PROCESSING 183
Ibrahim DEMIRCI, Ismail SARITAS

MECHANICAL OBJECT PARTS DETECTION USING DEEP LEARNING BASED YOLO MODELS 205
Mübarek Mazhar ÇAKIR, Gökalp ÇINARER

METAL ORGANIC FRAMEWORK PHOTOCATALYST SYNTHESIS BY MECHANOCHEMISTRY 227
Hüsnü Arda YURTSEVER

NEW ARCHITECTURE IN DEEP LEARNING: CAPSULE NETWORKS (CapsNet) 245
Sumeyra Busra SENGUL, Ilker Ali OZKAN

PHYSICAL AND BEHAVIOUR BIOMETRIC AUTHENTICATION METHOD 263
Ümmügülsüm MENGUTAYCI

RENEWABLE ENERGY AND TYPES 275
İbrahim BOZ

EFFECTS OF DIGITALIZATION ON SUPPLY CHAIN MANAGEMENT 291
Batin Latif AYLAK

MECHANICAL OBJECT PARTS DETECTION USING DEEP LEARNING BASED YOLO MODELS

Mübarek Mazhar ÇAKIR[1], Gökalp ÇINARER[2]

INTRODUCTION

Artificial intelligence is very popular these days. The development of artificial intelligence has increased the use of machines and smart systems in many fields. Especially in the industry, their use has started to become widespread day by day, as they make fewer mistakes than people and can produce more serially and with higher quality. The ability of machines to perform certain tasks by interacting with the outside world primarily depends on their perception of the objects in their environment. The perception processes of the machines are carried out with auxiliary tools such as sensors, keys and cameras. With deep learning, where more complex structures can be resolved compared to machine learning, studies in this direction continue to progress rapidly.

Inventory management and planning are of great importance for the uninterrupted continuation of mass production in the machinery industry. Companies with high production capacity and high daily production are required to deliver products to customers on time and to use their production resources in the best way to meet the periodical expected demand. It is necessary to determine the number of materials and mechanical parts to be used in production and to calculate the stock amount accordingly (Sarıca 1998). Mechanical parts such as bolts, nuts, bearings, etc., which are inevitable to be used in every machine, are of vital importance for

1 Mr. Mübarek Mazhar ÇAKIR1, Yozgat Bozok University, Department of Mechatronics Engineering, Türkiye, 0000-0001-7771-1271, mashar66@gmail.com
2 Asst. Prof. Gökalp ÇINARER2, Yozgat Bozok University, Department Of Computer Engineering, Türkiye, 0000-0003-0818-6746, gokalp.cinarer@bozok.edu.tr

machine assembly and production. The absence of one of the mechanical parts that must always be in stock and this situation cannot be detected in a timely manner; It causes some problems such as not being able to complete the machine, delaying the deadline, not reaching the targeted production number. In order to minimize these problems and correctly classify the products in stock, deep learning method and object detection algorithm are used. Object detection not only detects whether there are objects in that area, but also helps to detect the correct product between different objects.

Objects in the images have distinctive features like shape, texture, color and brightness. These features are the distinctive attribute information of the objects. By identifying and using this information, desired objects can be detected in the images. However, problems such as blurred object images, closeness of the image background and object color, and complexity in the image make it complexity to detect the object with classical image processing methods.

Object detection and tracking with deep learning algorithms has made it easier to overcome these difficulties. Although there are different studies for object detection in recent years, the number of studies on the detection of mechanical parts is very limited.

In this study, Yolo algorithm-based models were created to facilitate the use of mechanical parts in the industrial field and to ensure the correct classification of products. The results obtained from each model were examined comparatively.

2. RELATED WORKS

Cites	Works Detailse	Object Detection Algorithms Used
(Bayram 2020)	The Mask R-CNN algorithm was trained with 430 plate images and 3140 characters in the images. Afterwards, an automatic license plate recognition trial was performed with 130 images and a 98.46% success rate was obtained as a result of the trial.	Mask R-CNN
(Daş et al. 2019)	The algorithm was trained with a special dataset consisting of eight classes. Object detection and counting were done via photograph, video and webcam. As a result of the training, a success rate of 98% was achieved.	Faster R-CNN
(Corovic et al. 2018)	In the study using the Berkley Deep Drive dataset, the data set was set to 70.000 trains and 30.000 validations. It has been studied on five different classes. As a result of the training, a success rate of 46.6% was achieved.	Yolov3
(Aktaş et al. 2020)	In the study, Yolov3 and DenseNet models were combined. The dataset consisting of 1200 image. The best result was obtained with a success rate of 92% compared to the Yolov2 and Yolov3 algorithms.	Yolov2, Yolov3, Yolov3-Dense
(Özel et al. 2021)	In the study, existing cracks in suspension parts were detected in real time. 80% of the created data set is reserved as train and 20% as test. As a result of experimental studies, Yolov4 was found to be more successful than other versions with a success rate of 96.3%.	Yolo-Tiny, Yolov2, Yolov3, Yolov4
(Ieamsaard et al. 2021)	The dataset consists of three classes, masked, unmasked and false masked. 682 images are reserved for train and 86 images for test. A total of 853 images were studied. Yolov5 success rate of 96.5% was obtained.	Yolov5

3. MATERIAL AND METHOD

3.1. Deep Learning

Deep learning is an extremely popular application area as a subtitle of machine learning (O'Mahony et al. 2020). The use of deep learning method, which successfully solves more complex tasks compared to classical machine learning, has become very common in recent days and has many application places like medical diagnosis, natural language and image process (Shinde and Shah 2018).

In traditional programming, as shown in Figure 1, a data is given by the user to a code block written by the programmer, the given data goes through the desired operations in the code lines and the output is taken.

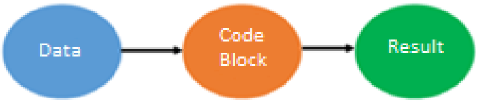

Figure 1: Traditional Programming Approach

In line with the outputs, the programmer makes an observation about achieving the desired result. When the result is not expressed correctly or the desired result is not obtained with the programmer's observations, the programmer tries to improve the result by editing the codes has created.

In machine learning, in addition to the data, the information about the desired results is given to a learning algorithm together. The machine goes through a learning process using data and result information and creates a model in this learning process. This approach to machine learning is visualized in Figure 2.

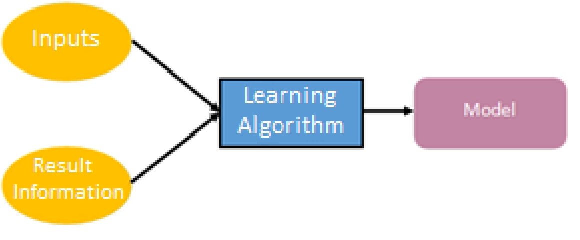

Figure 2: Machine Learning Approach

The process in machine learning can be defined as automatically generating the codes written by the programmer in traditional programming in a way that gives the most accurate results. Supervised, Unsupervised, and Reinforcement Learning are machine learning types (Arslankaya and Toprak 2021).

Supervised learning labels (result information) are given to the learning algorithm along with the dataset and the machine creates the model by solving the mathematical relationship between the data and the results.

