1. Introduction

Fault Detection and Classification of Metal Nuts Containing Anomaly by Deep Learning Techniques

**Hasan GOKKAYA1, Murat KOMESLI2, Can AYDIN1**

1 Dokuz Eylül University, Institute of Social Science, Department of Management Information Systems,

Dokuzcesmeler Campus 24 Sk. No:2, 35160 Buca, Izmir, Turkey

hasan.gokkaya@ogr.deu.edu.tr, canaydin@ogr.deu.edu.tr

2 Yasar University, School of Applied Sciences, Department of Management Information Systems, Universite cad. 37, 35100, Bornova, Izmir, Turkey

murat.komesli@yasar.edu.tr (\*Corresponding author)

**Abstract:** In the smart communication network that came with the Fourth Industrial Revolution, it is important for businesses to develop efficient business models with data reporting, analysis and cost-effective management. This process requires industrial organizations to integrate their business processes with smart systems and transform them into modern automation units. In this sense, computer vision applications are one of the solutions that catch the future. Businesses that have gone through automation processes have begun to integrate these techniques into their own special operations. In this integration, the production process is especially important in terms of creating a common point with the main subjects of industrial enterprises such as raw materials, quality control and operational efficiency. This study proposes an intelligent approach to detect faulty products and optimize production efficiency in the industrial field.

The aim of this study is to develop an intelligent business intelligence application that detects broken products during production, contributes to reducing the error rate and increasing the quality, optimizing the use of raw materials, increasing production efficiency and providing lowcost operation of the process with computer vision in industrial enterprises producing metal nuts.

**Keywords:** Anomaly, Fault, Computer Vision, Deep Learning, Machine Learning, Object Detection, Classification

Today, industrial enterprises that do not adapt to the intelligent communication structure that emerged with the Internet of Things (Ashton, 2009) and spread throughout the industrial sphere with Industry 4.0; inadequate quality control, unforeseen production errors, increase in raw material consumption, cost increase resulting from inefficient production and control process, waste of potential benefit from the resulting information, problems that the application will contribute directly or indirectly. Computer vision techniques provide an important key to this error detection process as corporations are forced to invest in precision control systems that can detect even minor defects in the manufacturing process, along with increasingly demanding markets (Gonzalez and Woods, 1992; Russ,1995; Suetens et al., 1992).

The purpose of this study is to identify the broken products that occur during production in real time in industrial enterprises producing metal nuts, to determine the cause of error, to reduce the error rate, to increase quality, to save raw materials, to optimize production efficiency and to propose a computer vision application based on deep learning methods that help the process to function cost-effectively.

This paper is structured as follows: In the following section, a background information on data mining is provided. In section 3, a comprehensive literature review has been done. In section 4, materials and methods have been depicted. Conclusion and results are made in the conclusion section.

2. Background Information

Networks with a large number of nodes, consisting of a large number of pre-trained Artificial Intelligence working on big data, are called deep learning. One of the main features of deep learning applications is that the amount of nodes on the network is very large (Alpaydin, 2011). Deep learning explores the complex structure in large datasets using the back propagation algorithm to show how the internal parameters used to calculate the machine's representation on each layer from the representation created in the previous layer should be changed (LeCun et al., 2015). Aminanto and Kim (2016) use deep learning based on approaches guided by Deng (2014); They classified many deep learning models by dividing them into three subgroups as producer (unsupervised), distinctive (supervised) and hybrid (both supervised and unsupervised).

**2.1 Supervised (Discriminative) Learning**

Deep learning, which aims to distinguish parts of the data with the help of pre-trained data by labeling them for classification process (Deng, 2014), is one of the most common and most successful techniques, which is at the center of recent developments in machine learning, especially in solving human-level tasks such as image recognition (Stoica, 2019). In this study, the images in the data set are classified as "faulty" and "faultless" by applying the labeling process, and "CNN-Convolutional Neural Network", which has a special architecture, is used for this process.

**2.2 Convolution and Convolutional Neural Network (CNN)**

The idea of modeling biological neural networks was evaluated with the logic of proposition in the early stages, and then with the discovery of concepts such as conversation and back propagation applied to neural networks, neural networks became better results. Until GPU (Graphics Processing Units) were discovered, computers were not fast enough to implement multilayered neural networks. So it wasn't commercially viable. Thanks to the power of GPUs and more efficient algorithms, CNN's have become applicable in real life (Mane et al., 2020).

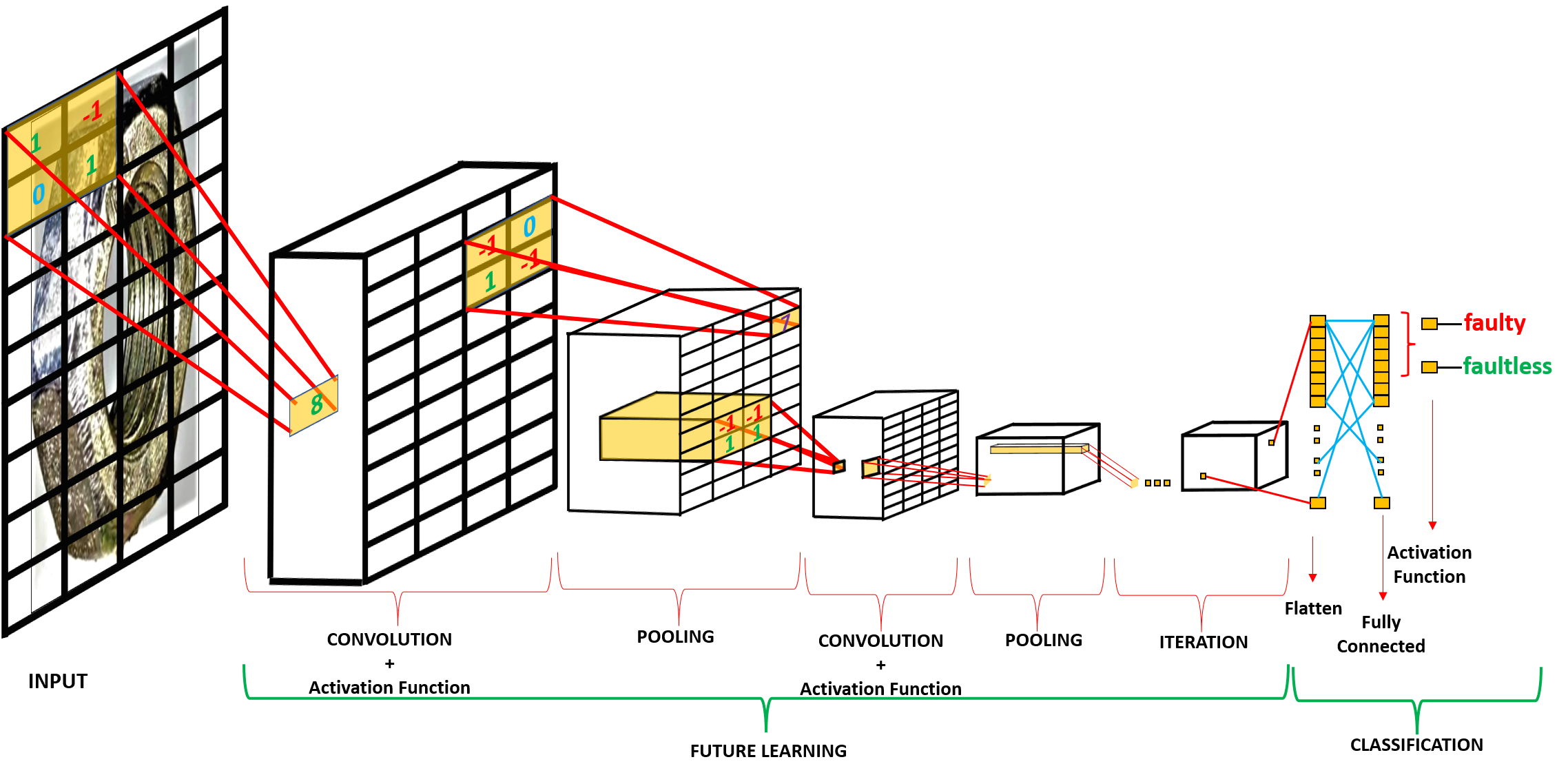
The development of computer science and artificial intelligence technologies has led to a remarkable performance improvement in the use of convolutional neural networks in image classification and computer vision applications (Russakovsky, 2015). As a result of these developments, computer vision applications, which are considered statistically, have become applicable with deep learning methods (Volulodimos et al., 2018).

