CONTAINER DAMAGE DETECTION AND CLASSIFICATION USING CONTAINER IMAGES

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Zeynep İMAMOĞLU

June 2019
İZMİR

We approve the thesis of **Zeynep İMAMOĞLU**

Examining Committee Members:

[illegible]

Assoc. Prof. Dr. Tuğkan TUĞLULAR
Department of Computer Engineering, İzmir Institute of Technology

[illegible]

Assoc. Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institute of Technology

[illegible]

Assoc. Prof. Dr. Derya BİRANT
Department of Computer Engineering, Dokuz Eylül University

[illegible]

Asst. Prof. Dr. Nesli ERDOĞMUŞ
Department of Computer Engineering, İzmir Institute of Technology

[illegible]

Prof. Dr. Oğuz DİKENELLİ
Department of Computer Engineering, Ege University

27 June 2019

[illegible]

Assoc. Prof. Dr. Tuğkan TUĞLULAR
Supervisor, Department of
Computer Engineering
İzmir Institute of Technology

[illegible]

Assoc. Prof. Dr. Yalın BAŞTANLAR
Co-supervisor, Department of
Computer Engineering
İzmir Institute of Technology

[illegible]

Assoc. Prof. Dr. Tolga AYAV
Head of the Department of
Computer Engineering

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

First of all, I would like to thank my Supervisor, Assoc. Prof. Dr. Tuğkan Tuğlular and Assoc. Prof. Dr. Yalın Baştanlar , who advised and encouraged me all the way during this thesis study. I am thankful for their assistance, patience, continuous support and for sharing their knowledge with me. Their guidance helped me in all the time of research and writing of this thesis.

I would like to thank company BIMAR and ARKAS LOGISTICS for providing the container image data.

I am grateful to my friends for all the support and their motivation during busy and tiring times.

Finally, I would like to express my infinite gratitude to my parents and my husband Hikmet Efe Imamoğlu for their unconditional love and endless support without any expectations. I dedicate this thesis work to them.

ABSTRACT

CONTAINER DAMAGE DETECTION AND CLASSIFICATION USING CONTAINER IMAGES

In the logistics sector, digital transformation is of great importance in terms of competition. In the present case, container warehouse entry / exit operations are carried out manually by the logistics personnel including container damage detection. During container warehouse entry / exit process, the process of detecting damaged containers is carried out by the personnel and several minutes are required to upload to the system. The aim of this thesis is to automate detection of damaged containers. This way, the mistakes made by the personnel in this stage will be eliminated and the process will be accelerated.

In this thesis, we propose a machine learning method which detects damaged containers using the container images to perform statistical damaged / undamaged estimation. We modeled the problem as a binary classification problem, which considers a container as damaged or undamaged. The result obtained from the undertaken studies shows that there is no single best method for visual classification. It is shown how the dataset was created and how the parameters used in the layered structure impact the most suitable model could be created for this study.

ÖZET

KONTEYNER GÖRÜNTÜLERİNİ KULLANARAK HASAR TESPİTİ VE SINIFLANDIRMASI

Lojistik sektöründe, dijital dönüşüm rekabet açısından büyük önem taşımaktadır. Mevcut durumda konteyner depo giriş / çıkış işlemleri sırasında konteyner hasar tespiti lojistik personeli tarafından elle yürütülen bir süreçlerdir. Konteyner depo giriş / çıkış işlemi sırasında, hasarlı konteynerleri tespit etme işlemi lojistik personeli tarafından gerçekleştirilir ve sisteme yüklenmesi için zaman gereklidir. Bu tezin amacı hasarlı konteynerlerin tespitinin otomatik hale getirilmesidir. Bu sayede lojistik personelinin bu aşamada yaptığı hatalar ortadan kalkacak ve süreç hızlandırılacaktır.

Bu tez çalışmasında, istatistiksel hasarlı / hasarsız tahmini yapmak için konteyner görüntülerini kullanarak hasarlı konteynerleri tespit eden bir makine öğrenme yöntemi önermekteyiz. Farklı yöntemler kullanarak tahminleme için en uygun yaklaşımı oluşturup hasarlı veya hasarsız olarak kabul edilen ikili bir sınıflandırma problem olarak modelledik. Yapılan çalışmalardan elde edilen sonuç, görsel sınıflandırma için tek bir en iyi yöntem olmadığını göstermektedir. Veri setinin nasıl oluşturulduğu ve katmanlı yapılarda kullanılan parametrelerin bu çalışma için en uygun modeli nasıl etkilediği gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
ReLu	Rectified Linear Unit
FC	Fully Connected
CNTK	Microsoft Cognitive Tool Kit

CHAPTER 1

INTRODUCTION

Machine learning is a science that deals with the design and processes of computer algorithms that model a problem according to the data [14]. The shortest expression of machine learning is to find patterns in the data. Then it uses these patterns to predict. Let's consider virtual frauds, for example. Based on past fraud data, potential patterns or patterns are found, and the process can be judged to be fraudulent or actual. There are many examples of machine learning in different fields and for different purposes.

Machine learning can be simulated by personal learning. For example, the person (he or she) learns to read. First, the letters or curves that make up the letters, then letters and letter patterns together with words have been defined in our minds. And when we see these patterns again, recognition of what we see happens. The more data you have, the more accurate the result. When you have so much data that manpower cannot afford, software is needed. In this case machine learning is required.

Recently, machine learning is one of the popular topics. There are several reasons for this. Huge amount of data makes the machine learning more accurate. More data is needed to produce more accurate estimates. Nowadays we live in the data age. In order to use the data correctly and to make sense of this data, various studies have been carried out. Many years have been spent on how the data will be used by the researchers. But so much data density also required computer power. Considering that we live in the cloud age, we have this power.

Machine learning starts with data containing patterns. With this data, the machine learning algorithm is fed and a model is generated as a result of the algorithm. This model can recognize patterns when a new data is presented. The applications then use this model to see if they match the known patterns when a new data is presented.

1.1. Motivation

Considering the increasing number of visual data (photos, videos), the recognition and interpretation of the objects in this image and creating a meaningful output have created a great awareness and created a new need in the sectoral and academic circles. In this way, thousands of visual data with a large set of education, to make the image meaningful, especially the application of methods such as classification and object recognition helps. When the studies and systems are examined, the success rates are remarkable. To achieve this level of success and minimize the margin of error, the size of the dataset should be as large as possible.

There is an IT system in which the technology user, who is working in the logistics company, is currently loading container photos. The process followed by the user in this system involves the visual inspection of incoming containers by an expert person and the photographing of the damaged parts by a photographer if they are found damaged. As soon as the container enters from the door of the logistics company depot to determine whether the container is damaged or undamaged, it will accelerate the process and will reduce the work accidents that will occur during the detection of damage. In addition, it will reduce the error in the damage detection reports entered by the employees manually and decrease the discrepancies in the container panel of damaged / undamaged.

1.2. Thesis Goals and Contributions

This thesis aims to accelerate the container depot entering processes by making the distinction between damaged or undamaged containers. We employed a binary classification model to distinguish damaged and undamaged containers from each other. We analyzed the machine learning and deep learning approaches for visual classification. To estimate the damaged or undamaged class to which the container belongs, we evaluated several versions of convolutional neural networks with varying methods, structures and hyperparameters.

1.3. Outline of Thesis

This thesis is arranged as follows. In Chapter 2, a literature survey has been carried out and an overview is presented. Chapter 3 provides theoretical information about machine learning, deep learning, deep learning layers and their applications. As described in Section 4, we present our dataset including Container Photographs and the experiments we perform in Section 5. The final chapter, Chapter 6, discusses the final explanations and future studies.

CHAPTER 2

RELATED WORK

Artificial intelligence, machine learning, deep learning are one of the popular research areas. There are many approaches and experimental studies for these methods. There is more than one method specific to the problem and problem within the field of deep learning. deep learning modeling is done by methods such as Convolutional Neural Network, Reinforcement Learning and Long Short Term Memory. In 1998, the method of convolutional neural network used by Yann LeCun et al. [7] can be applied to problems such as classification, object recognition, tracking, still transfer.

Deeper networks have been established that will perform better as the computer power used increases and volume of the data grows. After the study in 2007, the use of the term 'Deep Learning' has become widespread. In 2006, Geoffrey Hinton has shown that deep neural networks can be effectively trained by greed-layer pre-training [25]. For the first time in 2012, deep learning had a major impact on the world of science. Biggest Visual Recognition Competition (ImageNet) (Competition 2012), the largest competition in the field of object identification, was won by the Convolutional Neural Network [1], which was considered the basic architecture of deep learning. This has been an incredible rise of deep learning. Artificial intelligence, machine learning, deep learning issues are published with various scientific articles or developed by the university methods, optimization algorithms, regularization methods, such as libraries containing approaches. Examples include libraries such as Keras, Tensorflow, Caffe, CNTK (Microsoft Cognitive Tool Kit). These libraries offer an easy-to-use development interface, as well as enhanced GPU support, a fast-growing structure, and multiple advantages with previously trained model content. CNTK is a powerful computational-graph-based Microsoft's deep learning tool to train deep neural networks. Some Microsoft products use, for example, Cortana speech models and web ranking CNTK. CNTK is scaled to multiple GPU servers and is designed for efficiency [32].

Artificial Intelligence and deep learning methods can be applied to problems with different algorithms and different solutions in different fields. The adoption of data-intensive machine-learning methods can be found throughout science, technology and commerce, leading to more evidence-based decision-making across many walks of life, including health care, manufacturing, education, financial modeling, policing, and marketing [9]. Experimental studies of different groups are available in different sectors. For example, in a study in the health sector, a solution is proposed for deep learning methods for the determination of skin cancer (Melanoma). With this solution, automated systems which can determine the disease at the point where human expertise is limited, can save lives, prevent unnecessary interventions to the patient and reduce the cost [6]. At the same time, financial and mobile banking transactions such as fraud, portfolio management can be done by machine learning method. Converting a normal text to handwriting was done with deep learning [28]. Using the deep learning method, it is decided whether the input is an eye or not and the eye is detected through the visual [29].

In the last few years, the use of deep learning models in the industry has been greatly increasing. In this context, there is a great potential for the automotive industry. Driverless vehicles, automatic parking systems, security systems are solved as deep learning problems. Grime methods that interact with machines are redefined with deep learning. Although Artificial Neural Networks have been studied since the 1950s, the work gained momentum with the machine power gained in the last few years. In the last 10 years, deep learning has become more mature with systems such as the automotive sector and voice assistants [33]. Another study in the automotive field is car damage classification. The issue of visual classification is an important development for vehicle owners and insurance companies. With the solution developed based on deep learning, damage classification is done in some vehicles. At the beginning of this study, it was observed that a small data set did not yield good results with CNN training and its effect was investigated by fine-tuning. It is observed that more successful results are obtained with fine-tuning process. A large amount of waste of money is encountered in the car insurance industry as a result of incorrect assessment of damage. There are delays in requests through visual inspections. To avoid these delays and reduce money loss, insurance companies invest in deep learning methods. The samples taken from the vehicle regions and the damage types are taken into consideration and the dataset is created. Since there is no public dataset for this problem, preparing the dataset is a difficult step in solving the problem. In this study, it was found that the most successful result was obtained when transfer learning process was applied by applying fine-tuning process to the dataset obtained [34]. In another study aimed at reducing insurance costs after traffic accidents, vehicle damage is tried to be predicted. It is desired to create a model in which the damaged vehicle photos taken from the mobile phone will be uploaded to the system and insurance compensation transactions will be performed automatically. However, when it is desired to create a dataset on the basis of problems, different problems such as accident scene, ambient lighting, another object outside the target and reflection for metallic vehicles are evaluated. For this reason, instead of standard approaches, they proposed a method to record scratch damage on the vehicle, recording a 3D CAD model (basic fact) of the undamaged vehicle on the image of the damaged vehicle. In this study, 3D CAD models are used in order to obtain the undamaged condition of the lightly damaged vehicle in the photograph [35].

There are also studies for the detection of damages after the earthquake. The resulting high resolution satellite images are an important data source for disaster management. Using this data source, there are methods for automatic detection and classification of buildings after earthquake. It is provided to automatically detect the position information of buildings before and after the images. [36].

Another study is Structural Health Monitoring (SHM). The SHM usually focuses on the yes / no damage detection and estimate of the damage size. In terms of health, it is very important to understand the characterization of damage, the severity of damage and the risk of damage growth. A formula and practice in the field of health is recommended by taking into account the recent studies in the field of deep learning, machine learning and image recognition. With the study conducted in the article, a method for recognizing the cracked damage on the composite material is proposed. The established multi-layer neural network ensures high accuracy [37].