In unsupervised learning, a classified dataset is not given for learning. In this case, the algorithm groups the data according to the similarities and differences between the data (Mahesh 2020).

Reinforcement learning is a goal-oriented type. As a result of the correctness of the movements, feedback is given with rewards. While maximizing the rewards, the model approaches the most correct course of action.

3.2. Artificial Neural Networks (ANN)

ANN are mathematically modeled by examining the structure and learning of the human brain. The point where machine and deep learning differ is the structures of neural networks. When Figure 3 is examined, the structure difference in the neural networks can be understood more clearly.

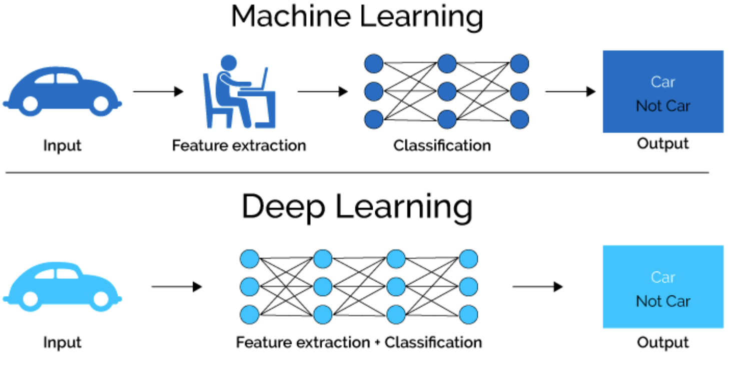

Figure 3: Neural Networks (Odi and Nguyen 2018)

Neural networks in machine learning consist of several layers. In deep learning, there are three main layers. For input layer where the inputs are taken, in hidden layer where inputs are processed, and the output layer where the results are (Lecun et al. 2015). In the hidden layers, where main layers, can sometimes be thousands. The neuron structure of neural networks is given in detail in Figure 4.

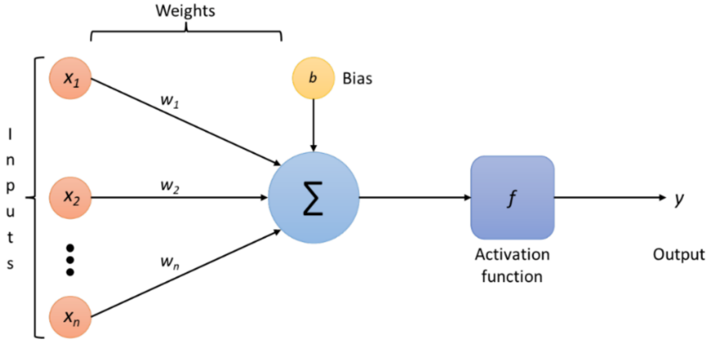

Figure 4: Structure Of A Neuron In A Neural Network (Sánchez et al. 2022)

Neurons in neural networks receive input from the outside world, multiply by weight, then add up with the bias value and produce result. Then result is given for activation function and an output is generated. Weight values are created by analyzing the structure of the data set to reduce the deviation, and the process of finding the optimum weight value is called training the network (Öztemel 2012). The bias value is used to change the range of results obtained by multiplying the input by the weight (Yılmaz and Kaya 2021).

$$y = \text{Activation Function}\left(\sum(\text{Weight}*\text{Input}) + \text{Bias}\right) \tag{1}$$

The activation function, as seen in equation 1 is a mathematical equation that determines the output. The output produced by the neuron is transferred to the next neuron in the neural network, and these processes repeat themselves in the neural network. As a result of these operations, non-linear results are obtained. Obtaining nonlinear results here provides a complex learning opportunity. In supervised learning, the process of creating the appropriate weight and bias values by analyzing the data set of the neural networks is realized by calculating the error with the loss function and minimizing the error with the optimization algorithm (Lillicrap et al. 2020).

3.2.1. Convolutional Neural Networks (CNN)

In classical ANN, the abundance of information for data set to increase processing load in training the network, which causes processing times and computational difficulties. Instead of retrieving all the data on the image in operations such as object detection, training the neural network with the features in the desired region alleviates the processing load on the network. The process of obtaining information in the desired regions on the image is called feature extraction (Wu 2017). Convolutional neural networks perform feature extraction with layers like convolution, pooling and fully-connected. (O'Shea and Nash 2015).

3.2.2. Convolution Layer

Images are processed in matrix form. In this layer, a feature map of the image is created by applying a filter to the image. We have a matrix of the image and a matrix of the filter. Matrix multiplication is performed in the convolution layer as in Figure 5. Multiplication is performed between the matrix of the image and the matrix of the filter.

$$\begin{bmatrix} 2 & 7 & 3 \\ 1 & 9 & 4 \\ 3 & 5 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 19 & 23 \\ 18 & 20 \end{bmatrix}$$

Figure 5: Applying Filters To The Image Matrix

In the above operation, a 3x3 matrix is multiplied by a 2x2 matrix and the result is a 2x2 matrix. Figure 6. shows how a 3x3 filter is applied to a 7x7 matrix.

Figure 6: Applying the Filter to the Image (Albawi et al. 2018)

The operation in Figure 5 is the 2D convolution operation. Color images consist of 3 channels. In color images, the filter should be applied for all 3 channels as shown in Figure 7.

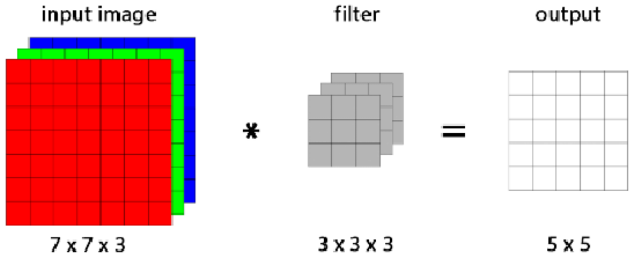

Figure 7: Applying The Filter To A Color Image (Hristov et al. 2019)

3.2.3. Pooling Layer

In this layer size of the input matrix is reduced without destroying its distinctive features, thereby reducing the processing load (Kattenborn et al. 2021). There are two types, maximum pooling and average pooling (O'Shea and Nash 2015). While the average pooling process was applied to the matrix on the left and maximum pooling process was applied to the matrix on the right side of Figure 8.

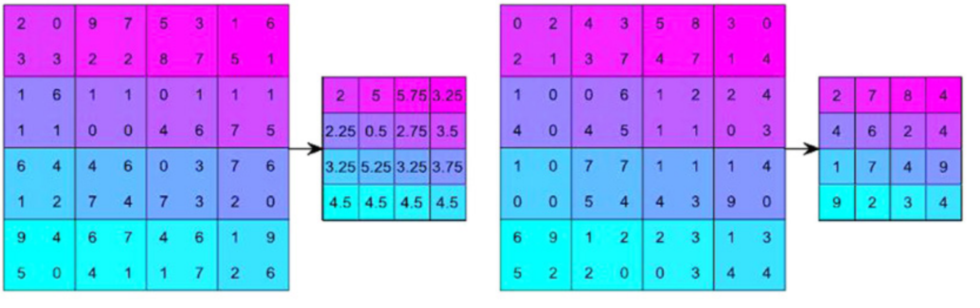

Figure 8: Max Pooling And Average Pooling (Li et al. 2019)

In the pooling process, a filter is applied as in the convolution layer. However, here, the largest of the values remaining in the frame where the filter is applied is taken (max. pooling) or the average of these values is taken (average pooling).