CNN is an artificial network that preserves the hierarchical structure by learning the internal feature representations of the neural network (Manaswi, 2018).

Instead of needing hand-designed features, a CNN includes a feature extractor that works automatically during the training process. The CNN feature extractor consists of special type neural networks whose weights are determined during the training phase, and since it has fewer parameters and connections than a similar feedforward network, the training process is facilitated (Krizhevsky et al., 2012).

**2.2.1 CNN Stages**

A typical CNN architecture consists of layers of input, convolution, pooling, activation and classification, respectively, as shown in   
Figure 1.



**Figure 1.** CNN Stages

In neural networks, the image used must be flattened and vectored. CNN is used to prevent the loss of corners and picture depth in the image during this process.

CNN is the first layer to extract image features and preserve pixel-to-pixel relationship using small frames of an input image. Image matrix and two input fields, such as filter or kernel, are a mathematical process (Gopikrishna, 2018). Instead of flattening and vectorizing the picture used as input, CNN uses filters to preserve the 3-dimensional structure of the picture by making shifts in the amount of stride value specified in pixels to all pixels of the picture. After applying the filter with the same depth as the layer depth to which itis applied, only one output is output that gives the output value of that region, and an activation map of the same size is obtained with the filter. The number of filters in the layer is applied individually, and the output depth is the same as the number of filters applied at the end.

With activation functions, multiple neuron layers are activated to make errors easily back propagation, thus making corrections to the exit of functions and bringing the truth closer. Thus, nonlinear real-world features are introduced to the evolutionary network, providing closer and stronger learning to the real world.

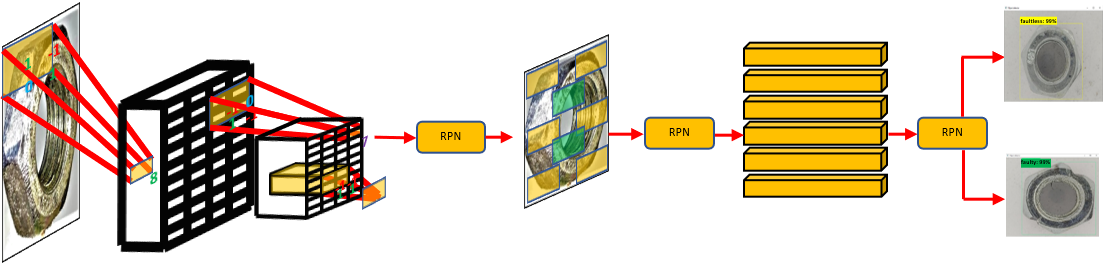
Pooling is applied to reduce the computational load for the next network layer and to provide better learning by preventing the system from memorizing. With the pooling process, when the input images are very large, the layer size is usually reduced by half, making the model smaller and reducing the number of parameters. Pooling is done at each independent depth measure to reduce the image size spatially. Therefore, the depth of the image remains unchanged, but there is a loss of data that does not cause a negative impact. The most common form of the pooling layer applied in general is maximum pooling (max pooling). The maximum pooling process is applied to each depth dimension of the convolutional output. The process performed on this layer is also called down sampling.

After pooling, the future mapping matrix is flattened to vector (x1, x2, x3...) and a future map is obtained. To create a model after this stage, these extracted properties are combined with fully connected layers. When convolution layers create 3D activation maps, data is needed about which class the image belongs to. When the forward pass through the output layer is complete, the weights and biases for error reduction begin to be updated and these values are moved to the RPN-Region Proposal Network to detect the relevant regions. RPNs, on the other hand, give the loss value for the object's location and whether an object can be found, and the loss value for that object's location if the object is present.

Once features for the input image from CNN are obtained, the RPN layer can be created with the zone recommendations (Anchors)/Bounding Box. The predicted region suggestions then estimate the offset values with the R-CNN via the RoIP (Pooling Region of Interest) layer, creating bounding boxes to classify the image within the proposed region and reshaping it according to the suggestions.

**2.2.2 Faster RCNN and RPN**

Figure 2 shows a high-level architecture for Faster R-CNN that uses RPN and that we apply in our study.



**Figure 2.**  Faster R-CNN Architecture

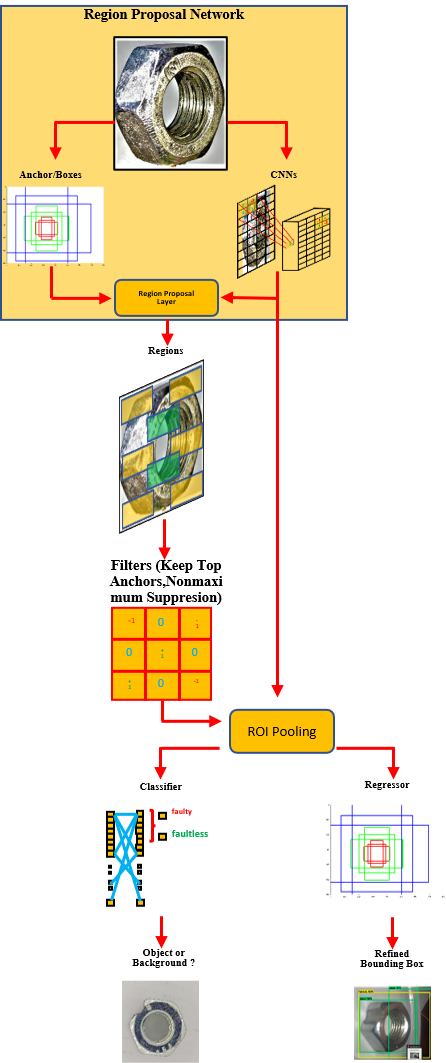
Faster R-CNN is a network that detects objects faster than the under lying RCNN and Fast RCNN. The Faster R-CNN network operates in the following basic step line:

• The image is run via CNN to get a Feature Map,

• Activation Map is run over a separate network called the Regional Recommendation Network, which produces interest boxes/regions,

* Several fully connected layers are used to extract Class + Bounding Box coordinates for interest boxes/regions from RPN.