CNN performs well in classification problems involving large-scale visuals. While learning CNNs, millions of parameters are estimated and a large number of classified data samples are required. This is an obstacle for problems with limited educational data. In the study, it is stated that limited educational data can be transferred to the information learned from the large scale data sets previously performed. A method is designed in which the trained layers in the ImageNet dataset can be reused for another problem. Although there are different data sets and different problems, the transfer of the acquired information yields significantly successful results [38]. The transferable property representation learned by CNN minimizes the over-fitting effect of small volume tagged datasets [39].

The problem covered by this thesis is a problem of the logistics sector. Deep learning method is used in different problems related to this sector and helps to create a competitive environment in the sector. Deep learning methods are used to reduce costs and increase efficiency and make faster decisions in problems such as planning, customer segmentation and potential forecasting.

CHAPTER 3

BACKGROUND ON DEEP LEARNING

In this section, ‘Machine Learning’, ‘Machine Learning Methods’ and ‘Deep Learning’ and ‘Convolutional Neural Network’ and concepts are introduced.

3.1. Machine Learning

Machine learning is a subset of the artificial intelligence used in today's highly popular and popular areas. In other words, all machine learning problem can be evaluated as artificial intelligence problem but all artificial intelligence problems are not considered as machine learning problem. Machine learning is dynamic, which is capable of changing itself when more data is obtained. No human intervention is required to make changes. In a sense, machine learning adapts to themselves in response to the data they are exposed to. Basically, it functions as learning and self-development.

Machine learning is a process that is difficult to learn. A high amount of complex data is studied, meaningful and predictable patterns are tried to be extracted from this data. Understanding machine learning is the understanding of machine learning processes, and these processes are in fact encouraged. Everything is done over and over again in small steps so it should be repeated. Success, which is the result of self-sacrificing efforts and time spent, is generally useful. Repetition of steps shows the solution way or solves problems. But it may not always be circulated with the right results. The most important point to be encountered here is to ask the right question. When selecting the right question, it is necessary to have the right data. For example, the assessment of whether the transaction by credit card is fraudulent or that the customer is a landlord or tenant can be an important information. Finally, it will be a model to make predictions against the new data. Determining the success criterion for predictions may take place among the topics that may be considered before starting.

There are two important languages for machine learning. These are R and Python. R is an open source programming language and supports more with machine learning.

Python also offers open-source technology for machine learning and is increasingly popular.

3.1.1. Machine Learning Methods

Machine learning algorithms are generally classified by three methods.

- **Supervised Learning**

In this method, algorithms estimate based on the learned data. In other words, the categories, labels or classes of data and data to be used during the training are already known. The system developed in accordance with this known knowledge learns by making meaningful molds from the data and predicts the incoming data with the patterns that it has learned before. The most time consuming step here is to prepare the data. The more accurate the dataset is, the better the system will learn and estimate correctly.

Supervised Learning is generally used to solve ‘Classification Problems’ and ‘Regression Problems’. For the classification problem, the dataset of the problem is divided into groups according to the determined categories. With the dataset allocated to the groups, the system is trained and classifies new incoming data in a similar way as it learns. Classification method can be categorized as binary classification, multiclass classification. Naive Bayes, K-Nearest Neighbors, Decision Tree, Support Vector Machine algorithms can be used to apply classification problems. For example, in the Naive Bayes algorithm, the probability data in the training data is performed while the data in the K-Nearest Neighbors algorithm is the closest to which data.

As with the method of classification, regression problems are also a Supervised method. The structure is similar to the classification method, but the estimation result is not a category but a numerical value.

- **Unsupervised Learning**

In contrast to Supervised Learning, we do not have a dataset that is labeled in this method. By working on the data, the data between the data are captured and data close to each other are clustered. As a result of this process, the new incoming data is assigned to that cluster if it is closest to which clusters are formed as a result of the algorithms.

This method is used for clustering problems. For the clustering problems, each data in the dataset is divided into groups according to their similarity values. Each group refers to a set and data in clusters are similar. Data that do not show similarity should be included in different groups. Thus, similarity between clusters is minimized. A detailed summary of supervised and unsupervised learning can also be found in [31].

- **Reinforcement Learning**

The basis of this method is based on behavioral psychology. Learning process takes place with the feedback received. The result of the system is evaluated as true or false. If the result is the target, the reward signal is taken, if not, the penalty signal is taken. The system does not repeat this move once the penalty signal is received. Learning the system continues by taking advantage of the step taken as a reward. Provides continuity of learning by always aiming for more awards.

For example, the developed artificial intelligence AlphaGo had defeated world champion Ke Jie in the game in 2017. There was Reinforcement Learning behind AlphaGo's logic of learning [12].

3.2. Deep Learning

Deep learning is a subfield of machine learning. Although deep learning is a new field, it has been an important artificial intelligence subtitle with achievements in different fields. With the many data processed behind deep learning, very successful results are observed.

Deep learning uses neural network architecture. Artificial neural networks are composed of many layers. So, the depth is directly proportional to the number of layers. In a classical network, the neural network consists of 2-3 layers and in deep learning this number can be up to 150. With the learning process that requires GPU computing power, the data is processed in these layers. Nowadays, artificial intelligence is used in areas such as object classification, object finding, face recognition, voice recognition [27].

3.2.1. Deep Learning Methods

Convolutional Neural Network (CNN) is an artificial neural network that seems to blend biology and computer concepts. It is a preferred and highly successful mechanism for image recognition actions.

Unique features that make an image into that image are used to recognize images in Convolutional Neural Network. It works in a similar way to the functioning of the human brain. In our brain, we can define an object that unconsciously distinguishes images. Similarly, a similar discrimination is performed in the CNN. Distinguishes an object or image from its properties and defines it in an abstract manner.

When running the CNN algorithm, there are more than one factor that can be changed or tested if the expected result is not obtained or if the received values do not meet the system expectations. This is one of the difficulties encountered. There is no clear view of what should be set in order to obtain the best result in effective machine learning. The first step in this process is to make sure that the algorithm works well with the training data. Therefore, the outcome of the training data should be evaluated in terms of admissibility. For some systems this expectation may be close to human perception, but this is a variable depending on the application.

After obtaining a good result with the training dataset, it is expected to obtain a good result on the verification data serial and test dataset. After the results of all datasets, the cost function arises. The algorithm can be revised to match the training set if it is decided to make a correction after the result of the cost function. For example, a better optimization algorithm can be used for example Adam Optimization algorithm. The arrangements that can be made and the parameters that can be changed will be described in the Section 3.3.2 CNN Hyperparameters.

Another possible situation may mean a need for a larger dataset if the algorithm does not match the validation dataset while performing well in the training dataset, or if it works well in the validation dataset and does not meet the expectation in the test dataset. Because if it doesn't fit the validation set and it doesn't fit the test set, there is probably an overfitting problem, so a larger validation series must be created. In spite of all these arrangements, either the validation set or the cost function can be changed if there is still no good result. Because if your test dataset does not meet the good result needed in a compatible algorithm, either the distribution dataset is not selected correctly or your cost function is not calculating the right thing.

One of the most important things in this process, if you can look at the established system and the results obtained, and if it can be determined that there is a missing point, it may mean one step closer to the successful result. Exactly where is missing or error is often not seen clearly.

Modern CNNs are able to achieve superior performance by employing a very deep hierarchy of layers. CNNs are widely used in a variety of applications including image understanding, speech recognition, game play, robotics, etc. [2].

3.2.2. Convolutional Neural Network

CNN mainly uses the standard Neural Network for the classification process, but for feature detection and differentiation, it performs the image processing in different layers in the background as shown in Figure 3.1.

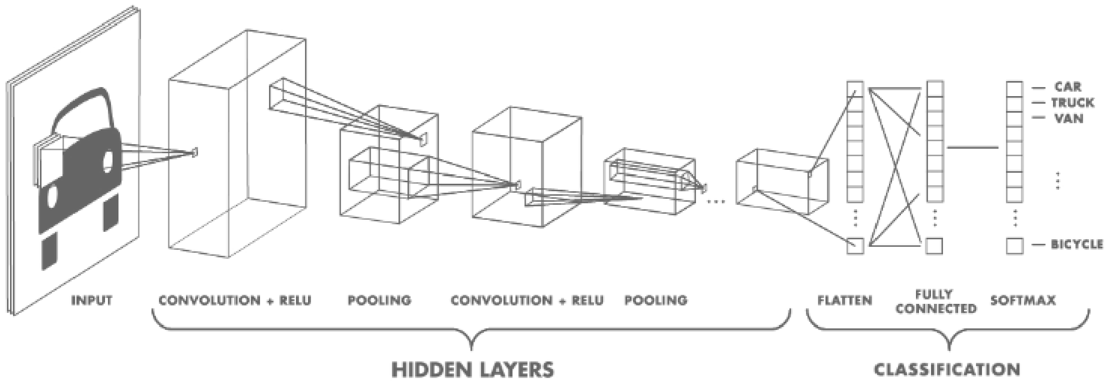


Figure 3.1. An illustration of the layers of an example CNN (Source: [19])

These layers can be summarized as:

Convolutional Layer: Used to extract features by applying filters to the image

Non-Linearity Layer: Applied to convert network to a non-linear structure

Pooling (Downsampling) Layer: Aims to reduce the size of data generated for output

Flattening Layer: Prepares input data for the fully connected layer

Fully-Connected Layer: Standard Neural Network for Classification

3.2.2.1. Input Layer

This layer is the first layer of CNN. In this layer, input is visually entered into the artificial neural network as raw. The size of the input data is important for model success. If high-dimensional image is used, high memory needs, high train time and test time need. But it can also increase network success. If a low-dimensional image is preferred, the memory requirement and duration are reduced proportionally, but the depth and success of the established network may be low.

It is important to choose the right size to reach the optimum result. To give an example of some models that could be considered as the basic structure as the deep learning Alexnet filter, which is another popular model when using 4x4 size 7x7 has been used as a drift net ZF 11x11 and 2x2 shift the filter size. ZF Net is also preferred in different sizes, the first layer with smaller filter input data to ensure the preservation of many original pixel information.

3.2.2.2. Convolutional Layer

The main building block for CNN can be called the conversion layer. This layer extracts features by applying filters to the image. These filters are mostly multidimensional and have pixel values. The conversion is based on the image data of the selected size filter. You create an output data by applying a filter operation to the data from the previous layer and result in a Feature Map as seen in Figure 3.2. This map is the region where each filter is discovered.

Filters can be created in different sizes. For example, 5x5x3 dimensions are created. 5x5 indicates the height and width of the matrix, and 3 is the depth.

Let's assume that our input data is 256x256. Firstly, the specified filter starts from the top left corner and the image is scanned (1x1 scroll (stride)). When the first line is completed, a pixel is passed to the bottom line and the same operation is repeated for this line. All image (all pixels) is scanned so that an output matrix is created. When creating the output matrix, the indices between the image and the filter are multiplied by one another and this multiplication is collected.

The output matrix is smaller than the original image matrix. The output matrix is often called Feature Map and properties are detected. Multiple filters are used to explore multiple properties. This can be found in more than one Convolutional Layer in a CNN. Each filter used is actually a neuron of the artificial neural network. The values in the filter correspond to the weight. Since the feature map is smaller than the original image matrix, the original matrix value is maintained by adding zero values with the padding operation.