3.2.4. Fully-Connected Layer

The data converted to vector in the flattening layer of the matrices from the convolution and pooling layers are given as input to the fully-connected layer. It depends on each of the fields in the previous layer. After this layer, the classification process of the data takes place with the classification layer (İnik and Ülker 2017).

3.3. Yolo Algorithm

Yolo (You Only Look Once) is a CNN-based algorithm capable of object detection and object tracking. It is very good balance of speed and accuracy (Li et al. 2022). Other object detection algorithms perform detection in two stages, as it determines possible regions in the image where the object is located and then executes CNN classifiers in these regions one by one. The popularity and speed of Yolo is that it passes the object detection process that other object detection algorithms do in two stages, through the neural network in a single stage (Chen et al. 2018).

While detecting objects, the Yolo algorithm divides the image into cells (grid) as in Figure 9, and each cell checks whether the object exists in its own part of the image; If the object exists, it is responsible for finding the object's class, center point, height, and width (Redmon et al. 2015).

Figure 9: Splitting The Image Into Cells (Grid) (Lan et al. 2018)

The found objects are enclosed in boxes called bounding boxes. However, there is more than one bounding box for each object and a prediction vector. Prediction vector confidence score (c), x coordinate of the object's center point (b_x), y coordinate center point (b_y), height (b_h), width (b_w) and contains the class probabilities (p_c) found in the model (Vardhan et al. 2022). If we call the prediction vector created for the bounding box y, we can represent the vector as follows:

$$y = (p_c, b_x, b_y, b_h, b_w, c) \tag{2}$$

The problem of having more than one bounding box for an object is solved by a technique called non-maximum suppression. The non-maximum suppression technique deletes the bounding boxes with a low confidence score and highest confidence score remains (Garg et al. 2018). By deleting the bounding boxes in a low confidence score, only one box remains around the object, as seen in Figure

10. ... ing the existence of this obj ... actual bounding box of the

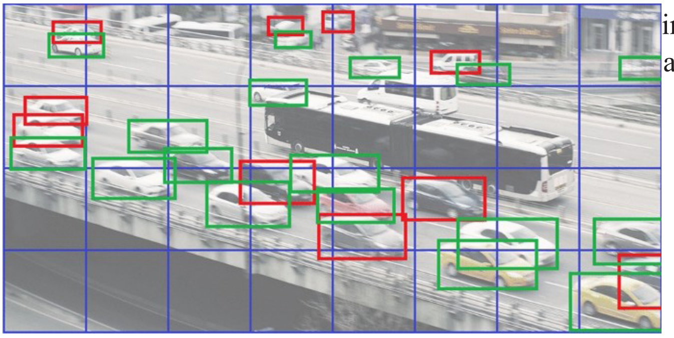

Figure 10: An Output From The Yolo Algorithm (Hurtik et al. 2022)

The Yolo algorithm uses loss functions to optimize accuracy for calculate to the differentiation among the actual value of the object and estimated value during the object detection process (Huang et al. 2019).

$$Loss = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} [(b_{x_i} - b_{\hat{x}_i})^2 + (b_{y_i} - b_{\hat{y}_i})^2] \tag{3}$$

$$+\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} [(\sqrt{b_{w_i}} - \sqrt{b_{\hat{w}_i}})^2 + (\sqrt{b_{h_i}} - \sqrt{b_{\hat{h}_i}})^2] \tag{4}$$

$$+\sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \tag{5}$$

$$+\lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{A} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \tag{6}$$

$$+\sum_{i=0}^{s^2} 1_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \tag{7}$$

Here, the sum of the equations of the loss functions is given. Equations 3 and 4 represent the localization loss, equations 5 and 6 represent the confidence loss and equation 7 represent the classification loss.

Considering that an image is divided into cells in the form of SxS, the letter A here denotes the bounding boxes. The actual and predicted states of the bounding boxes may be very close to each other. To solve this problem, the variable λ is used. The λ variables in the equation are used to increase the salience in the boxes where the object exists and to decrease the salience in the bounding boxes where the object does not exist. 1_{ij}^{obj}, value 1 for an object in cell i; if there are no objects in cell i, it means 0. 1_{ij}^{noobj}, value 0 for an object in this cell i; takes the value 1 if there is no object it means i. The architectural structure and convolution layers of the Yolo algorithm are given in Figure 11.

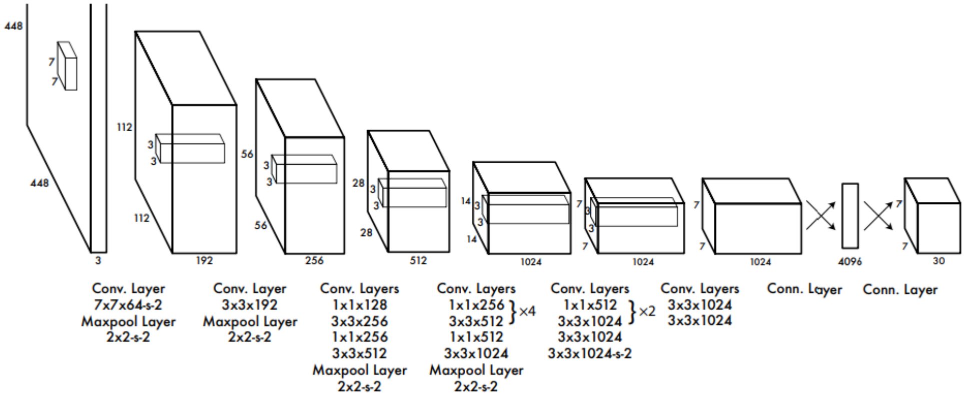

Figure 11: Yolo Architecture (Redmon et al. 2015)

In Yolo, we mentioned that it is accountable for object detecting in the area covered by the cell on the image. In case of multiple objects in the same cell, Yolo can make errors in object detection. As a solution to the problem, Yolov2 using anchor box was proposed (Redmon et al. 2015), thus increasing accuracy and performance. With Yolov2, improvements have also been made to remove the recall errors existing in Yolo and the accuracy rate has been increased by adding batch normalization to each convolution layer (Atik et al. 2022). Yolov2 uses the Darknet19 classification algorithm. As seen in Figure 12, the Darknet19 classification algorithm consists of 19 convolution layers, 5 maximum pooling layers and 1 softmax layer.

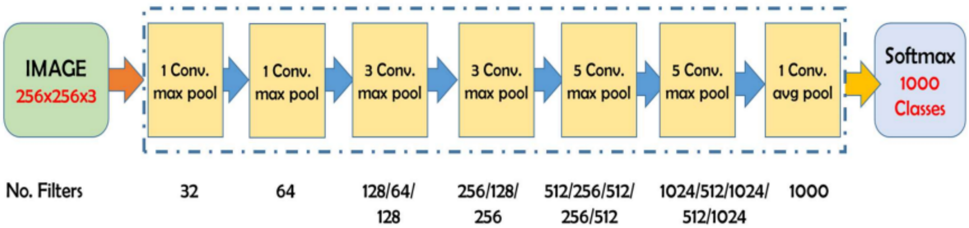

Figure 12: Darknet19 Architecture (Al-Haija et al. 2021)

With Yolov3, the understanding that "every object belongs to a class" in previous versions has been removed, and the understanding that an object can have more than one class label has been introduced (Atik et al. 2022). For example, a bearing on the image can be included in both the bearing class and the mechanical part class. While Darknet 19 performs the feature extraction of images in Yolov2, Darknet-53, whose architecture is given in Figure 13, performs these operations in Yolov3.