Figure 3 shows RPN and Faster R-CNN Architecture. The reason Faster R-CNN is fast is because it uses a different network to determine zone recommendations, rather than using a selective search algorithm in the future map to predict region recommendations.



**Figure 3.** RPN and Faster R-CNN Architecture

After input image is passed through CNN and the feature map is mapped, instead of receiving region recommendations with selective search at this stage, these recommendations are made within the network and speed gain is achieved. After the classification process is carried out, 4 different parameters appear. Both the network that suggests the region and the network where normal convolution operations are performed need to be trained.

Here, the RPN has two tasks:

• For each suggestion, there is "object or not?" decide,

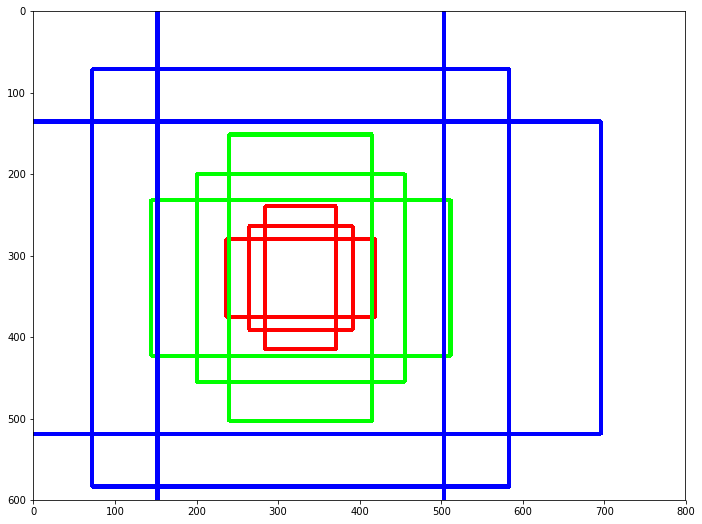
• At the same time, determine the window size of the recommendations.

There are two tasks to be done in our original network:

* "Is there an object or not?" to determine,

• To determine the boundaries of the object it finds (Cebeci,2019).

An n x n bounding sliding window is run on the feature maps created after the convolution operations, and three anchors of three different sizes with the same center are produced.



**Figure 4.**  Anchor Production

Finally, this evolution feeds a smaller network with two tasks, such as 3×3 spatial properties, classification and regression, extracted from feature maps. The output of the regressor determines a bounding box, while the classification subnet has a probability "p" indicating whether the specified box contains an object. Because RPN determines ROI, it is an ideal method for implementing transfer learning (Ferguson et al., 2018). When RPN trains an object detection network on a large dataset with many classes, it learns to identify sub-features of an image that objects are likely to contain. In the work presented in this article, by applying a pre-trained model using the Microsoft Common Objects in Context (COCO) dataset to a faulty metal nut image with transfer learning, RPN immediately detects casting defects between other regions of the image.

**2.2.3 Faster R-CNN and ROI (Region of Interest) Layer**

After RPN, region recommendations of different sizes are obtained. Regions of different sizes mean CNN feature maps of different sizes. Since it is difficult to create an efficient structure by working on features of different sizes, the ROI pool and feature maps are reduced to the same size, simplifying the request and increasing the processing speed. When there are many object suggestions in the framework, the feature map can be applied for each, thus speeding up.

**2.2.4 Faster R-CNN and R-CNN Layer**

Region-based convolutional neural network (R-CNN), which is the last step in Faster R-CNN structure, tries to imitate the final stages of classification of CNNs, in which a fully connected layer is used for classification after receiving object suggestions by processing the extracted feature map with RPN.

R-CNN has two different purposes:

• Classifying suggestions as one of the classes and a background class (to remove false suggestions),

• Better set the suggestion's bounding box to the predicted class.

Recent advances in object detection applications have been made possible by the success of region suggesting methods (RPM) and R-CNN (Rhinderetal., 2016).

3. Literature Review

When conducting the literature review, the applied in the study carried out; studies on computer vision, deep learning, transfer learning, error detection and object detection have been examined.

In this study conducted by Turkoglu et al. (2019), decisional tree, classification and clustering methods of real-life data of cold forging machines used in an industrial company between 2014 and 2017 were applied to data mining techniques such as “yes/no” failure prediction was performed. The results of the tested algorithms are compared with those obtained using the “hypothesis testing” method. In the study, a “Rule Model” was prepared with RapidMiner and the data were used as input in the decision tree model. Data loss was also obtained with the Naive Bayes classification function. The generated model estimation was obtained by induction method and 1442 events from the 2008 record were classified with an accuracy of 71.8%. After working with RapidMiner, IB1 instance-based classifier, KStar Beta Version, Naive Bayes Classifier, Localy weighted learning and J48 pruned tree algorithms were applied to the data using WEKA program and the best result was obtained with J48 algorithm with 77% accuracy. According to the results of the applied Confusion Matrix, it was seen that 28 errors were detected as a result of the application of this algorithm to the data of 2014-2017, and in the light of these results, the company was recommended to use the J48 algorithm with WEKA.

Saurabh (2018), in her study using machine learning, proposed a steel plate defect detection system using different classifiers such as support vector classifier, random forest classifier, gradient boosting, k-nearest neighbors, decision trees and AdaBoost (adaptive boosting). With the application of the Gradient boosting classifier and precise adjustment of hyperparameter values, the desired classification value of dents, attachments, patches, pitted surfaces and scratches with 92.5% accuracy was obtained.

Ferguson et al. (2018) has proposed a kind of automatic quality control application with the ability to simultaneously detect and segmentate defects occurring in metal castings and welding process. This proposed system was developed with a 0.957 mean precision (mAPbbox) with pre-trained ImageNet weights and pre-trained Ms COCO data set as feature extractor using a number of powerful paradigms such as transfer learning, dataset augmentation, and multitasking learning for machine learning.

Campos et al. (2018) performed a computer vision system for several types of metal components used in the automotive industry. They performed quality control on three types of defects such as complete or partial closure in the central hole in perforated metal parts, flattening of bolt teeth or burrs in welding connections. The results obtained by applying different grayscale image analysis algorithms to various defects were examined. Circular Hough Transformation, which first found many circle shapes in the picture, was used for closed hole error, while a bounding box was applied as another method. In order to determine the defective teeth and burrs, the pixels between the two images were compared to successfully determine the faulty area.

Hajizadeh et al. (2016), in their study to detect defective rail surfaces in the Dutch railway network, applied semi-supervised learning, including unlabeled images, in a way that the imbalance in the data could be prevented. used a dataset that also includes other rail sections that also contain other potential defects such as welds, insulated joints, benign defects, minor cracks.

Staar et al. (2019) demonstrated how deep metric learning, which was applied for the first time for surface anomaly detection in the proposed study, can be used. Here, feature extraction occurred in the feature space that CNN learned. Where it differs from similar CNN-based studies is that networks are trained to explicitly learn similarity measures for surface structures using their advances in deep metric learning. Three data sets were used: Kylberg Texture, DAGM and CIFAR100 and classification was performed over 10 classes. Performance has been shown to depend on the surface type. In addition, it was observed that the data set used had a high effect on system performance, and in this context, the samples taken from the CIFAR-100 data set were more performance than the samples taken from the DAGM data set.