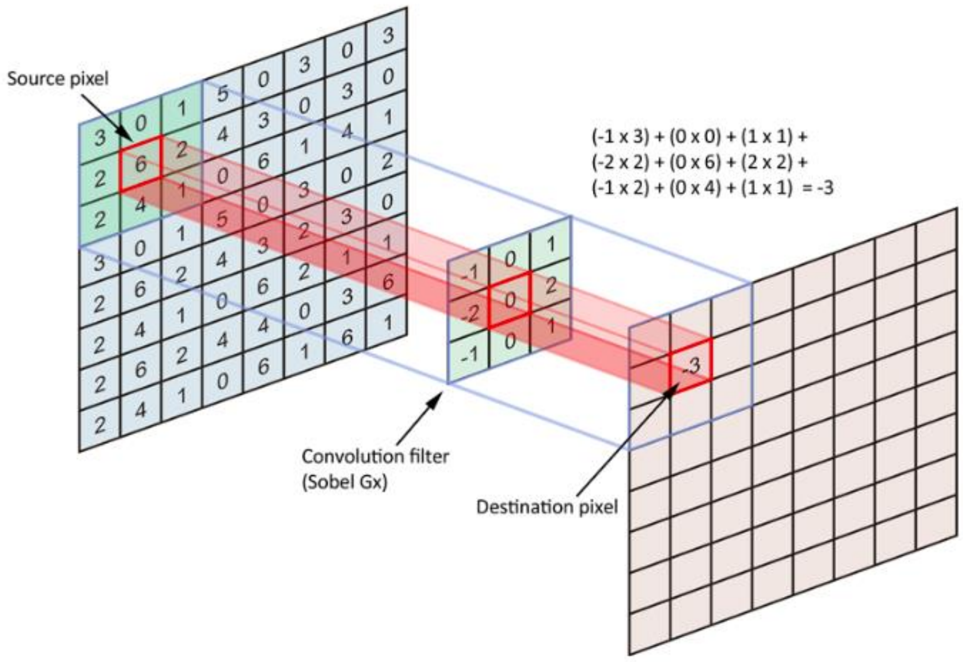

Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Figure 3.2. Visualization of 3 x 3 filter convolving around an input (Source: [21])

3.2.2.3. Non-Linearity Layer

It uses activation functions in this layer. Activation functions are divided into linear and non-linear functions. Non-linear functions are the most commonly used activation function, and commonly the ReLu (Rectified Linear Unit) function, which produces the best results in terms of speed, is preferred.

The network is linear because it performs mathematical operations during filtering on convolutional layer. This layer is applied to convert this network to a non-linear structure, and as a result of this layer, the network is trained more quickly. Since the output of the feature map can reach very large dimensions, it is necessary to draw the output results to a certain range.

ReLu Function = f (x) = max (0, x)

In the following example as shown in Figure 3.3, black values in Feature Map are negative. The ReLu function is applied to Feature Map to replace negative values to 0.

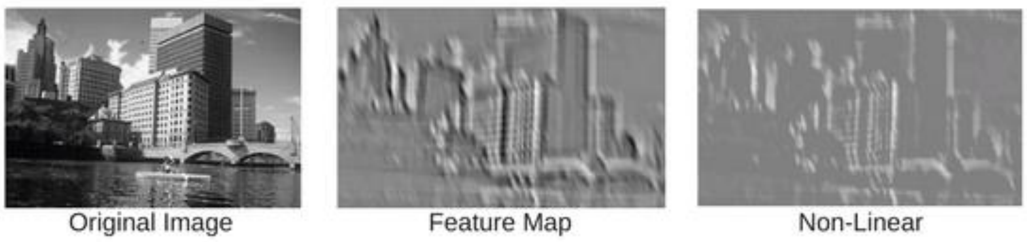


Figure 3.3. ReLu function results (Source: [23])

As you can see from the chart in Figure 3.4, R (z) is equal to zero when z is less than zero for ReLu.

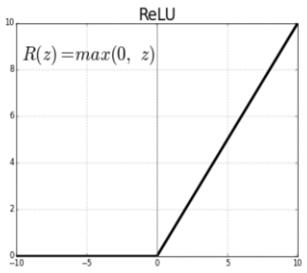


Figure 3.4. Representation of ReLu function (Source: [18])

In cases where it is greater than zero, f (z) is equal to z. Therefore, the result range is between zero and infinite as shown in Figure 3.4.

3.2.2.4. Pooling

The pooling process is a filtering process on the kernel output. A variety of computations that reduce the dimensionality of a feature map are referred to as pooling. [2] The Pooling layer is usually located after the ReLu function and is intended to reduce the size of the data generated in this layer output. Pooling does not affect the depth of the image.

Generally, the max pooling method is used. However, techniques such as min pooling, average pooling can also be used. With max pooling, the filter takes the largest number in the region covered by the matrix on which it is applied as shown in Figure 3.5. This results in smaller matrix outputs that contain enough information for the network.

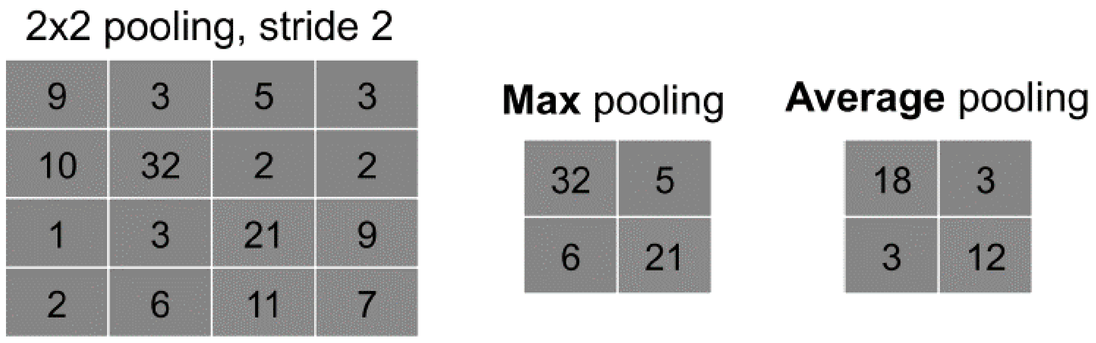


Figure 3.5. Some Pooling Methods (Source: [18])

As can be seen from the above example, the matrix size decreases as a result of this process. This reduction is thought to lead to data loss, but this downsizing will reduce the computational load on subsequent layers. In addition, problems such as system memorization and overfitting are prevented.

The pooling operation is performed for each image on the dataset as much as the number of filters produced by the first layer. In addition, this layer can be considered as an applied layer in the CNN architecture.

3.2.2.5. Flattening Layer

Input data for the last layer, fully connected layer, is prepared in this layer. Artificial neural networks generally use a one-dimensional array as input data can be seen in Figure 3.6. In this layer the data from the previous layer is converted to a one-dimensional array.

3.2.2.6. Fully-Connected Layer

This layer is the latest layer for CNN. The output resulting from the flattening layer is taken as input data and performs the learning process via neural network.

3.2.3. CNN Hyperparameters

Hyperparameters vary according to the problem and dataset. There may be different groups of hyperparameters where the model provides high performance. In this section, hyperparameters will be explained.

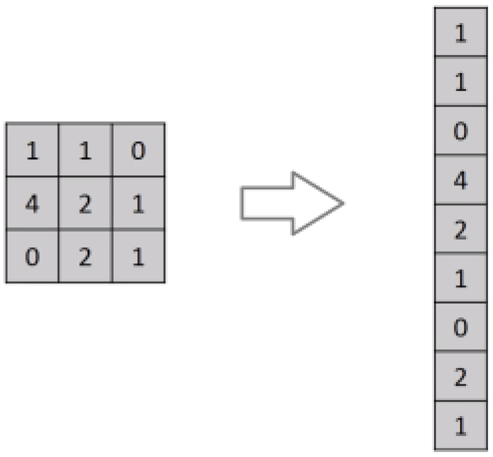


Figure 3.6. Flattening (Source: [22])

3.2.3.1. Number of Layers

According to other artificial neural networks on complex problems, the most important feature that holds the deep learning method forward for success is the layer number. The number of layers actually corresponds to the concept of depth for deep learning. Although there is no mathematical relationship between the number of layers and the success rate, deep learning provides better results thanks to the layered structure.

It is observed that in the first layers belonging to an artificial neural network, the visual lines are learned and the other features of the data are learned in deeper layers.

However, there is no general rule about what the number of layers will be. If a new network is being established, the success rate can be followed by starting with a two-layer network and working on hyperparameters. Then, the number of layers can be increased in a controlled manner. However, as the number of layers increases, the structure of the model may be changed. Because the first established low-layer structure may not support a deeper neural network. Because the number of layers, the depth increases, so the backpropagation effect will be less able to reach the first layers. also increasing. If a non-GPU environment is present, or if there are no sufficient GPUs, the number of neurons should be considered. It is an important hyperparameter to determine how many layers to use and how many layers to use when building CNN architecture.

The concept of depth corresponds to the number of layers that make up the architecture, which gives a higher success rate in solving more complex problems by forming the deep learning method and differentiating from other artificial neural networks. It is observed that the model gives better results by increasing the number of layers, because the properties learned in each layer are detailed.

The intuition behind this network is that each convoluted layer learns a more detailed representation of the images (property map) than before. For example, the first layer can recognize very simple forms or colors and a more complex forms such as full faces for example.

Kernel dimensions are an important parameter on learning. In the CNN, each layer is used to process the kernels on the matrix. The size of the kernel is determined by how wide the size of the data will affect each other. Using kernel in large sizes will cause loss of data and the output will be small and usually 3X3 kernel is used.

3.2.3.2. Learning Rate

In deep learning, parameters are performed with backpropagation. With the chain rule, the difference is obtained and the difference is found and the difference value and learning rate parameter are multiplied and the resultant weight value is calculated and the new weight value is calculated. The learning rate value can be set as a fixed value, or the step by step can also be determined as an incremental value.

If the learning rate is chosen as a large value, this means that it is more affected by the data. However, choosing a small value causes learning to take too long. For this reason, the optimal solution at the beginning would be to reduce the learning rate by keeping it high as it progresses, because initially the smallest value may cause it to fail at a local level and not reach a global degree. By default, it is accepted as ‘Learning Rate = 0.01’ and then it is reduced to 0.001 after a certain number of epoch.

3.2.3.3. Optimization Algorithms

Deep learning is essentially an optimization problem. Different optimization methods are used to find the optimum value for nonlinear problems. In deep learning projects, Adam, Adamax, Stochastic Gradient Descent algorithms are widely used (Figure 3.7). These algorithms differ in speed and success. Stochastic Gradient Descent algorithm can give bad results in image recognition problems. These algorithms are self-learning and dynamic.

3.2.3.4. Number of Epoch

While a model is being trained, not all of the data is used at the same time. It is divided into a number of clusters. The first cluster is trained and the success value is evaluated and the weight values are updated with the backpropagation process. When creating the model, batch size is determined. The number of data is determined by the established neural network model. Each passage is called epoch.

When weight values are calculated step by step in a deep learning problem, the success rate will be increased by increasing the number of steps while the success value is lower in the first epoch. but after a certain step the learning situation will no longer be reduced. In other words, increasing epoch may not provide better efficiency for each scenario, but may cause overfitting of training data.

In some scenarios, the training period of the model is quite long, and this is considered usual for deep learning. However, it is possible to decrease the value of number of epoch hyperparameters in cases where this period is desired to be reduced.

Some problems (RNN, CNN problems) expect a high number of epoch numbers, because this number increases to a point proportional to success. As mentioned earlier, this success will start to increase at a smaller rate after a certain success rate. At this point training can be terminated.

It would be more meaningful to refer to thousands of faces for the dataset at hand for visual classification. In addition, if the similarity between classes is greater (if the passivity is higher), the success graph may be bumpy. In this case, playing with the batch value can increase the chances of success.

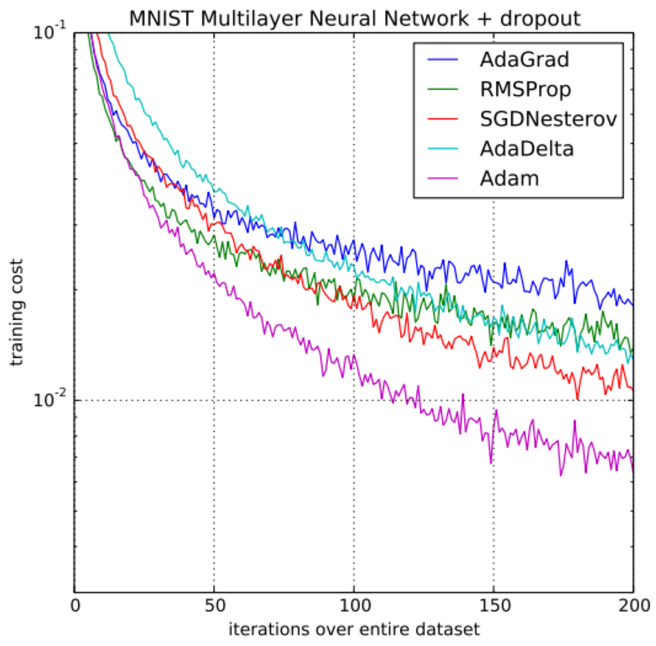


Figure 3.7. The speed of training cost of different optimization algorithms on MNIST dataset (Source: [24])

3.2.3.4.1. Early stopping

When the model train starts, the difference between the validation loss and the test loss increases, so the overfitting occurs. In this case, when the validation error begins to increase, the training is stopped and the previous epoch is returned.