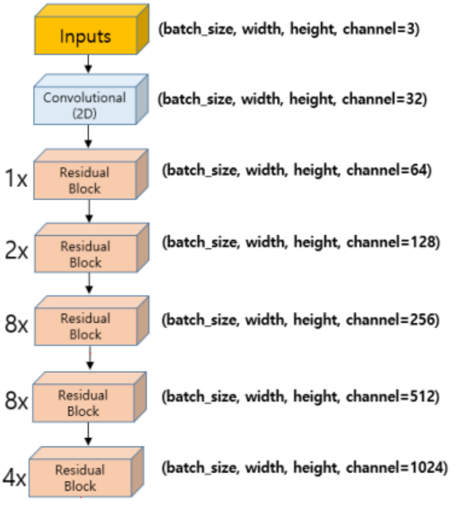

Figure 13: Darknet-53 Architecture (Kim et al. 2019)

Darknet-53, which consists of 53 convolution layers, provides more detailed information from the image. However, Yolov3 is less successful in detecting medium and large objects than Yolov2 (Atik et al. 2022). FPN (Feature Pyramid Network) was used to obtain more information on the image by scaling in Yolov3 (Afif et al. 2020). Figure 14 symbolizes the FPN structure and Figure 15 symbolizes the use of the FPN structure in Yolo.

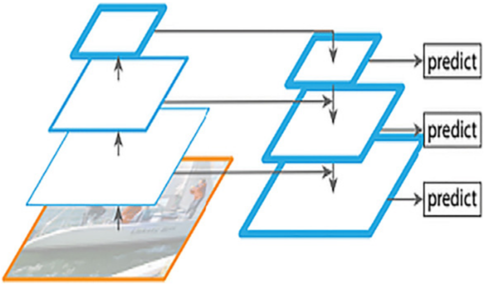

Figure 14: FPN Structure (Afif et al. 2020)

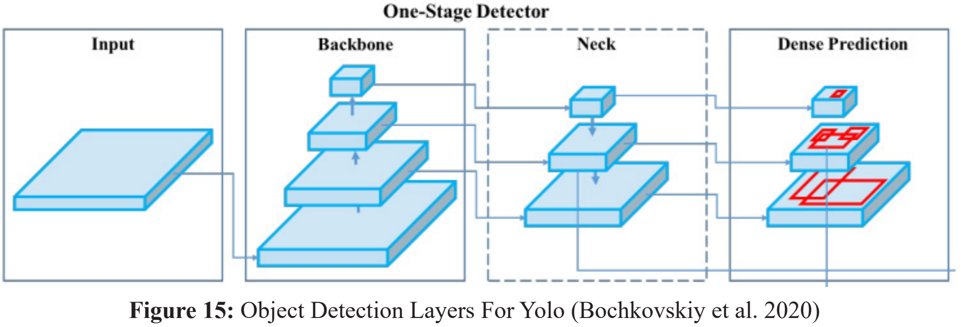

Figure 15: Object Detection Layers For Yolo (Bochkovskiy et al. 2020)

CSPDarknet53 model was used in the backbone layer, where feature extraction was performed in the Yolov4 algorithm, and it was seen that this model gave better results than the Darknet-53 model (Bochkovskiy et al. 2020). In addition, SAM, PAN, SSP are used in Yolov4 in the layer (neck) that enables more information to be obtained while object estimation is being made (Bochkovskiy et al. 2020). Maximum and average pooling operations are not applied on the SAM implementation used in Yolov4. In addition, while the addition was performed in the original PAN application, the merge operation was performed in the modified PAN application. These processes have been carried out to obtain high accuracy and high performance.

Yolov5 is the version of Yolo released in 2020 by Glenn Jocher. Yolov5 uses CSPDarknet53 in the backbone layer and PANet in the neck layer, as in the previous version Yolov4 (Ieamsaard et al. 2021). The different aspects of the developed version from previous versions used in Pytorch as the framework and the use of Focus structure in the backbone layer, as seen in Figure 16, where the Yolov5 architecture is shown (Nepal and Eslamiat 2022). As a result of the additions and improvements, the training time was shortened and the storage size of the model was reduced (Nelson and Solawetz 2020). In addition, with the addition of the Focus layer, the required Cuda memory and the number of layers have been reduced, and forward and backward propagation has increased (Nepal and Eslamiat 2022).

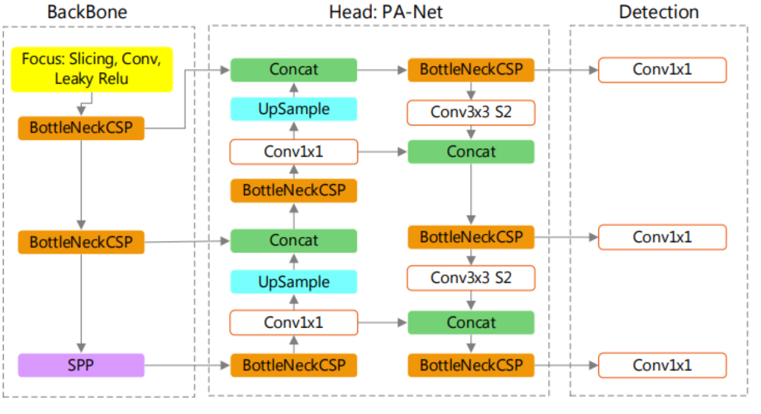

Figure 16: Yolov5 Architecture (Ieamsaard et al. 2021b)

There are five different basic models of the Yolov5 algorithm. These models can be listed as Yolov5-n (nano), Yolov5-small, medium, large and xlarge respectively. The Yolov5-n model, which is the smallest model among the models, works faster than the other models and the weight file created and it's train result is smaller than the other models. Nano model is mostly designed for use in embedded systems and mobile applications. The Yolov5-s model is larger than the Yolov5-n model, but still a small model when compared to other models. Smaller models allow image feature extraction via the CPU.

The model that ranks third in terms of size is the Yolov5-m model. Small models are faster but less accuracy rate than large models. Large models, on the other hand, have higher accuracy but slower than small models. While creating the Yolov5-m model, a balance was maintained between operating speed and accuracy. In other words, optimum speed and optimum accuracy ratio are presented in a balanced way in this model.

Finally, the two biggest models of Yolov5 are Yolov5-l and Yolov5-x, respectively. These models contain more parameters than others but run slower. Large models are good at detecting small objects. The Yolov5-x model is the model with the highest performance rate, but also the slowest.