Fernandez-Robles and others (2017) examined the inserts in the shaft of the system that processes the poles in the milling machines used for the processing of metal poles of wind towers in the study, and proposed a system that allows them to be classified as "broken” and "not broken". While performing the application, the contrast quality was improved by the adaptive histogram equalization method with contrast limited applied to the images, and edge detection was facilitated, and circular screw shapes were determined with the standard hough tranform method.

Song et al. (2018) proposed a technique based on a deep convolutional neural network to detect defects such as damage, dirt, and crushing on the surface of metal screws, and according to the results of the application performed with the CNN model trained with screw images, it was found to provide 98% detection accuracy. According to the results obtained, the superiority of the proposed SYSTEM over traditional machine vision-based techniques is seen.

**4. Materials and Methods**

In this section, a system that performs error detection and classification on metal nuts containing anomaly with deep learning is proposed. TensorFlow library's successful pre-trained models in the Object Detection API have been retrained with the data set we have prepared using the transfer learning method.

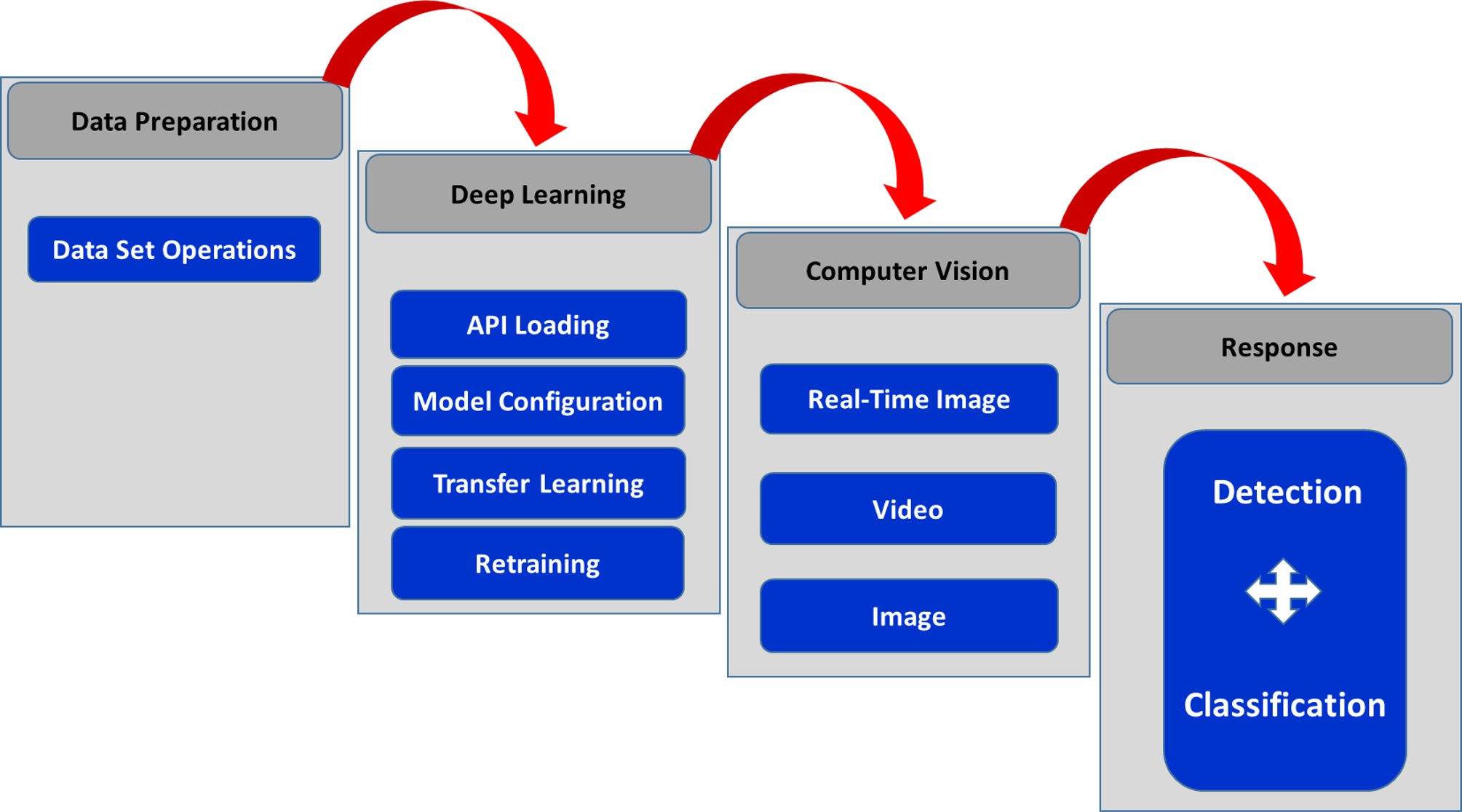
**4.1 Tools Used in Education**

The training of our deep learning model, which requires the processing power of they, was carried out quickly through the GPU offered by Google with Collaboratory, a free Jupyter laptop environment that it provides as a cloud service. The most important feature that distinguishes Colab from other cloud services that offer free service; Gpu usage is free of charge. With the Tesla K80 GPU it serves, it's fast for training the model in deep learning applications. The tensorboard platform, which can also be operated on Google Colab, was used to visualize the data obtained during the training phase. Through tensorboard, it is decided whether the training is successful by following the loss values of the training. With tensorboard, training parameters, metrics, hyper parameters or any statistics of the neural network can be visualized. In this way, the possibility of direct immediate intervention to the model can be obtained with instant follow-up. During learning, it is seen how the effects of neural network weight updates on neural networks are compared with weight loadings. The training process was completed by performing transfer learning through the Faster-RCNN-Inception-V2 model, which is trained with the Tensorflow Object Detection API, which enables the use of pre-trained object detection models using the transfer learning method, or the MS COCO data set used for object detection, which enables the creation and training of new models.

**4.2 Method Used in Education**

The model specified in Figure 5 was used when developing the application.

