CONVOLUTIONAL NEURAL NETWORKS (CNN) BASED BINARY CLASSIFIERS FOR CONSTRUCTION MACHINERY DETECTION

BAHADIR TATAR

SEPTEMBER 2022

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF CIVIL ENGINEERING
MASTER'S THESIS IN
CIVIL ENGINEERING

CONVOLUTIONAL NEURAL NETWORKS (CNN) BASED BINARY CLASSIFIERS FOR CONSTRUCTION MACHINERY DETECTION

BAHADIR TATAR

SEPTEMBER 2022

ABSTRACT

CONVOLUTIONAL NEURAL NETWORKS (CNN) BASED BINARY CLASSIFIERS FOR CONSTRUCTION MACHINERY DETECTION

TATAR, Bahadır
Master of Science in Civil Engineering

Supervisor: Assist. Prof. Dr. Seda YEŞİLMEN
Co-Supervisor: Dr. Halil Fırat ÖZEL
September 2022, 107 pages

Monitoring construction activities with artificial intelligence is a substantial mission for efficiency of construction site application. Therefore, this subject has attracted considerable attention in the literature. In construction sites that are observed and optimized with artificial intelligence supported computer vision technologies, the size of the construction site affects the efficiency and success of the work. This situation determines the type of methods and tools to be used in the study. Construction site monitoring studies in large construction sites can be carried out with image classification algorithms trained with construction machinery images. The use of drone footage may be insufficient in construction site monitoring applications performed for large areas. In this thesis, satellite image classification has been performed for construction machinery detection. A dataset that contains construction machinery images created from scratch using Google Earth was used to train convolutional neural networks. A total of 23 different pre-trained convolutional neural network models were modified with the transfer learning method and their performance was evaluated.

Keywords: Deep Learning, Convolutional Neural Networks, Transfer Learning, Object Detection, Satellite Imagery, Image Classification.

ÖZ

İNŞAAT MAKİNESİ TESPİTİ İÇİN EVRİŞİMLİ SİNİR AĞLARI (ESA) TABANLI İKİLİ SINIFLANDIRICILAR

TATAR, Bahadır
İnşaat Mühendisliği Yüksek Lisans

Danışman: Dr. Öğr. Üyesi Seda YEŞİLMEN
Ortak Danışman: Dr. Öğr. Görevlisi Halil Fırat ÖZEL

Eylül 2022, 107 sayfa

İnşaat faaliyetlerinin yapay zeka ile izlenmesi şantiye operasyonlarındaki verimlilik için önemli bir vazifedir. Bu nedenle işlenen konu literatürde oldukça ilgi görmüştür. Farklı çeşitlilikteki görevleri izleyerek ve tespit ederek inşaat alanlarındaki operasyonları başarılı bir şekilde eniyileştirmek, şantiye işlerinde kullanılabilen araçları belirlemede önemli bir rolü olan şantiye alanının boyutuna bağlıdır. Yapay zeka algoritmalarının inşaat makinelerini algılaması için eğitilerek, görüntü sınıflandırma algoritmaları aracılığıyla geniş alanları kapsayan bir izleme görevi yüksek verimlilikle gerçekleştirilebilir. İnsansız hava araçlarından alınan görüntülerin kullanılması çok geniş bir bölgedeki inşaat operasyonlarını tespit etme açısından verimsiz kalabilir. Dolayısıyla bu tezde, iş makinelerinin tespit edilmesi için uydu görüntüsü sınıflandırılması yapılmıştır. Evrişimli sinir ağlarını eğitmek için Google Earth kullanılarak sıfırdan oluşturulan ve inşaat makineleri görüntüleri içeren bir veri seti oluşturulmuştur. Toplamda 23 adet önceden eğitilmiş evrişimli sinir ağı modeli öğrenme aktarımı yöntemi kullanılarak modifiye edilmiştir ve performansları değerlendirilmiştir.

Anahtar Kelimeler: Derin Öğrenme, Evrişimli Sinir Ağları, Öğrenme Aktarımı, Nesne Algılama, Uydu Görüntüsü, Görüntü Sınıflandırma

ACKNOWLEDGEMENT

I would like to express my deepest and incomparable gratitude to my family who built the path I walk and Assist. Prof. Dr. Seda YEŞİLMEN, who is the brightest light that illuminates my path.

I would like to thank Dr. Halil Fırat ÖZEL for his unique suggestions on my thesis journey.

I would like to thank my dear friend Hüseyin Basri ÇERÇİ, who supported me whenever I needed help during my thesis work.

I would like to express my gratitude to Halil İbrahim ANDİÇ and Kadir Can ERKMEN for contributing to my undergraduate and graduate journey with their vast knowledge and experience.

I convey my regards to the esteemed members of the thesis jury, who have spared their precious time and honored this thesis with their valuable perspectives.

This thesis would never have existed without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

AI	:Artificial Intelligence
ANN	:Artificial Neural Networks
CM	:Construction Machine
CMD	:Construction Machinery Dataset
CNN	:Convolutional Neural Network
COCO	:Common Objects in Context
CPU	:Central Process Unit
DT	:Decision Tree
FN	:False Negative
FP	:False Positive
GPU	:Graphics Processing Unit
ILSVRC	:ImageNet Large Scale Visual Recognition Challenge
IOU	:Intersection Over Union
NCM	:Non-Construction Machine
ReLU	:Rectified Linear Unit
RF	:Random Forests
SGD	:Stochastic Gradient Descent
TN	:True Negative
TP	:True Positive
UAV	:Unmanned Aerial Vehicle

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

The increasingly popular term deep learning is essentially a machine learning technique. Many artificial neurons are brought together and artificial neural networks are created. This network, which has a similar structure to the human brain, analyzes the data and makes its own inferences. A neural network needs a sufficient amount of data to work efficiently and properly. This data can be numerical values, images, or sounds. With these datasets, artificial neural networks are trained and an artificial mind is created that can respond to input data to obtain an output.

Studies in convolutional neural networks are always taking artificial intelligence one step further. While the world is in great global digitalization, civil engineers living in the 21st century should also contribute to artificial intelligence studies. As the power of artificial neural networks increases, the potential research topics that will arise in civil engineering increase. Computer vision is a system that enables to develop real-world applications with artificial intelligence supported software. Visual data such as city surveillance camera records, digital camera images, satellite images, and video recordings are used to train artificial intelligence. In other words, computer vision aims to experience seeing and sensing like a real human eye. Figure 1.1 shows an artificial neural network architecture.

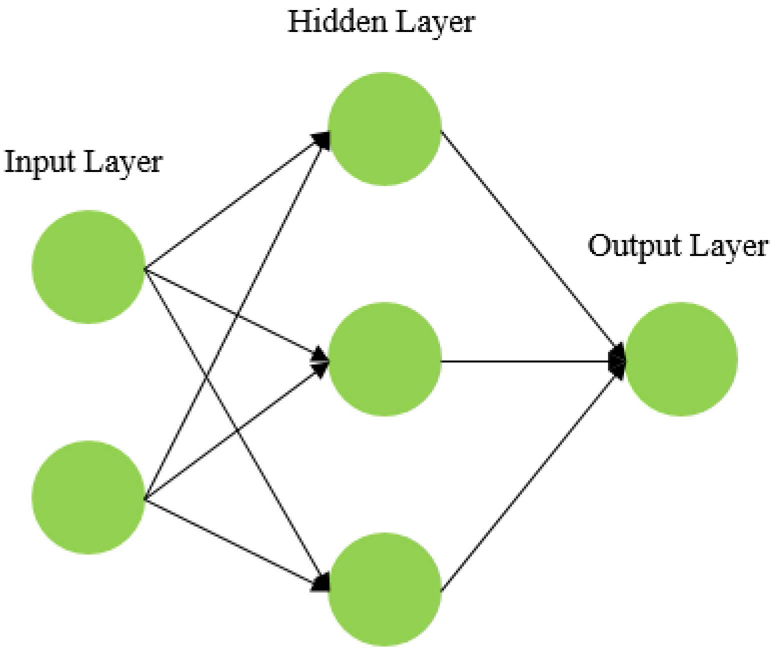


Figure 1.1: An Artificial Neural Network Architecture.

Artificial neural networks are deep learning algorithms consisting of digital neurons connected to each other. The neural network consisting of input, hidden, and output layers is processed with algebraic calculations called forward propagation and backward propagation. In addition, convolutional neural networks are a subset of artificial neural networks. Just like artificial neural networks, it consists of input, hidden, and output layers. The biggest advantage of convolutional neural networks is the convolutional layers. Convolutional layers produce a feature map of each image entering the network. Thus, specific details of the images are processed to be learned. Developing and implementing civil engineering applications with artificial intelligence become important in different professions in the literature. Some of these studies include predicting the diameters of jet grouted columns [1], predicting the residual flexural strength of fiber-reinforced concrete [2], and predicting the drying shrinkage of alkali-activated blast furnace-fly ash mortars [3].

When the research topic is construction sites, different priorities need to be examined. While it is undeniable that artificial intelligence will contribute to the construction industry technologies, this important power can also be used to make construction sites safer. Construction sites are very prone to major accident risks and unfortunately, these accidents can also be fatal. 20% of all fatal accidents in the USA are caused by accidents on construction sites [4].

It has always been a well-known fact that construction machinery has always created a great risk environment on construction sites. To avoid accidents during work on the construction site, there must be perfect harmony between the construction machinery operators and the workers in the field [5].

1.2 PROBLEM STATEMENT

In the time period from the past to the present, humanity has tended to build or demolish a structure in different areas, especially the need for shelter. Primitive equipment used for drilling, cutting and shaping in the first ages of humanity has left its place to technological machines over time. Larger living spaces have been created with larger construction sites to meet the needs arising from the increase in the human population.

In construction sites, there must be perfect harmony between managers, design engineers, field engineers, workers, and construction machine operators. Especially with the increase in the number of equipment used in construction sites belonging to large infrastructure projects, a detailed examination of the types of construction machines and their location in the region can be a factor that increases the efficiency of the project. In addition, construction site monitoring applications performed with drones have brought a different perspective to this issue.

Today, construction machines are not only used in legal construction but also operate in many areas such as illegal sand mining. Considering both the legal and illegal use of construction machinery, approaches such as camera systems and drone applications may be insufficient for detecting and tracking construction machinery.

Considering the global impact of satellite images, it is seen that satellite images have access to points that cameras and drones cannot reach. This thesis aimed to save manpower and time by aiming to eliminate the inadequacies and problems in real-world applications of construction machine detection with the contribution of using satellite images.

1.3 OBJECTIVE

Construction site monitoring applications supported by artificial intelligence are one of the most important ways to safely analyze the process from the very beginning to the end of a project. Although today's applications are generally carried out by drones, it is a very possible scenario that drones can be insufficient at some point. Some of these inadequacies may be due to the fact that drones, unlike satellite imagery, require a pilot and there is a limit to their hovering capacity caused by refueling. A solution that eliminates the manpower factor, such as using satellite imagery, could make construction site monitoring faster and cheaper.

The ultimate objective of this thesis is to detect construction machines in different locations with artificial intelligence using satellite images. In this thesis, two classes, 'construction-machine' and 'non-construction-machine', were determined for the classification of satellite imageries. Thus, satellite imageries that were fed to artificial neural network algorithms as input were classified into two separate classes 'construction-machine' or 'non-construction-machine' as output.

1.4 CONTRIBUTIONS TO THE LITERATURE

The contributions of this thesis to the literature are listed below.

- Detection of construction machinery using satellite images for the first time in the literature.
- Creating the Construction Machinery Dataset from scratch, which was collected only from satellite imagery.
- Comparison of the performances of different open source deep learning models.

1.5 THESIS STRUCTURE

Chapter II includes a literature review that flows from general studies to specific studies. In this chapter, firstly, general artificial intelligence studies are explained in chronological order. Afterward, image classification and computer vision studies, which are more detailed subjects, are examined. Finally, a review was made about the most special concept, the construction vehicle detection studies.

In Chapter III, after giving detailed information about the dataset used in this thesis and giving the models, general evaluation metrics are shown. Finally, this chapter is concluded with hardware and software information.

In the Chapter IV, detailed training results are given in both tables and graphs, along with general explanations of each artificial intelligence model trained for this thesis.

In the Chapter V, six satellite images of different difficulty levels were shown to the models for the first time and the responses of the models were examined.

Finally, in chapter VI, a conclusion has been prepared in which all the lines of this thesis are explained. In addition, after discussing how the obtained results in this study can be improved, information about potential future works is given.

CHAPTER II

LITERATURE REVIEW OF ARTIFICIAL INTELLIGENCE STUDIES

2.1 STATE-OF-THE-ART STUDIES

Computer vision technology has been in great development and rise in recent years. It is used in almost every field, from the prevention of fatal accidents in the construction industry to self-driving cars in the automobile industry, from space missions to diagnostics in the medical field, from finding the traffic speed in the transportation industry to quality control in manufacturing [6]. Although the methods, datasets, and purposes change, the only thing that does not change is that computer vision automates many works and industries rather than being manual. The development of computer vision studies that has started with the leadership of SIFT + FVs, continue with AlexNet, ZFNet, Five Base + Five HiRes, SPPNet, VGG-19, Inception V2, ResNeXt-101, Dual Path Network, NASNet-A (6), PNASNet-5, AmoebaNet- A, ResNeXt-101 32x48d, FixResNeXt-101 32x48d, EfficientNet-B7, BiT-L (ResNet), EfficientNet-L2, ViT-H / 14, EfficientNet-L2-475 and Meta Pseudo Labels (EfficientNet-L2), respectively.

When computer vision and deep learning are included as research topics, a strong dataset must be used. ImageNet is one of the most popular datasets and makes a great contribution to the training and evaluation of artificial intelligence models. ImageNet is a quite large dataset with approximately 1000 categories of data created for image processing and composed of more than 1 million labeled high-resolution images collected from the internet. Human taggers were preferred to label each image on ImageNet. Amazon's Mechanical Turk crowd-sourcing tool which is named after The Turk, the first Automation Chess Player in history, was used for the labeling process [7] [8]. ImageNet is one of the most important sources for the global evaluation of image classification models. The state-of-the-art models participate in the ImageNet

Large Scale Visual Recognition Challenge (ILSVRC) since 2012 and demonstrate their strength to be the most powerful image classification model ever created.

The magnificent era of ImageNet in computer vision and image processing began with Fisher Vector and Scale Invariant Feature Transform (SIFT + FVs) in 2011 [9]. SIFT, Scale Invariant Feature Transform, detects distinctive features in images and can offer reliable matching to images in very large-scale datasets. SIFT, also defined as an identifier, is used for object recognition applications [10]. The Fisher Vector used in the image classification splits images into parts and performs the algorithm for each part. Square-rooting and L2-normalizing the Fisher Vector provide high classification accuracy. Bag-of-visual-words (BOVW) is a feature extraction method used in image classification algorithms [11]. A significant advantage concerning the bag-of-visual-words is that high-dimensional discriminative signatures can be obtained even with small vocabularies, and therefore at a low CPU cost [12].

The current state-of-the-art model changed when the AlexNet model was announced in 2012. The importance of deep convolutional neural networks for the future of image processing technology was seen with AlexNet. The recently developed dropout regularization method was used to reduce overfitting in this fully supervised learning model. Another important achievement of AlexNet is its first place in the ILSVRC-2012, ImageNet Large Scale Visual Recognition Challenge, competition with a top-5 test error rate of 15.3% [7].

The ZFNet model was published in 2013 and it was examined why convolutional neural networks are such an effective method. Thanks to their novel techniques that visualize the activity within the model, they obtained a model that performed better than *Krizhevsky et al.*'s AlexNet model. Moreover, this study introduced how existing models can be made more efficient [13].

When Andrew G. Howard published their article in 2013, Some Improvements on Deep Convolutional Neural Network Based Image Classification, they presented three methods that will improve the existing state-of-the-art structure. First, the short edge of the images was adjusted to 256 pixels with the method of adding more image transformations to the training data. 224x224 pixels images were randomly cut from these images for later use in the training dataset. Secondly, they emphasized the efficiency of three different scaling operations on predicting by adding more transformations at the test time method. The last method is the higher resolution models method. With this method, for higher resolution models, they cut 128x128 pixels sized patches from the images with the short edge of 256 pixels instead of adjusting the short edges of the images to 448 pixels and cutting patches of 224x224 pixels. They then rescaled the 128x128 pixels sized images to 224x224 pixels. Thus, the Five Base + Five HiRes model became the new state-of-the-art [14].

SPPNet convolutional neural network model, which took the first state-of-the-art throne in 2014, uses a new pooling layer, unlike other CNN models. This method, called Spatial Pyramid Pooling, creates a fixed-size model without focusing on the input image size. SPPNet extracts feature maps of any image once and saves large processing times. SPPNet's incredible approach to pooling layers contributed to all convolutional neural network-based image classification methods. This approach greatly increased the accuracy of CNN models [15].

According to *Simonyan et al.*, the depth of an algorithm is a fact that positively affects large-scale image classification accuracy. This great approach made the VGG-19 the new state-of-the-art model with its success in ImageNet Challenge 2014 [16].

The deepening of a deep learning network causes a snowball effect in the parameters of the neural networks. The reason is that each layer in neural networks is fed by the previous layers. Due to this fact, the change of input distribution directly affects all connected layers and makes it difficult to train a deep neural network. It has been stated that the results found on mini-batches based on a group logic are more efficient than individual calculations. Main focus of Inception V2 architecture is the normalization of activations for use directly in the network structure. Thus, Inception V2 became the state-of-the-art model of 2015 with the acceleration created by the idea of batch normalization [17].

ResNeXt-101 is constructed by repeating a building block that aggregates a set of transformations with the same topology. In addition to the depth and width dimensions, another dimension called cardinality, which is the size of the set of transformations, was revealed in this study. It has been observed that if cardinality increases while model complexity is constant, classification accuracy increases. In the experiments conducted in this study, it was observed that the accuracy does not only increases when the model complexity is constant but also increases when the number of parameters is constant. In the light of all these results, the ResNeXt-101 model will inevitably be a state-of-the-art model in 2016 [18].

Dual Path Network (DPN) is an architecture designed for image classification. In that study takes advantage of residual neural networks and densely connected networks. Dual Path Network performs high accuracy with low computational cost, has a small model size, and low GPU memory consumption. Thanks to these advanced features, the Dual Path Network has become the state-of-the-art model of 2017 [19].

In the study in which NASNet-A (6), Neural Architecture Search Networks, is introduced, an architecture that learns the required model structure in its own dataset is presented and showed how insignificant the human factor is in building neural networks. An architecture was introduced that creates its own neural networks by focusing on the main point in the dataset instead of manually created neural networks as a result of human manipulation. Instead of starting this learning process directly in the ImageNet dataset, first, the best convolutional layer for the CIFAR-10 dataset is searched and then transferred to ImageNet. The space named NASNet Search Space is the most important point that performs this transfer procedure. As a result of the experiments, it was seen that the created architectures gave better results than all human-designed structures and became a state-of-the-art model [20].

PNASNet-5 (Progressive Neural Architecture Search) uses a method called sequential model-based optimization. This method speeds up the process of finding the best Convolutional Neural Network in the search space. This new approach makes PNASNet-5 much faster and much more efficient in terms of classification accuracy. PNASNet-5 becomes the state-of-the-art model by surpassing all of the latest models [21].

An Evolutionary Algorithm architecture that surpasses human-designed models was developed with AmoebaNet-A. Aging evolution which simulates the evolutionary logic of living beings has been suggested for AmoebaNet-A. During this period, a tournament selection environment was created. The process in which older genotypes died and younger ones survived was initiated. It was found that the Evolutionary Algorithm gave better results as a result of the search speed experiments conducted on the Evolutionary Algorithm, Reinforcement Learning Algorithm, and

Random Search Algorithm. Inevitably, AmoebaNet-A became the state-of-the-art model of 2018 [22].

Facebook AI team conducted a study with transfer learning to predict the hashtags of 3.5 billion Instagram photos. In the study, it was seen how high the large-scale hashtag prediction results were. It was emphasized that, unlike traditional methods, data cleaning is not a required procedure for the success of this study. It has been verified that selecting a logical label space rather than increasing the size of the dataset affects the success of the model, too. Thus, the image classification and object detection level of the ResNeXt-101 32x48d model surpassed the previous state-of-the-art model [23].

With the FixResNeXt-101 32x48d model proposed by Facebook AI, a way to significantly reduce the training time is presented. The proposed fast and cheap strategy is based on data augmentation. Since the importance of data augmentation for convolutional neural networks is known, it has been shown that classification accuracy will increase with correct parameter adaptation. Thus, the FixResNeXt-101 32x48d model became a state-of-the-art model in 2019 due to its superior performance and increased efficiency [24].

Which came first the chicken or the egg? Or, which came first a teacher or a student? These two questions, the answers of each are hidden within themselves, created the state-of-the-art model of 2019. With a semi-supervised learning approach named as NoisyStudent (EfficientNet-B7), the EfficientNet model is trained as a teacher on labeled images and creates pseudo labels for unlabeled images. Then, with these outputs, a stronger EfficientNet model is trained as a student. This figurative student, who is stronger than the teacher, is put in a loop again as a teacher. Thus, stronger students turn into stronger teachers. Thanks to this approach, the NoisyStudent (EfficientNet-B7) model has become a state-of-the-art model [25].

The two most important phenomena that affect the success of the BiT-L (ResNet) model are supervised pre-training and fine-tuning of the target tasks. The components of these phenomena for transfer learning efficiency are upstream and downstream, respectively. Due to the high performance not only in ImageNet but in more than 20 datasets, the BiT-L (ResNet) model became the state-of-the-art model [26].

With the combination of EfficientNet and Fixing Resolution, a new model called FixEfficientNet-L2 has been developed. Thus, higher performance was achieved without changing the number of parameters. Due to the nature of Fixing Resolution, it can work with any convolutional neural network structure to classify. Therefore, the EfficientNet-L2 model, optimized with FixRes, became the state-of-the-art model by eliminating all other models [27].