3.4. Method Of Evaluating The Results Of Object Detection Algorithms

Many algorithms have been created for the detection of an object through the image and many improvements have been made on them. When object detection is performed by object detection algorithms, the results obtained need to be evaluated. First of all, we need to understand what the confusion matrix. Figure 17 is gives the describing of accuracy between actual situation and the predicted situation in the classification process (Shultz et al. 2011). For example the bolt on an image will be detected. True positive that the algorithm calls the bolt in the image a bolt (TP), true negative if the algorithm says not a bolt to a non-bolt (TN), false negative if the algorithm says the bolt is not a bolt (FN), if the algorithm calls the non-bolt a bolt, it is evaluated as false positive (FP).

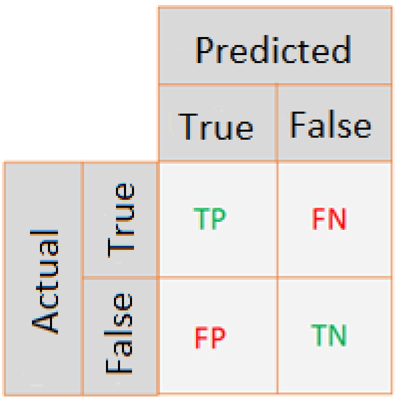

Figure 17: Confusion Matrix

Based on the confusion matrix, we can find the accuracy of our model by the ratio of correct predictions to all predictions.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

However, the success of the model cannot be determined with accuracy. Here, the dispersion of the classes in dataset on which the model is trained is also important.

Consider a two-class dataset consisting of 990 bolts and 10 nuts, a total of 1000 mechanical parts. In this case, the accuracy rate in bolt detection is calculated as 99%. This shows that the success of the model cannot be determined by the accuracy value alone. In addition to the accuracy value, it is necessary to examine the precision and recall values.

The precision value shows how many of the objects the model predicts correctly are actually correct (Zhang and Su 2012).

$$Precision = \frac{TP}{TP + FP}$$

The recall value gives how many of the objects we actually need to detect correctly and that are correctly detected by the model (Zhang and Su 2012).

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

F1 score is a value that includes precision and recall values. It is the harmonical mean of these values (Aktaş et al. 2020).

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Mean average precision (mAP) is used to evaluate precision, recall, F1 score and IoU (Intersection over Union) values in a single number (Aktaş et al. 2020).

$$mAP = \int_0^1 P(R)dR$$

3.4. Dataset

One of the most important steps when creating a dataset is the correct labeling of the dataset. The correct labeling of objects in the data set and the size of the data set directly affect the success rate of object detection, making learning parameters easier to use. In this study, an open source roboflow mechanical parts dataset consisting of four classes, bearing, nut, gear, and bolt, was prepared. Labeling processes were done manually on the Roboflow website. A few labeled image from the dataset are shared in Figure 18.

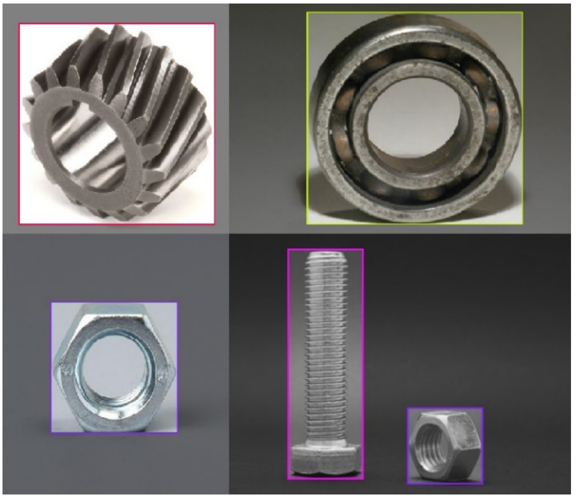

Figure 18: Images Of Dataset

The dataset consists of a total of 2250 images obtained by downloading from various internet platforms. Among the images in the dataset, there are 714 images with bearings, 632 images with bolts, 616 images with gears and 586 images with nuts. A total of 10597 manual labeling processes were carried out in the dataset, including 2099 labels belonging to the bearing class, 2734 labels belonging to the bolt class, 2662 labels belonging to the gear class and 3102 labels belonging to the nut class. As can be seen in Figure 19, the distribution of labels and images according to classes has become more complex and difficult to detect due to more than one labeling process in some images and the fact that objects belonging to two different classes are in the same image.

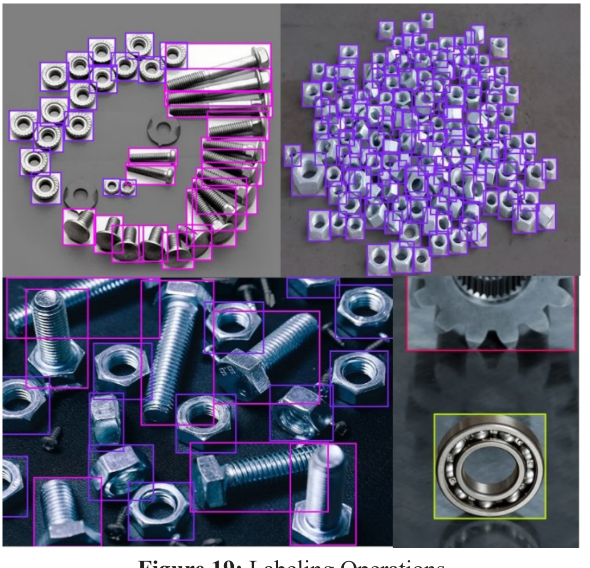

Figure 19: Labeling Operations

3.5. Training Process

Different models of the Yolov5 version of the Yolo algorithm are trained. The training of the models was carried out on the Google Colab platform. Google Colab is a virtual environment that allows users to use high RAM and GPU for operations such as deep learning and object recognition. In this environment, data processing was carried out using Python-based Tensorflow, Keras and PyTorch libraries. The dataset is reserved for 80% train, 10% validation and 10% test operations. Basic parameters such as study training parameters, epoch number, learning rate, optimization algorithm, activation function were determined in accordance with the mechanical parts dataset. Basic parameters for the models are given in Table 1.

All training processes were run over 100 epochs. The dimensions of all images given as input are set to 640x640. At the beginning of the training, the transfer learning process was carried out using the weights of the Coco dataset.

Table 1: Hyperparameters

Model	Learning Rate	Momentum	Activation Function	Optimization Algorithm	Epoch
Yolov5-n	0.01	0.937	SiLU	SGD	100
Yolov5-s	0.01	0.937	SiLU	SGD	100
Yolov5-m	0.01	0.937	SiLU	SGD	100
Yolov5-l	0.01	0.937	SiLU	SGD	100
Yolov5-x	0.01	0.937	SiLU	SGD	100

Learning rate can be evaluated as the coefficient of error reduction to zero. Here, the optimization algorithm is used to reduce the error to zero. In some optimization algorithms, the learning rate is not taken as a fixed value, but is determined by the training process of the model.

The SGD (Stochastic Gradient Descent) optimization algorithm and SILU (Sigmoid Linear Units) activation function used in all models in the study uses a fixed value as the learning rate. If the learning rate is too large, it causes the zero point to be missed at the point of reducing the error to zero, while the it is too small, causing the processing to take too long and negatively affect the performance.

The use of momentum together with SGD in the study is to reduce the oscillation by increasing the speed. In Table 2, the training times of all models are given in detail.