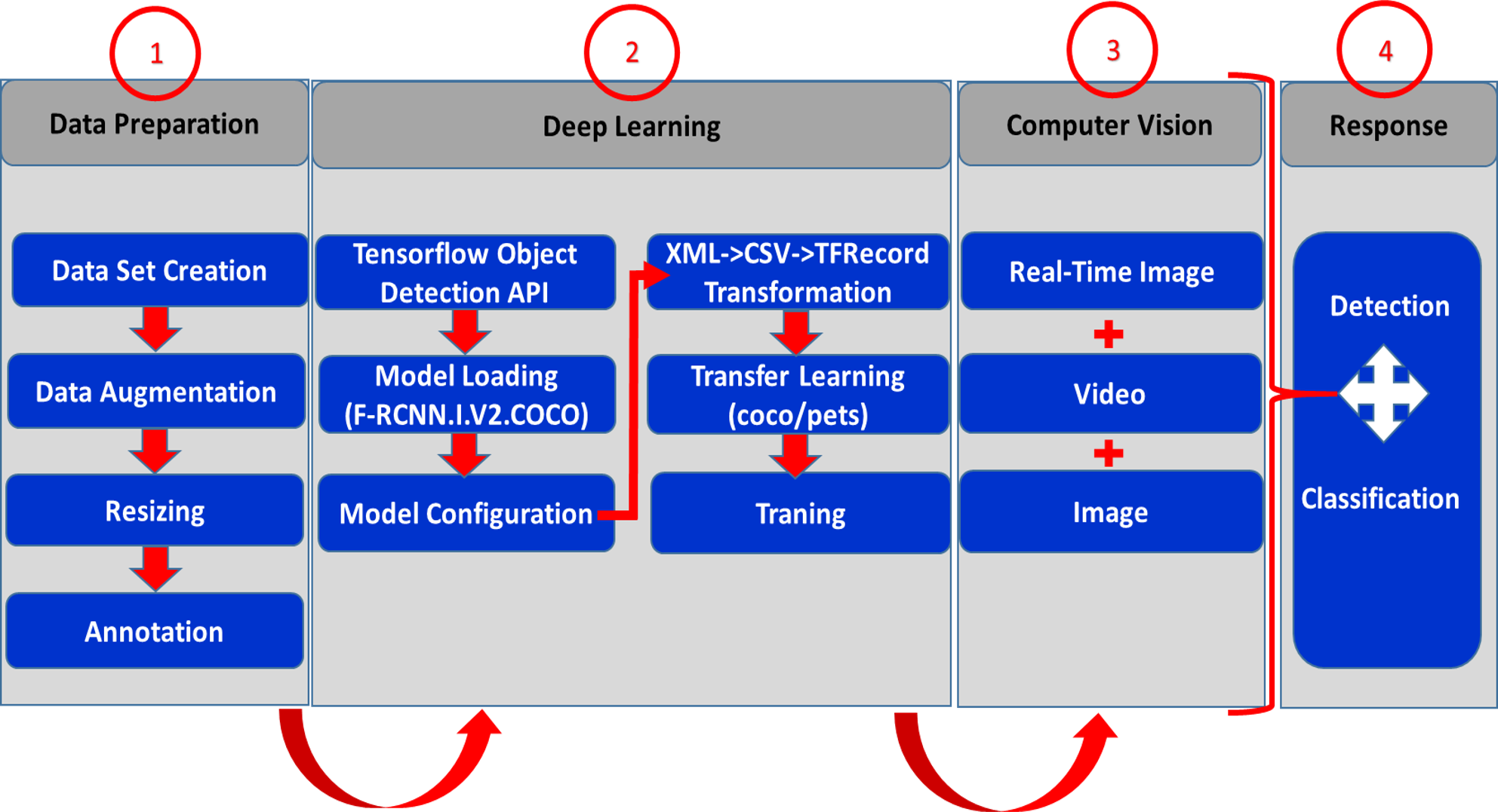
The proposed system works like a picture localization and object detection system with anomaly detection and classification processes on metal nuts.



**Figure 5.** Application Development Model

Since the Faster-RCNN model has higher accuracy data compared to similar CNN models, the transfer learning method was applied using the Inception-V2 model, which was trained with the Microsoft COCO dataset, in the training process within the scope of our study.

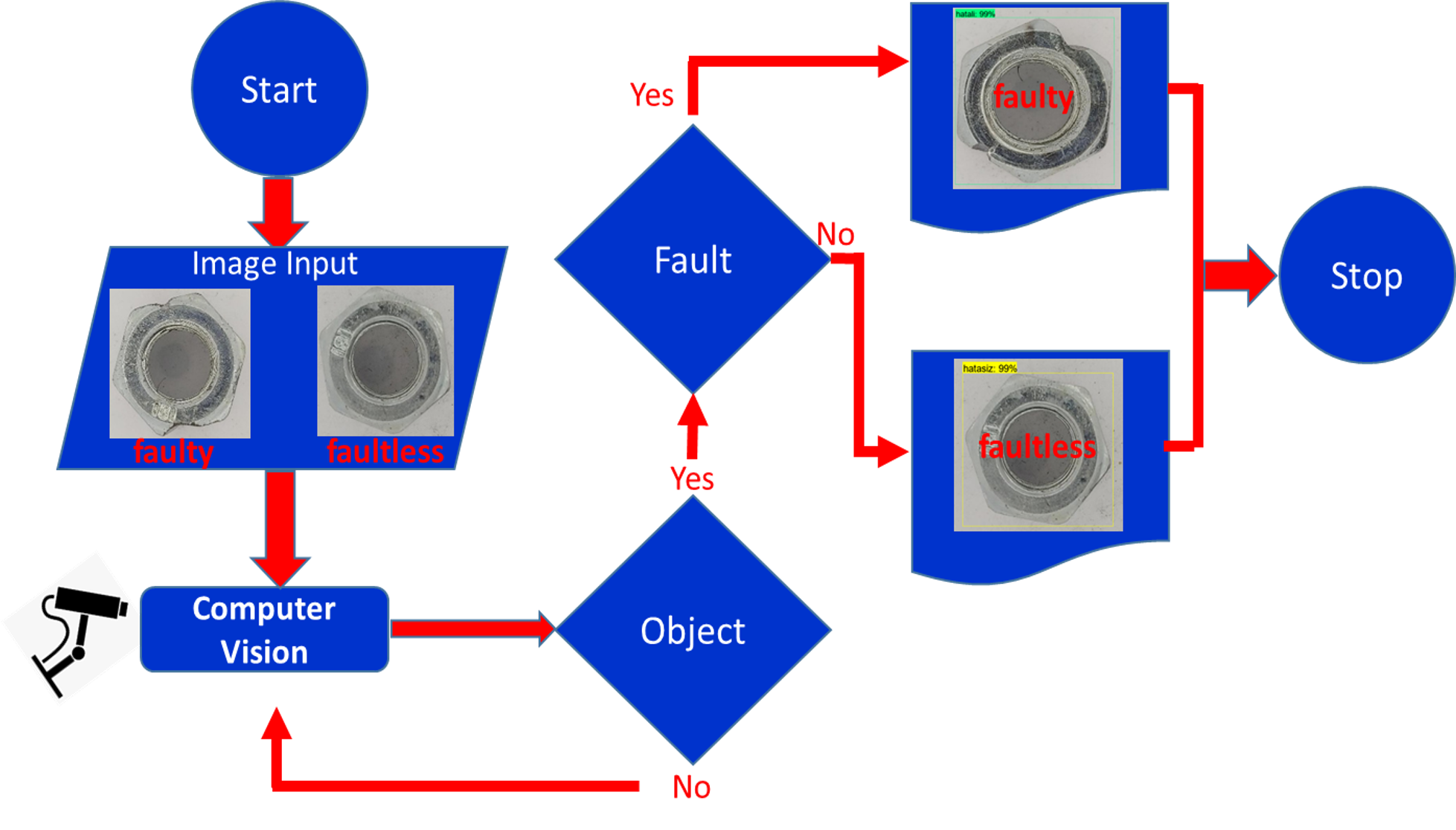
The workflow specified in Figure 6 has been applied to perform the error detection and classification of metal nuts containing anomaly.



**Figure 6.** Error Detection and Classification Application Development Workflow

In the test phase and use of the application, the algorithm given in Figure 7 was applied.

**Figure 7.** Error Detection and Classification Algorithm



In the Data Preparation Phase,a unique data set has been prepared as there is no ready-made data suitable for our study. Data increment was carried out to increase the number of data in the data set and to the success of our model. After this process, with the completion of the data set, the metal nuts were labeled as “faulty" and "faultless". While creating the dataset, images with a resolution of 3492x3492 px were taken at a ratio of 1:1, using a 12 Mp camera resolution LG G7 ThinQ smartphone to obtain suitable images with faulty and error-free nuts, and only from the top and side angles, which were considered suitable for our object detection application. In order to increase the number of metal nut images, our original data set consisting of 2000 images was prepared by applying data augmentation to error-free metal nut images. 2000 photographs in our dataset were divided into two groups as 400 tests and 1600 trains.