Very large networks can be trained with early stop applied together with backpropagation without significantly overfitting [4]. With early stopping, it is aimed not to stop at a specific end point determined by the beginning of the network, it is aim to stop at the most ideal point [4].

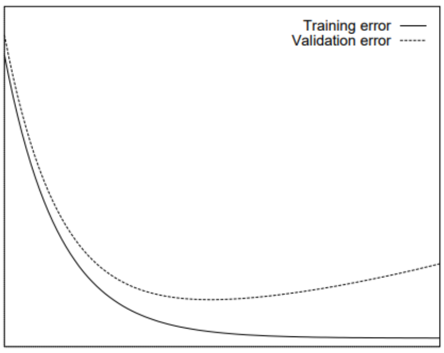


Figure 3.8. Idealized training and validation error curves. Vertical: errors; horizontal: time (Source: [5])

How to do early stopping using validation: Train only on the training set and evaluate the per-example error on the validation set once in a while, e.g. after every epoch. When the error value obtained from the validation set is greater than the previous error value, the training is stopped as seen in Figure 3.8 [5].

3.2.3.5. Batch Size

In a large data series, the memory costs to process all data at the same time. With backpropagation in each iteration, backward gradient descent calculation is performed on the network and weight values are updated as a result of this calculation. And the larger the data, the longer the calculation takes. As a solution, the dataset is divided into groups and training is applied to these small data groups. These groups are divided into groups and called the mini-batch. The value given to the batch parameter during model design indicates how much data will be processed during a iteration. Batch operation increases the loss value, but it also saves time and performance.

When a mini-batch is applied, there are fluctuations in the error value. Because each iteration is processing different data, and some parameters may not be suitable for some data when the selected parameters are appropriate. However, it is observed that these fluctuations decrease as the iterations increase.

This fluctuation can be reduced by determining the learning rate. The mini-batch value can be set to 1 and all data in the cluster. Setting this value to 1 means that each iteration requires only one data operation. This may cause the model to learn the data and fluctuations in the graph may increase. A global value can never be reached and a optimization algorithm can be mistaken for this reason. In addition, a loss of time will occur because the network can go the wrong way. If the mini-batch value is set to be equal to the total number of data, then all data will be included in each iteration. This will result in less fluctuation in the modeled model and faster progress in the optimization process. But all this will cause time and performance loss. Therefore, when choosing this value, neither too small nor too large should be selected. Learning in this way will be faster. There are some clear determinations about the batch size. These are;

● Batch value is selected to be multiples of 2 to fit in the GPU memory. If this is not chosen, there may be a sudden drop in success.

● Batch is usually selected between 64-512.

● If the training set is small (<2000) the number of all elements in the dataset can be used as batches.

● Batch for CNN is an important parameter. So small changes in this value can affect the success of the CNN result.

● Batch and data can be grouped together. This creates a correlation. This will result in high success in the selected test dataset from the dataset and memorizing overfitting. To prevent this, the dataset should be shuffled before disassembling. Therefore, the data should be selected randomly in batch operation.

3.2.3.6. Weight Initialization

The determination of the weight values of the model directly affects the teaching and speed of the model. These weight values can be determined to be 0 or they can be determined to have a distribution between 0.5 - 0.9. The weight values of an earlier model can also be used as weight values of the model. Weight values can be determined by different methods.

If the weight value is initially set to 0, the matrix multiplication is a total, so the input values appear as output. Therefore, this value should not be initially set to 0.

Since y = f (x, w) -> w.x, the values of the matrix w are 0, so the x-containing matrix collection will return x.

Weight values can be started at the beginning of the model to be randomly small numbers and the model works in small networks. However, it causes a non-homogeneous distribution of the inter-layer activation.

3.2.3.7. Activation Functions

Activation functions add non-linearity to our deep learning model. Provides activation functions for the nonlinear connection that connects the input sum obtained with the result of matrix multiplication with output. That is, the matrix function is multiplied with the linear function in hidden layer and the weight value of the neurons is calculated and the output is converted to a non-linear value. Because deep learning problems are mostly non-linear. Non-linear problems and deep learning are more effective on the solution of these problems. In order to perform gradient decent calculation, hidden layer outputs are normalized with activation functions. Commonly used activation functions such as Sigmoid, ReLu, PReLu, tanH functions are shown in Figure 3.9. In these functions, the most choices is ReLu. However, if negative values are important for our problem, PReLu should be preferred. PReLu captures the negative values that ReLu has missed.

The Sigmoid function was more popular when the classification process became more popular among deep learning problems, but as the depth increased, ReLu became a better solution. With increasing depth, ReLu was the most preferred function to avoid derivative transactions of negative values.

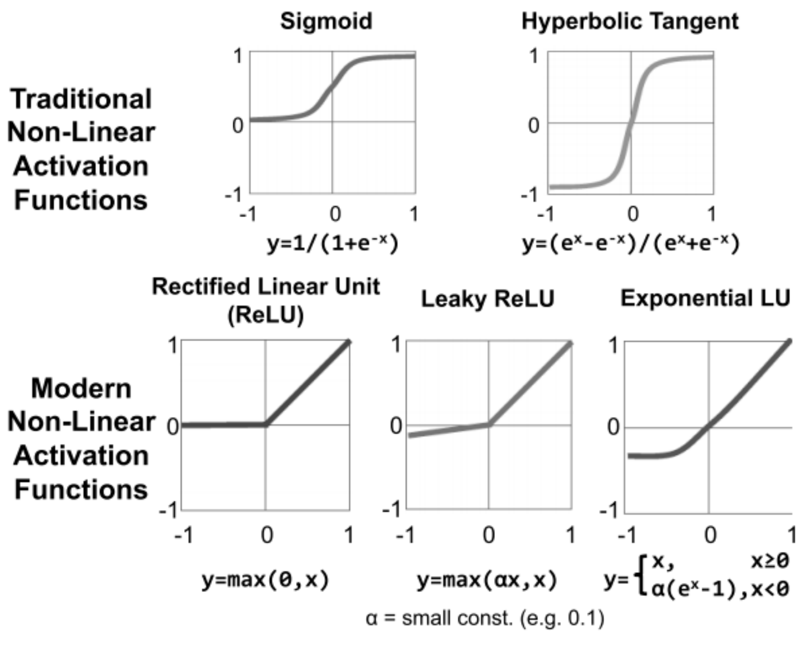


Figure 3.9. Various forms of nonlinear activation functions (Source: [18])

A function similar to the Relu function was introduced by Google, and this function, called Swish, has been described by some sources to improve performance by a very small 0.001 percent. But Swish operates 20% slower than the ReLu function.

3.2.4. CNN Regularization Hyperparameters

Whether the model to be designed by using machine learning and deep learning methods will be successful will emerge as a result of experimental studies. Whether the data to be studied is sufficient or whether the model will cause overfitting is the result of these studies. Regardless of the results of the model or model with regularization hyperparameter, the performance rate to be applied with improvement methods can be increased. Regularization (e.g., early stopping) is critical for nets of all sizes - not just ones that are too big [4]. In this section, regularization methods will be discussed.

3.2.4.1. Dropout

In CNN, if there is a risk of overfitting in the established network, Dropout method should be applied to prevent memorization. If the network is large, the training period is too long, or if there is not enough data, this method can be applied. With this application, some nodes on the network are removed. The neurons which are “dropped out” in this way do not contribute to the forward pass and do not participate in backpropagation (Figure 3.10). So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights [1].

The Dropout method can only be implemented in fully-connected layers. Without dropout, our network exhibits substantial overfitting [1]. It is observed that success is increased by disabling the nodes under the threshold value set in the Fully Connected layers. This threshold is generally accepted as 0.5. That is, forgetting the information below 0.5 increases training. It is not necessary to use the same dropout value in all layers. Different values can be used in each layer.

During training with dropout, neurons are made independent and transformed into more efficient.

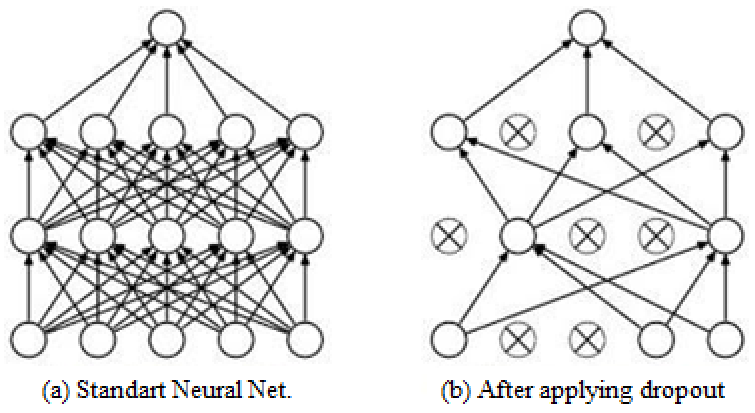


Figure 3.10. Dropout Neural Network Model (Source: [3])

3.2.4.2. Dataset Augmentation

The more training for training the model, the more successful the model will be. The high number of data will also reduce or even prevent overfitting. Therefore, to increase the size of the existing raw data, the method of producing synthetic data is preferred. Different versions of the data in a dataset will be in accordance with deep learning. For example, for visual data, the data size can be increased by rotating the picture at different angles, random translations, rotations, flips of the input [17], taking a portion of the visual data, playing with brightness or vividness settings.

Although the size of the data is increased by means of the synthetic data generation method, it only increases the performance at a certain rate. Because although different versions belonging to a visual are in the data table, there is still a relationship between the original image and its version, the diversity does not increase in full. Therefore, the dataset augmentation method shows better performance when used in combination with other regularization methods.

3.2.4.3. Regularization Strength

There may be one different weight vector that produces the same cost for an input value. In this case, only one of them should be preferred. In model improvement, models with small weight values are assumed to be more understandable and simpler than models with large weight values. For this reason, lower weights are preferred and success will be increased and memorization will be prevented. For this purpose, by using penalty functions, weights can be kept at small values.

3.2.5. Transfer Learning

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [10]. It is the learning of a new problem by transferring learning to improve learning by transferring knowledge from a previously learned problem. For this reason, the pre-trained models provide great benefits for the new problems to be solved for many reasons. Firstly, it saves time in the problems that will be applied since a previously trained model is used. The previously trained model spends a lot of resources and time for feature learning. Transfer learning method to be applied new problems that save time and resource needs. In other words, transfer learning is an optimization that provides a faster progress and improved performance when modeling a second task.

Many machine learning methods only operate under a common assumption: training and testing data are obtained from the same property area and the same distribution. When the distribution changes, most statistical models need to be re-made from scratch using the newly collected training data. In many real-world applications, collecting the necessary training data again and rebuilding models is expensive or impossible. It would be good to reduce the need for re-collecting training data. In such cases, it may be desirable to learn to transfer or transfer information between task fields [11]. Transfer modeling with visual data is a widely used approach. For such problems, the pre-trained deep learning model is widely used for a major task such as the ImageNet 1000 class classification competition. In practice, very few people train an entire convolutional neural network from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pretrain a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest. Convolutional neural networks features are more generic in early layers and more original-dataset-specific in later layers [13].

The advantages of transfer learning are:

1. There is no need for an extremely large training dataset.

2. Too much calculation power is not required. We need to learn about the weight of the last few layers and we only use pre-trained weights.

With the transfer learning method, more capable models can be developed for problems that do not have too much data. But the choice of source data or model is very important. With the comparative experiments, the source model to be used for problem solving should be determined.

3.2.6. Specifying the Dataset

Dataset size and variety of data is the most important factor in deep learning applications. The larger the dataset, the better it will be realized and a good result will be obtained. However, having a large dataset also means that more time will be spent on learning, and the model that arises as a result of learning grows at the same rate. This time can be ignored for datasets that will not be trained very often and will not cause problems in storage. However, this size should be taken into consideration in the datasets that will require more frequent training and create problems in terms of storage. Apart from the size of the dataset, diversity is also important. Data diversity is also important in model success. There is actually a breakpoint for the size of the dataset. Because the infinite dataset should not mean very accurate results. The concept of size should also have a limit in terms of the problem. The breakpoint of the dataset should also be explored under the deep learning problem. If we have a small dataset, this problem may not be suitable for deep learning. The dataset should be increased with data augmentation or the attribute can be upgraded with Transfer Learning.

CHAPTER 4

CONTAINER DAMAGED - UNDAMAGED CLASSIFICATION

Digitalization is important in logistics sector. In the present system, the personnel carrying out the damage assessment of the containers entering the warehouse, photographing the damages detected, recording these photos and container numbers together and entering these reports into the IT system are carried out by the personnel. Instead of the all of these processes, deep learning solution to be developed by the system is intended to be automated.