Table 2: Training Times

Model	Training Time (minute)
Yolov5-n	56.1
Yolov5-s	62.1
Yolov5-m	81.7
Yolov5-l	121.8
Yolov5-x	222.3

Considering the duration of education, it is seen that there is an opposite proportion between the model structure and the speed of the model. That is, as the model grows, the speed of the model decreases; as the structure of the model gets smaller but the speed increases.

In this direction, it is seen that the largest model, the Yolov5-x model, is the slowest and the smallest model, the Yolov5-n, is the fastest model.

4. RESULT AND DISCUSSION

Five different models of the Yolov5 algorithm were run with the mechanical parts dataset to identify the parts used in machine manufacturing and compare their performance. As a result of the training, the models used for object detection. The training results of the applied algorithms are shown in Table 3.

Table 3: Training Results for Yolov5 Models

Model	Precision	Recall	F1	Weight File Size (MB)	mAP
Yolov5-n	0.861	0.788	0.822	3.72	0.858
Yolov5-s	0.885	0.803	0.842	13.77	0.892
Yolov5-m	0.915	0.833	0.872	40.27	0.901
Yolov5-l	0.913	0.836	0.873	88.56	0.906
Yolov5-x	0.922	0.841	0.879	165.11	0.911

When the size of the file of the weights created by the Yolov5 models as a result of the training is examined, it is seen that the file sizes are directly proportional to the model sizes. The Yolov5-x model with the largest model structure has the largest file size with 165.11 MB. Compared to other models, the Yolov5-n model, which has the smallest file size with a file size of 3.72 MB, is the smallest model for all models in terms of structure.

When the mAP values obtained by the algorithms of four different classes in the mechanical parts dataset were examined, it was seen that the highest value was obtained in the Yolov5-x model with 0.911. This model was followed by the Yolov5-l and Yolov5-m models, respectively. According to the mAP results, the Yolov5-n model gave the lowest result with 0.858. When Figure 20 is examined, it is observed that the mAP values of the models increase as the epoch increases. Increasing mAP values tend to be more stable at the end of each epoch. It is seen that all models tend to increase in direct proportion to the number of epochs.

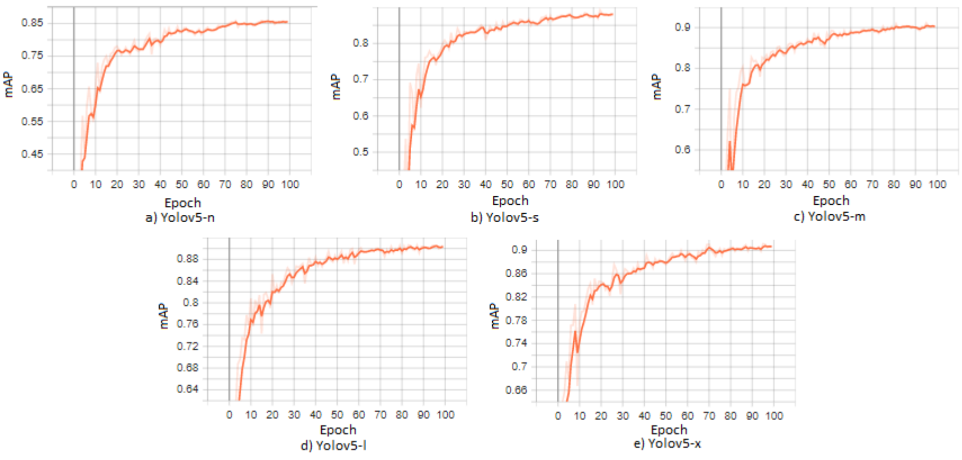

Figure 20: Epoch-mAP Graphics of Yolov5 Models

After the training process of the algorithms was completed, the test images were given to the models after the training process to evaluate the results. In the testing phase of the Yolov5 models, 225 images that were not used in the training phase were used. The performance results of the tested models are detailed in Table 4.

Table 4: Test Results Of Yolov5 Models

Model	Precision	Recall	F1	mAP
Yolov5-n	0.887	0.819	0.851	0.882
Yolov5-s	0.884	0.830	0.856	0.901
Yolov5-m	0.920	0.846	0.881	0.901
Yolov5-l	0.903	0.858	0.879	0.910
Yolov5-x	0.916	0.870	0.892	0.919

When Table 4 is examined, an increase in the mAP value of the model is observed. According to the table the mAP value of the Yolov5-x model increased by approximately 0.9% as a result of the test and reached 0.919. As a result of the test, the model showing the highest increase rate with a 2.8% increase in mAP value is the Yolov5-n model. In this case, it is seen that Yolov5-x model obtained the best result according to mAP values. On the contrary, Yolov5-n model gave the lowest result.

The precision and recall curves of the test results of the five models used are given in Figure 21. The area under the curve in the graph represents the mAP value. In other words, a large curve on the graph means that the model performance is also good. Conversely, the smaller the curve, the lower the performance of the model due to the smaller area under it.

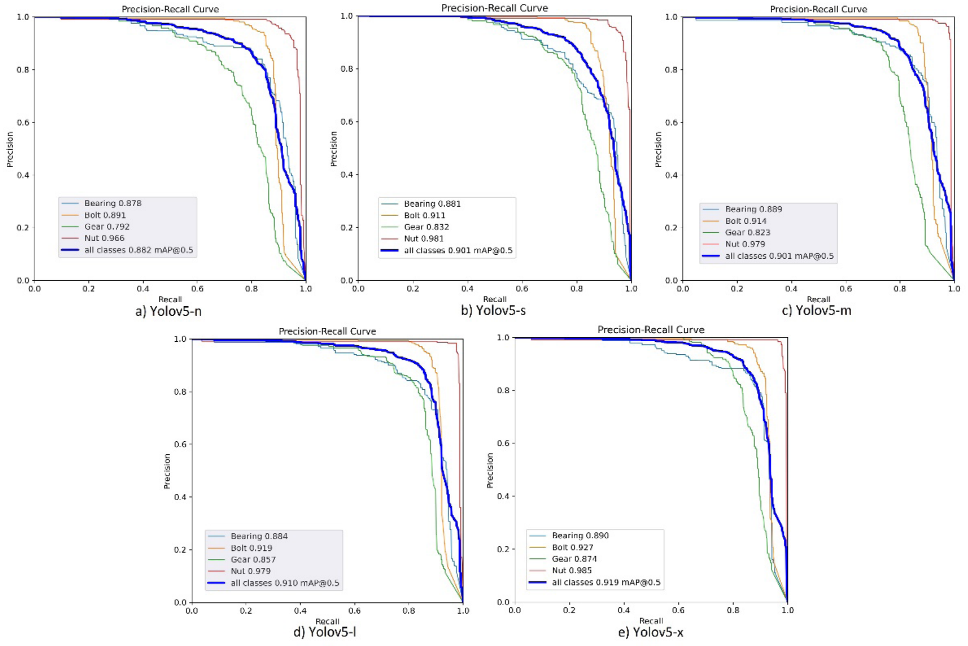

Figure 21: Precision-Recall Curves Of Models

In the test phase, the models are tested with images that they did not see during the training phase, that is, they did not learn and they try to detect the objects in these images.

As a result of the test, the correct and incorrect prediction rates of the models emerge. The correct prediction rates of the five model for four class given in Table 5.