In order to increase the deep learning quality of the neural network, data augmentation was performed by making changes on the images belonging to the same class in the training data set. Image data augmentation is usually applied to the training dataset, not the validation or test dataset. Data augmentation differs from data preparation such as resizing and pixel scaling in that it must be applied consistently across all datasets (Brownlee, 2019). By editing the imagesizes*,* the image dimensions are reduced to 600 x 600 px to prevent data loss while filtering between layers of the images that our model will use as inputs and to shorten the lengthy training time when training pictures of  
3492 x 3492 px size.

In order to label the images, two classes have been created as “faulty” and “faultless” for the system we have implemented to detect and classify metal nuts. In order for the detection and classification process to be carried out successfully, our model needs to be trained to learn this. This is possible by labeling the wrong and error-free metal nut images correctly and giving them to the model. Tagging was performed with the LabelImg graphic image tagging tool, and the label information for the meat-enclosed images was saved as .xml files in pascal voc format used by ImageNet.

During the deep learning phase, the Installation and configuration of the Faster-RCNN-Inception-V2 model, which is trained with the MS COCO dataset, where we will perform transfer learning for this API, isperformed by converting the label information that occurs after the labeling process into tfrecord files that will understand the tensorflow, and then the training of the model with the original data set.

Tensorflow helps us create neural network models such as CNN to automatically detect images. In this context, Tensorflow presents two approaches to image recognition;

•Classification: CNN is trained to recognize categories of objects such as cats, dogs, cars, etc. The trained system classifies the image as a whole according to these categories.

•Object Detection: stronger than classification and detects many objects within the same image. It also tags the objects it detects and shows the locations of the objects on the image.

In this study, both steps were applied and fault detection and classification of metal nuts were carried out.

Our training was carried out until step 28.850 as stated in Figure 8.

Since the MS COCO model is trained, the loss value should fall steadily to 0.05 and gold.

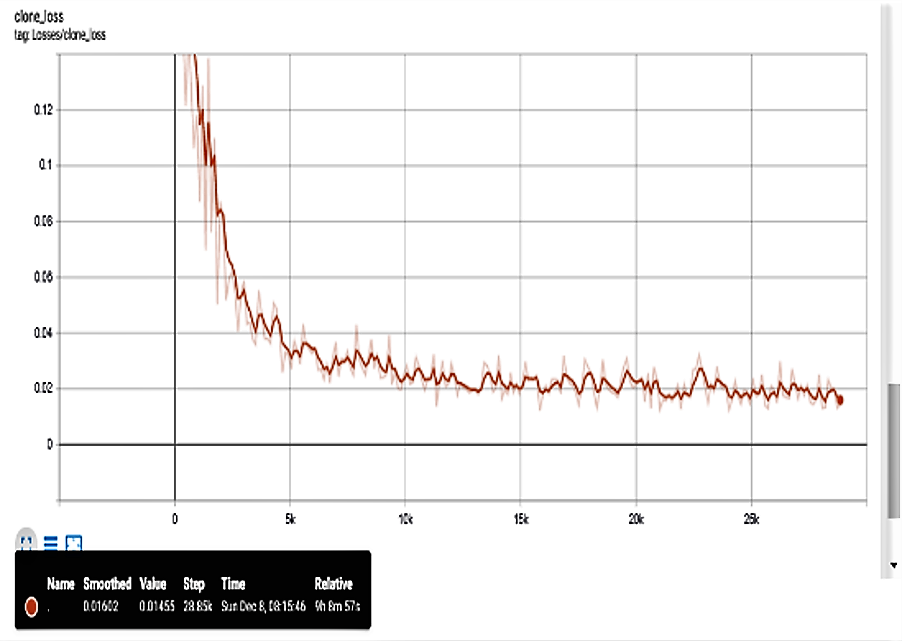


Figure 8. Loss Graphic

After we trained 28,850 steps, our loss value started to be 0.01. Since the loss value is within the framework we want, our training was terminated in step 28,850 and frozen and ready for use in order to prevent overfitting of the model.

During the computer vision phase, our modules were operated to detect errors and class-nect on image and video files or on patterns obtained from real-time camera.

In the response stage, our application has succeeded in classifying the metal nuts it sees as "faulty" or "faultless".

As can be seen from the pictures given in Figure 9 below, it is understood that there is only one metal nut with the system performed and that the detection and classification is done correctly up to 99% in the images taken from angles suitable for the data set under appropriate ambient conditions.