This section describes a deep learning method to perform damaged / undamaged estimation using visual data to be taken from containers while entering logistics warehouses. The estimation problem is modeled as classification problem. In order to model this problem, the definitions of damaged and undamaged Container have been learned. In accordance with this information, the visual dataset was divided into two classes and experiments were performed in this section.

In the following subsections, the general architecture of the chosen method is explained, and the steps of dataset preparation are mentioned.

4.1. Problem and Challenges

‘Damaged’ or ‘Undamaged’ estimation to be performed correctly using visual data is an important factor for the companies involved in the logistics sector. Nowadays, container damage detection is carried out depending on the personnel experience. As a result, negative conditions such as failure to detect damaged containers may occur. With the failure to detect the damaged container, financial procedures may be applied to logistics companies.

During the process which is carried out today as shown in Figure 4.1, container damage detection is performed by authorized personnel during the logistic depot entry or exit. The photographs of the container that are considered damaged are taken by the relevant personnel and these photos are uploaded to the IT system together with a container number.

The photos that will be included in the reports to be prepared from among the uploaded photographs are separated and a ‘Damage Determination Form’ is created. Container maintenance processes are then initiated.

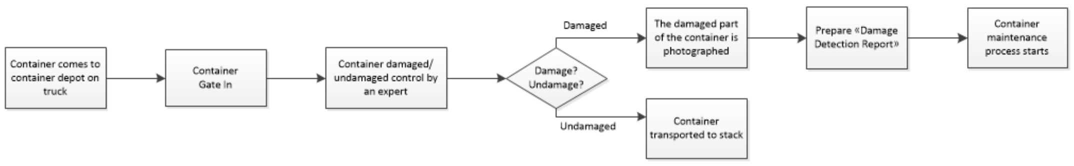

Container comes to container depot on truck
Container Gate In
Container damaged/ undamaged control by an expert
Damage? Undamage?
Damaged
Undamaged
The damaged part of the container is photographed
Prepare «Damage Detection Report»
Container maintenance process starts
Container transported to stack

Figure 4.1. Container warehouse workflow

It is important to create the right model and achieve high success rates with the method to be determined for this problem which will affect the financial results. Thanks to the solution to be developed, a system will be established to integrate with the IT systems that logistic companies are currently using and the processes that are carried out manually will be automated.

In addition to the target of automating, I / O processes will be accelerated in container warehouses and will enable new demands and developments on digitization in the logistics sector.

4.2. Constructed CNN Model

The proposed approach for the damaged / undamaged estimation of the container image is based on a convolutional neural network based classification model. Determining that the visual sections to be taken from the container warehouse belonging to the logistics company belong to the damaged or undamaged class will accelerate the processes and will minimize the human error.

This research deals with the classification method for determining whether the container image coming into the system belongs to the damaged (1 / Positive) or not (0 / Negative) class. Determining the approach for this solution which will work on visual data and choosing the method for the classification process is an important decision. The selected method uses visual data and deep learning methods are used to train the model. The model is used to estimate whether the container image is damaged or not.

For the dataset containing container visual data, the CNN model with the below-mentioned layers was created. In the first two layers, there is a two-dimensional convolutional layer with an input size of 50x50. Again, the convolutional layer was added with filter size = 64. max pooling was applied as 2x2 and stride = 2x2 and then 0.25 dropout was applied again.

Flattening was performed by adding a fully connected layer. ReLU activation consisting of 256 neurons was added to the FC layer. After applying batch normalization, the Softmax activation function was performed to summarize the model.

Adam Optimizer was chosen as the optimizer by using the Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999) equation.

Accuracy was monitored and callback was assigned so that the learning rate was updated dynamically. After the observed results of the first trials, the number of epochs was selected as 50. As a last step, batch size for each epoch was modeled and the model was created.

Transfer Learning method was applied after these steps. In cases where there is not enough data, storage area or computer power when traditional learning is desired, more effective and correct results are obtained by using transfer learning method. Previously, the information acquired by using models that have been trained in millions of images of several classes in high-power GPUs for several days can be transferred to our model. As shown in Figure 4.2, we can train our own problem model using Transfer Learning with a small volume dataset.

For this method, the previously trained VGG16 model was used for use with the Container visual dataset. Using this model, we will observe how experimental patterns will affect the results of the freezing of existing model layers.

4.3. Dataset Preparation & Training

Container photographs that the visual dataset of this study (BO equipment type & Arkas Line company) obtained from the container warehouses belonging to Arkas Logistics were used as the dataset. All container data is stored on file server in the system with no memory problem to be installed.

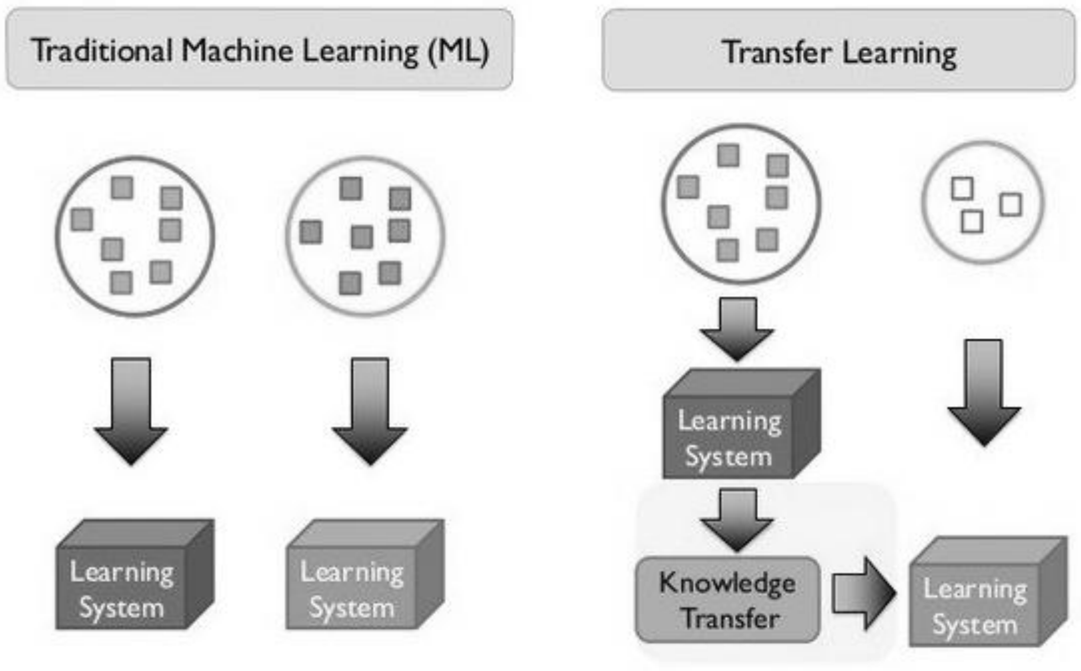


Figure 4.2. Traditional Machine Learning vs. Transfer Learning (Source: [30])

The studies to be carried out within the scope of the thesis are carried out in the PyCharm workspace on the Windows operating system. CPU performance is taken into consideration considering the performance requirements.

To carry out the experiments without transfer learning to be performed, container visual data obtained from the container stores were converted to the appropriate csv format to be modeled. In the matrix prepared for educational purposes, the first column represents the class of visual data. However, this column is not included in the data matrix for testing purposes.

For the input layer of the CNN, the container images were resized as 50x50. Container photographs taken from different angles and distances are divided into 2 classes, ‘Damaged’ and ‘Undamaged’, utilizing the experience of the operation personnel. In order to ensure the consistency of the dataset, the most preferred company in the warehouses was preferred as the container company.

Dataset preparation started with the allocation of container photos to class-based folders. Within the scope of the thesis, photographs were grouped between two separate folders with the name DAMAGED (1) and UNDAMAGED (0).

Initial dataset folder path was given to read the container photos folder. Image manipulation was performed to increase the size of the train dataset. Different rotations were applied on the images. For each Image, 1 original 1 rotated (180 degree) image was created. Also +-25 degree random rotation applied for each image. When considering the side panels belonging to the container, the probability of testing data in the other degrees was low. The brightness and saturation settings were changed randomly. The dataset size has been expanded with these steps. All image read / write operations were performed in python language using OpenCV library. This process was done on PyCharm ide.

Classes of visual data will be expressed as 1 and 0. 1 indicates the class damaged, 0 indicates the class undamaged (Figure 4.3).

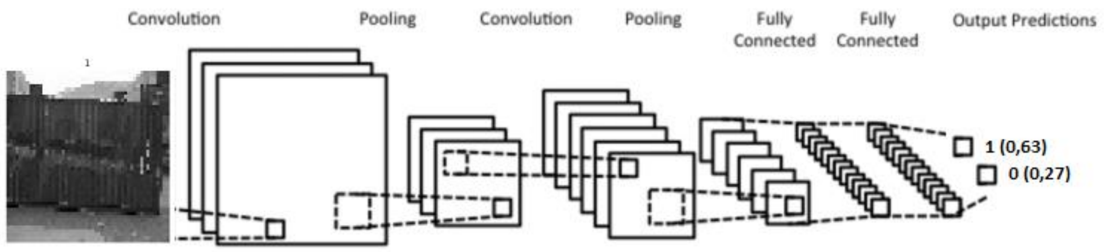


Figure 4.3. Sequential model architecture for Container Dataset

The algorithm that will be established in the following steps will form a mathematical model that associates the visual data contained in the matrices with the classes they belong to. This algorithm can recognize the corresponding class when a test visual data is not included in the previously installed model. The model to be prepared will consist of sequential layers.

CHAPTER 5

EXPERIMENTAL RESULTS

Experimental studies with the data collected in this section are explained. Container visual data were collected through a company in the logistics sector and the data collected from the personnel and the visual data were divided into two separate groups. Experiments were performed to calculate the extent to which the test data to be given to the system with the collected data could be estimated as damaged or undamaged. In addition, the preparation of the training and validation dataset will be mentioned.

Finally, the results of the iterative evolving steps are compared.

5.1. Experiments without Transfer Learning

In this section, the experiments without Transfer Learning carried out on the dataset generated from the container visual data and the evaluation of the test results are described. In the first step of the application, the classes contain as much visual data as the quantities shown in the Figure 5.1.

Visual samples taken from container full side panel were cropped and dataset was created. Since there were no images of damaged and undamaged containers, parts were taken from the container panel. In addition, a standard approach could not be established in the visuals obtained by taking the container panel from the whole. Other containers in the background, loaded trucks, other objects took place in the visual accuracy was low value. As a result of the tests, the accuracy value of the whole used container panels was observed as 0.67. In addition, this equilibrium was achieved by the crop method, since no equal amount of damaged and undamaged classes could be obtained.

Training dataset contains 9088 examples. The visuals were resized in 50x50x3 dimensions. There are 4544 samples per class. An example of an index with damaged and undamaged container parts is as follows as given in Figure 5.2.

The validation dataset contains 909 visual data. Approximately 10% of the validation dataset was allocated over the train dataset. Overall, there is little data to learn for a deep learning problem. Therefore, experiments were performed with different parameters and different model structures in order to make the best use of less data. Damaged container images can be seen in Figure 5.3.

```
1    4544
0    4544
Name: label, dtype: int64
```

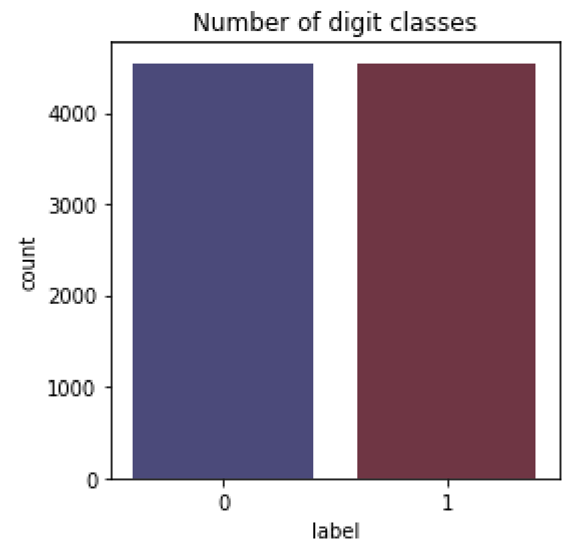
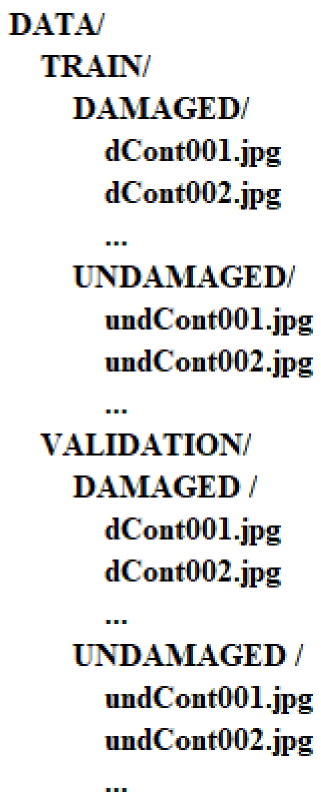


Figure 5.1. Number of classes for first iteration. (1 shows ‘Damaged’ 0 shows ‘Undamaged’)

CNN is one of the best models for problems with small dataset and educating with zero model structure can still give reasonable results. If the results are not good enough, the model can be retrained with minor modifications to the finetuning. Data pre-processing and data augmentation make the best use of dataset. Random rotation was applied to each class to have 4544 data. Within the scope of the problem, first rotation was made between + -25 degrees and then 180 degrees. These values are preferred because the container vertical lines will never be horizontal in the future test data. When the container structure is considered, the side panels are made of vertical corrugated structure.