One of the biggest contributions of the ViT-H / 14 (Vision Transformer) model on computer vision applications is that Transformer Architecture has shown that it is not only a method for natural language processing but also a productive solution for computer vision. It has been observed that a series of images processed by the Standard Transformer Encoder give much more successful results on large datasets compared to previously interpreted images. This holistic approach has made the ViT-H / 14 the state-of-the-art model of the year 2020 [28].

The EfficientNet-L2-475 (SAM / Sharpness-Aware Minimization) model minimizes loss value and loss sharpness and improves generalization. SAM searches for parameters lying in neighborhoods. In the resulting case, the neighborhoods have low loss values and the parameters increase the success of the gradient descent. The success of SAM has been empirically evaluated on large datasets, and successful results have made SAM a state-of-the-art model [29].

Semi-supervised learning-based Meta Pseudo Labels (EfficientNet-L2) model includes a teacher and a student network. The purpose of the teacher in logarithms is to generate pseudo labels using unlabeled data. Then the teacher teaches this data to the student. Differently, the student's performance in labeled data is given to the teacher as feedback and the teacher is constantly updated. Due to its progressive approach, the Meta Pseudo Labels (EfficientNet-L2) model has become the state-of-the-art model as the best image classification algorithm ever for the ImageNet dataset [30].

2.2 IMAGE CLASSIFICATION AND COMPUTER VISION STUDIES

One of the most popular applications of computer vision studies is image classification and it can be used in almost every industry. Thus, image classification with satellite imageries has become a large research area. Satellite imagery classification is commonly used for land classification [31], image scene classification [32], vehicle detection [33] and tree species classification [34].

In the study by *Castelluccio et al.* [31], land-use classification was made with satellite images. UC-Merced Land Use Dataset and Brazilian Coffee Scenes Dataset were used as the dataset, while GoogLeNet and CaffeNet were used as convolutional neural network models. According to UC-Merced, the GoogLeNet model with fine-tuning design method has the highest accuracy at 97.10%. This proposed method also has the highest accuracy among other articles published in its area. According to the Brazilian Coffee Scenes, the GoogLeNet model with the from-scratch design method has the highest accuracy with 91.83%. This proposed method also has the highest accuracy among other articles published in this area.

In another study [35] conducted in 2015, satellite image classification methods are explained. The classification methods are divided into three parts. These are automated, manual, and hybrid methods. The automated method is divided into two supervised and unsupervised. It is stated that artificial neural networks, binary decision trees, and image segmentation are used in major supervised classification methods. They created a detailed table comparing the satellite image classification studies performed by nine different researchers. In that table, the classification methods, datasets, and which of the methods work more successfully are shown.

The aim of the study [36], published by *Sharma et al.* in 2017, is remote sensing image classification using the convolutional neural networks. The land cover classification was performed with Landsat 8 satellite images. Compared to the other four networks used for area classification, the deep patch-based convolutional neural network achieved 85.60% accuracy, while pixel-based network achieves 62.34%, pixel-based convolutional neural network achieves 63.01% and a patch-based neural network achieves 73.17%. According to the number of iterations, the accuracy is as follows, 9999 iterations: 72.93%, 49999 iterations: 79.12%, 89999 iterations: 83.06%, 129999 iterations: 84.67%, 149999 iterations: 85.60%.

According to the study of *Nogueira et al.* in 2017, they focused on three strategies for exploiting existing convolutional neural networks in different scenarios which are full training, fine tuning, and using as feature extractors. They experimented on six popular convolutional neural networks (OverFeat Networks, AlexNet, CaffeNet, GoogLeNet, VGG-16, and PatreoNet) and three widely used datasets (UC Merced Land Use, RS19, and Brazilian Coffee Scenes). Their objective was to understand the best way to obtain the most benefits from these state-of-the-art deep learning approaches to problems. According to the UC Merced dataset, CaffeNet, AlexNet, and VGG-16 achieved highest average accuracy values of over 93% compared to other global descriptors. According to the RS19 dataset, Convolutional Neural Networks gave the highest average accuracy values of above 90% compared to other global descriptors. According to the Brazilian Coffee Scenes dataset, BIC and ACC global descriptors gave the highest average accuracy values with 87.03% compared to convolutional neural networks. According to the experimental results, the fine-tuning method tends to be the best strategy under different scenarios [37].

Li et al. created a binary classification system as Cloud and Non-Cloud by modifying the VGG-16 network according to their own subjects in 2020. Among the different methods, the proposed WDCD method based on CAM with GCP + LPP gave the best overall accuracy and F1 Score with 0.9666 and 0.8855 [38].

CNN-Relief-SVM, a new hybrid feature extractor, has been developed for the recognition of satellite images. UC-Merced Land Use Dataset was used for the operations. The last fully connected layers of pre-trained architectures such as AlexNet, VGG16, VGG19, GoogLeNet, ResNet, and SqueezeNet were used to create this model. The features obtained from the last layers were given to the support vector machine separately and their classification performances were measured. Accuracy was measured as 99.29% for 80% training ratio and 98.76% for 50% training ratio [39].

Khan et al. modified the ResNet-50 in 2020 to process large-scale images at once, instead of processing images in small-scale patches. They replaced ResNet-50's prediction layer classifier with a 1x1 convolutional layer classifier to process satellite images of any size. They trained the model with satellite images collected from Google Earth. The proposed model reduces the processing time by 99.9% by keeping the accuracy at the same level. For example, while ResNet-50 in a sliding window manner processes an image with an image size of 14882x14848 in 11.9 hours, the proposed model processes in 39 seconds [40].

This [32] review article discusses papers with more than 160 deep learning methods for remote sensing image scene classification. They discussed different methods under three headings. These are Autoencoder-Based Remote Sensing Image Scene Classification, CNN-Based Remote Sensing Image Scene Classification, GAN-Based Remote Sensing Image Scene Classification. They give brief information about 13 datasets. These are UC Merced, WHU-RS19, RSSCN7, Brazilian Coffee Scene, SAT-4 / -6, SIRI-WHU, RSC11, AID, NWPU-RESISC45, RSI-CB 128 / -CB256, OPTIMAL-31, EuroSAT, BigEarthNet. Then, detailed information about UC Merced, AID, NWPU-RESISC45, and overall accuracy values are given. CNN-based CNN-CapsNet method gives the highest accuracy with 97.59% at 50% training ratio in UC Merced Dataset, while the convolutional neural network-based ADSSM method gives the highest accuracy with 99.76% at 80% training ratio. The CNN-CapsNet method gives the highest accuracy with 93.79% at a 20% training ratio in AID Dataset, while the CNNs-WD method gives the highest accuracy with 97.24% at a 50% training ratio. Hydra method gives the highest accuracy with 92.44% at a 10% training ratio in NWPU-RESISC45 Dataset, while the DNE method gives the highest accuracy with 96.01% at a 20% training ratio.

In the study [41], a new convolutional neural network based on single shot multi-box detector (SSD), to detect vehicles on high-resolution images is proposed in 2020. They compared the proposed model with the Faster R-CNN, SSD300, SSD512, and YOLOv3. The proposed model achieved an average precision of 90.40%, higher than other models. They used UCAS-High Resolution Aerial Object Detection Dataset in this study. They added a batch normalization layer to the detection layers in the detection module to prevent overfitting and increase system speed.

A deep learning algorithm is used by *Tan et al.* to detect vehicles in high-resolution satellite remote sensing images. They used the AlexNet model to classify satellite images. Then Faster R-CNN algorithm is tested and optimized by the method of model pruning and quantization. After these actions, the model is applied to the practical application of vehicle detection at an intersection. According to the study results, the rate of missed detection and false detection was 0%. The highest values are 4.5% and 2.7%. The values show that vehicle detection based on deep learning has high accuracy [42].

Advanced deep learning techniques, multilevel feature fusion, and sample mining are investigated in 2020 to realize vehicle detection in remote sensing images. They presented CycleGAN-like to realize simultaneous super-resolution and object detection for low-resolution images. They used four datasets which are Potsdam, VEDAI, DLR Munich, and UCAS-AOD. According to VEDAI Dataset, the VDM method has the highest average precision with 0.458, the highest average precision of 0.856 with 0.5 IoU (Intersection over Union) threshold, the highest average precision of 0.457 with 0.75 IoU threshold, and the highest mean recall rate (mRecall) with 0.573. According to Potsdam Dataset, the VDM method has the highest average precision with 0.668, the highest average precision of 0.793 with a 0.75 IoU threshold, and the highest mean recall rate with 0.740. YOLOv3 has an average precision value of 0.904 with the highest threshold of 0.5 IoU. According to Munich Dataset, the CVDM method has the highest average precision with 0.599, the highest average precision of 0.889 with 0.5 IoU threshold, the highest average precision of 0.684 with 0.75 IoU threshold, and the highest mean recall rate with 0.648. According to UCAS-AOD Dataset, the CVDM method has highest average precision with 0.572, the highest average precision of 0.885 with 0.5 IoU threshold, the highest average precision of 0.637 with 0.75 IoU threshold, and the highest mean recall rate with 0.653. Their study showed that their system surpasses state-of-the-art methods [43].

In the study [44] conducted in 2020, a multi-source active fine-tuning vehicle detection (Ms-AFt) framework is proposed. Ms-AFt contains transfer learning, segmentation, and active classification. The proposed model employs fine-tuning network to generate a vehicle training set from the unlabeled dataset. Then, a multi-source-based segmentation branch is designed to construct additional candidate object sets. Open ISPRS datasets were used for this study. VGG-19, GoogLeNet, Cascadenet18, and ResNet18 were used to compare classification performances. According to ISPRS Vaihingen Dataset, ResNet18 gave the best precision, recall, and F1-Score values with 0.96, 0.86, and 0.91. According to ISPRS Potsdam Dataset, ResNet18 gave the best precision, recall, and F1-Score values with 0.99, 0.90, and 0.92. According to DLR SAI-LCS Dataset, ResNet18 gave the best precision, recall, and F1-Score values with 0.97, 0.76, and 0.86. When SSD-Ms-AFt, YOLO1-Ms-AFt, YOLO2-Ms-AFt, FRCNN-A-Ms-AFt, FRCNN-B-Ms-AFt, and R-FCN-Ms-AFt methods are compared according to the threshold value of 0.6, R-FCN-Ms-AFt gave the best precision, recall, and F1-Score values of 0.4759, 0.8012, and 0.5971 in ISPRS Vaihingen Dataset, 0.5779, 0.9106, and 0.7071 in ISPRS Potsdam Dataset, 0.8313, 0.8612, and 0.8463 in DLR SAI-LCS Dataset. According to the performance values, it is seen that the Ms-AFt method gives the best results compared to Fine-Tuning, Segmentation & Attention, and VIS-AFt methods in all three datasets.

The process of counting plants and detection of plantation-rows operations were performed by using Cornfield and Citrus Orchard datasets with the VGG19 model in 2021. In the corn plantation dataset, their mean absolute error (MAE) is 6.224, mean relative error (MRE) is 0.1038, precision and recall values are 0.856 and 0.905, and F-Score is 0.876. For the plantation-row detection, their precision, recall, and F-Score scores are 0.913, 0.941, and 0.925. In the Citrus Orchard dataset, their MAE is 1.409 citrus trees, MRE is 0.0615, precision is 0.922, recall is 0.911 and F-measure is 0.965. For the citrus plantation-row detection, their precision, recall, and F-Score are 0. 965, 0.970 and 0.964. Compared to HRNet, Faster R-CNN, and RetinaNet networks, the proposed approach in this study provides superiority over all of them [45].

Remote sensing scene classification still struggles to overcome some difficult tasks. *Bi et al.* proposed a multiple scale staking attention pooling called MS2AP to solve these difficulties and classification of satellite images in 2021. Three datasets, UC-Merced, AID, and NWPU, were used in this study. They used 50% and 80% training ratios in the UCM dataset, 20% and 50% training ratios in the AID dataset, 10% and 20% training ratios in the NWPU dataset. They verified their MS2AP with AlexNet and VGG-16 CNN models. The overall accuracy values in the UCM dataset are 98.38% at a 50% training ratio and 99.01% at an 80% training ratio for Alex_MS2AP. For VGG_MS2AP, 99.09% at 50% training ratio and 99.45% at 80% training ratio. The overall accuracy values in the AID dataset are 92.19% at a 20% training ratio and 94.82% at a 50% training ratio for Alex_MS2AP. For VGG_MS2AP, 95.42% at 20% training ratio and 96.86% at 50% training ratio. The overall accuracy values in the NWPU dataset are 87.91% at a 10% training ratio and 90.98% at a 20% training ratio for Alex_MS2AP. For VGG_MS2AP, 92.27% at 10% training ratio and 93.91% at 20% training ratio [46].

A new RS-DCNN method [47] is introduced for processing large satellite images in 2021. The proposed approach consists of two main parts. First, to create a training set by dividing large satellite images into small pieces, and then to process a supervised classification algorithm called Maximum Likelihood. The second is to use RS-DCNN to classify large satellite images. To achieve parallelism, they used asynchronous distributed stochastic gradient descent. They obtained large satellite images from SPOT-6/7 sensors. Compared to generative adversarial networks (GAN), random forests (RF), artificial neural networks (ANN), and decision tree (DT), RS-DCNN gives the highest overall classification accuracy with 92.06%. Kappa value of RS-DCNN is 0.883.

In this article [48] conducted in 2021, SceneNet is proposed. The proposed SceneNet is an evolutionary algorithm-based neural architecture search approach for the remote sensing image scene classification task. The most efficient network is automatically determined based on the dataset. No human manipulation is required in this process. SceneNet was compared to AlexNet, VGG16, ResNet34, and GoogLeNet, designed by human experts, to prove its design approach. UC Merced, NWPU45, and AID were used as a dataset in the experiments. SceneNet_UCM achieved the highest overall accuracy with 99.10% in the UC Merced dataset. The Kappa value of SceneNet_UCM is 0.9905. SceneNet_NWPU45 achieved the highest overall accuracy in the NWPU45 dataset with 95.219%. The Kappa value of SceneNet_NWPU45 is 0.9511. SceneNet_AID achieved the highest overall accuracy in the AID dataset with 89.58%. The Kappa value of SceneNet_AID is 0.8927.

The purpose of the study [34] conducted in 2021 is the classification of the major tree species scats pine, Norway spruce, birch, and European aspen. They compared the performance of 3D-CNNs with the support vector machine, random forest, gradient boosting machine, and artificial neural network in individual tree species classification. They collected hyperspectral and LiDAR data from the study area located in the southern boreal zone in Finland. The best-performing 3D-CNN achieved an F1-Score of 0.91 for aspen, an overall F1-Score of 0.86, and overall accuracy of 87%, while the lowest-performing 3D-CNN achieved an F1-Score of 0.83 and accuracy of 85%. The support vector machine achieved an F1-Score of 0.82 and an accuracy of 82.4%. The artificial neural network achieved an F1-Score of 0.82 and an accuracy of 81.7%. According to these results, 3D-CNN is the most efficient option.

In the study [49], the main fact is vehicle detection in aerial images based on deep neural networks and 3D feature maps. They also investigated the effect of 3D feature maps in increasing the performance of DNN structures. They used YOLOv3, which they modified with different structures such as Darknet-53, SqueezeNet, MobileNet-v2, and DenseNet-201, to detect trucks, semi-trailers, and trailers. For this study, they used the dataset they obtained using UAV. According to the results, Darknet-53 gave the most successful result with 93.4% precision. It is shown that 3D features improve the performance of vision-based deep neural networks and its F1-Score is 95.72%. 3D features improved the precision of DNNs from 88.23% to 96.43% and from 97.10% to 100%.

2.3 CONSTRUCTION VEHICLE DETECTION STUDIES

Few articles conduct construction vehicle detection research [50] [51] [52]. Mainly drone images were used and high accuracy was achieved in these articles [33]. None of these studies perform construction vehicle detection using satellite imagery. *Arabi et al.* proposed the SSD MobileNet object detection model, which is suitable for embedded devices, to detect construction vehicles in 2020. ImageNet, Common Objects in Context (COCO), and Open Image are used as large-scale datasets. AIM dataset, which is a subset of ImageNet, is also used for its construction machine images. AIM dataset contains street view of excavators, loaders, rollers, concrete mixer trucks, and dump trucks images. The average precision values of the model are 92.31% for dump truck, 83.70% for excavator, 93.86% for grader, 93.77% for loader, 96.94% for mixer truck, and 86.65% for roles. The mAP value is 91.20% [50].

The purpose of this study [51] conducted in 2018 is to detect excavators and workers on construction sites using Improved Faster Regions with Convolutional Neural Network Features. The proposed model’s accuracy for workers and excavators is 91% and 95%, respectively. Precision and recall values for workers are 98% and 79%. 99% and 81% for excavators. A custom dataset created by authors with images collected from construction sites to train CNN model is used in the study. The proposed method can also detect unsafe actions and unsafe conditions.

Although computer vision technologies have been developed to automate the investigation of construction sites or building environments, using computer vision technologies in real construction projects remains a major challenge. 119 articles have been reviewed by *Kim et al*. According to the table of existing object detection and tracking algorithms given in the study, the best construction equipment detection performance belongs to single shot multi-box detector with 98.8% accuracy [52].

Fixed-placed cameras with a certain angle that can monitor the construction site perform construction vehicle classification and environmental risk analysis cheaper and more practical. Hence, orientation-aware feature fusion single-stage detection (OAFF-SSD) is created by *Guo et al*. In the study, VGG-16 was used as a feature extraction module. The modified version of VGG-16 was used instead of the original version. First, the kernel size and stride of the fifth max pool layer are transformed from (2×2, 2) to (3×3, 1). The sixth and seventh fully connected layers are transformed into convolutional layers with 3×3 and 1×1 kernels. Two more convolutional parts were added to the five convolutional parts of the original VGG-16, the sixth, and the seventh convolutional parts. These two parts are using 1×1 and 3×3 kernels for the first and second layers. The stride in the second layer of the sixth convolutional part is set to 1 so that the feature map size in the second layer of the seventh convolutional part is 10×10. The third layer of the fourth convolutional part, the seventh fully connected layer, and the second layer of the seventh convolutional part are used as the fusion base layers. Only layers larger than 10×10 sizes are selected because feature maps smaller than 10×10 have a negligible effect on fusion effects. Batch normalization has a great contribution because its effect reduces training time and improves prediction accuracy. Two important hyperparameters, β, and δ were used to measure the success of the model. β is the threshold of OA-IOU and δ is the threshold of intersection over union. Intersection over Union (IOU) is a common evaluation method for object detection models. To get the maximum benefit from the model, β and δ values were found to be 0.25-0.35 and 0.2-0.3, respectively. In the light of this information, the average precision value was found as 98.8% [33].

CHAPTER III

MATERIALS AND METHODS

In this thesis, construction machinery detection is performed using satellite image classification. The custom dataset consists of different kind of construction machinery placed on different ground types is created from scratch using Google Earth. The types of construction machinery present in the dataset are excavator, backhoe, boom lift, articulated hauler, dumper, bulldozer, grader, mobile crane, wheel loader, and skid steer loader. Satellite images from various states of The United States of America were used to train convolutional neural networks with various architectures.

Table 3.1: Detailed Information of Construction Machinery Dataset.

Class	Training	Validation	Testing	Total
Construction Machine	1494	186	186	1866
Non Construction Machine	1012	186	186	1384

In the dataset, there are 3250 satellite images in total, 1866 of which belong to the 'construction-machine' class and the rest 1384 belong to the 'non-construction-machine' class. The Construction Machinery Dataset (CMD) was split into training, validation, and testing dataset. 80% of the total dataset was dedicated to training. The validation and testing datasets were equally divided into two groups. The remaining 20% was dedicated to validation as 10% and testing as 10%. Figure 3.1 is an example of construction machinery class, while Figure 3.2 is an example of non construction machinery class. Table 3.1 summarizes the overall data-splitting details. In addition, the maximum and minimum dimensions of the satellite images in the dataset were examined in Table 3.2.

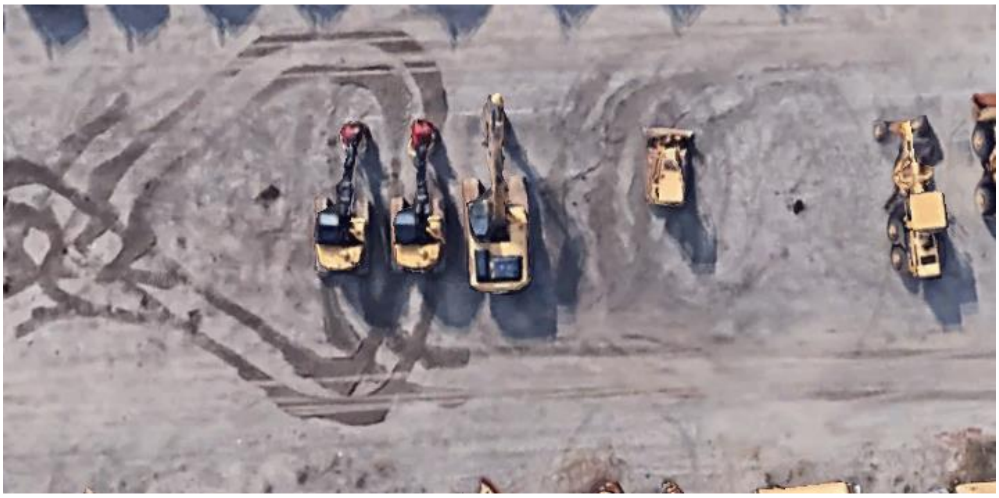


Figure 3.1: An example satellite image of construction machinery class in the dataset.