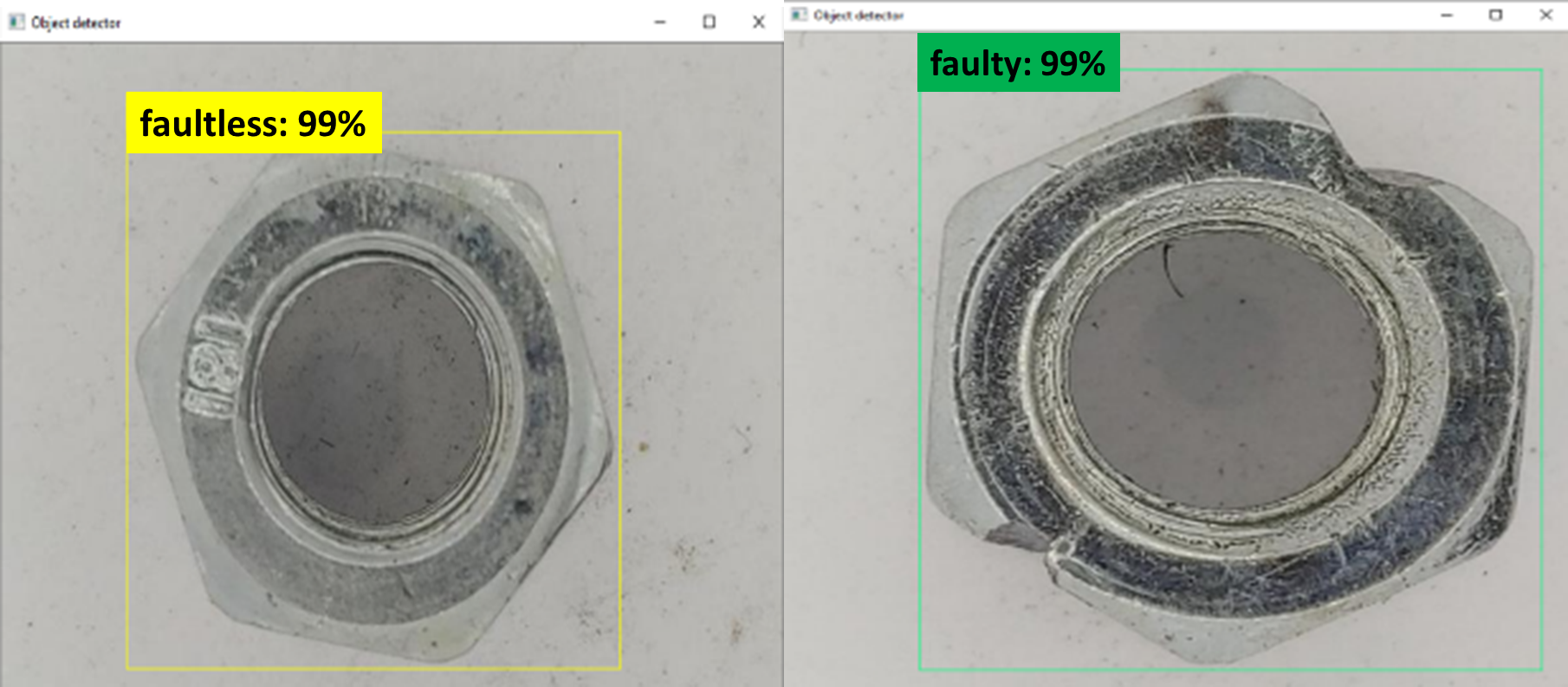


Figure 9. Object Detection and Classification

As shown in Figure 10, it is understood that the correct detection and classification rate of the nuts at the appropriate angles to the data set is 90% on the image with more than one object from different angles.

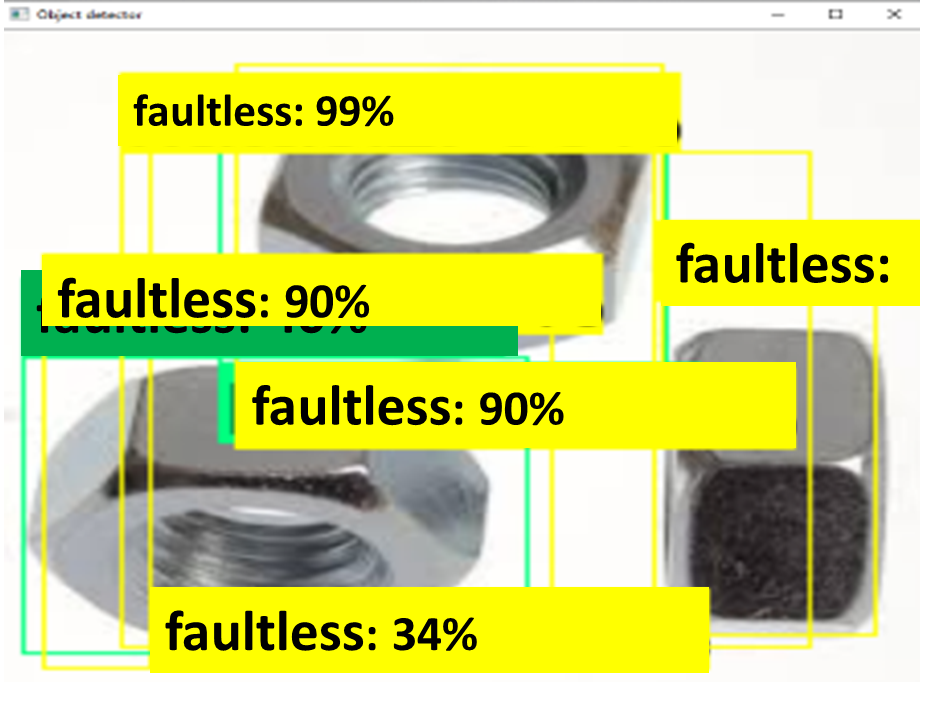


Figure 10. Multi-Object Detection and Classification

According to the data set, it was determined that there were no problems with object detection in images from incompatible angles, but errors occurred in the classification process.

Since all of the images in our data set are created with images taken from only two angles, from the top and side, the detection and classification accuracy of metal nut images taken from the upper and side angles is up to 99% in the error detection and classification application.

Apart from the images at the upper and side angles in the dataset, RPN (region proposal network) tries to detect and localize the images from different angles by creating bounding boxes. Therefore, it is seen that “faulty” or “faultless” labeling occurs in different areas of the metal nuts in the picture. A similar process occurred on real-time images of nut images taken via webcam. In the images of the upright or side metal nut, it is understood that error detection and classification with an accuracy in the range of 90% to 99% if the lighting is sufficient, as in Figure 11, but the accuracy of the system decreases in cases in Figure 12, where the light is insufficient, and incorrect classifications occur due to insufficient lighting, especially in metal nut images from side angles.

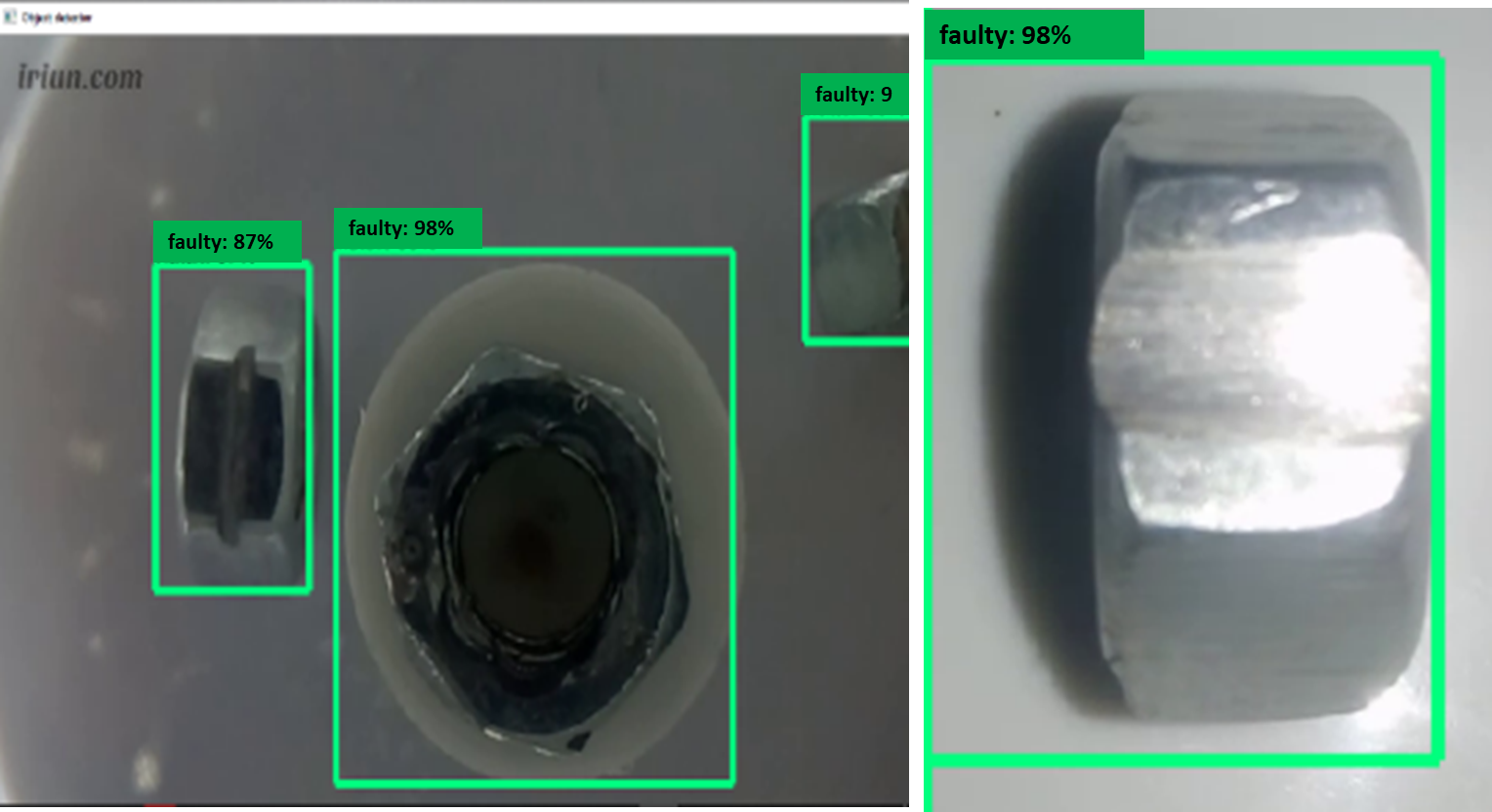


Figure 11. Detection and Classification Under Appropriate Environmental Conditions