There may be slight differences in the camera images to be taken, but 90 degrees of darkness will never happen. After rotation, brightness and contrast were adjusted randomly (-25.5) and uniform (0.2.1.8), respectively. In addition, during the model train, we performed data augmentation with the ImageDataGenerator class of the Keras library. The horizontal flip parameter was set to true. Also randomly shifted images horizontally and vertically 5%. Randomly zoomed image 5%. After all these operations, CNN model was established.



```
DATA/
   TRAIN/
      DAMAGED/
         dCont001.jpg
         dCont002.jpg
         ...
      UNDAMAGED/
         undCont001.jpg
         undCont002.jpg
         ...
   VALIDATION/
      DAMAGED /
         dCont001.jpg
         dCont002.jpg
         ...
      UNDAMAGED /
         undCont001.jpg
         undCont002.jpg
         ...
```

Figure 5.2. Container dataset file index

The constructed model as seen in Figure 5.4 has a shallow layer structure as shown in Figure. This model structure was preferred to evaluate the results by keeping the complexity low. It is similar to the VGG16 structure and has dual CNN layers followed by Pooling and Dropout layers. In the last layer, ReLu and Sigmoid activation functions were applied and accuracy rates were obtained for two classes of problems.

Data augmentation method is the most effective method to prevent overfitting problem. The CNN model can store a lot of information, but there is a risk of storing irrelevant information. However, the model, which can store the information limited by its shallow structure, will have the possibility to make the most general and most relevant and the best generalization by focusing on the most important features found in the data. In this iteration, it’s used a small convnet with few layers and few filters per layer, alongside data augmentation and dropout. Dropout also helps reduce overfitting, by preventing a layer from seeing the same pattern, thus acting in a way. Both dropout and data augmentation tend to disrupt random correlations occuring in data.

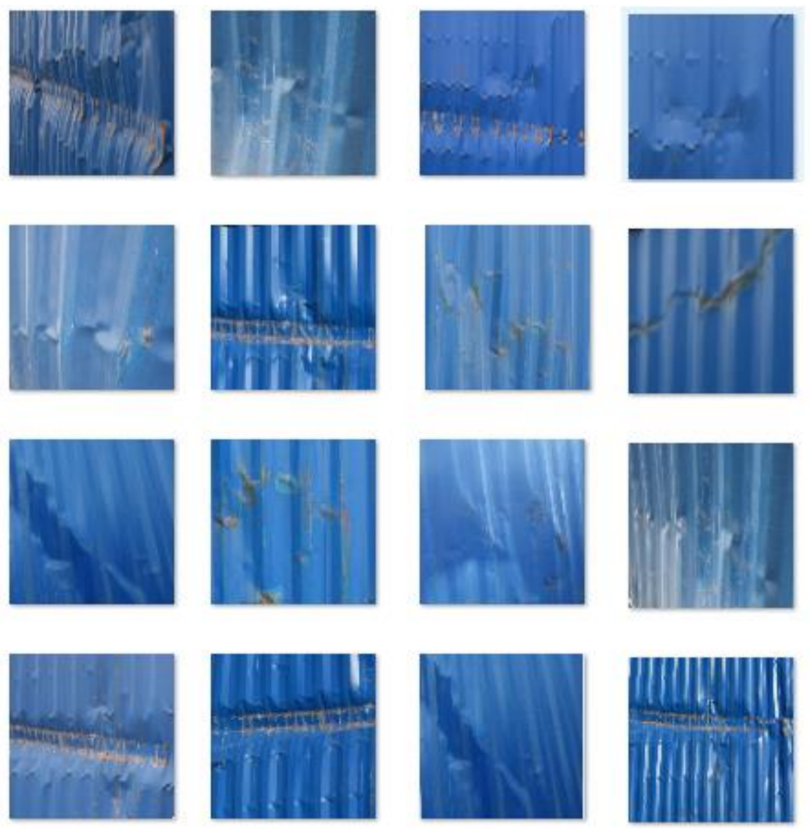


Figure 5.3. Visual sample selected randomly from dataset (50 x 50)

As it can be seen in Figure 5.4, CNN architecture contains 4 convolution layers with a ReLU activation and followed by max-pooling layers for each 2 convolutional layers. Fully connected layer is added on the CNN model.

After applying the ReLu activation function first, this layer is ended with a sigmoid activation. After the CNN model structure is prepared, using Keras library deep learning methods, generated batches of data and their labels. Because of the small dataset and shallow CNN model, it's possible to run model on CPU.

System features:

Processor :Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz

Installed Memory (RAM) Ç 16,0GB

System type: 64-bit Operation System, x64-based processor

The features of the environment in which the experiments are carried out are as shown in above.

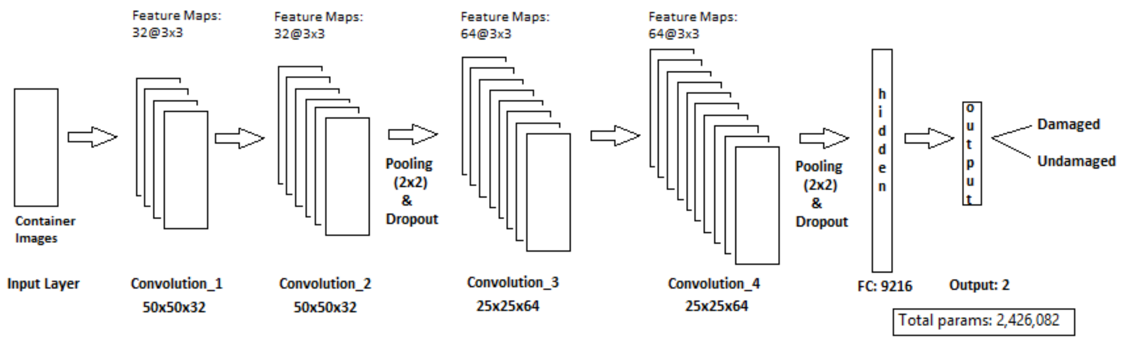
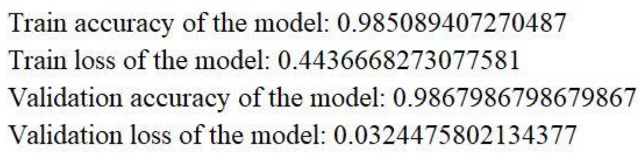


Figure 5.4. Constructed CNN model

For a CNN model with a shallow model and low number of parameters, training and test times were observed at reasonable levels. There is a total training time of approximately 2440 seconds and less than 20 second test time for test dataset. It’s observed that, validation accuracy is 0.98 after 50 epochs as seen in Figure 5.5.

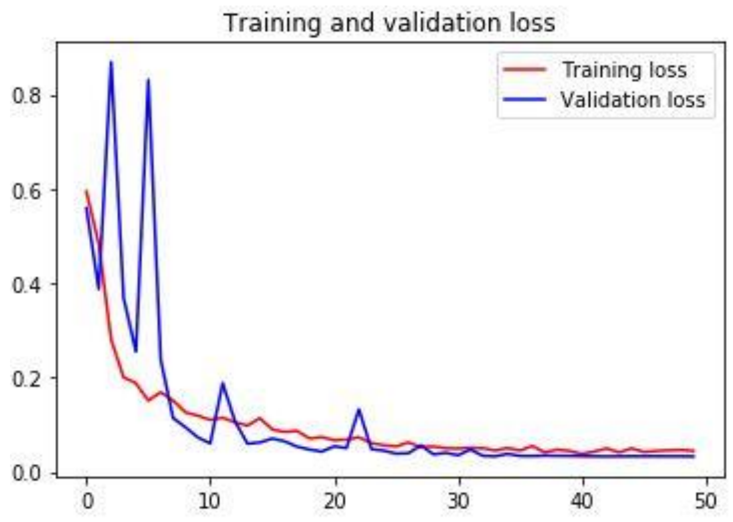


```
Train accuracy of the model: 0.985089407270487
Train loss of the model: 0.4436668273077581
Validation accuracy of the model: 0.9867986798679867
Validation loss of the model: 0.0324475802134377
```

Figure 5.5. Accuracy results for first iteration

The accuracy and loss results for the training and the validation datasets can be seen in Figure 5.6.

When preparing the test dataset, container images not included in the train dataset were used. Test dataset containing 120 data in total was prepared with crops taken from 20 container side panels. This test dataset contains 70 damaged and 50 undamaged images.



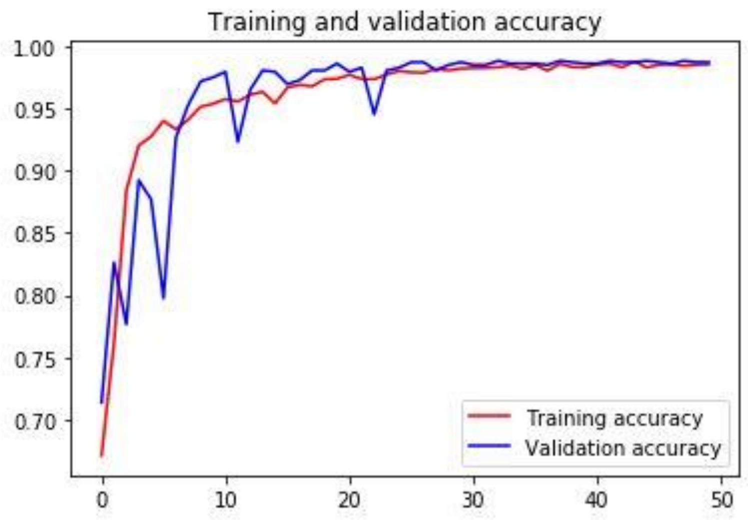


Figure 5.6. Train / Validation accuracy and loss result for first iteration

Tests were performed on the test dataset using the CNN model, which was established without using a pretrained model. While 70 damaged test data were classified as 64 damaged and 6 undamaged, 50 undamaged test data were classified as 37 undamaged and 13 damaged as shown in Figure 5.7. When the test results are examined, there are many estimates with a prediction value close to 0.5. Therefore, the threshold value was given and tuning was not performed on the results.

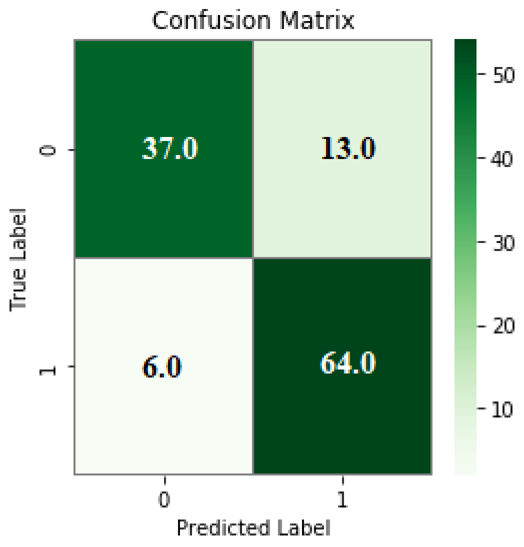


Figure 5.7. Confusion matrix for test results of first iteration

5.2. Experiments with Transfer Learning

In this iteration, the existing train - validation test container datasets prepared for the experimental studies in the first iteration were used without any revision. Transfer learning method was applied on the container dataset and used pre-trained VGG_16. First of all layers were frozen and then frozen the layers of the VGG16 model up to the last convolutional block and then compared results. Since the features learned by low-level convolutional layers are more general, less abstract than those found higher-up, so it is sensible to keep the first blocks fixed and only fine-tune the last block.