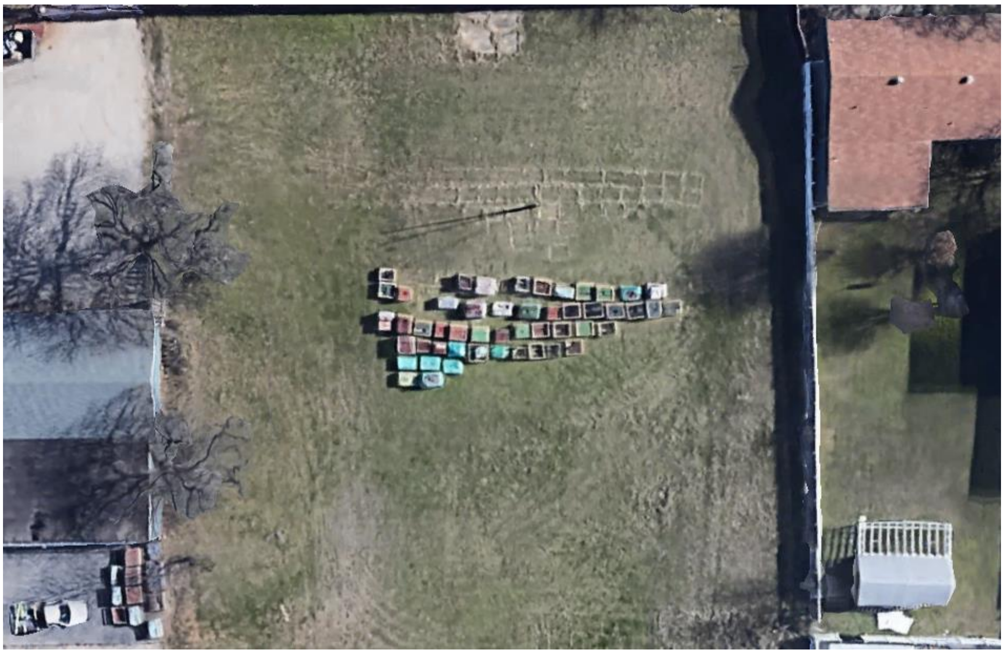


Figure 3.2: An example satellite image of non construction machinery class in the dataset.

Table 3.2: Detailed Image Properties of Construction Machinery Dataset.

Class	Maximum Height (Pixel)	Minimum Height (Pixel)	Maximum Width (Pixel)	Minimum Width (Pixel)
Construction Machine	810	77	1846	69
Non Construction Machine	807	60	1821	83

Python, a popular programming language, is often preferred for training and testing deep learning algorithms. PyTorch, an open-source deep learning framework, was used in this thesis. The coding processes were performed with the PyTorch library and PyTorch's built-in functions. In addition, the following models were prepared using the PyTorch library. Open-source pre-trained artificial intelligence models provide advantages in terms of both time and efficiency. For this reason, 23 different open-source pre-trained models are used in this thesis. These models are AlexNet, VGG16, VGG19, ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, MobileNetV2, MobileNetV3 Large, MobileNetV3 Small, DenseNet121, DenseNet161, DenseNet169, DenseNet201, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5, EfficientNetB6 and EfficientNetB7. Therefore, this thesis focused on these popular models. The number of parameters of each model used in this thesis is available in Table 3.3. Figure 3.3 shows a neural network architecture to perform construction machinery classification.

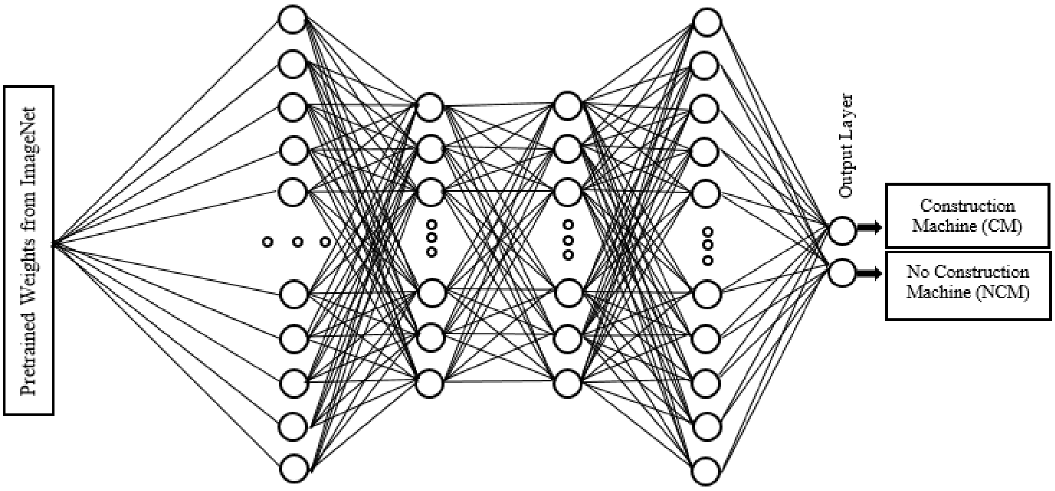

Pretrained Weights from ImageNet
Output Layer
Construction Machine (CM)
No Construction Machine (NCM)

Figure 3.3: A Neural Network Architecture of Construction Machinery Classification.

23 different pre-trained deep learning models have been utilized. The weights of the models come directly from ImageNet. When working with pre-trained models, a common method used in the literature is transfer learning. Transfer learning is the reuse of pre-trained models on a new dataset by modifying them according to the purpose of the study. Figure 3.6 shows the transfer learning scheme.

The stochastic gradient descent optimizer was used to update the weights. In addition, early stopping was used in this study. Early stopping is a regularization method that ends the training at an efficient point by following the validation data and realizing that the model does not improve during the training.

AlexNet architecture consists of 8 layers, 5 convolutional and 3 fully connected layers. In addition, AlexNet uses ReLU as activation function. The numbers at the end of the VGG16 and VGG19 models indicate the number of layers in the relevant model. VGG16 has 13 convolutional layers and 3 dense layers, while VGG19 has 16 convolutional layers and 3 dense layers. ResNet architectures have residual blocks that allow one layer to be connected to more than one other layer. There are 5 different variants of the ResNet architecture in this thesis, namely ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152. The numbers at the end of each ResNet architecture indicate the total number of layers in the model.

There are linear bottlenecks between the layers in the MobileNetV2 model. Bottlenecks are connected to each other by shortcuts. This collaborative architecture enables MobileNetV2 to train faster and generate successful accuracy values. The difference of MobileNetV3 architectures from MobileNetV2 is that MobileNetV3 uses fewer layers and reaches the accuracy values that MobileNetV2 reaches.

The feature that distinguishes DenseNet architectures from traditional CNN architectures is the number of layers. In a traditional CNN model, there is one connection between each layer. This means that there are the same number of connections as the number of layers. However, this situation is different in DenseNet architectures. Each layer in DenseNet architectures is connected to all subsequent layers.

Until EfficientNet architectures emerged, scaling image width, image depth and image resolution on classical CNN models was not an automatic process. Thanks to the compound scaling method used in EfficientNet architectures, not every input image is treated the same. Instead, scaling is made to reveal the efficiency of the model in the best way.

Transfer learning applications in image classification studies is quite popular. Transfer learning methods are increasing the performance of models, easy to implement, and fast to work with pre-trained models [53]. Transfer learning is commonly applied in the literature. *K. Nogueira et al.* [37], *S. N. Khan et al.* [40], and *S. Javadi et al.* [49] are examples of transfer learning studies in the literature. At the end of this thesis, the performances of each pre-trained model are compared.

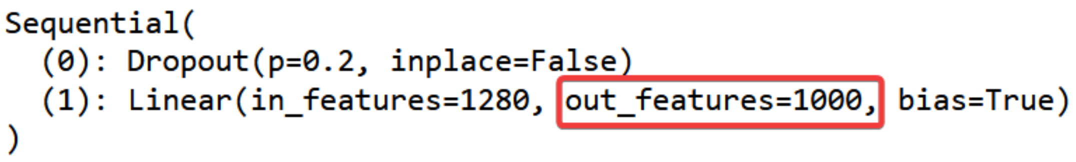
The detailed detection of objects that are the target of a study in neural network algorithms is specialized in the last layer of the network. This means that in order to apply transfer learning in a new study, it is necessary to change the last layers of a model.

Since all of the pre-trained models in this thesis were pre-trained with the ImageNet dataset with 1000 classes, the classifier layers were not suitable for binary classification for the new classes. One of the most important reasons for using pre-trained models is to take advantage of the weights of the models trained with the ImageNet dataset.

In order to perform the transfer learning process, the classifier layers of all models were removed and the layers in Figure 3.5 were added instead. The classifier layer structure of all models in this thesis is not the same. While the classifier layers of ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, DenseNet121, DenseNet161, DenseNet169, and DenseNet201 models are linear layers, all other models are sequential layers. The same layers were added to each architecture in this thesis because it is valuable for this study to evaluate the response of different models to the same transfer learning method.

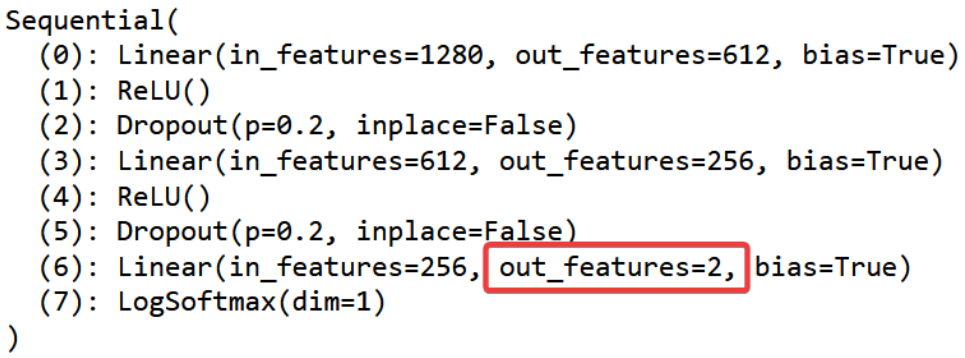
The nn.Linear module creates a layer with a certain number of inputs and outputs. Thus, the number of features belonging to the previous layer from nn.Linear can be transferred to new layers. ReLU, with its long name, Rectified Linear Unit is an activation function. The biggest feature of ReLU is that it converts negative values to zero. Thus, ReLU's computational cost will be less than other activation functions, so it works faster. Dropout is a regularization method used to prevent overfitting, which is very common in artificial neural network applications. Logarithmic softmax is another activation function. The biggest advantage of the logarithmic softmax used in the classifier layer over softmax is that it penalizes the model due to its mathematical nature when the model makes an incorrect class prediction. The purpose of using these layers is to make predictions for the new dataset using features from ImageNet.

For example, in the last classifier layer of the MobileNetV2 model, the part with 1000 classes of output specialized for ImageNet is shown with a red rectangle in Figure 3.4.



```
Sequential(
  (0): Dropout(p=0.2, inplace=False)
  (1): Linear(in_features=1280, out_features=1000, bias=True)
)
```

Figure 3.4: Last classifier layer of the original MobileNetV2.



```
Sequential(
  (0): Linear(in_features=1280, out_features=612, bias=True)
  (1): ReLU()
  (2): Dropout(p=0.2, inplace=False)
  (3): Linear(in_features=612, out_features=256, bias=True)
  (4): ReLU()
  (5): Dropout(p=0.2, inplace=False)
  (6): Linear(in_features=256, out_features=2, bias=True)
  (7): LogSoftmax(dim=1)
)
```

Figure 3.5: Last classifier layer of the modified MobileNetV2.

Since there are 2 classes in the Construction Machinery Dataset, the value representing 1000 classes in the last layers of the pre-trained models is coded as 2 to perform binary classification as seen in Figure 3.5.

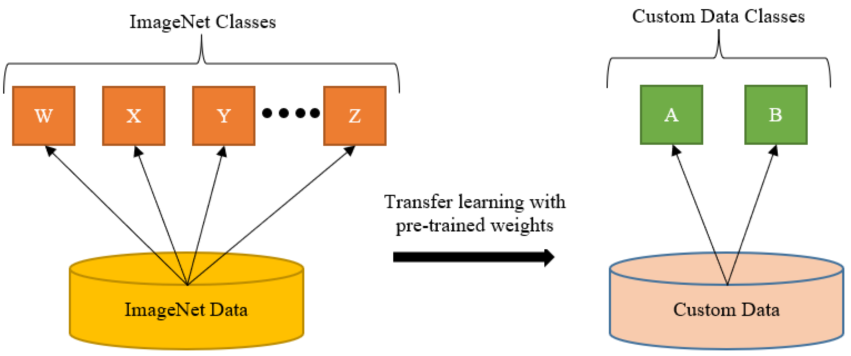


Figure 3.6: Transfer Learning Scheme.

Common metrics such as precision, recall, F1-Score, and accuracy were used to evaluate the success of the models. These metrics can be presented as:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

$$F1\,Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.3}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.4}$$

where TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative.

Table 3.3: Number of Total Parameters and Trainable Parameters of Each Model.

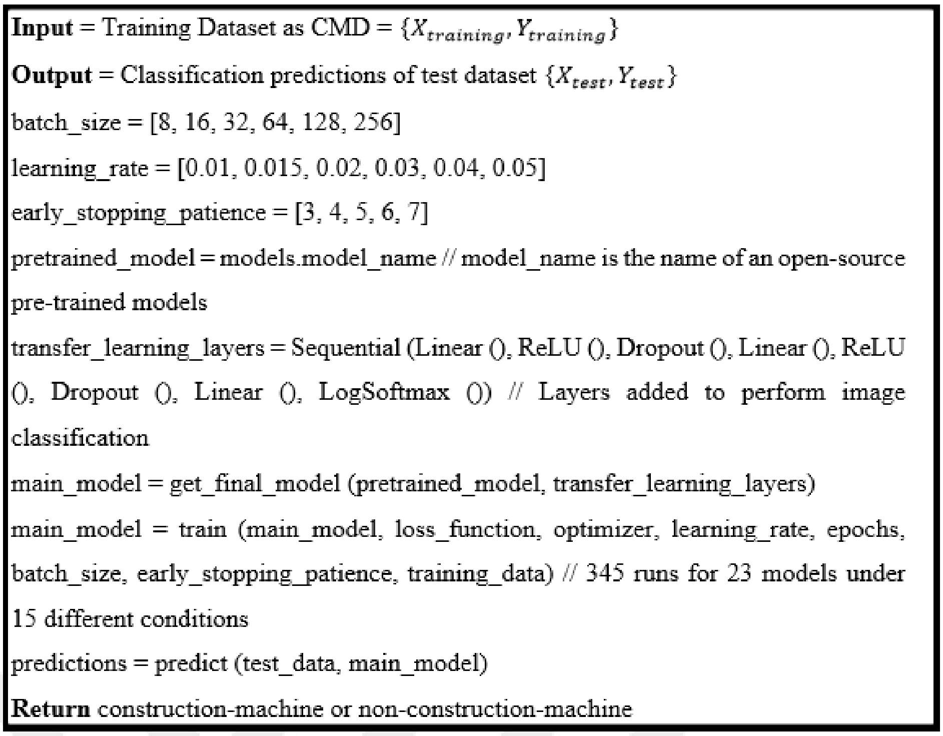
Models	Total Parameters	Trainable Parameters
AlexNet	8,267,942	5,798,246
VGG16	30,226,598	15,511,910
VGG19	35,536,294	15,511,910
ResNet-18	11,647,910	471,398
ResNet-34	21,756,070	471,398
ResNet-50	24,919,462	1,411,430
ResNet-101	43,911,590	1,411,430
ResNet-152	59,555,238	1,411,430
MobileNet V2	3,165,286	941,414
MobileNet V3 Large	3,717,526	745,574
MobileNet V3 Small	1,437,574	510,566
DenseNet-121	7,738,598	784,742
DenseNet-161	27,981,350	1,509,350
DenseNet-169	13,660,902	1,176,422
DenseNet-201	19,426,022	1,333,094
EfficientNet B0	4,948,962	941,414
EfficientNet B1	7,454,598	941,414
EfficientNet B2	8,720,744	1,019,750
EfficientNet B3	11,794,318	1,098,086
EfficientNet B4	18,803,374	1,254,758
EfficientNet B5	29,752,214	1,411,430
EfficientNet B6	42,303,806	1,568,102
EfficientNet B7	65,511,734	1,724,774

Grid search tables were created to clearly observe the working conditions of each model under different circumstances. Examining the graphics of the models trained with a learning rate of 0.01 in 19 of the 23 models in this thesis, excluding EfficientNetB3, EfficientNetB4, EfficientNetB5 and EfficientNetB6, it was seen that the models were easily trained. When the graphs of the models under other conditions are examined, the fact that the models could not be trained most of the time has made this situation important. Therefore, the 0.01 learning rate value led to future investigations.

In the trainings performed according to 32 batch size values, validation accuracy value gave the best results in all models except ResNet18, ResNet50, ResNet101, DenseNet121, and EfficientNetB2. That's why a batch size of 32 was selected as the best option for grid search.

Early stopping patience was taken as 5 in the examinations for the learning rate and batch size. It was seen that in architectures other than AlexNet, VGG16, VGG19, ResNet18, EfficientNetB4, and EfficientNetB7 models, having early stopping patience of 5 did not terminate the training process early. In order to better understand the effects of early stopping patience value, early stopping patience values of 3, 4, 6, and 7 were examined. Training was terminated early when early stopping patience value was 3 in ResNet34, ResNet101, ResNet152, MobileNetV2, DenseNet121, DenseNet161, DenseNet169, and EfficientNetB4 models. Early stopping occurred when early stopping patience value was 4 in ResNet34, ResNet50, ResNet101, ResNet152, MobileNetV2, DenseNet121, DenseNet169, DenseNet201, and EfficientNetB4 models. In ResNet50, ResNet152, DenseNet201, EfficientNetB4 models, early stopping was experienced when the patience value was 6. When the early stopping patience value was 7 in MobileNetV2 and EfficientNetB4 models, early stopping occurred. Thus, the reactions of 23 different models in this thesis under 15 different situations were examined and the corresponding training loss and training accuracy values were also examined. While creating grid search tables with 345 runs in total, the importance of having a high GPU was revealed.

Jupyter Notebook is a web-based application that provides fast and practical data visualization by showing the inputs and outputs of the codes in a user-friendly way. All models used in this thesis were trained and tested using the Python programming language and Jupyter Notebook. Intel(R) Core(TM) i7-10700F CPU 2.90GHz 32.0 GB RAM and NVIDIA GEFORCE RTX 3060 12 GB GPU were used in the training and testing phase of all models. The pseudocode of the functions used in this thesis is as follows:



```
Input = Training Dataset as CMD = $\{X_{training}, Y_{training}\}$
Output = Classification predictions of test dataset $\{X_{test}, Y_{test}\}$
batch_size = [8, 16, 32, 64, 128, 256]
learning_rate = [0.01, 0.015, 0.02, 0.03, 0.04, 0.05]
early_stopping_patience = [3, 4, 5, 6, 7]
pretrained_model = models.model_name // model_name is the name of an open-source pre-trained models
transfer_learning_layers = Sequential (Linear (), ReLU (), Dropout (), Linear (), ReLU (), Dropout (), Linear (), LogSoftmax ()) // Layers added to perform image classification
main_model = get_final_model (pretrained_model, transfer_learning_layers)
main_model = train (main_model, loss_function, optimizer, learning_rate, epochs, batch_size, early_stopping_patience, training_data) // 345 runs for 23 models under 15 different conditions
predictions = predict (test_data, main_model)
Return construction-machine or non-construction-machine
```

Figure 3.7: Pseudocode of The Study.

CHAPTER IV

RESULTS

The training efficiency of artificial neural networks is calculated by successive iterations. However, it is not possible to make an inference that running a model with too many iterations increases the accuracy of that model. This may result in a deep learning model being overfitted. If a model has very high accuracy at the end of training, but results in very weak predictions, it is overfit. The opposite situation is the use of very few iterations. This situation may cause the model to have learning difficulties. As a clear solution to these two scenarios, the early stopping method was used in this thesis. Early stopping is a deep learning method that understands that there is no improvement in the iteration calculations of the model during the training and terminates the training of the model. It is not a healthy method to carry out a study based only on the training accuracy value in grid search optimizations. If validation accuracy is not taken into account, it is difficult to see whether the trained model is overfit or not. In the tables reviewed in this chapter, the rows highlighted in light green show that the relevant conditions are the best working condition compared to the others in the table during training optimization.

4.1 GENERAL MODEL TRAINING RESULTS

4.1.1. AlexNet Architecture

Despite being one of the oldest models that pioneered deep learning studies in the literature, AlexNet can still compete with the most up-to-date models. One of the reasons why AlexNet is still used is that it uses Rectified Linear Units (ReLU) as its activation function. ReLU provides faster training of AlexNet compared to other activation functions. Another factor in reducing training time is the parallelization scheme that AlexNet uses. AlexNet can be trained using multiple GPUs. It reduces training time by using half of the neurons in a model on one GPU and the remaining half on another GPU. Especially when the training loss values are examined, AlexNet is one of the two models with the lowest loss among all models in this thesis with a loss of 0.1.

AlexNet was one of the models with the most difficult training process in this thesis. Although AlexNet had a good loss and accuracy value as a result of the training, the graph results said the opposite. Accuracy is expected to increase gradually as the number of epochs increases in the Accuracy vs Epoch chart. Conversely, as the number of epochs increases in the Loss vs Epoch chart, the loss is expected to decrease gradually. As the epoch progressed, the loss and accuracy values in the graphs were constantly parallel to the X-axis and AlexNet refused to train. This was especially observed at 16 and 32 conditions of batch size. As a result of these processes, AlexNet gave the healthiest graph with a 0.01 learning rate, 128 batch size, and 5 early stopping patience values. Looking at Table 4.2, AlexNet analyzed the images in the test dataset in a total of 7.10 seconds and reached the fastest test speed value among all models.