Table 5: Prediction Accuracy Of Yolov5 Models

Class	Model				
	Yolov5-n	**Yolov5-s**	**Yolov5-m**	**Yolov5-l**	**Yolov5-x**
Bearing	%86	%83	%85	%87	%88
Bolt	%87	%86	%87	%90	%89
Gear	%74	%78	%76	%80	%80
Nut	%96	%96	%98	%98	%98

When Table 5 is examined, the highest prediction accuracy rate of the bearing class belongs to the Yolov5-x model with 88%. This ratio shows that the mechanical parts dataset correctly knew 88% of the 206 labeled bearings in the test group, or 181 of them, but did not know the remaining 25. The highest prediction accuracy rate of the Bolt class belongs to the Yolov5-l model with 90%. The Yolov5-l model correctly knew 293 of the 326 labeled bolts, while 33 did not. The highest prediction accuracy in Gear class belongs to Yolov5-x and Yolov5-l models with 80%. Of the 319 labeled gears in the two models test data, 225 of them were correct and the remaining 94 were not.

Finally, the highest prediction accuracy of the Nut class with 98% belongs to the Yolov5-x, Yolov5-l and Yolov5-m models. The models predicted 244 of the 249 labeled nuts correctly and 5 of them incorrectly.

Ana Malta et al. (Malta et al. 2021) trained Yolov5-s and Yolov5-m models with their own dataset to detect and classify mechanical equipment in automobiles. In the study, it was stated that Yolov5-s is sufficient for the detection and classification of mechanical equipment of automobiles. Yadian Zhao et al. (Zhao et al. 2022) conducted a study to detect loose bolts at various attachment points. They worked with NPU-BOLT dataset and Yolov5-s, Yolov5-l, Faster-RCNN and CenterNet algorithms to detect loosened bolts. Yolov5-l model was recommended because of its higher accuracy than other models. Jianhui Zhao et al. (Zhao et al. 2022) apply a study on the detection of gear pump parts for an aid system for gear pump assembly. In their study, they used a dataset of mechanical parts such as gear, shaft and wedge of a gear pump. They stated that the Yolov5-l model is suitable for detecting the mentioned mechanical parts.

Yolo algorithms have performed quite well in object detection and recognition. Our research aimed to identify, classify and count equipment used in industry quickly and accurately. The results obtained mean that the Yolov5-x model can successfully meet its intended targets, with a 0.919 mAP value higher than other models.

In future studies, the classes in the data set can be increased and the images of the classes in the mechanical parts data set can be reproduced to increase the performance rate and the proposed model can be optimized. By modifying the hyperparameter of the Yolov5 algorithms, the effect of the deep learning model on the performance rate can be examined. In addition, the studies can be examined more comprehensively with the new yolo algorithm versions used in object selection. The mechanical parts dataset can be run on the last two versions of the Yolo series, Yolov6 and Yolov7, and compared with the Yolov5 model.

REFERENCES

Afif, Mouna, Riadh Ayachi, Edwige Pissaloux, Yahia Said, and Mohamed Atri. 2020. "Indoor Objects Detection and Recognition for an ICT Mobility Assistance of Visually Impaired People." *Multimedia Tools and Applications 2020 79:41* 79(41):31645–62. doi: 10.1007/S11042-020-09662-3.

Aktaş, Abdulsamet, Buket Doğan, and Önder Demir. 2020. "Derin Öğrenme Yöntemleri Ile Dokunsal Parke Yüzeyi Tespiti." *Journal of the Faculty of Engineering and Architecture of Gazi University* 35(3):1685–1700. doi: 10.17341/gazimmfd.652101.

Aktaş, Abdulsamet, Buket Doğan, and Önder Demir. 2020b. "Tactile Paving Surface Detection with Deep Learning Methods." *Journal of the Faculty of Engineering and Architecture of Gazi University* 35(3):1685–1700. doi: 10.17341/gazimmfd.652101.

Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. 2018. "Understanding of a Convolutional Neural Network." *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017* 2018-January:1–6. doi: 10.1109/ICENGTECHNOL.2017.8308186.

Al-Haija, Qasem Abu, Mahmoud Smadi, and Osama M. Al-Bataineh. 2021. "Identifying Phasic Dopamine Releases Using Darknet-19 Convolutional Neural Network." *IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*. doi: 10.1109/IEMTRONICS52119.2021.9422617.

Arslankaya, Seher, and Şevval Toprak. 2021. "Makine Öğrenmesi ve Derin Öğrenme Algoritmalarını Kullanarak Hisse Senedi Fiyat Tahmini." *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi* 13(1):178–92. doi: 10.29137/umagd.771671.

Atik, Muhammed Enes, Zaide Duran, and Roni Ozgunluk. 2022. "Comparison of YOLO Versions for Object Detection from Aerial Images." 9(2). doi: 10.30897/ijegeo.1010741.

BAYRAM, Fatih. 2020. "Derin Öğrenme Tabanlı Otomatik Plaka Tanıma." *Politeknik Dergisi* 23(4):955–60. doi: 10.2339/POLITEKNIK.515830.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection." doi: 10.48550/arxiv.2004.10934.

Chen, Yuan, Marlies C. Goorden, and Freek J. Beekman. 2018. "Understanding of Object Detection Based on CNN Family and YOLO." *Journal of Physics: Conference Series* 1004(1):012029. doi: 10.1088/1742-6596/1004/1/012029.

Corovic, Aleksa, Velibor Ilic, Sinica Duric, Malisa Marijan, and Bogdan Pavkovic. 2018. "The Real-Time Detection of Traffic Participants Using YOLO Algorithm." *2018 26th Telecommunications Forum, TELFOR 2018 - Proceedings*. doi: 10.1109/TELFOR.2018.8611986.

Daş, Resul, Berna Polat, Gürkan Tuna, Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Trakya Üniversitesi, Teknik Bilimler MYO, and Bilgisayar Teknolojileri Bölümü. 2019. "Derin Öğrenme Ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi." *Fırat Üniversitesi Mühendislik Bilimleri Dergisi* 31(2):571–81. doi: 10.35234/FUMBD.608778.

Garg, Dweepna, Parth Goel, Sharnil Pandya, Amit Ganatra, and Ketan Kotecha. 2018. "A Deep Learning Approach for Face Detection Using YOLO." *IEEE*. doi: 10.1109/PUNECON.2018.8745376.

Hristov, Anton, Maria Nisheva, and Dimo Dimov. 2019. "Filters in Convolutional Neural Networks as Independent Detectors of Visual Concepts." *ACM International Conference Proceeding Series* 110–17. doi: 10.1145/3345252.3345294.

Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. 2019. "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers." *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018* 2503–10. doi: 10.1109/BIGDATA.2018.8621865.

Hurtik, Petr, Vojtech Molek, Jan Hula, Marek Vajgl, Pavel Vlasanek, and Tomas Nejezchleba. 2022. "Poly-YOLO: Higher Speed, More Precise Detection and Instance Segmentation for YOLOv3." *Neural Computing and Applications 2022 34:10* 34(10):8275–90. doi: 10.1007/S00521-021-05978-9.

İnik, Özkan, and Erkan Ülker. 2017. "Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri." *GAZİOSMANPAŞA BİLİMSEL ARAŞTIRMA DERGİSİ (GBAD)* 6(3).