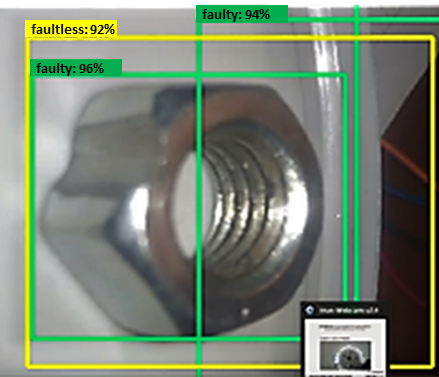


Figure 12. Detection and Classification in Unsuitable Environmental Conditions

In addition, when the background color of the same metal nuts changes, it is determined that bounding boxes begin to appear in different parts of the image.

5. Conclusion

With the implementation carried out, it is thought that promising results have been obtained for the solutions that the proposed system will provide in the industrial field in accordance with the objectives specified at the beginning of the study.

From an operational point of view, it has been seen that an intelligent automation process is achieved through computer vision with the application and will play an active role in determining improvements by contributing to the production process.

In the trials, it was seen that the posture angles of the metal nuts in the images obtained from the upper and side sides were important. Images taken at angles that were not suitable for the data set showed that the region recommendation network created multiple bounding boxes due to the desire to create a bounding box, and success in the classification process decreased.

In the detection and error classification processes carried out with the trainings applied to the data set consisting of photos taken from the upper and side angles of metal nut images with deep learning method and the images obtained in real time via webcam or taken from test pictures on the local computer, the process performed at the appropriate lighting and angles to the data set has yielded successful results with detection and classification accuracy between 90%-99%.

Today, it is clear that it is inevitable that computer vision applications will be integrated into production and control processes, as the need and interest in deep learning and computer vision solutions continues to increase.

The limitation of the method we used in our study is that the data set of a deep network training requires manually labeled data and this causes time increase and cost increase. The developed system is currently in the prototype phase. In order for the contribution of the application developed within the scope of the thesis to the industrial field to be continuous, in future studies,

* Increasing the number of data in the data set,
* Implementation of data increment on all data,
* Classification according to error types using mask RCNN,
* To provide the advantage of accurate detection and classification at any angle by getting rid of the angular appearance difficulties required by Faster RCNN using capsule networks,
* Making the application workable on mini-computers such as the Raspberry Pi,
* Using simultaneous images obtained through multiple cameras,
* Applying a self-learning design,
* Significant improvements, such as the integration of software that provides warnings and recommendations to the user, can be achieved by improving performance rate and improving system stability and availability.

References

Ashton, K. (2009). That ‘Internet of Things’ Thing, RFID Journal*.*  https://www.rfidjournal.com/articles/pdf?4986: 1, (03/07/2021).

Gonzalez, R.C., Woods, R.E. (1992). *Digital Image Processing*. New Jersey: Prentice Hall.

Russ, J.C. (1995). *The Image Processing Handbook 2dh edition.* Florida: CRC Press.

Suetens, P., Fua, P., Hanson, A.J. (1992). Computational Strategies for Object Recognition. *ACM Computing Surveys*. 24(1): 5-61.

Alpaydin, E. (2011). *Artificial Learning*. Istanbul: Bosphorus University Publishing House.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep Learning. *Nature.*  521(2015): 436-444.

Aminanto, M.E. and Kim, K. (2016). Deep Learning in Intrusion Detection System: An Overview. *Computer Science.* 2016: 4-6.

Deng, L. (2014). A Tutorial Survey Of Architectures, Algorithms and Applications for Deep Learning. *APSIPA Transactions on Signal and Information Processing.* 3(2014): 3-8.

Stoica, I. (2019). Solving System Problems with Machine Learning. *Studies in Informatics and Control.* 28(2): 119-132.

Deng, L., Yu, D.and diğerleri. (2014). Deep Learning: Methods and Applications. *Foundations and Trends R in Signal Processing*. 7(3–4): 197–387.

Mane, D. T.and Kulkarni, U. V. (2019). A Survey on Supervised Convolutional Neural Network and Its Major Applications. *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications* (pp. 1058-1071). United State of America: IGI Global. Russakovsky, 2015

Voulodimos, A., Doulamis, N., Doulamis A. and Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience.* 2018: 1-13

Manaswi, N.K. (2018). Convolutional Neural Networks. *Deep Learning with Applications Using Python* (pp. 91-96). California: Apress.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (pp. 1097-1105), Organizer: Neural Information Processing Systems Foundation Inc., Lake Tahoe, United State of America. 3-8 December 2012.

Cebeci, H.I. (2019). Face and Object Recognition | with Computer Vision R-CNN, SSD, GANs. https://www.udemy.com/course/bilgisayar-gorusu/, (16/10/2019).

Ferguson, M., Ak, R., Lee, Y.and Law, K.H. (2018). Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning.  *Smart and Sustainable Manufacturing Systems.*  2(1): 137-164.

Ren, S., He, K., Girshick, R.and Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 39: 117-1149.

Saurabh, G. (2018). Use Machine Learning to Detect Defects on the Steel Surface. https://software.intel.com/en-us/articles/use-machine-learning-to-detect-defects-on-the-steel-surface, (24/07/2021).

Campos, M., Martins, T., Ferreira, M.and Santos, C. (2008). Detection of Defects in Automotive Metal Components Through Computer Vision. IEEE International Symposium on Industrial Electronics (pp. 860-865), Organizer: The Institute of Electrical and Electronics Engineers (IEEE) Cambridge, United Kingdom. 30 June-2 July 2008.

Hajizadeh, S., Nunez, A., Tax, D.M.J. (2016). Semi-supervised Rail Defect Detection from Imbalanced Image Data. IFAC (International Federation of Automatic Control), 49(3): 78-83.

Xian, T., Dapeng, Z., Wenzhi M., Xilong L., De, X. (2018). Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks. *Applied Sciences.* 8(9):1575.

Staar, B., Lütjen, M., Freitag, M. (2019). Anomaly Detection with Convolutional Neural Networks for İndustrial Surface Inspection. 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering (pp. 484-489), Organizer: CIRP, Gulf of Naples, Italy. 18-20 July 2018.

Fernandez-Robles, L., Azzopardi, G., Alegrea, E., Petkov, N., (2017). Machine-vision-based identification of broken inserts in edge profile milling heads. *Robotics and Computer–Integrated Manufacturing.*  44: 276-283.