Table 4.1: Grid Search Performances of AlexNet.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	8	0.13	95.37	85.67
0.015	32	5	9	0.1	96.49	83.29
0.02	32	5	7	0.13	95.33	85.07
0.03	32	5	7	0.13	95.45	87.02
0.04	32	5	5	0.35	83.72	86.08
0.05	32	5	7	0.14	94.73	84.77
0.01	8	5	7	0.12	95.45	84.12
0.01	16	5	6	0.16	93.85	80.58
0.01	64	5	8	0.14	94.53	80.58
0.01	128	5	10	0.13	95.41	81.12
0.01	256	5	15	0.35	83.06	84.53
0.01	32	3	5	0.13	95.09	84.57
0.01	32	4	5	0.18	92.74	81.35
0.01	32	6	7	0.09	96.69	83.39
0.01	32	7	8	0.19	92.82	86.87

Table 4.2: Test Results of AlexNet on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
AlexNet	82.25	7.10	0.34	0.83	0.93	0.87

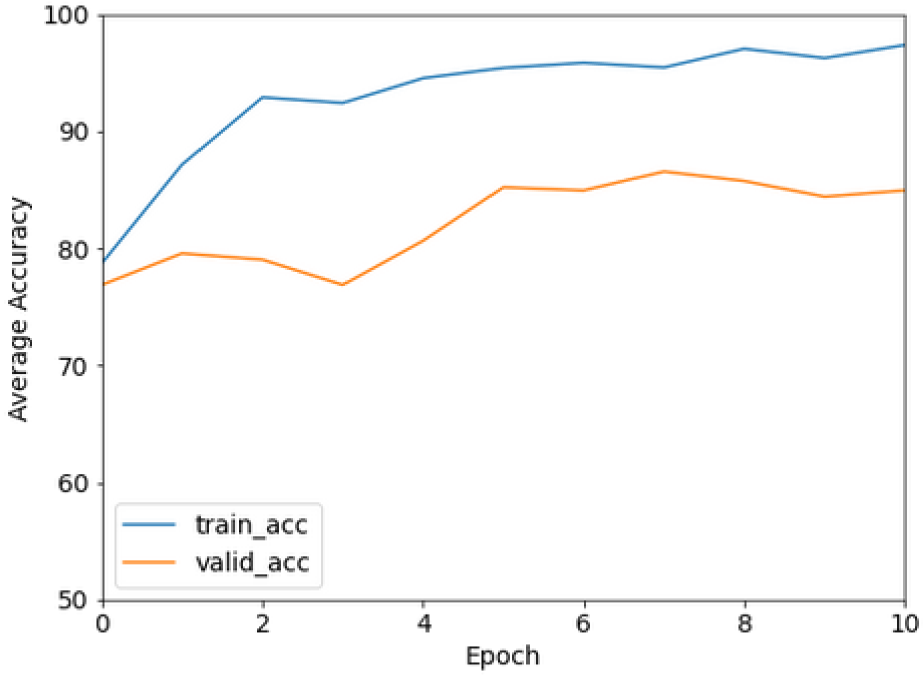


Figure 4.1: Accuracy vs. Epoch Graph of AlexNet.

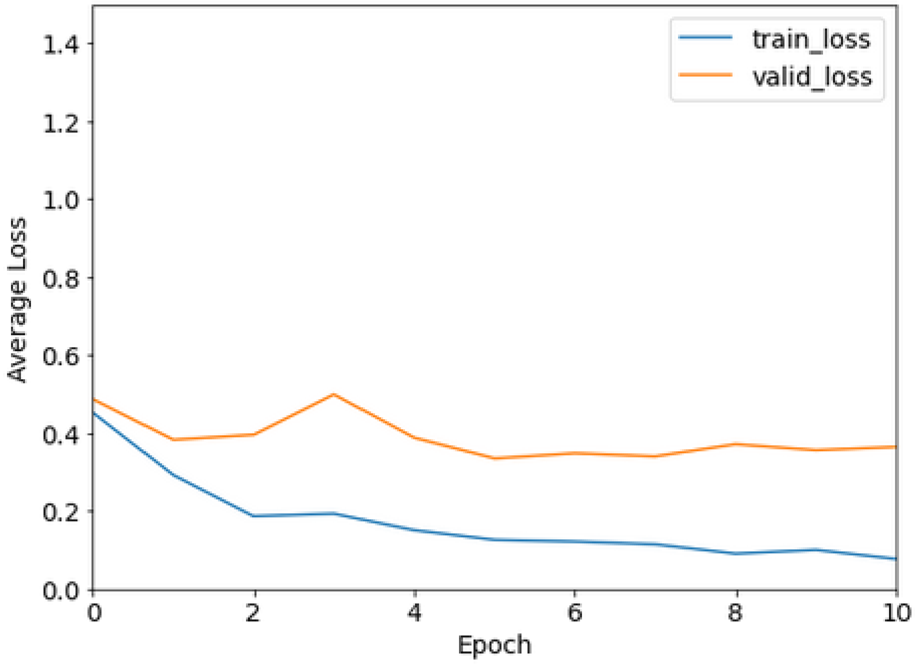


Figure 4.2: Loss vs. Epoch Graph of AlexNet.

4.1.2. VGG Architectures

In this section, training and test results of VGG16 and VGG19 models are examined. The idea that makes VGGs a popular model frequently used by researchers is to understand the effect of the depth of convolutional neural networks on accuracy.

Using very deep convolutional neural networks, high accuracy values were achieved. In the literature, it has been stated that the accuracy value increases as the model depth increases with the VGG models.

The power of the GPU used during the training determines the speed of the training as well as the amount of the batch size. Thanks to the powerful GPU used in this thesis, 256 batch size calculations could be made.

As can be seen in Table 4.3, early stopping was not activated in 128 and 256 batch size values of VGG16 model. Although all the accuracy values at the end of the training were above 90%, unfortunately, the graphics in all cases did not appear as in Figure 4.3 and Figure 4.4. In some cases, the graphs progressed parallel to the X-axis, while in some cases, the graphs showed very sudden uptrends and downtrends. VGG16 gave the most successful graphics at a 0.01 learning rate, 256 batch size, and 5 early stopping patience values.

Table 4.3: Grid Search Performances of VGG16.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	7	0.15	94.69	88.56
0.015	32	5	7	0.13	94.89	85.08
0.02	32	5	7	0.13	94.85	83.25
0.03	32	5	6	0.17	93.77	83
0.04	32	5	8	0.09	96.65	84.47
0.05	32	5	6	0.18	92.66	82.09
0.01	8	5	6	0.16	93.42	84.11
0.01	16	5	7	0.13	94.61	81.17
0.01	64	5	12	0.09	97.09	80.56
0.01	128	5	15	0.22	90.59	88.41
0.01	256	5	15	0.18	94.98	87.13
0.01	32	3	5	0.14	94.41	86
0.01	32	4	6	0.15	94.57	85.51
0.01	32	6	12	0.06	97.96	84.7
0.01	32	7	8	0.2	93.46	86.22

Table 4.5 shows that VGG16 analyzed the test dataset in 7.88 seconds. Although the test accuracy value of 86.55% is an average value compared to other models, the recall value reaching 92% is enough to say that the VGG16 is a successful model.

VGG19 has a deeper architecture than VGG16. Therefore, it is expected to give better results than VGG16. Looking at the VGG19 test criteria, it outperformed VGG16 in all test accuracy, test loss, precision, recall, and F1 Score values. The only test criterion where VGG16 surpasses VGG19 is total test time. According to the results of the training time, VGG16 completed the training in 664.83 seconds, while VGG19 completed it in 549.16 seconds, despite having a deeper architecture.

Table 4.4: Grid Search Performances of VGG19.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	7	0.15	94.49	85.26
0.015	32	5	7	0.15	94.61	84.43
0.02	32	5	5	0.35	84.52	88.3
0.03	32	5	6	0.18	93.26	81.14
0.04	32	5	10	0.08	97.09	83.82
0.05	32	5	5	0.34	85.28	88.44
0.01	8	5	6	0.17	93.26	81.15
0.01	16	5	7	0.12	95.57	82.01
0.01	64	5	8	0.12	95.33	83.38
0.01	128	5	12	0.11	95.53	81.73
0.01	256	5	15	0.16	95.59	87.19
0.01	32	3	5	0.15	94.13	88.72
0.01	32	4	7	0.13	94.89	87.19
0.01	32	6	8	0.15	94.45	82.19
0.01	32	7	11	0.1	96.33	90.02

Table 4.5: Test Results of VGG16 and VGG19 on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
VGG16	86.55	7.88	0.26	0.87	0.92	0.89
VGG19	90.86	9.55	0.22	0.89	0.95	0.92

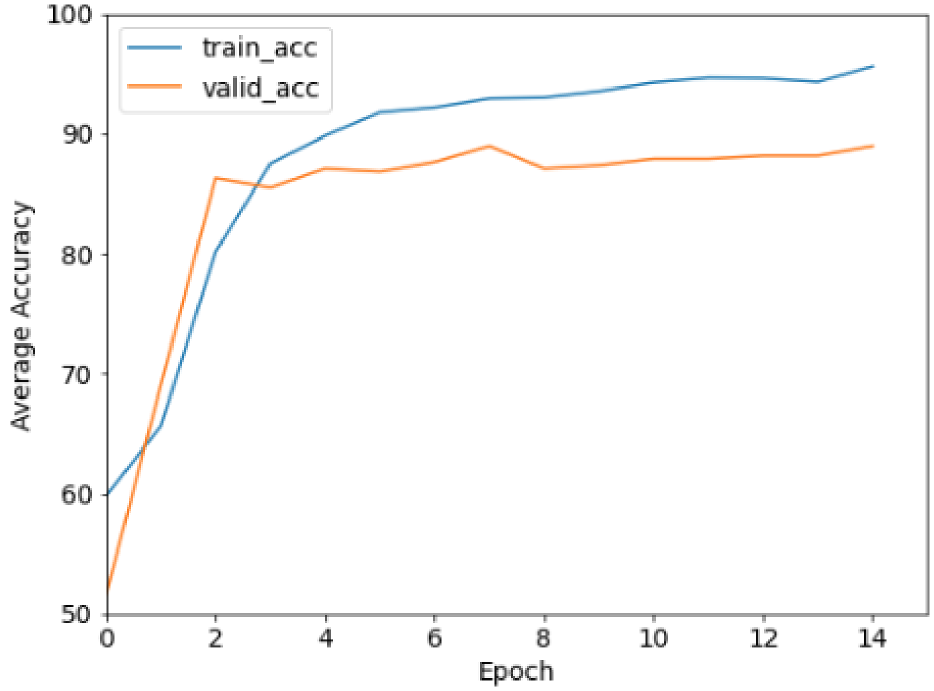


Figure 4.3: Accuracy vs. Epoch Graph of VGG16.

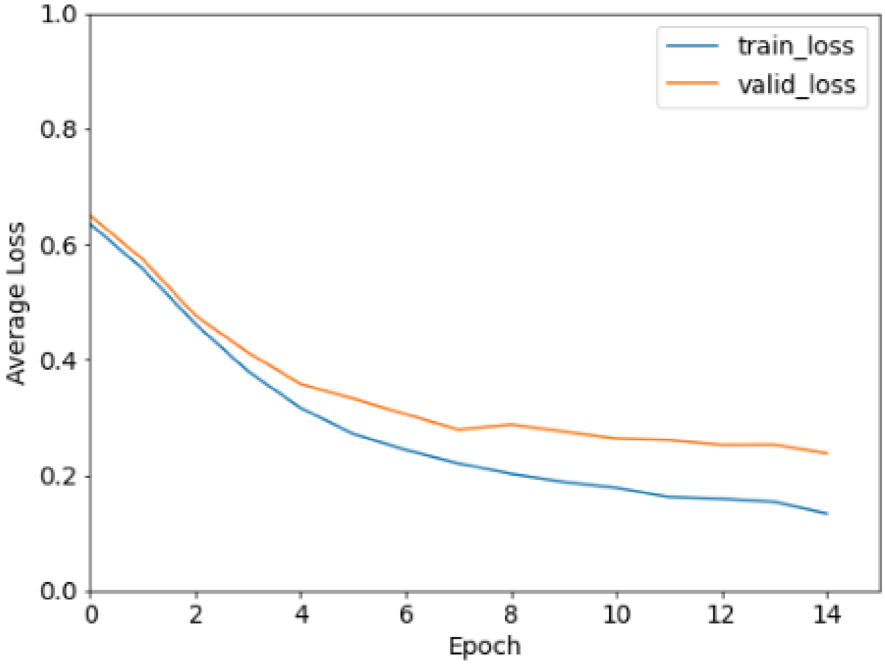


Figure 4.4: Loss vs. Epoch Graph of VGG16.

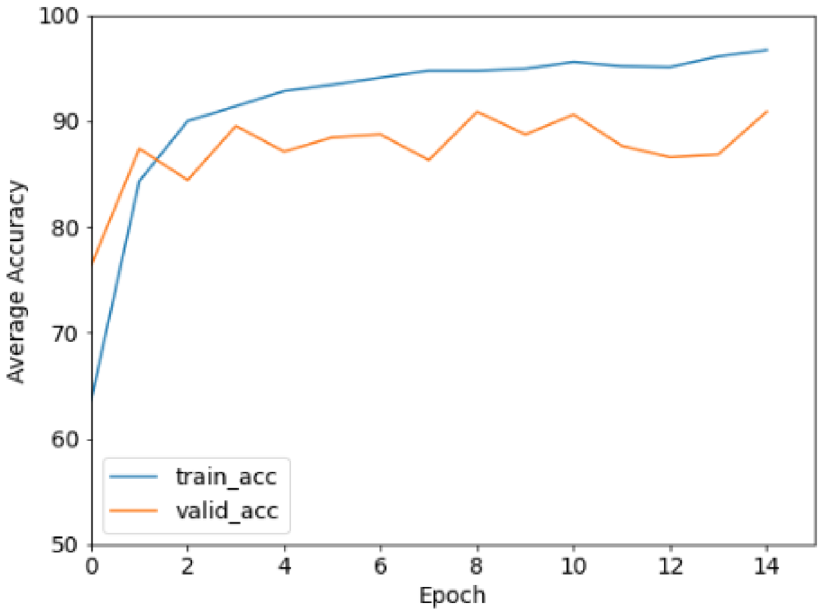


Figure 4.5: Accuracy vs. Epoch Graph of VGG19.

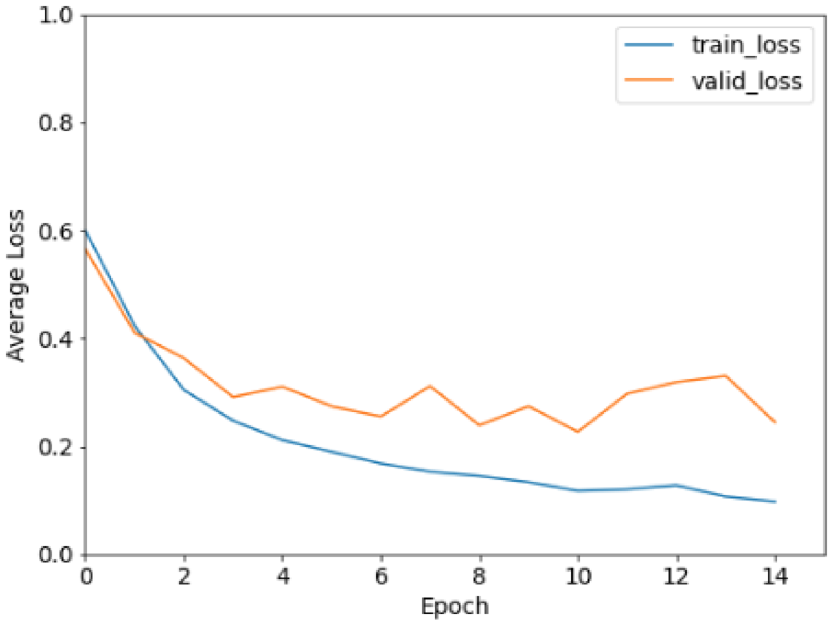


Figure 4.6: Loss vs. Epoch Graph of VGG19.

4.1.3. ResNet Architectures

In this section, training and test results of ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 models are examined. It is an undeniable fact that the training phase of neural networks is the part that takes the most time for researchers. As the depth and complexity of the network increase, the training time inevitably gets longer. ResNets have lower complexity compared to other deep learning models and they are easy to optimize. Hence, deeper models can be trained with lower training loss value. As such, with deep residual networks, it is aimed to shorten the training times of models even if the model complexity is high [54].

Model depth is increasing in ResNet architectures from ResNet18 to ResNet152. This situation naturally creates an expectation for good results as the model depth increases in the training and test results. When Table 4.6, which is ResNet18's grid search table, is examined, it is seen that the model gives the most efficient graph in 128 batch size. While the epoch 15 and early stopping patience value were 5, the training of the model took 15 epochs. This means that there was no early stopping during model training. When the test loss value of ResNet18 is examined, it is seen that it has the lowest test loss value compared to all other ResNet architectures. When Table 4.11 is examined, it is seen that the test values of ResNet18 have many variations. While ResNet18 has the lowest test accuracy and test loss values with 80.10% and 0.46, respectively, among all other ResNet architectures, its recall value has the highest value with 0.95.

Table 4.6: Grid Search Performances of ResNet-18.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	11	0.25	89.78	80.03
0.015	32	5	8	0.23	90.9	86.8
0.02	32	5	6	0.31	86.95	85.6
0.03	32	5	10	0.19	93.22	81.14
0.04	32	5	8	0.2	91.98	81.49
0.05	32	5	10	0.17	93.1	85.45
0.01	8	5	15	0.22	91.4	86.98
0.01	16	5	13	0.16	93.26	82.47
0.01	64	5	13	0.17	93.06	86.87
0.01	128	5	15	0.18	91.25	87.25
0.01	256	5	15	0.3	87.9	83.97
0.01	32	3	5	0.31	88.15	88.9
0.01	32	4	15	0.26	88.98	85.04
0.01	32	6	15	0.26	88.71	80.53
0.01	32	7	15	0.25	89.52	80.93

According to Table 4.7, the most successful graphics of the ResNet-34 model came with a learning rate of 0.01 and a batch size of 32. Early stopping patience value is 7, but model training did not stop until the maximum epoch value of 15. When the training loss value of ResNet-34 is examined, it is seen that it has the highest value with 0.22 among other ResNet architectures. A similar situation applies to training accuracy. But this time, ResNet-34's 90.17% training accuracy is the lowest not only among the ResNet architectures, but among all other models. Looking at Table 4.11 for the test values of ResNet-34, ResNet-34 has the fastest total test time of 7.37 seconds among other ResNet architectures. When the F1 score values are examined, ResNet-34 has the best value with 0.89.

Table 4.7: Grid Search Performances of ResNet-34.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	12	0.26	88.44	88.89
0.015	32	5	15	0.25	88.71	81.92
0.02	32	5	15	0.24	88.44	85.15
0.03	32	5	15	0.24	89.52	86.81
0.04	32	5	15	0.22	90.32	81.68
0.05	32	5	15	0.22	89.25	81.97
0.01	8	5	15	0.28	89.25	82.79
0.01	16	5	15	0.22	89.25	84.33
0.01	64	5	13	0.18	93.18	87.26
0.01	128	5	15	0.26	88.44	80.25
0.01	256	5	15	0.32	86.29	88.71
0.01	32	3	11	0.19	92.62	85.2
0.01	32	4	14	0.17	93.42	86.13
0.01	32	6	15	0.26	88.17	84.02
0.01	32	7	15	0.22	90.17	86.3

When the grid search results of ResNet-50 in Table 4.8 are examined, it is seen that ResNet-50 gives the best graphics under the same conditions as ResNet-18. Under the same conditions, ResNet-18's training loss is lower than ResNet-50, while ResNet-50's training accuracy is higher. ResNet-34's training time is lower than ResNet-18, while ResNet50's training time is higher than ResNet-34. When the test results of ResNet-50 are examined in Table 4.11, it is seen that the F1 score value is the best value in ResNet architectures with 0.89 like ResNet-34.

Table 4.9 provides information about ResNet-101's grid search. According to this information, ResNet-101 gave its best graphics in 64 batch size. According to these results, when the training accuracy values of ResNet architectures are examined, it is seen that ResNet-101 has the highest training accuracy value among other ResNet models with 93.17%. When Table 4.11 is examined for the test values of ResNet-101, ResNet-101 has the second worst value with 9.14 seconds in total test time. On the other hand, ResNet-101's test accuracy, test loss, and precision are the second best among other ResNet models.

Table 4.8: Grid Search Performances of ResNet-50.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	13	0.21	91.54	84.84
0.015	32	5	15	0.23	90.32	86.77
0.02	32	5	11	0.2	92.42	82.82
0.03	32	5	15	0.21	91.4	82.62
0.04	32	5	15	0.21	90.59	84
0.05	32	5	15	0.21	91.4	83.23
0.01	8	5	15	0.22	90.59	81.05
0.01	16	5	12	0.22	90.74	81.99
0.01	64	5	15	0.25	89.52	88.46
0.01	128	5	15	0.21	92.71	88.85
0.01	256	5	15	0.46	85.48	85.8
0.01	32	3	15	0.24	89.52	83.43
0.01	32	4	11	0.19	92.18	87.77
0.01	32	6	11	0.22	91.62	86.14
0.01	32	7	15	0.22	89.78	82.59

Table 4.9: Grid Search Performances of ResNet-101.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.17	93.5	81.56
0.015	32	5	12	0.19	92.98	85.94
0.02	32	5	9	0.22	91.5	84.77
0.03	32	5	15	0.24	90.05	86.81
0.04	32	5	15	0.25	90.05	87.02
0.05	32	5	11	0.18	92.78	86.17
0.01	8	5	13	0.23	91.02	81.99
0.01	16	5	15	0.23	89.78	80.09
0.01	64	5	15	0.19	93.17	88.96
0.01	128	5	15	0.33	86.83	83.97
0.01	256	5	15	0.46	83.6	86.81
0.01	32	3	8	0.21	91.78	85.65
0.01	32	4	12	0.19	92.62	86.5
0.01	32	6	15	0.26	88.71	83.5
0.01	32	7	15	0.27	88.44	83.17

According to the training results of ResNet-152 in Table 4.10, the best graphics were formed in 16 batch sizes. This is a great value for low power computer setups. However, while the ResNet-101 model reached 93.17% training accuracy in 666.57 seconds, the ResNet-152 model reached 92.47% training accuracy in 818.7 seconds. When the test results of ResNet-152 are examined in Table 4.11, it is seen that ResNet-152 has the highest value among other ResNet architectures with 94.62% test accuracy. While ResNet-152 has the highest value with 10.38 seconds among other ResNet architectures on the basis of total test time, it has the lowest value with 0.14 on the basis of test loss. ResNet-152's precision has the highest value of 0.92 among other ResNet architectures. On the other hand, recall and F1 score values have the lowest value among all other models in this thesis.