Ieamsaard, Jirarat, Surapon Nathanael Charoensook, and Suchart Yammen. 2021. "Deep Learning-Based Face Mask Detection Using YoloV5." *Proceeding of the 2021 9th International Electrical Engineering Congress, IEECON 2021* 428–31. doi: 10.1109/IEECON51072.2021.9440346.

Ieamsaard, Jirarat, Surapon Nathanael Charoensook, and Suchart Yammen. 2021b. "Deep Learning-Based Face Mask Detection Using YoloV5." *Proceeding of the 2021 9th International Electrical Engineering Congress, IEECON 2021* 428–31. doi: 10.1109/IEECON51072.2021.9440346.

Kattenborn, Teja, Jens Leitloff, Felix Schiefer, and Stefan Hinz. 2021. "Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing." *ISPRS Journal of Photogrammetry and Remote Sensing* 173:24–49. doi: 10.1016/J.ISPRSJPRS.2020.12.010.

Kim, Kwang Ju, Pyong Kun Kim, Yun Su Chung, and Doo Hyun Choi. 2019. "Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data." *IEEE Access* 7:78311–19. doi: 10.1109/ACCESS.2019.2922479.

Lan, Wenbo, Jianwu Dang, Yangping Wang, and Song Wang. 2018. "Pedestrian Detection Based on Yolo Network Model." *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018* 1547–51. doi: 10.1109/ICMA.2018.8484698.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature 2015 521:7553* 521(7553):436–44. doi: 10.1038/nature14539.

Li, Chuyi, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. 2022. "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications."

Li, Zhi, Shui Hua Wang, Rui Rui Fan, Gang Cao, Yu Dong Zhang, and Ting Guo. 2019. "Teeth Category Classification via Seven-Layer Deep Convolutional Neural Network with Max Pooling and Global Average Pooling." *International Journal of Imaging Systems and Technology* 29(4):577–83. doi: 10.1002/IMA.22337.

Lillicrap, Timothy P., Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. 2020. "Backpropagation and the Brain." *Nature Reviews Neuroscience 2020 21:6* 21(6):335–46. doi: 10.1038/s41583-020-0277-3.

Mahesh, Batta. 2020. "Machine Learning Algorithms-A Review." *International Journal of Science and Research (IJSR)* 9(1). doi: 10.21275/ART20203995.

Malta, Ana, Mateus Mendes, and Torres Farinha. 2021. "Augmented Reality Maintenance Assistant Using YOLOv5." *Applied Sciences* 11(11):4758. doi: 10.3390/APP11114758.

Nelson, Joseph, and Jacob Solawetz. 2020. "Responding to the Controversy about YOLOv5." Retrieved December 5, 2022 (https://blog.roboflow.com/yolov4-versus-yolov5/).

Nepal, Upesh, and Hossein Eslamiat. 2022. "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs." *Sensors* 22(2). doi: 10.3390/S22020464.

Odi, Uchenna, and Thomas Nguyen. 2018. "Geological Facies Prediction Using Computed Tomography in a Machine Learning and Deep Learning Environment." *SPE/AAPG/SEG Unconventional Resources Technology Conference 2018, URTC 2018*. doi: 10.15530/URTEC-2018-2901881.

O'Mahony, Niall, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. 2020. "Deep Learning vs. Traditional Computer Vision." *Advances in Intelligent Systems and Computing* 943:128–44. doi: 10.1007/978-3-030-17795-9_10/COVER.

O'Shea, Keiron, and Ryan Nash. 2015. "An Introduction to Convolutional Neural Networks." doi: 10.48550/arxiv.1511.08458.

Özel, Muhammed Abdullah, Selim Sefa Baysal, and Mustafa Şahin. 2021. "Derin Öğrenme Algoritması (YOLO) Ile Dinamik Test Süresince Süspansiyon Parçalarında Çatlak Tespiti." *European Journal of Science and Technology Special Issue* (26):1–5. doi: 10.31590/ejosat.952798.

Öztemel, Ercan. 2012. *YAPAY SİNİR AĞLARI*. 3rd ed. edited by R. Çölkesen. İstanbul: Papatya Yayıncılık Eğitim.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2015. "You Only Look Once: Unified, Real-Time Object Detection." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December:779–88. doi: 10.48550/arxiv.1506.02640.

Sánchez, Gutiérrez, Eduardo Máximo, Pérez González, and Pablo Pedro. 2022. "Multi-Class Classification of Medical Data Based on Neural Network Pruning and Information-Entropy Measures." *Entropy* 24(2):196. doi: 10.3390/E24020196.

Sarıca, Mustafa. 1998. "Seri İmalat Yapan Bir İşletmede Üretim Planlama Ve Kontrolü." İstanbul Teknik Üniversitesi, İstanbul.

Shinde, Pramila P., and Seema Shah. 2018. "A Review of Machine Learning and Deep Learning Applications." *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*. doi: 10.1109/ICCUBEA.2018.8697857.

Shultz, Thomas R., Scott E. Fahlman, Susan Craw, Periklis Andritsos, Panayiotis Tsaparas, Ricardo Silva, Chris Drummond, Charles X. Ling, Victor S. Sheng, Chris Drummond, Pier Luca Lanzi, João Gama, R. Paul Wiegand, Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Jun He, Sanjay Jain, Frank Stephan, Sanjay Jain, Frank Stephan, Claude Sammut, Michael Harries, Claude Sammut, Kai Ming Ting, Bernhard Pfahringer, John Case, Sanjay Jain, Kiri L. Wagstaff, Siegfried Nijssen, Anthony Wirth, Charles X. Ling, Victor S. Sheng, Xinhua Zhang, Claude Sammut, Nicola Cancedda, Jean-Michel Renders, Pietro Michelucci, Daniel Oblinger, Eamonn Keogh, and Abdullah Mueen. 2011. "Confusion Matrix." *Springer* 209–209. doi: 10.1007/978-0-387-30164-8_157.

Vardhan, Ala Vishnu, Sai Sowjanya, and K. Siva Kumar. 2022. "Improvement of Object Detection Using YOLO Method in OpenCV." *JAC : A Journal Of Composition Theory* 15(3):123–29.

Wu, Jianxin. 2017. "Introduction to Convolutional Neural Networks." *Lamda Group*.

Yılmaz, Atınç, and Umut Kaya. 2021. *Derin Öğrenme*. 3rd ed. edited by G. Atlı. İstanbul: İnkılab Kitapevi Yayın San. Tic. A.Ş.

Zhang, Peng, and Wanhua Su. 2012. "Statistical Inference on Recall, Precision and Average Precision under Random Selection." *Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012* 1348–52. doi: 10.1109/FSKD.2012.6234049.

Zhao, Jianhui, Xiaolin Mei, Jinsheng Ma, Rui Wang, and Ruyi Zhang. 2022. "Research and Implementation on Key Technologies of Assembly Assistance System Based on Augmented Reality." *Journal of Physics: Conference Series*. doi: 10.1088/1742-6596/2196/1/012017.

Zhao, Yadian, Zhenglin Yang, and Chao Xu. 2022. "NPU-BOLT: A Dataset for Bolt Object Detection in Natural Scene Images." doi: 10.48550/arxiv.2205.11191.