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” [26]. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes [15].

VGG-16 [26] is a simple network model and unlike the other models, convolutional layers are used as two or three can be seen in Figure 5.8. The modeled calculation of approximately 138 million parameters is made. It has similar structure with other models and decreases the height and width dimensions of the matrix from the input layer to the last layer. Depth value i.e. the number of channels increases.

In VGG-16 model [26] different weight filters are calculated at each convolutional layer output. As the number of layers increases, the attributes of the filters represent the depth of the image.

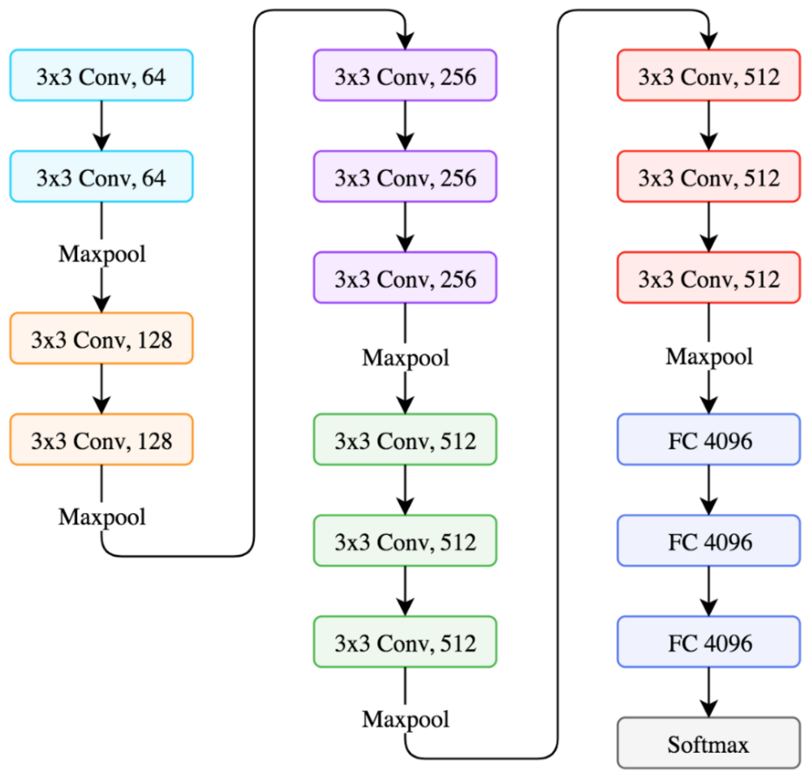


Figure 5.8. VGG16 Model (Source: [20])

The VGG16 class belonging to the Keras library takes the parameters mentioned below. These parameters can be set according to the model architecture to be created for the problem [16].

- **include_top** : Whether or not to include the output layers for the model. This parameter is not needed these if it is modelled on your own problem.
- **weights** (‘imagenet‘): Which weights to load. If you are interested in training the model yourself from scratch, this parameter could be specified none to not load pre-trained weights
- **input_tensor** : A new input layer if you intend to fit the model on new data of a different size.
- **input_shape** : The size of images that the model is expected to take if you change the input layer.
- **pooling** : The type of pooling to use when you are training a new set of output layers.
- **classes** : The number of classes (e.g. size of output vector) for the model.

The VGG-16 model was loaded using ImageNet weights for the second iteration of the problem. As a first step of this iteration, the last fully connected layer that performs the classification task was not loaded by setting the Include_top parameter to false. Only convolutional layers were loaded as first as shown in Figure 5.9.

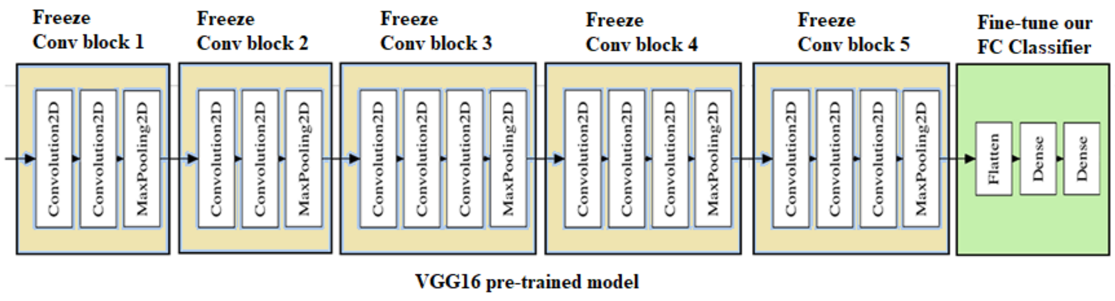
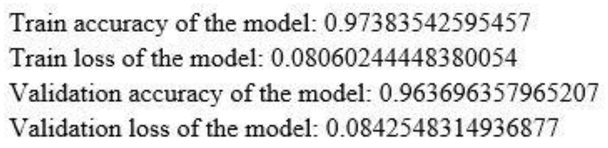


Figure 5.9 Freeze all layer of VGG16 pretrained model

The final layer, the fully connected layer, also created a simple layer with Softmax to achieve prediction values. The established network has been fit for training. RMSprop optimization algorithm was used. Batch_size = 50 and the epoch number is 50 was selected in this iteration. It’s observed that, validation accuracy is 0.96 after 50 epochs as seen in Figure 5.10.



```
Train accuracy of the model: 0.97383542595457
Train loss of the model: 0.08060244448380054
Validation accuracy of the model: 0.963696357965207
Validation loss of the model: 0.0842548314936877
```

Figure 5.10. Accuracy results for second iteration freeze all layer of VGG16

As a second step of this iteration, the last convolutional block was unfrozen and last fully connected layer that performs the classification task was not loaded. Representation of freezing model is shown in Figure 5.11.

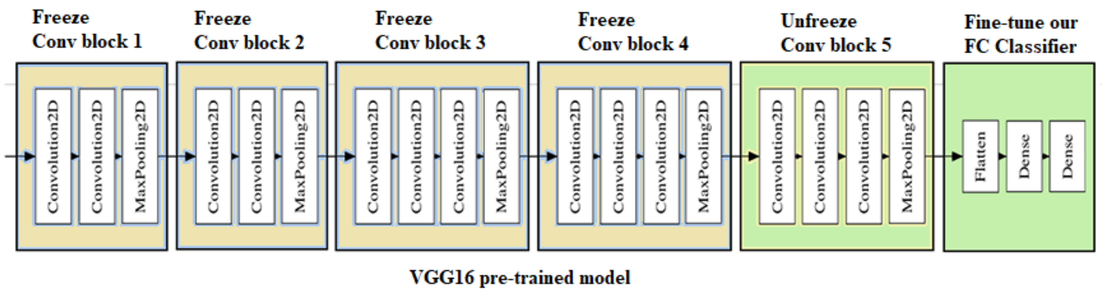


Figure 5.11. Except the last 4 layers (3Conv layer + 1 MaxPooling) the other layers freeze of VGG16 pretrained model

For the values of training & value accuracy and training & validation loss values, visualization was performed by using model history can be seen in Figure 5.12.

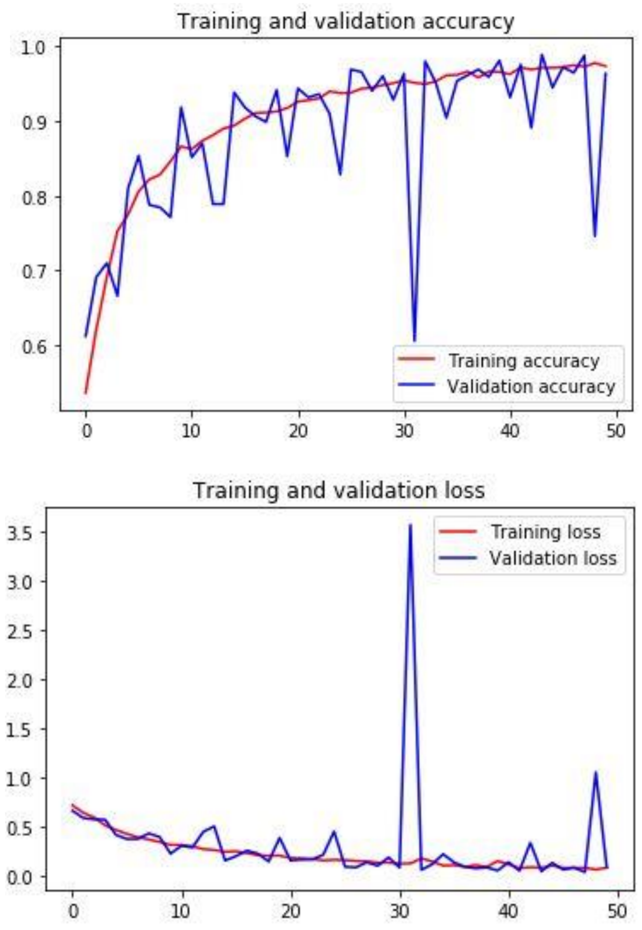


Figure 5.12. Train / Validation accuracy and loss result for second iteration freeze all layer of VGG16

The model was train by adding a fully connected layer to be the same as the previous step. There is a total training time of approximately 8800 seconds and less than 1 minute test time for test dataset. It is observed that the decrease in the accuracy and accuracy is 0.88 after 50 epochs as seen in Figure 5.14.

For the values of training & value accuracy and training & validation loss values, visualization was performed by using model history can be seen in Figure 5.15. Faulty predicted images are illustrated in Figure 5.13.

Transfer learning method was tested by using pretrained VGG16 model with train dataset used in the first iteration. When performing tests for this method, the test dataset used for the first iteration was used without any changes.

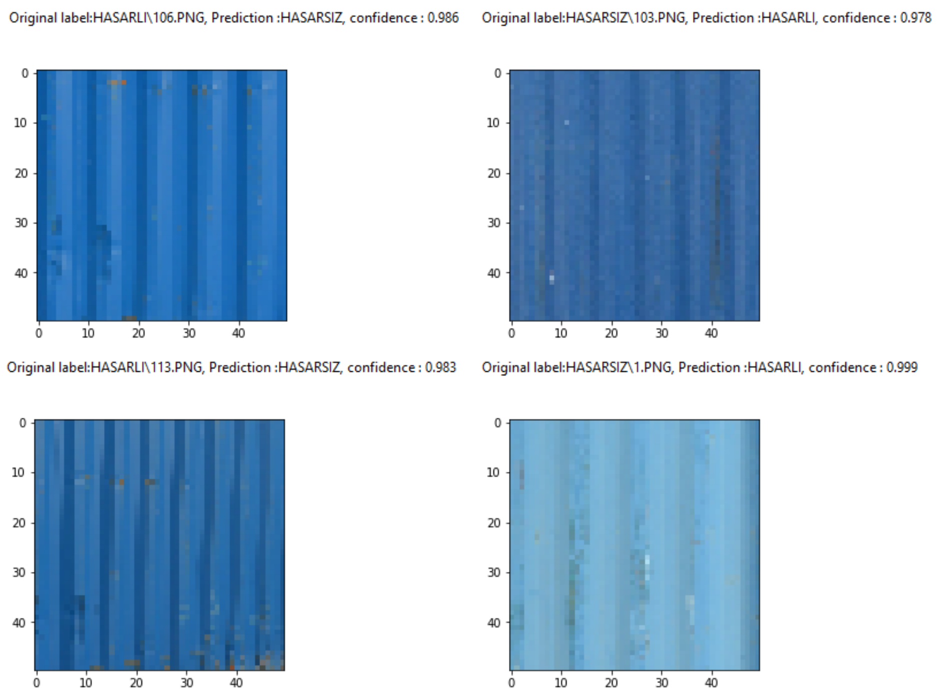
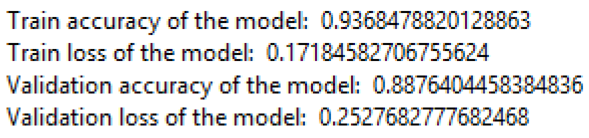

Original label:HASARLI\106.PNG, Prediction :HASARSIZ, confidence : 0.986
Original label:HASARSIZ\103.PNG, Prediction :HASARLI, confidence : 0.978
Original label:HASARLI\113.PNG, Prediction :HASARSIZ, confidence : 0.983
Original label:HASARSIZ\1.PNG, Prediction :HASARLI, confidence : 0.999
0
10
20
30
40

Figure 5.13. Incorrect predictions for second iteration

In VGG16 model, except the last 4 layers (3Conv layer + 1 MaxPooling), the other layers were frozen and tests were performed on the model created model. While 70 damaged test data were classified as 66 damaged and 4 undamaged, 50 undamaged test data were classified as 28 undamaged and 22 damaged.



```
Train accuracy of the model:  0.9368478820128863
Train loss of the model:  0.17184582706755624
Validation accuracy of the model:  0.8876404458384836
Validation loss of the model:  0.2527682777682468
```

Figure 5.14. Accuracy results for second iteration freeze all layer of VGG16 except the last 4 layers

When the test results were examined, it was found that the damaged container was correctly estimated as damaged and had an estimated value of 0.70 and above. Only 1 damaged sample was estimated to be 0.64 Damaged. Similarly, when the undamaged container was estimated to be damaged, it was found that there were 12 images with a prediction value of 0.70 and below.