Table 4.10: Grid Search Performances of ResNet-152.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.2	92.74	90.09
0.015	32	5	15	0.22	91.13	86.54
0.02	32	5	10	0.21	91.9	87.29
0.03	32	5	15	0.19	93.01	86.58
0.04	32	5	7	0.25	90.3	80.83
0.05	32	5	11	0.16	93.97	80.71
0.01	8	5	14	0.2	91.5	88.81
0.01	16	5	15	0.19	92.47	89.38
0.01	64	5	15	0.23	91.4	88.28
0.01	128	5	15	0.28	88.98	80.25
0.01	256	5	15	0.48	83.87	87.2
0.01	32	3	12	0.19	92.34	87.38
0.01	32	4	14	0.17	93.34	80.78
0.01	32	6	11	0.21	92.1	87.62
0.01	32	7	15	0.21	90.86	80.37

Table 4.11: Test Results of ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
ResNet-18	80.10	8.65	0.46	0.82	0.95	0.88
ResNet-34	82.79	7.37	0.35	0.84	0.94	0.89
ResNet-50	89.78	8.26	0.27	0.89	0.89	0.89
ResNet-101	91.93	9.14	0.15	0.90	0.80	0.85
ResNet-152	94.62	10.38	0.14	0.92	0.77	0.83

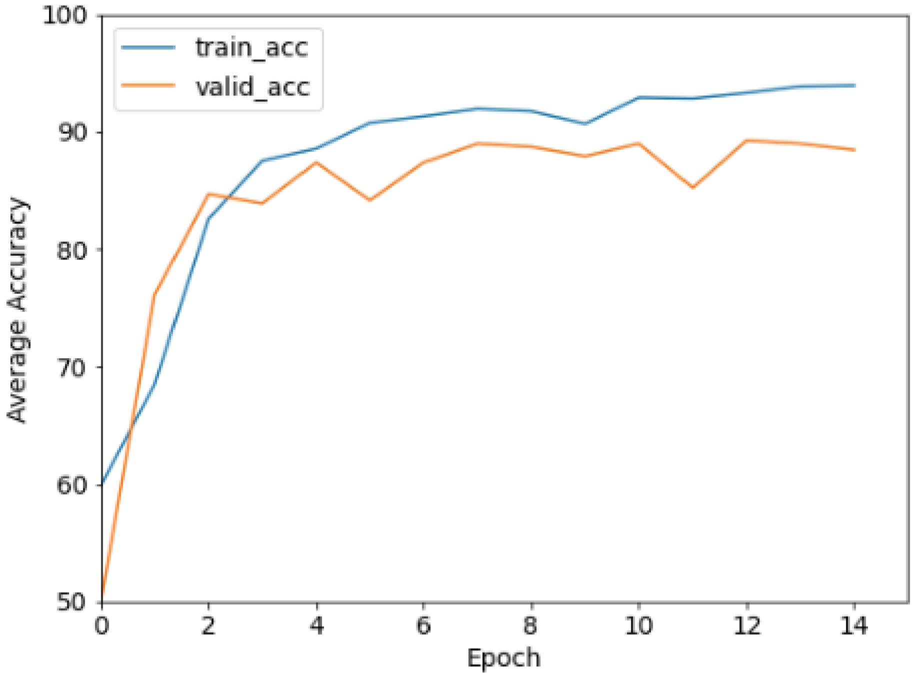


Figure 4.7: Accuracy vs. Epoch Graph of ResNet-18.

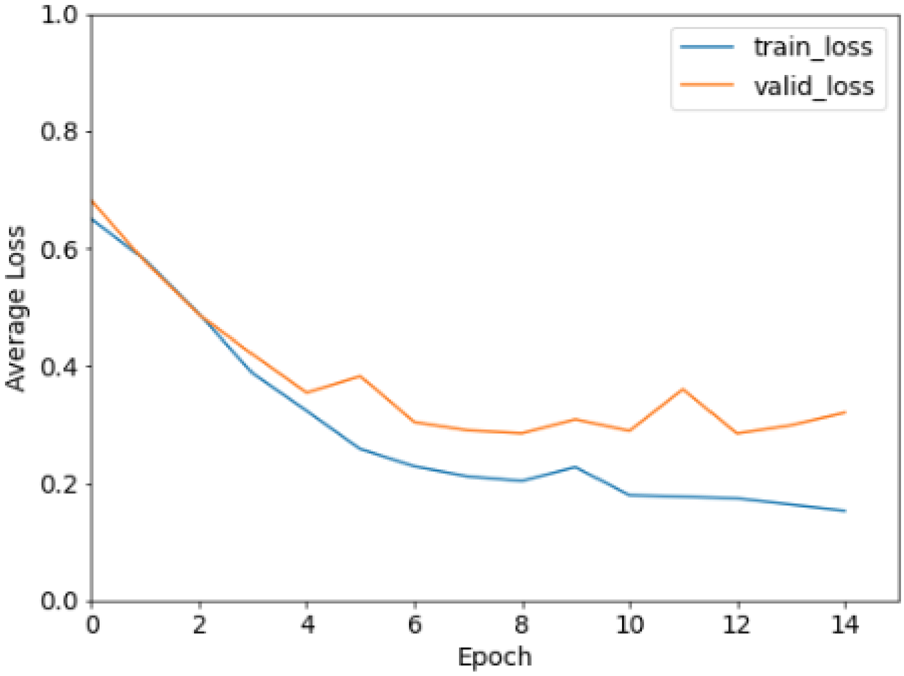


Figure 4.8: Loss vs. Epoch Graph of ResNet-18.

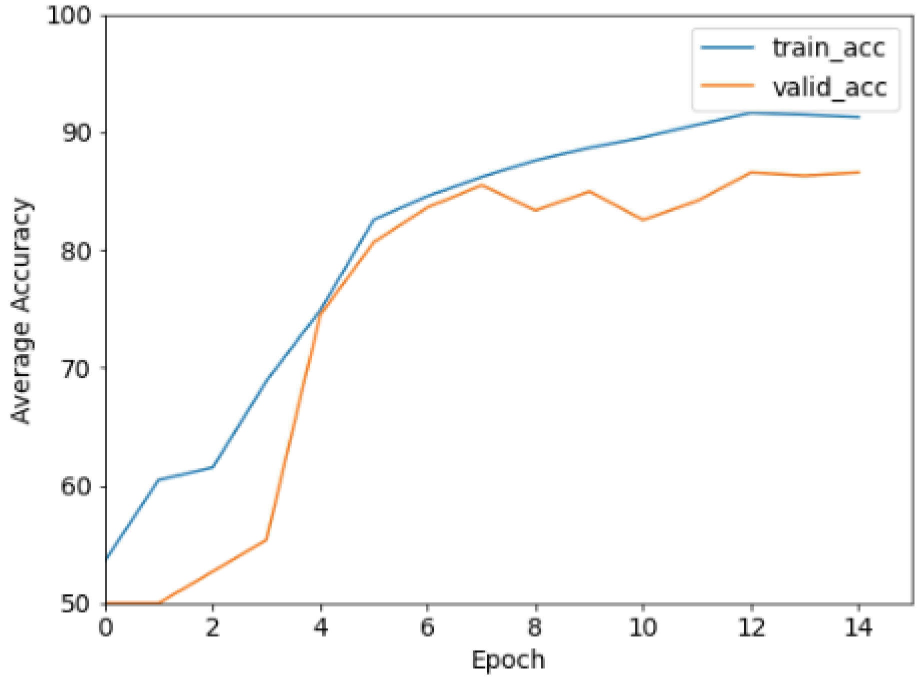


Figure 4.9: Accuracy vs. Epoch Graph of ResNet-34.

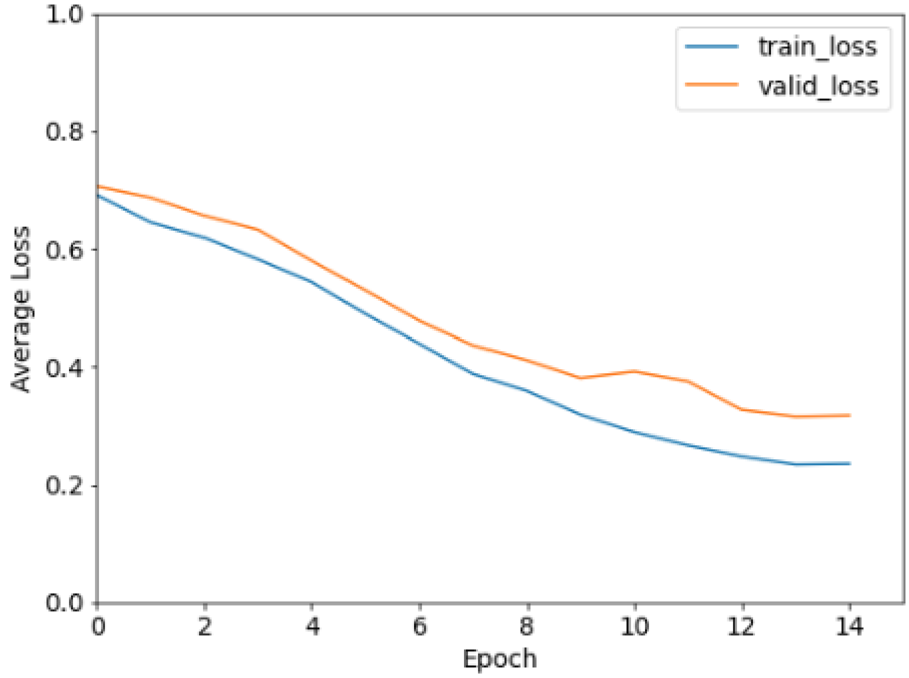


Figure 4.10: Loss vs. Epoch Graph of ResNet-34.

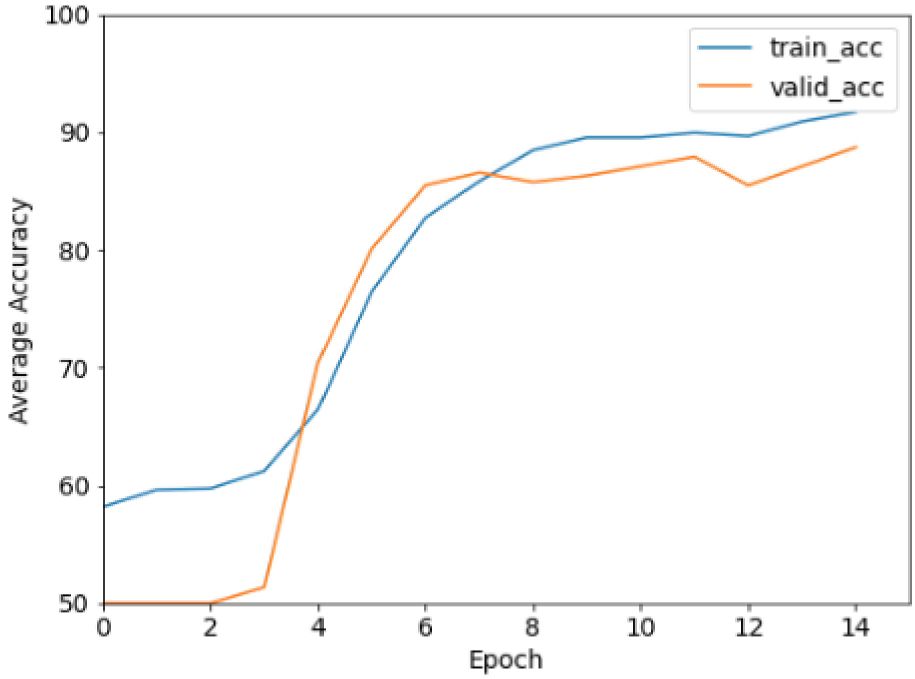


Figure 4.11: Accuracy vs. Epoch Graph of ResNet-50.

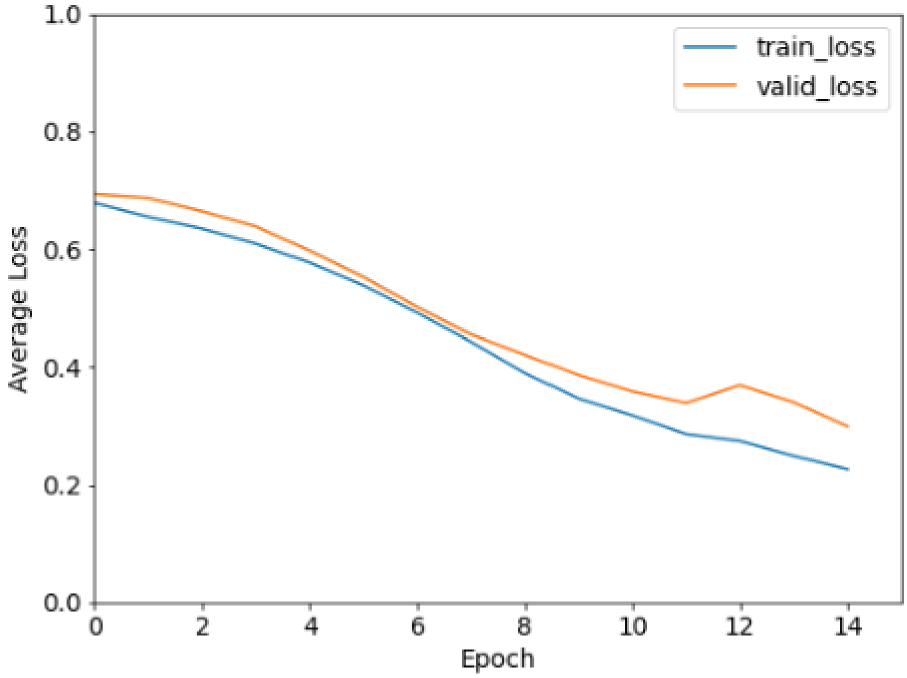


Figure 4.12: Loss vs. Epoch Graph of ResNet-50.

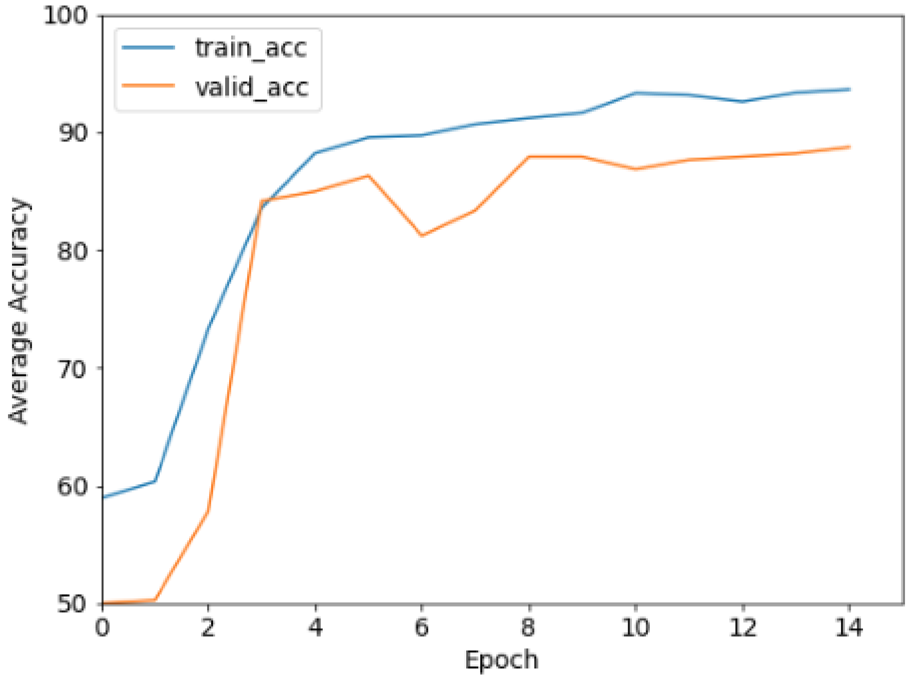


Figure 4.13: Accuracy vs. Epoch Graph of ResNet-101.

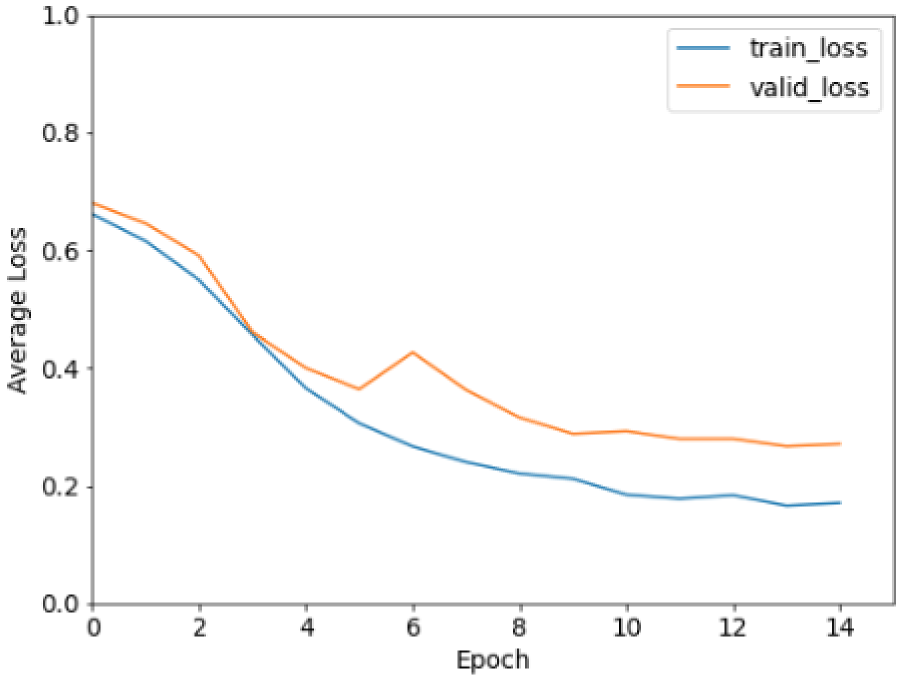


Figure 4.14: Loss vs. Epoch Graph of ResNet-101.

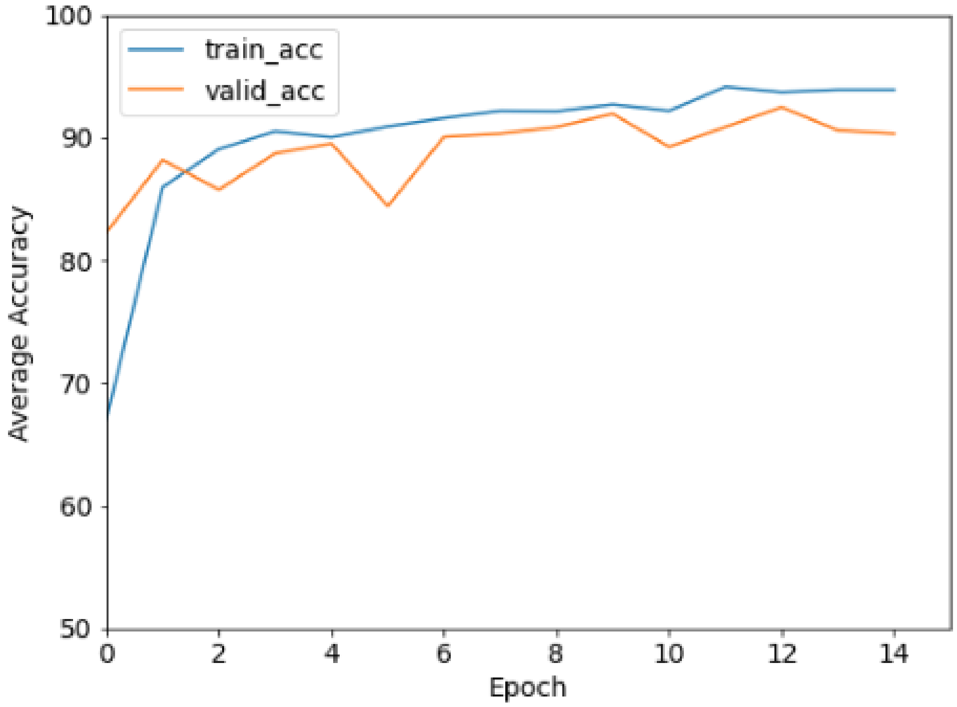


Figure 4.15: Accuracy vs. Epoch Graph of ResNet-152.

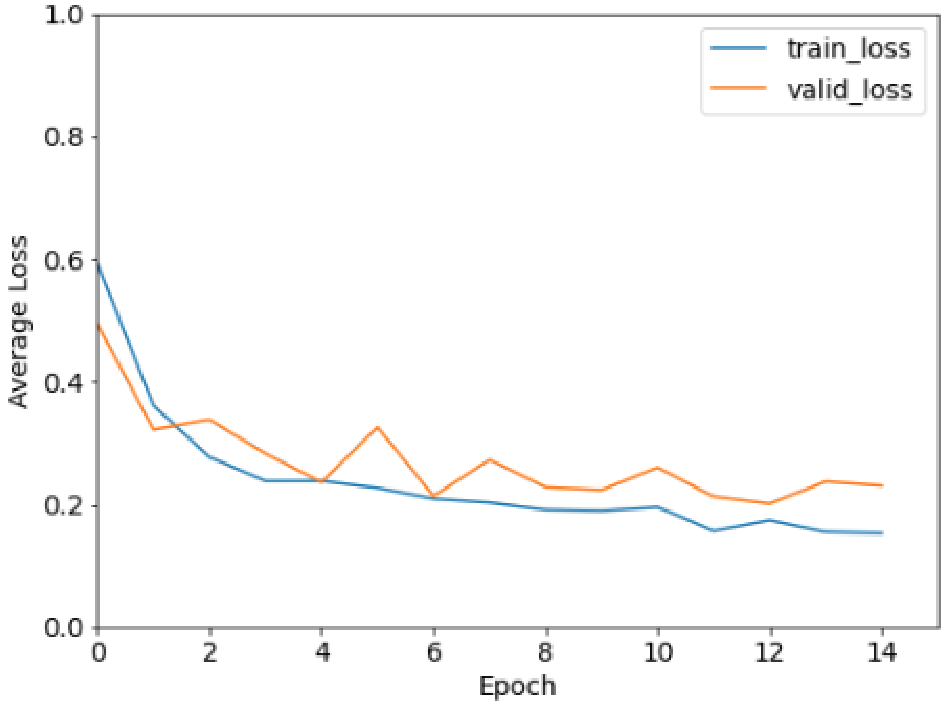


Figure 4.16: Loss vs. Epoch Graph of ResNet-152.

4.1.4. MobileNet Architectures

In this section, training and test results of MobileNet V2, MobileNet V3 Large, and MobileNet V3 Small models are examined. MobileNet is a deep learning model that uses depth-wise separable convolutions for mobile and embedded vision applications [55]. In addition, MobileNet introduced a revolutionary structure in mobile applications with a module named the inverted residual with the linear bottleneck. It provides great savings in computational cost by reducing the memory needed by mobile computer vision models without changing the accuracy [56]. Thanks to its low number of parameters, MobileNet [57] both takes up little space in memory and has a fast-training structure. As can be seen from Table 3.3, MobileNet architectures have the least number of parameters compared to other models. The MobileNetV2 model gave its best graphics at 0.01 learning rate, 16 batch size and 5 early stopping patience values. When the data other than the highlighted best case in Table 4.12 were examined, the graphics were not very successful, although there were good training results. The fact that MobileNetV2 reaches its best form at 16 batch size makes it easy to train the model especially in low computer setup conditions. When the status of MobileNetV2 on the test dataset is examined in Table 4.15, it is seen that the test accuracy, precision, recall, and F1 score values are below 90%. Although these values are not bad, it is an undeniable fact that there are more promising models.

Table 4.12: Grid Search Performances of MobileNet V2.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	13	0.18	93.06	87.46
0.015	32	5	11	0.16	94.37	88.33
0.02	32	5	15	0.24	88.71	86.7
0.03	32	5	15	0.23	90.59	87.45
0.04	32	5	13	0.14	94.97	86.34
0.05	32	5	8	0.2	92.06	87.57
0.01	8	5	12	0.24	90.22	86.17
0.01	16	5	10	0.18	92.1	86.62
0.01	64	5	15	0.26	88.71	86.6
0.01	128	5	15	0.31	86.83	81.4
0.01	256	5	15	0.46	84.41	87.04
0.01	32	3	12	0.17	93.34	82.55
0.01	32	4	11	0.24	88.71	83.3
0.01	32	6	15	0.24	89.52	82.61
0.01	32	7	13	0.19	92.02	83.56

Table 4.13: Grid Search Performances of MobileNet V3 Small.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.27	89.52	86.16
0.015	32	5	15	0.26	87.63	80.19
0.02	32	5	14	0.18	92.94	80.77
0.03	32	5	15	0.22	90.86	83.84
0.04	32	5	15	0.21	91.13	83.61
0.05	32	5	15	0.23	91.13	85.56
0.01	8	5	15	0.26	89.78	85.76
0.01	16	5	15	0.24	88.98	80.2
0.01	64	5	15	0.22	91.02	85.56
0.01	128	5	15	0.44	81.18	83.52
0.01	256	5	15	0.64	54.3	83.98
0.01	32	3	15	0.26	86.56	84.89
0.01	32	4	15	0.27	87.9	83.16
0.01	32	6	15	0.26	88.44	84.46
0.01	32	7	15	0.26	88.44	84.94

In Table 4.13, it is seen that MobileNet V3 Small model gives the best graphics at 64 batch size, which is four times that of MobileNetV2. Looking at all the number of epochs before stopping values in the table, it is seen that the MobileNet V3 Small model is not prone to early stopping. Due to this situation, MobileNetV2 reached 92.1% training accuracy in 428.12 seconds, while MobileNet V3 Small reached 91.02% training accuracy in 590.55 seconds. When the test results of the MobileNet V3 Small model are examined in Table 4.15, although it has lower test accuracy and higher test loss value than MobileNetV2, it is slightly ahead of MobileNetV2 in total test time, recall and F1 score values.

MobileNetV3 Large, on the other hand, gave its best graphics at 32 batch sizes, unlike other MobileNet architectures. The MobileNetV3 Large model gave the highest training accuracy with little difference, among the MobileNet architectures. When the calculation times of MobileNet architectures for 1 epoch in training are examined, it is seen that MobileNetV2 takes 38.92 seconds, MobileNetV3 Small takes 42.18 seconds and MobileNetV3 Large takes 37.57 seconds. Although MobileNetV3 Large is the fastest MobileNet architecture to calculate 1 epoch, MobileNetV3 Large could not outperform MobileNetV2 in training speed, since the number of epochs before stopping value of MobileNetV2 is 10. When Table 4.15 is examined, the test accuracy of MobileNetV3 Large gave a much more successful result with 92.47% compared to other MobileNet architectures. MobileNetV3 Large outperformed other MobileNet architectures in loss, precision, recall, and F1 score values in test results.

Table 4.14: Grid Search Performances of MobileNet V3 Large.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	14	0.19	92.2	91.32
0.015	32	5	15	0.18	93.55	87.61
0.02	32	5	15	0.18	92.2	84.77
0.03	32	5	15	0.15	94.35	84.93
0.04	32	5	15	0.14	94.62	87.53
0.05	32	5	15	0.15	94.09	87.76
0.01	8	5	15	0.18	93.55	81.4
0.01	16	5	15	0.18	93.28	86.96
0.01	64	5	15	0.23	90.59	84.94
0.01	128	5	15	0.35	86.56	80.91
0.01	256	5	15	0.59	70.16	81.8
0.01	32	3	15	0.18	91.94	88.41
0.01	32	4	15	0.19	91.4	86.64
0.01	32	6	15	0.19	92.2	86.03
0.01	32	7	15	0.2	92.2	81.28

Table 4.15: Test Results of MobileNet V2, MobileNet V3 Large, and MobileNet V3 Small on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
MobileNet V2	88.70	8.81	0.21	0.86	0.86	0.85
MobileNet V3 Small	87.63	7.12	0.29	0.86	0.87	0.86
MobileNet V3 Large	92.47	7.27	0.15	0.91	0.88	0.89

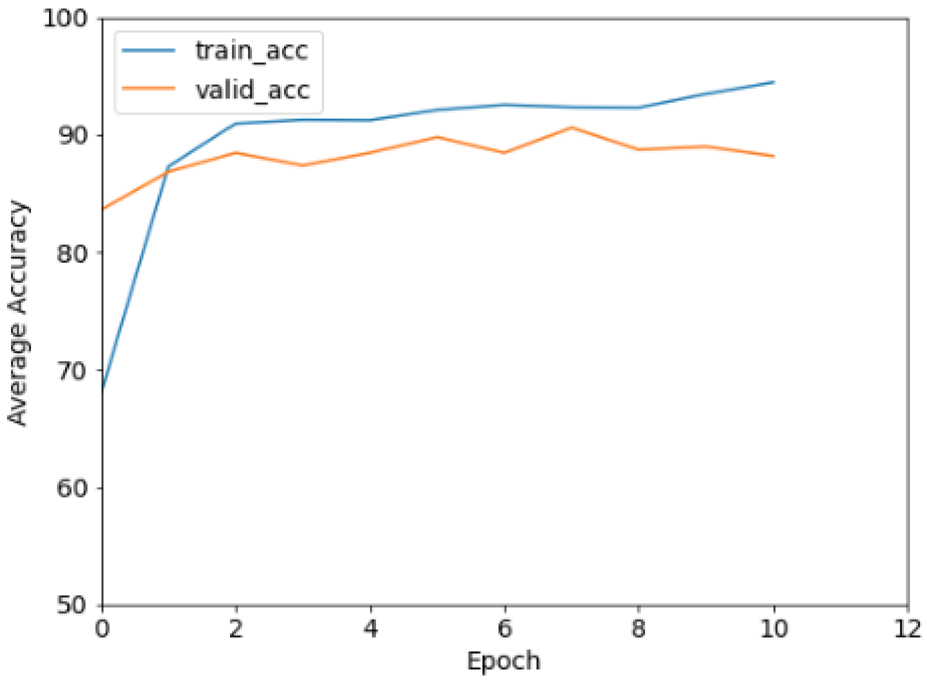


Figure 4.17: Accuracy vs. Epoch Graph of MobileNet V2.

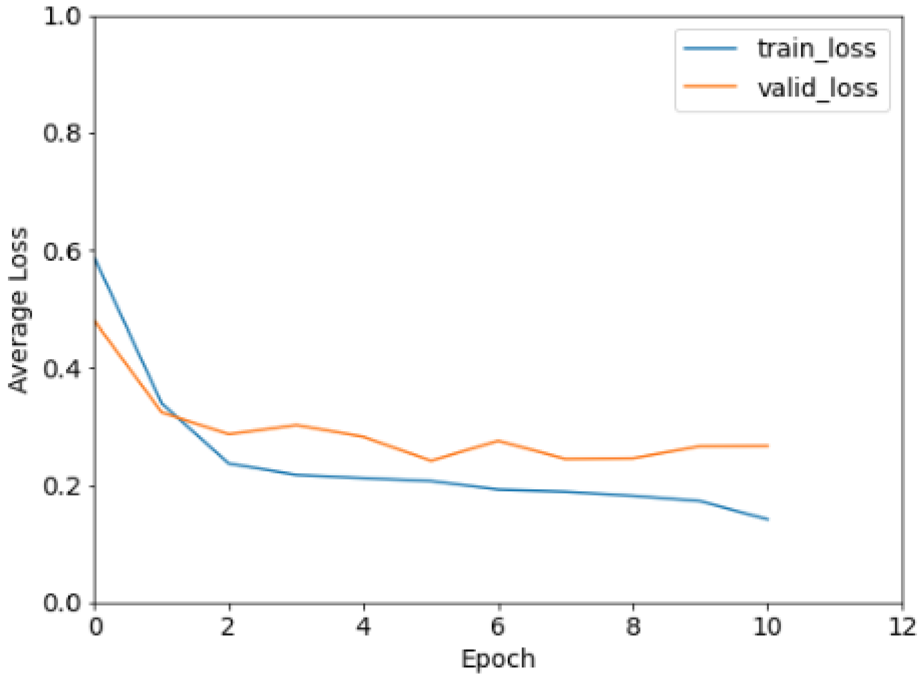


Figure 4.18: Loss vs. Epoch Graph of MobileNet V2.

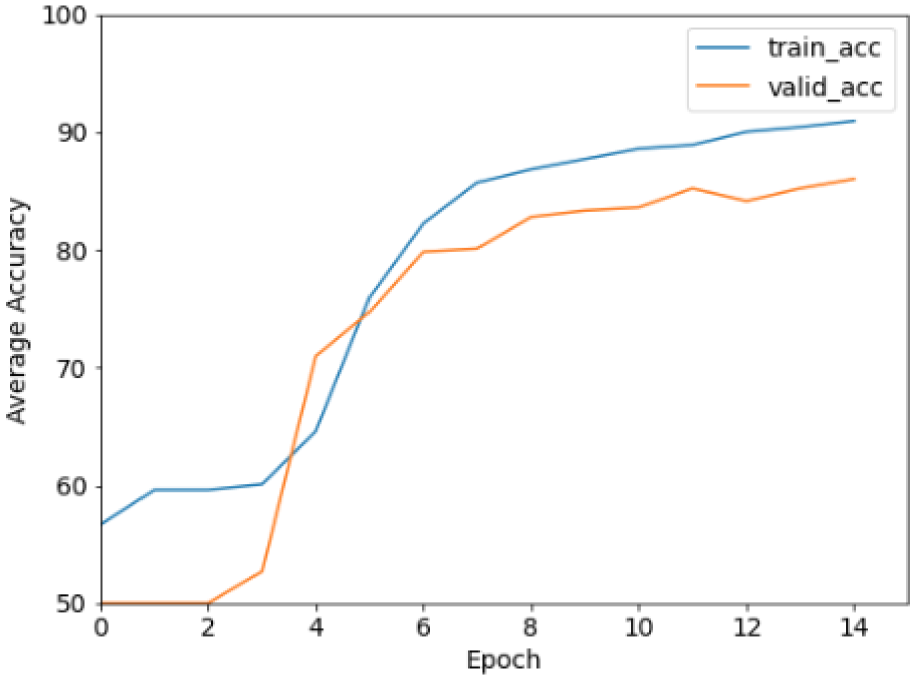


Figure 4.19: Accuracy vs. Epoch Graph of MobileNet V3 Small.

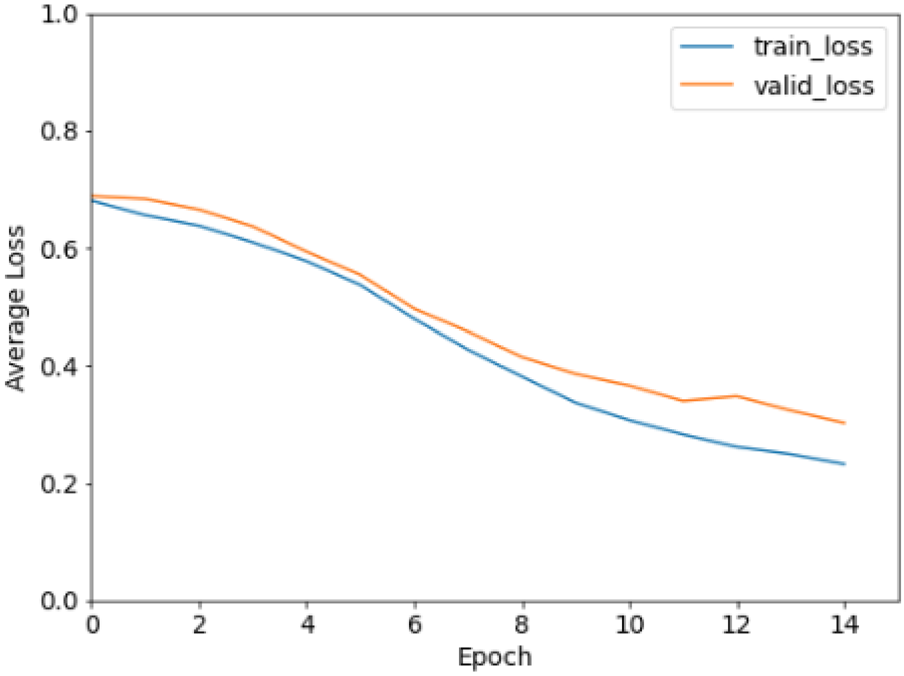


Figure 4.20: Loss vs. Epoch Graph of MobileNet V3 Small.

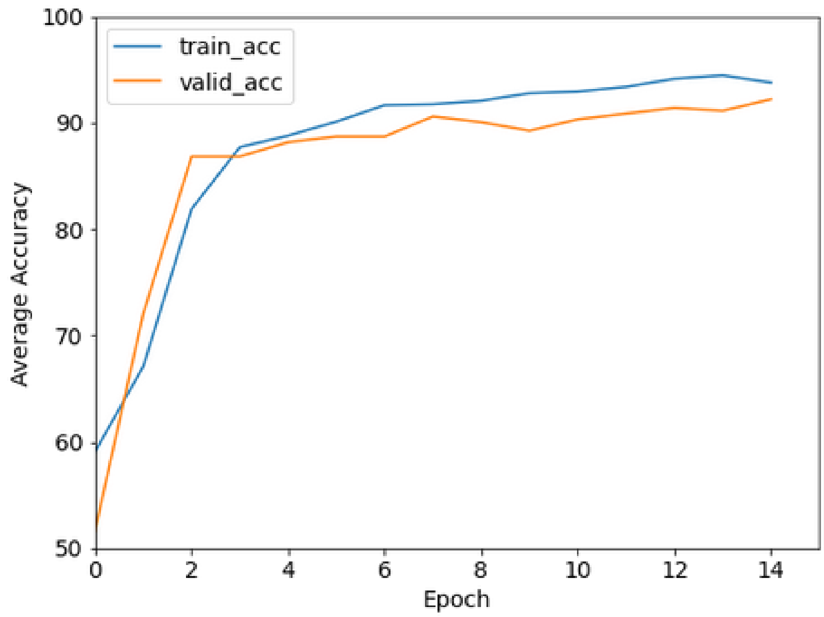


Figure 4.21: Accuracy vs. Epoch Graph of MobileNet V3 Large.

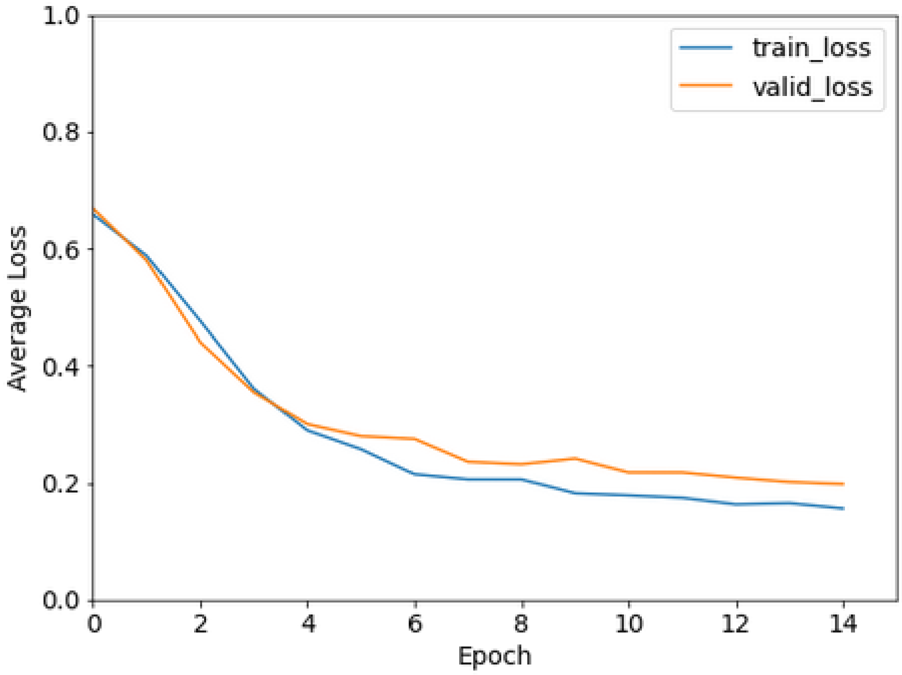


Figure 4.22: Loss vs. Epoch Graph of MobileNet V3 Large.

4.1.5. DenseNet Architectures

In this section, training and test results of DenseNet-121, DenseNet-161, DenseNet-169, and DenseNet-201 models are examined. According to their number of parameters and depth, DenseNets have high efficiency. Compared to other CNN models, DenseNets need fewer number of parameters to train a model and use the number of parameters more wisely for almost identical accuracy level. Another advantage of DenseNets is its architecture, making it easy to train the model. Every layer in the architecture has direct connection from loss function to gradients. This connection network with different points makes DenseNets deeper and easier to train. Unlike conventional Convolutional Neural Networks, DenseNets do not have exactly the same number of connections as the number of layers. Because with the Dense Convolutional Network approach, $\frac{L(L+1)}{2}$ connection was used in a model with an L layer. With dense blocks, a high-accuracy architecture can be created without any performance loss [58].

When Table 4.16 is examined, it is seen that DenseNet 121's training is finished without early stopping and it gives the best graphics at 0.01 learning rate and 64 batch size. The absence of early stopping during the training of DenseNet 121 extended the training period of DenseNet 121. Although DenseNet 121's 0.19 training loss and 92.25% training accuracy values were very successful, DenseNet 121 could not achieve the same success on test data. Especially when the test accuracy, test loss and precision values are examined, DenseNet 121 gave the most unsuccessful results not only among DenseNet architectures but also among all models. DenseNet 121's failing behavior in test results can be improved with a better prepared dataset and a larger grid search.

Table 4.16: Grid Search Performances of DenseNet-121.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.25	89.52	87.02
0.015	32	5	15	0.24	89.78	81.74
0.02	32	5	15	0.24	89.25	83.54
0.03	32	5	15	0.24	90.05	85.94
0.04	32	5	15	0.25	89.25	82.54
0.05	32	5	15	0.22	91.4	87.87
0.01	8	5	15	0.21	91.67	86.16
0.01	16	5	15	0.25	88.98	90.11
0.01	64	5	15	0.19	92.25	85.42
0.01	128	5	15	0.31	87.63	82.95
0.01	256	5	15	0.44	84.95	87.84
0.01	32	3	11	0.19	93.02	87.91
0.01	32	4	12	0.16	94.33	88.1
0.01	32	6	15	0.26	88.71	86.49
0.01	32	7	15	0.24	90.59	89.03

In Table 4.17, it is seen that DenseNet-161 gives the healthiest graph in 128 batch size. A 128 batch size training process requires a high power computer setup and an average power GPU may be insufficient. Just like DenseNet-121, DenseNet-161 did not have early stopping. Since DenseNet-161 has a deeper network than DenseNet-121, this has resulted in higher training time. DenseNet-121 completed the training process in 695.05 seconds, while DenseNet-161 completed it in 867.1 seconds. When the test results of DenseNet-161 are examined in Table 4.20, DenseNet-161 gave a much more successful result in test accuracy than DenseNet-121. But on test loss basis, the difference in loss of 0.07 between DenseNet-121 and DenseNet-161 is not that promising.