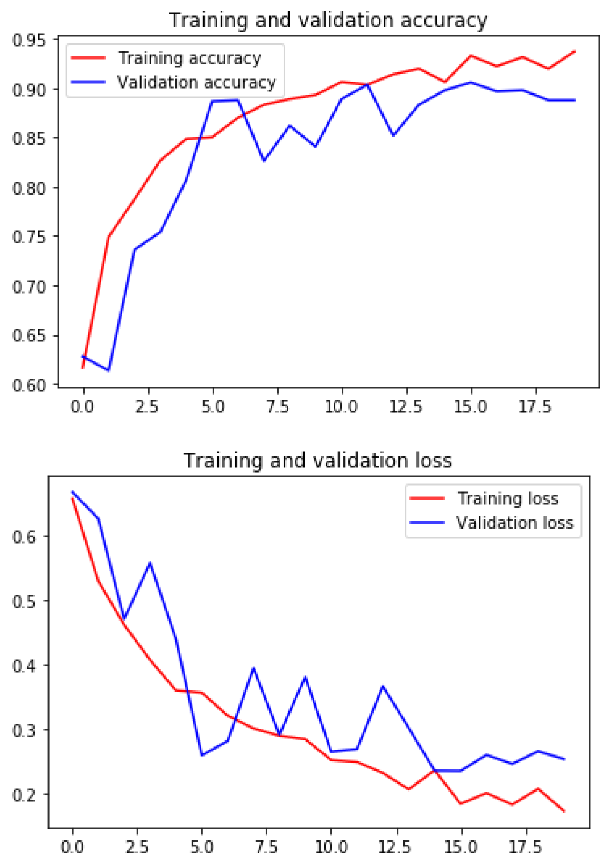


Figure 5.15. Train / Validation accuracy and loss result for second iteration freeze all layer of VGG16 except the last 4 layers

Therefore, when a threshold value of 0.70 is given while performing the tests, the results changed as 65 damaged and 5 undamaged for 70 damaged test data, 12 undamaged and 38 undamaged classification for 50 undamaged test data, thus increasing the accuracy of the test as shown in Figure 5.16.

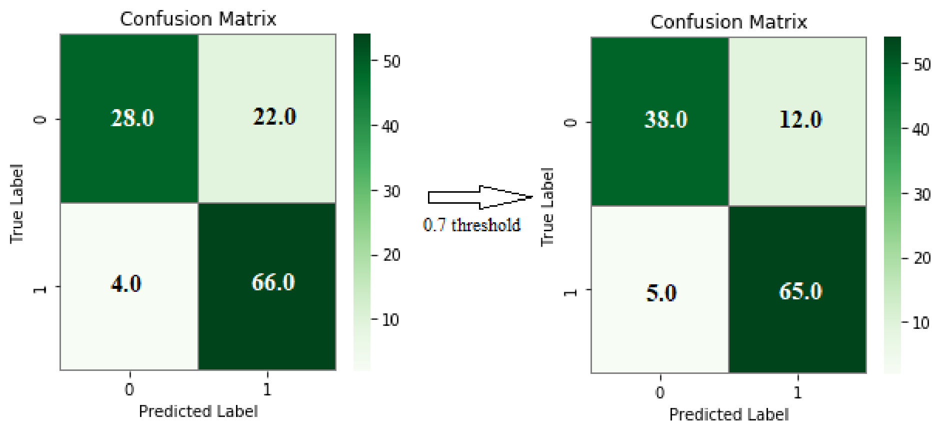


Figure 5.16. Confusion matrix for test results of second iteration

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

In this study, the classification method with CNN approach, which is one of the deepest learning methods, is proposed in order to determine the damages that may occur as a result of transportation or loading on container side surfaces. Two methods were selected for classification and the results were shared. In the first iteration, the training was carried out with the CNN model we established, while in the second iteration transfer learning method was tried. The results were evaluated using the same training and test sets for both iterations.

The damaged and undamaged images taken from the side surfaces of the containers are grouped according to the classes to which they belong. Each group refers to the class of that image. Some of the images obtained by the same method were separated as a test data set without labeling. A total of two classes, namely damaged and undamaged classes, were identified and binary classification was performed.

Experimental studies were started by preparing data set first. Crops were taken from the images of the container panel obtained from Arkas Logistics personnel and container warehouses. In this way, the objects that are included in the visuals belonging to the whole panel of the container, such as truck, personnel, another container, were eliminated and the container object was focused. In addition, since there is no balance in the number of damaged and undamaged containers, this equality was achieved by taking crops. In the real system to be formed, this method will be automated and the side panels of the container coming to the container warehouse will be automatically cropped and the undamaged classification of each crop part will be realized. If any part of the panel is found to be damaged, it will be concluded that the container panel is damaged. In addition, this method constitutes the infrastructure for the study on the region where the damage of the container in the future studies section will take place.

In the first iteration of the experiments without transfer learning section, the power factor of the computer was evaluated and the dimensions of the data sets were changed to 50x50. Training was performed with the CNN model established by the methods described in Section 5.1 and the model was tested with the test dataset. The validation accuracy value was 0.98 and the test accuracy value was approximately 0.85. When the confusion matrix and incorrectly estimated container crop parts were examined, it was found that the difference of angle and light affected the results, but at least one part of the damaged container parts was estimated to be damaged. Despite incorrect estimates of cropped parts, the performance of the damaged / undamaged estimate of the whole panel is within acceptable values.

One of the popular methods for deep learning, transfer learning method was applied. Pretrained VGG16 model consisting of double and triple convolutional layers was used to implement this method. First of all, all layers except the FC layer were frozen. Then, the remaining layers except the FC layer and the last 4 layers (3 Conv layers + 1 MaxPooling) were frozen. The results obtained from the experiments performed with the same train and test datasets used in the first iteration are described in Section 5.2. When the obtained results were evaluated, it was observed that higher accuracy value was obtained in the first iteration. When the CNN model, which is constructed without using pretrained model, is tested with test dataset, VGG16 pretrained model gives lower accuracy value.

In this study, it has been observed that while tuning and augmentation procedures applied while training the convolutional neural network without using pretrained model, the success of the model increased in good direction. When the same dataset was trained by using the transfer learning method using pretrain model, it was observed that the obtained accuracy value was lower. Considering the fact that there is no clear definition in the transfer learning method to select the model appropriate for the problem, the result was considered an acceptable result.

6.2. Future Works

Existing methods can be improved by increasing the size of available data for future studies and selecting a different model for transfer learning practice. When improvements are observed in the results obtained, this study can be expanded with different perspectives.

The model can be presented to the web sites of logistics companies as web services and run dynamically. Integration allows the CNN model to automatically train itself, so that newly added visual data can be dynamically classified.

When the obtained container images are examined, multi-class classification estimation can be made for different damage types. In this way, more detailed reports can be prepared for logistics companies by making the data more meaningful. Since different maintenance types are applied to different types of damage, estimation of the damage class can prevent financial losses.

Since cropped parts of the container are used, the damaged container part can be identified by addressing the damaged area. Thus, the reports generated can be transferred to the online environment and the damaged area can be marked on the container panel simulation to save time for logistics companies.

REFERENCES

1. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

2. Sze, Vivienne, et al. "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE 105.12 (2017): 2295-2329.

3. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine learning Research 15.1 (2014): 1929-1958.

4. Caruana, Rich, Steve Lawrence, and C. Lee Giles. "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping." Advances in neural information processing systems. 2001.

5. Prechelt, Lutz. "Early stopping-but when?" Neural Networks: Tricks of the trade. Springer, Berlin, Heidelberg, 1998. 55-69.

6. Arik, Alper, Mesut Gölcük, and Elif Mine Karslıgil. "deep learning based skin cancer diagnosis." 2017 25th Signal Processing and Communications Applications Conference (SIU). IEEE, 2017.

7. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

8. Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

9. Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.

10. Torrey, Lisa, and Jude Shavlik. "Transfer learning." Handbook of research on Machine learning applications and trends: algorithms, methods, and techniques. IGI Global, 2010. 242-264.

11. Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2010): 1345-1359.

12. "Research Blog: AlphaGo: Mastering the ancient game of Go with Machine learning". Google Research Blog. 27 January 2016.

13. Yosinski, Jason, et al. "How transferable are features in deep neural networks?" Advances in neural information processing systems. 2014.

14. ATALAY, Muhammet, and Enes ÇELİK. "Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamalari-artificial intelligence and machine learning applications in big data analysis." 2019.

15. Neurohive, "VGG16 – Convolutional Network for Classification and Detection" https://neurohive.io/en/popular-networks/vgg16/ , Nov 20, 2018.

16. Jason Brownlee, "How to Use The Pre-Trained VGG Model to Classify Objects in Photographs" Machine Learning Mastery, https://machinelearningmastery.com/use-pre-trained-vgg-model-classify-objects-photographs/ , November 8, 2017.

17. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

18. Y. Jia, et al., "Caffe: Convolutional architecture for fast feature embedding," in Proc. ACM Int. Conf. Multimedia, pp. 675–6782014.

19. Shyamal Patel, Johanna Pingel "Introduction to deep learning: What Are Convolutional Neural Networks?" https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html Mar 24, 2017.

20. Dertat Arden "Applied Deep Learning - Part 4: Convolutional Neural Networks" https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 , Nov 8, 2017.

21. Professor Ben Barres, Professor Staci Bilbo, Professor Beth Stevens "Course Notes: Idempotent Productions" Stanford University https://web.stanford.edu/class/cs379c/resources/lectures/index.html 2019.

22. “Convolutional Neural Networks (CNN): Step 3 – lattening” SuperDataScience ,https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening Aug 18, 2018.

23. Rob Fergus, Honglak Lee, Marc'Aurelio Ranzato, Ruslan Salakhutdinov , Graham Taylor , Kai Yu “Deep Learning Methods for Vision” , https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/ June 21, 2017.

24. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

25. Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

26. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015.

27. Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks." Pattern Recognition 77 (2018): 354-377.

28. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.

29. Karahan, Şamil, and Yusuf Sinan Akgül. "Eye detection by using deep learning." 2016 24th Signal Processing and Communication Application Conference (SIU). IEEE, 2016.

30. Pan, Sinno Jialin, et al. "Domain adaptation via transfer component analysis." IEEE Transactions on Neural Networks22.2 (2010): 199-210.

31. Baştanlar, Yalin, and Mustafa Özuysal. "Introduction to machine learning." miRNomics: MicroRNA Biology and Computational Analysis. Humana Press, Totowa, NJ, 2014. 105-128.

32. Seide, Frank, and Amit Agarwal. "CNTK: Microsoft's open-source deep-learning toolkit." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

33. Singh, Kanwar Bharat, and Mustafa Ali Arat. "Deep Learning in the Automotive Industry: Recent Advances and Application Examples." 1906.08834 (2019).

34. Patil, Kalpesh, et al. "Deep learning based car damage classification." 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2017.

35. Jayawardena, Srimal. "Image based automatic vehicle damage detection." (2013).

36. Samadzadegan, F., and H. Rastiveisi. "Automatic detection and classification of damaged buildings, using high resolution satellite imagery and vector data." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (2008): 415-420.

37. Sarkar, Soumalya, et al. "Deep learning for structural health monitoring: A damage characterization application." Annual Conference of the Prognostics and Health Management Society. 2016.

38. Oquab, Maxime, et al. "Learning and transferring mid-level image representations using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

39. Yosinski, Jason, et al. "How transferable are features in deep neural networks?" Advances in neural information processing systems. 2014.