Table 4.17: Grid Search Performances of DenseNet-161.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	7	0.11	95.73	88.5
0.015	32	5	15	0.24	90.05	85.78
0.02	32	5	15	0.26	88.44	89.04
0.03	32	5	15	0.23	91.13	80.85
0.04	32	5	15	0.23	90.86	84.56
0.05	32	5	15	0.24	90.59	87.16
0.01	8	5	14	0.18	92.22	85.06
0.01	16	5	15	0.24	90.32	88.37
0.01	64	5	15	0.27	88.71	84.48
0.01	128	5	15	0.2	93.22	83.74
0.01	256	5	15	0.47	84.41	86.29
0.01	32	3	14	0.12	95.77	82.61
0.01	32	4	15	0.25	88.17	88.12
0.01	32	6	15	0.26	90.05	89.16
0.01	32	7	15	0.24	89.52	84.83

When Table 4.18 is examined, it is seen that DenseNet-169 gives its most successful graphics under the same conditions as DenseNet-121. Although DenseNet-169 has a deeper architecture than DenseNet-161, it lags behind DenseNet-161 with 92.4% training accuracy. DenseNet-169 also completed its training in 15 epochs like DenseNet-161 and DenseNet-121. However, despite the fact that the network has a deep architecture, the training time was 123.69 seconds shorter than DenseNet-161, contrary to expectations. Although DenseNet-169 has the same training loss value as DenseNet-121, it gave a slightly better result with 92.4% in terms of training accuracy. In Table 4.20, DenseNet-169's test accuracy and total test time value lagged behind DenseNet-161's. When only the total test time value of DenseNet-169 is examined, it is seen that it is the slowest model that analyzes the test data with 11.08 seconds in this thesis. However, DenseNet-169's recall value is 0.97, which is the highest value among other models.

Table 4.18: Grid Search Performances of DenseNet-169.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	10	0.24	90.22	90.05
0.015	32	5	15	0.19	91.4	85.93
0.02	32	5	14	0.12	94.97	81.1
0.03	32	5	15	0.19	91.94	89.56
0.04	32	5	15	0.18	93.28	80.5
0.05	32	5	15	0.19	92.74	87.75
0.01	8	5	15	0.2	91.67	80.32
0.01	16	5	15	0.2	92.47	86.98
0.01	64	5	15	0.19	92.4	89.94
0.01	128	5	15	0.27	89.52	89.34
0.01	256	5	15	0.45	87.37	83.21
0.01	32	3	6	0.25	90.78	89.6
0.01	32	4	12	0.15	94.13	86.29
0.01	32	6	15	0.2	91.94	88.41
0.01	32	7	15	0.2	92.2	89.52

Table 4.19: Grid Search Performances of DenseNet-201.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.18	93.9	85.16
0.015	32	5	14	0.13	95.05	83.51
0.02	32	5	15	0.23	89.52	86.4
0.03	32	5	15	0.23	91.67	82.44
0.04	32	5	15	0.2	92.2	82.44
0.05	32	5	15	0.2	92.74	89.01
0.01	8	5	15	0.21	91.94	80.69
0.01	16	5	15	0.22	89.78	84.82
0.01	64	5	15	0.26	89.25	83.38
0.01	128	5	15	0.33	87.1	83.97
0.01	256	5	15	0.47	84.14	84.82
0.01	32	3	15	0.15	94.37	84.49
0.01	32	4	9	0.18	92.74	84.05
0.01	32	6	14	0.24	88.98	86.49
0.01	32	7	15	0.23	89.78	85.59

Unlike other DenseNet models, DenseNet201 gave its most successful graphics in 32 batch sizes, but it took 15 epochs to train like other DenseNet models. When the training loss and training accuracy values of DenseNet201 were examined, it was seen that it surpassed all other DenseNet models. DenseNet201 has the deepest structure among DenseNet architectures. Therefore, it is expected to have a longer training time at the same epoch value. DenseNet201 has the longest training time of 895.58 seconds among other DenseNet architectures. However, the time it takes to calculate 1 epoch is 59.47 seconds, 2.47 seconds better than DenseNet161. When Table 4.20 is examined, DenseNet201's success on test data can easily be seen. While DenseNet201 is more successful than other DenseNet architectures in test accuracy, total test time, test loss, and precision values, it does not have the same success in recall and F1 score values.

Table 4.20: Test Results of DenseNet-121, DenseNet-161, DenseNet-169, and DenseNet-201 on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
DenseNet-121	75.80	10.12	0.47	0.79	0.95	0.86
DenseNet-161	83.33	9.98	0.40	0.84	0.95	0.89
DenseNet-169	82.79	11.08	0.38	0.84	0.97	0.90
DenseNet-201	87.09	8.03	0.29	0.85	0.90	0.87

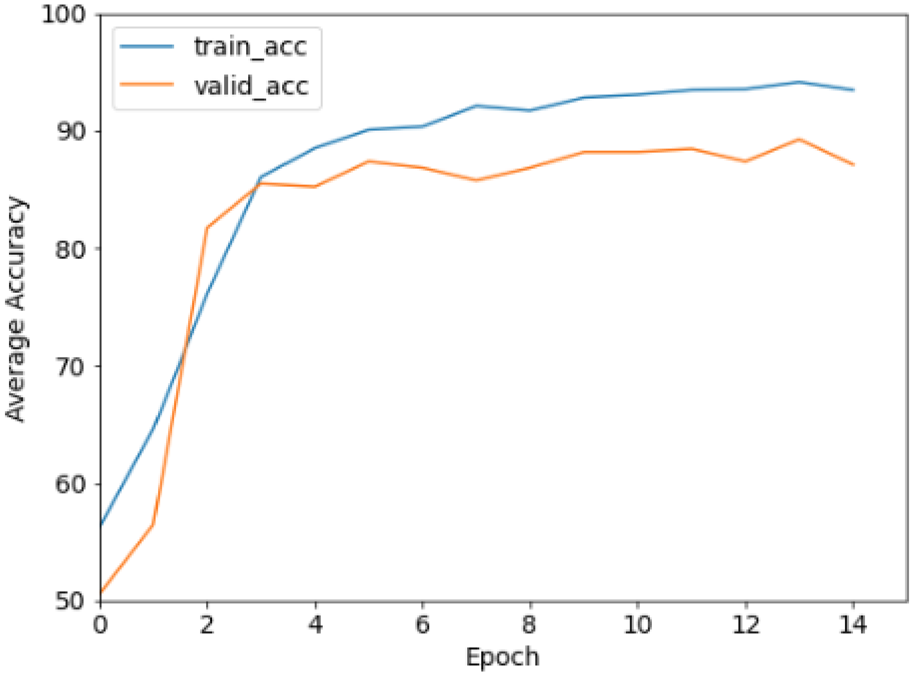


Figure 4.23: Accuracy vs. Epoch Graph of DenseNet-121.

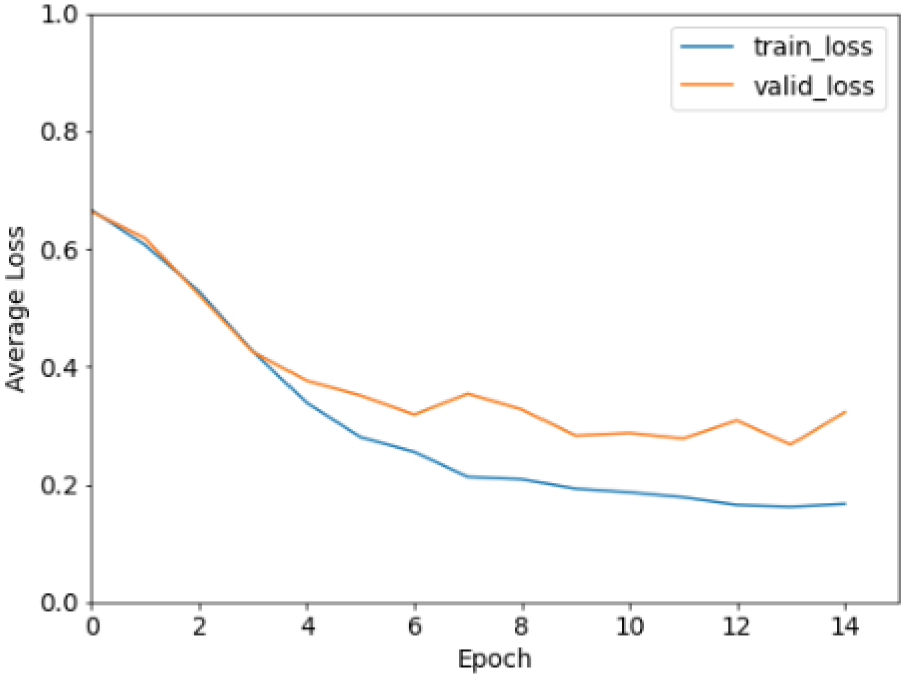


Figure 4.24: Loss vs. Epoch Graph of DenseNet-121.

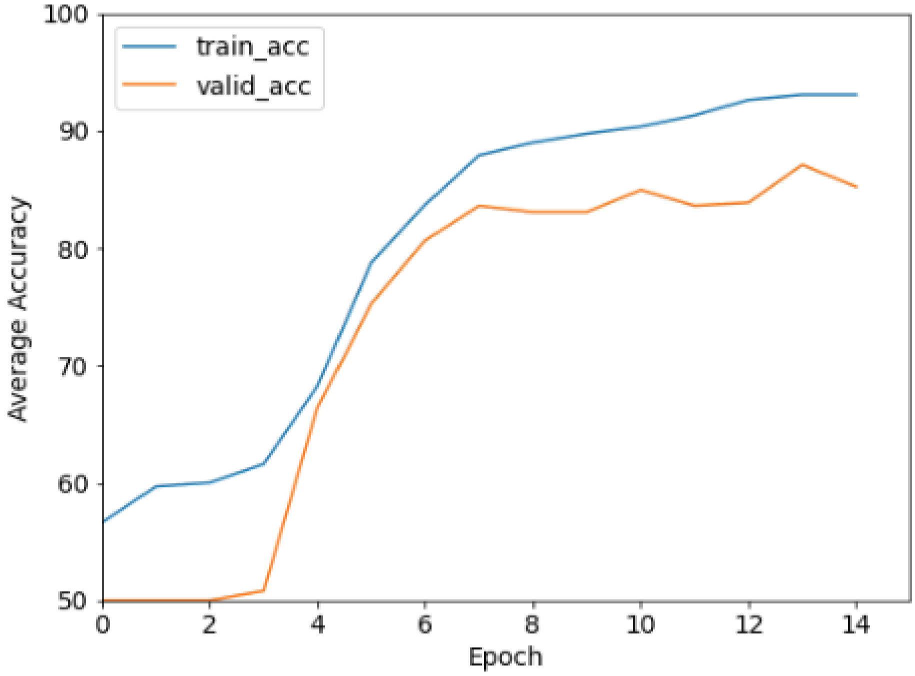


Figure 4.25: Accuracy vs. Epoch Graph of DenseNet-161.

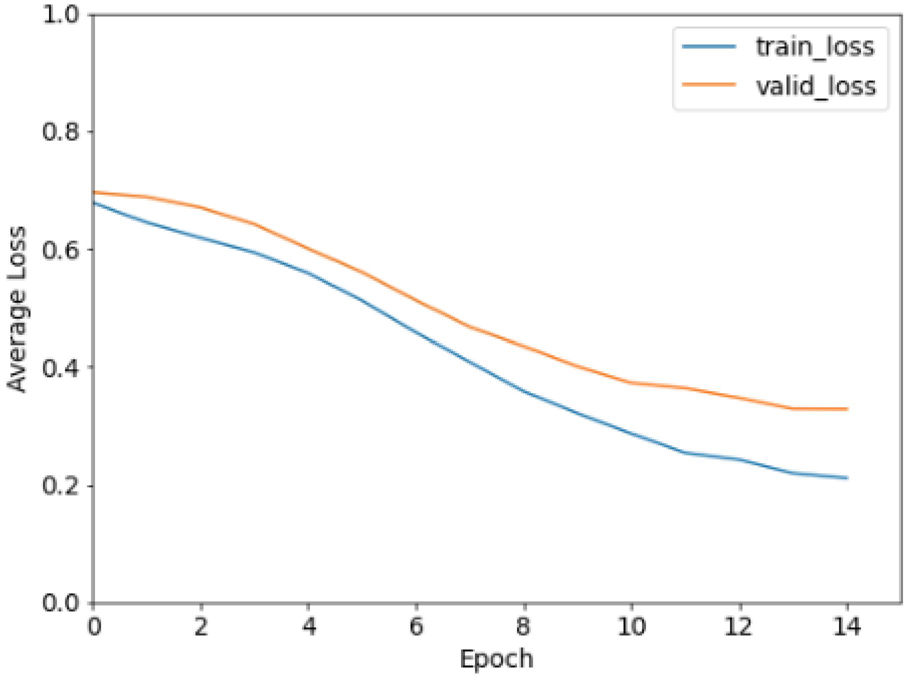


Figure 4.26: Loss vs. Epoch Graph of DenseNet-161.

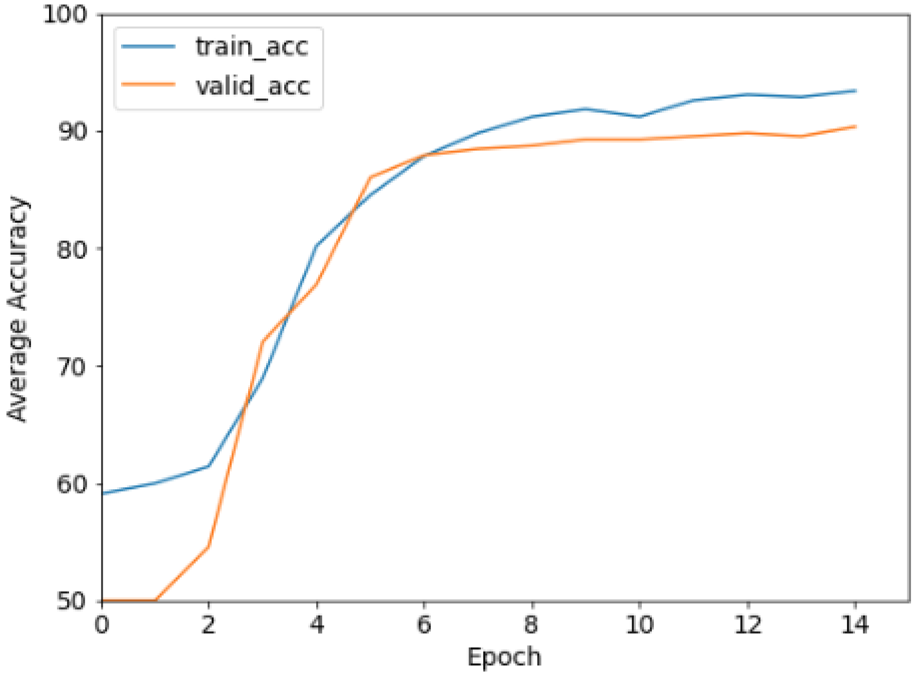


Figure 4.27: Accuracy vs. Epoch Graph of DenseNet-169.

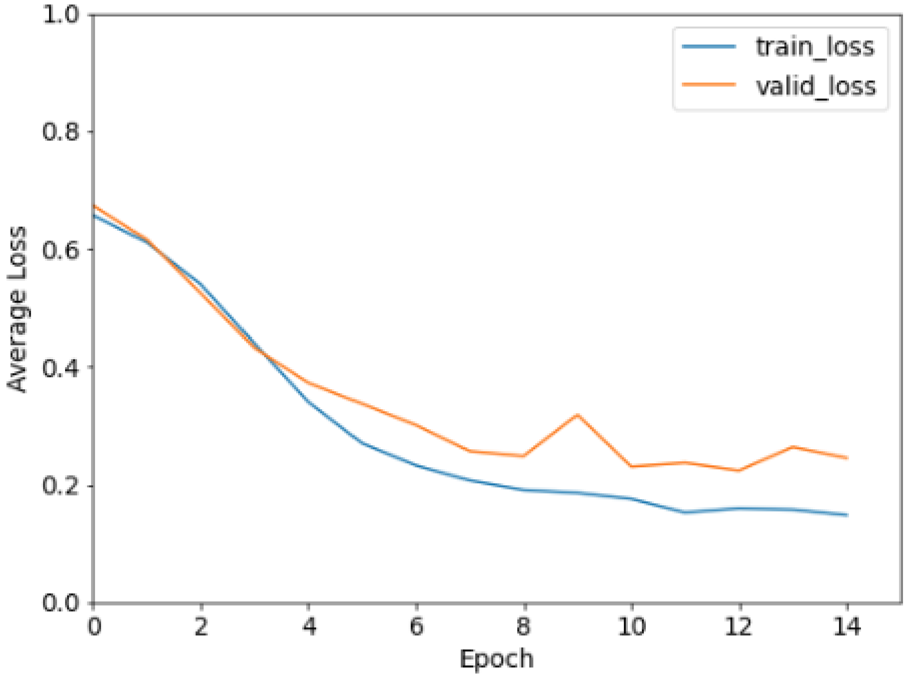


Figure 4.28: Loss vs. Epoch Graph of DenseNet-169.

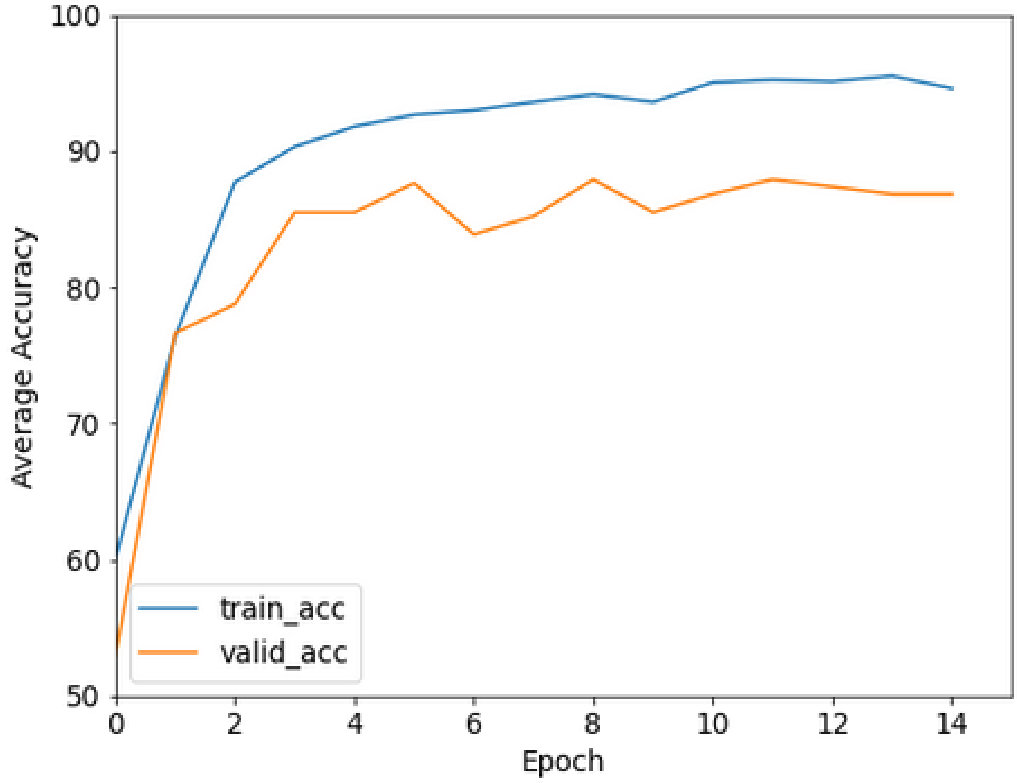


Figure 4.29: Accuracy vs. Epoch Graph of DenseNet-201.

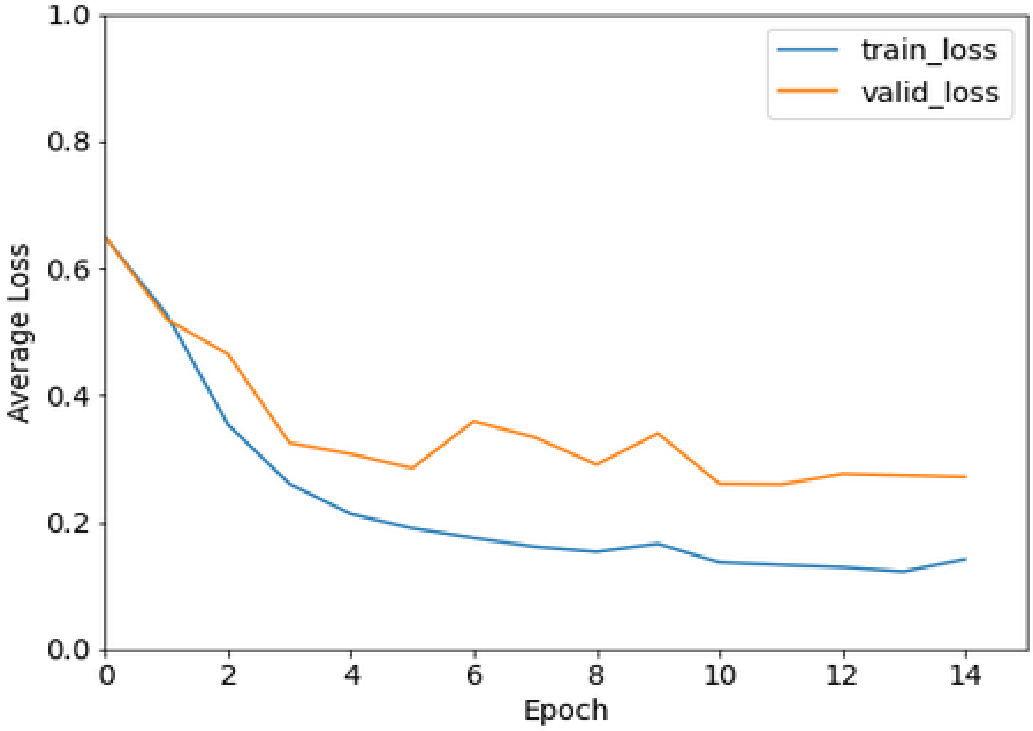


Figure 4.30: Loss vs. Epoch Graph of DenseNet-201.

4.1.6. EfficientNet Architectures

In this section, training and test results of EfficientNet B0, EfficientNet B1, EfficientNet B2, EfficientNet B3, EfficientNet B4, EfficientNet B5, EfficientNet B6, and EfficientNet B7 models are examined. There are some main factors such as the selection of the dataset according to its purpose and the sufficient architectural depth to make the training phase of convolutional neural networks efficient. The importance of the harmony between network width, depth, and resolution in creating a healthy model is presented with the compound scaling method. The success of this approach has been very clearly demonstrated by the EfficientNet-B7 in literature [59]. The EfficientNet models trained in this thesis showed some differences among themselves.

Table 4.21: Grid Search Performances of EfficientNet B0.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.25	88.98	90.22
0.015	32	5	15	0.26	87.1	86.26
0.02	32	5	15	0.25	88.71	81.25
0.03	32	5	14	0.17	93.06	86.46
0.04	32	5	15	0.21	89.78	81.25
0.05	32	5	13	0.16	93.7	82.75
0.01	8	5	15	0.27	87.9	88.99
0.01	16	5	15	0.21	91.32	86.84
0.01	64	5	15	0.35	87.1	90.18
0.01	128	5	15	0.58	73.39	87.22
0.01	256	5	15	0.67	50	50
0.01	32	3	15	0.28	88.71	81.54
0.01	32	4	15	0.26	88.44	80.69
0.01	32	6	15	0.25	90.32	80.25
0.01	32	7	15	0.26	87.9	82.01

EfficientNet B0 gave the best graphics at 0.01 learning rate, 16 batch size and 5 early stopping patience. EfficientNet B0 is one of the three models that gives the most successful graphics at 16 batch sizes among all the models in this thesis. The other two models are ResNet152 and MobileNetV2. When Table 4.21 is examined, it is seen that there is no early stopping in any case, except for the learning rate values of 0.03 and 0.05. Despite this situation, the training time of 638.75 seconds is not bad among other EfficientNet models. Examining Table 4.29 for the test results of

EfficientNet B0, it is seen that the lowest precision value among other EfficientNet models belongs to EfficientNet B0.

Table 4.22: Grid Search Performances of EfficientNet B1.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.23	92.63	86.48
0.015	32	5	15	0.32	88.44	88.68
0.02	32	5	15	0.29	88.98	86.95
0.03	32	5	15	0.3	88.44	87.17
0.04	32	5	15	0.28	88.98	86.68
0.05	32	5	15	0.27	89.52	87.9
0.01	8	5	15	0.34	85.48	81.81
0.01	16	5	15	0.25	89.58	84.92
0.01	64	5	15	0.4	85.22	85.12
0.01	128	5	15	0.6	62.37	50.58
0.01	256	5	15	0.69	50.32	52.73
0.01	32	3	15	0.34	88.17	83.73
0.01	32	4	15	0.34	87.37	85
0.01	32	6	15	0.34	86.29	85.15
0.01	32	7	15	0.34	86.56	89.62

As seen in Table 4.22, EfficientNet B1 model gave its most efficient graphics at 0.01 learning rate, 32 batch size and 5 early stopping patience. When the number of epochs before stopping values are examined, it is seen that there is no early stopping in the EfficientNet B1 model under any circumstances. When this situation is considered on the basis of training time, since the EfficientNet B1 model is deeper than the EfficientNet B0, it creates a higher training time expectation. Despite this, EfficientNet B1's training time was 548.07 seconds, 90.68 seconds shorter than EfficientNet B0. When the training loss values are examined, the EfficientNet B1 model has the highest value with 0.23 not only among the EfficientNet architectures but also among all the models. When the test results in Table 4.29 are examined, it is seen that among other EfficientNet architectures, EfficientNet B1 is the fastest model to examine test data on a total test time basis with 7.20 seconds.

When Table 4.23 is examined, it is seen that EfficientNet B2 gives the best graphics under the same conditions as EfficientNet B1. In addition, EfficientNet B2 and EfficientNet B1 have the same training loss as 0.23. However, these similarities change when Table 4.29 is examined. EfficientNet B2's test accuracy is the lowest of all EfficientNet architectures at 82.25%. When the test results of all EfficientNet models are examined, the EfficientNet B2 model has the highest value compared to other EfficientNet architectures with a test loss value of 0.33. On the other hand, EfficientNet B2's recall value is 0.93, which is the highest value among EfficientNet architectures.

Table 4.23: Grid Search Performances of EfficientNet B2.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.23	91.75	83.86
0.015	32	5	15	0.3	87.37	89.72
0.02	32	5	15	0.27	87.1	87.08
0.03	32	5	15	0.28	86.29	83.82
0.04	32	5	15	0.26	89.25	86.46
0.05	32	5	15	0.27	87.63	86.16
0.01	8	5	15	0.28	87.37	90.53
0.01	16	5	15	0.29	86.56	82.29
0.01	64	5	15	0.4	85.22	85.73
0.01	128	5	5	0.68	59.46	50.1
0.01	256	5	8	0.68	59.98	52.73
0.01	32	3	15	0.31	86.83	87.41
0.01	32	4	15	0.29	86.56	84.26
0.01	32	6	15	0.28	86.29	84.15
0.01	32	7	15	0.3	86.83	84.39

When Table 4.24 is examined, it is seen that the EfficientNet B3 model gives the most successful graphics at 0.04 learning rate and 32 batch size values. Looking at the table, it is seen that the training accuracy in cases with a learning rate of 0.01 decreases as the batch size increases.

At the same time, no early stopping was experienced in any of the situations in the table. When the training loss value of EfficientNet B3 is examined, it is seen that it is lower than the EfficientNet B0, B1, and B2 models. In addition, the EfficientNet B3's training accuracy is higher than that of the EfficientNet B0, B1, and B2 models.

The test results of the EfficientNet B3 model are excellent. When Table 4.29 is examined, it is seen that the test accuracy of EfficientNet B3 is 98.38%. This value is not only the highest among the EfficientNet architectures, but the highest among all other models. This is the same for test loss and precision values. EfficientNet B3's test loss and precision values are 0.08 and 0.97, respectively. These values are the highest values among all models.

Table 4.25 states that the EfficientNet B4 model gives the healthiest graphics at 0.03 learning rate and 32 batch size. While the EfficientNet B4 model achieved 91.78% training accuracy in 696.59 seconds, the EfficientNet B3 achieved 92.86% training accuracy in 670.73 seconds. In this case, the high training accuracy value expected from the model depth of EfficientNet B4 could not be met.

Looking at the test results of the EfficientNet B4 model in Table 4.29, it is seen that the test loss value is the second lowest value between the EfficientNet architectures with 0.17. At the same time, the EfficientNet B4 model is the second-best model among the EfficientNet architectures in terms of test accuracy, precision, and F1 score.

Table 4.24: Grid Search Performances of EfficientNet B3.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.3	87.1	87.62
0.015	32	5	15	0.28	90.59	87.71
0.02	32	5	15	0.26	88.71	82.75
0.03	32	5	15	0.25	88.98	89.61
0.04	32	5	15	0.19	92.86	86.47
0.05	32	5	15	0.22	90.05	85.07
0.01	8	5	15	0.26	88.17	84.77
0.01	16	5	15	0.26	89.78	83.88
0.01	64	5	15	0.48	82.53	85.7
0.01	128	5	15	0.61	59.94	51.81
0.01	256	5	15	0.65	59.62	52.8
0.01	32	3	15	0.3	87.37	88.86
0.01	32	4	15	0.28	89.25	80.7
0.01	32	6	15	0.3	88.44	83.37
0.01	32	7	15	0.29	87.63	87.84

According to Table 4.26, the EfficientNet B5 model gave its best graphics under the same conditions as the EfficientNet B4 model. Although the training loss values of EfficientNet B5 and EfficientNet B4 are the same, the training accuracy of EfficientNet B5 is higher than EfficientNet B4. However, as an important point, the EfficientNet B5 model achieved 93.92% training accuracy in 1179.78 seconds, while the EfficientNet B4 model achieved 91.78% training accuracy in 696.59 seconds. The training time of the EfficientNet B5 model is the longest of all models. When Table 4.29 is examined for the test results of EfficientNet B5, the total test time value is 9.29 seconds, which is the second worst model among the EfficientNet architectures.

Table 4.25: Grid Search Performances of EfficientNet B4.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	5	0.69	57.66	87.52
0.015	32	5	15	0.33	87.63	82.38
0.02	32	5	15	0.29	89.52	89.64
0.03	32	5	15	0.2	91.78	87.12
0.04	32	5	15	0.28	88.71	84.07
0.05	32	5	12	0.25	90.1	90.18
0.01	8	5	15	0.29	87.9	86.22
0.01	16	5	15	0.31	88.44	80.95
0.01	64	5	5	0.69	56.62	51.84
0.01	128	5	5	0.69	58.06	52.73
0.01	256	5	5	0.69	59.86	52.21
0.01	32	3	3	0.68	59.66	52.42
0.01	32	4	3	0.69	55.27	50.38
0.01	32	6	6	0.69	58.74	51.48
0.01	32	7	7	0.69	57.02	51.8

Table 4.26: Grid Search Performances of EfficientNet B5.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.34	87.11	86.51
0.015	32	5	15	0.35	85.75	81.5
0.02	32	5	15	0.31	88.44	88.26
0.03	32	5	15	0.2	93.92	86.45
0.04	32	5	15	0.29	88.17	81.87
0.05	32	5	12	0.21	91.78	85.38
0.01	8	5	15	0.4	85.75	80.51
0.01	16	5	15	0.26	89.47	82.49
0.01	64	5	15	0.69	59.02	52.23
0.01	128	5	15	0.69	59.7	52.78
0.01	256	5	7	0.69	58.62	51.23
0.01	32	3	15	0.39	86.56	84.92
0.01	32	4	15	0.41	83.87	88.63
0.01	32	6	15	0.4	85.22	89.48
0.01	32	7	15	0.4	84.68	80.36

In Table 4.27, it is seen that the most successful graphics of the EfficientNet B6 model are formed at a learning rate of 0.05. Among all EfficientNet architectures, EfficientNet B6 is the best model for both training loss and training accuracy. Looking at Table 4.29 for the test results of EfficientNet B6, it is seen that EfficientNet B6 has the worst total test time with 9.90 seconds and the worst precision with 0.79.

Table 4.27: Grid Search Performances of EfficientNet B6.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.46	81.99	84.16
0.015	32	5	15	0.27	89.35	84.22
0.02	32	5	15	0.24	89.98	86.43
0.03	32	5	15	0.21	91.94	87.82
0.04	32	5	15	0.2	91.66	81.23
0.05	32	5	15	0.17	93.93	82.67
0.01	8	5	15	0.26	89.58	80.37
0.01	16	5	15	0.26	90.02	83.25
0.01	64	5	15	0.69	55.75	52.83
0.01	128	5	15	0.68	59.66	51.25
0.01	256	5	15	0.69	59.62	52.41
0.01	32	3	15	0.45	84.68	80.56
0.01	32	4	15	0.45	82.8	87.8
0.01	32	6	15	0.37	87.72	80.74
0.01	32	7	15	0.34	85.79	86.77

Table 4.28: Grid Search Performances of EfficientNet B7.

Learning Rate	Batch Size	Early Stopping Patience	Epochs Before Stopping	Training Loss	Training Accuracy (%)	Validation Accuracy (%)
0.01	32	5	15	0.23	90.9	80.73
0.015	32	5	15	0.28	88.75	84.17
0.02	32	5	15	0.24	90.34	86.91
0.03	32	5	15	0.21	91.94	86.02
0.04	32	5	15	0.18	92.58	82.94
0.05	32	5	15	0.19	92.18	82.96
0.01	8	5	15	0.27	89.27	80.24
0.01	16	5	15	0.26	89.07	80.53
0.01	64	5	5	0.69	58.94	52.83
0.01	128	5	6	0.69	50.6	51.48
0.01	256	5	9	0.69	59.38	51.93
0.01	32	3	15	0.69	57.86	51.35
0.01	32	4	15	0.35	87.15	87.26
0.01	32	6	15	0.37	87.03	86.72
0.01	32	7	15	0.35	86.95	90.33

According to Table 4.28, EfficientNet B7 gave the most promising graphics at 0.01 learning rate and 32 batch size. The training performance of EfficientNet B7, when compared to other EfficientNet models, was the model that gave the worst results with 0.23 training loss and 90.9% test accuracy. When Table 4.29 is examined in terms of test results, although the EfficientNet B7 model has a better total test time value than EfficientNet B6, it could not exceed EfficientNet B6 in test accuracy.

Table 4.29: Test Results of EfficientNet B0, EfficientNet B1, EfficientNet B2, EfficientNet B3, EfficientNet B4, EfficientNet B5, EfficientNet B6, and EfficientNet B7 on Construction Machinery Images.

Model	Accuracy (%)	Total Time (sec)	Loss	Precision	Recall	F1 Score
EfficientNet B0	83.33	8.31	0.31	0.81	0.89	0.84
EfficientNet B1	88.17	7.20	0.22	0.86	0.83	0.84
EfficientNet B2	82.25	7.41	0.33	0.83	0.93	0.88
EfficientNet B3	98.38	9.22	0.08	0.97	0.81	0.88
EfficientNet B4	89.78	8.32	0.17	0.89	0.86	0.87
EfficientNet B5	89.12	9.29	0.27	0.85	0.87	0.85
EfficientNet B6	89.78	9.90	0.26	0.88	0.79	0.83
EfficientNet B7	87.63	8.75	0.29	0.86	0.86	0.86

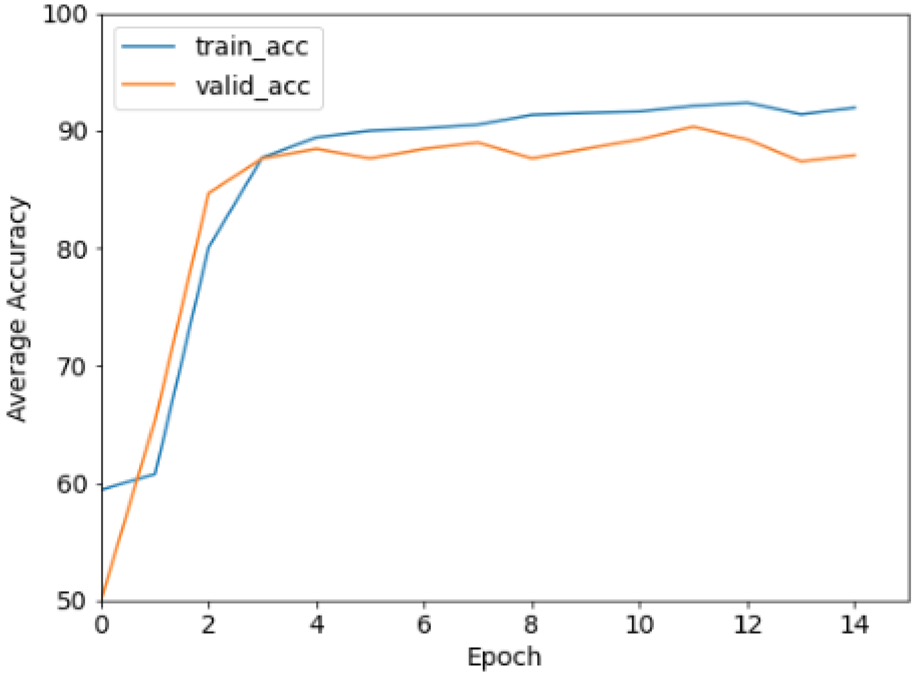


Figure 4.31: Accuracy vs. Epoch Graph of EfficientNet B0.

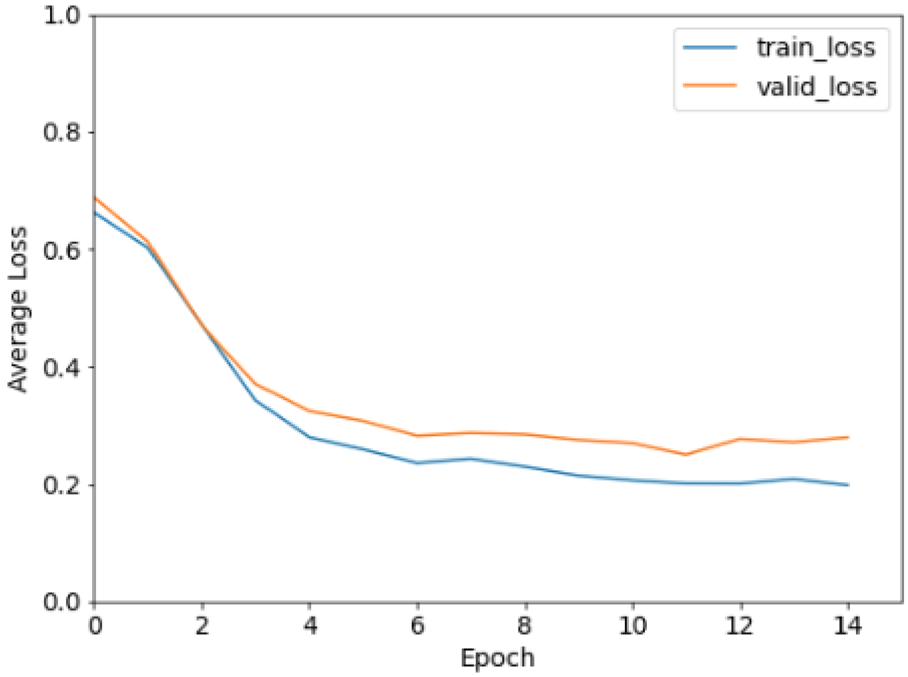


Figure 4.32: Loss vs. Epoch Graph of EfficientNet B0.

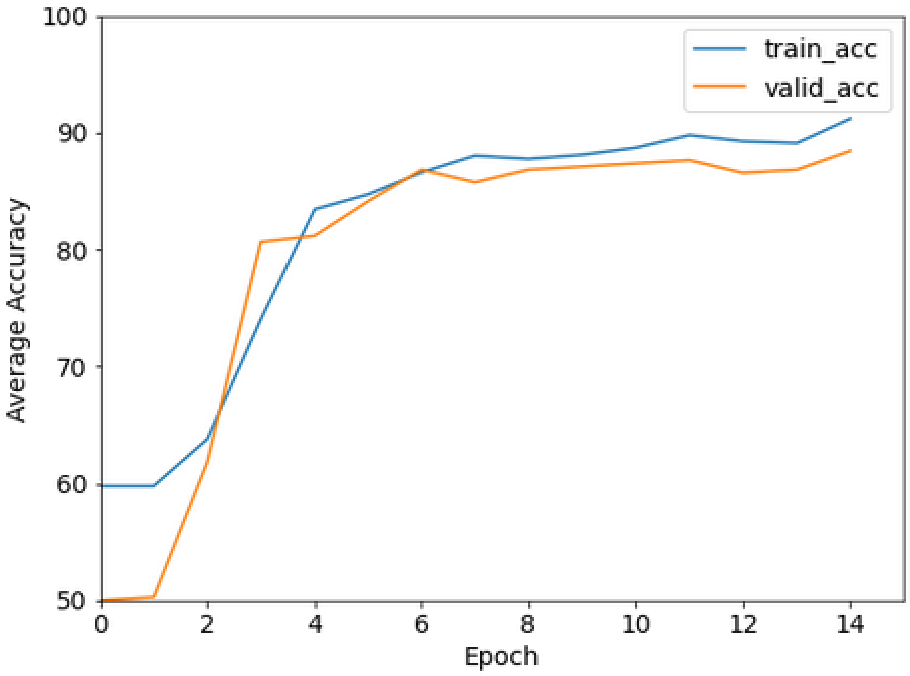


Figure 4.33: Accuracy vs. Epoch Graph of EfficientNet B1.

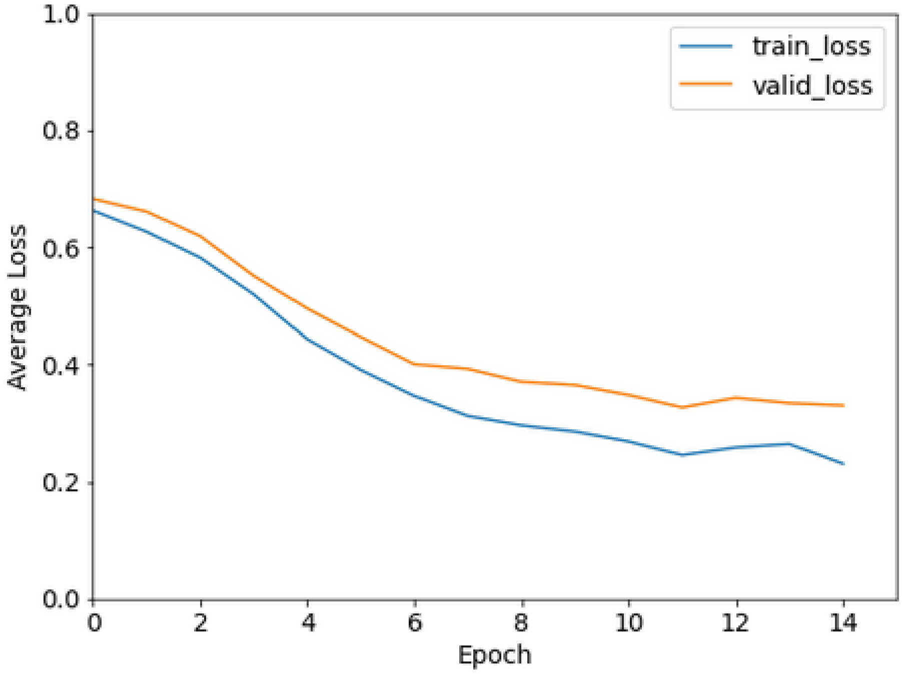


Figure 4.34: Loss vs. Epoch Graph of EfficientNet B1.

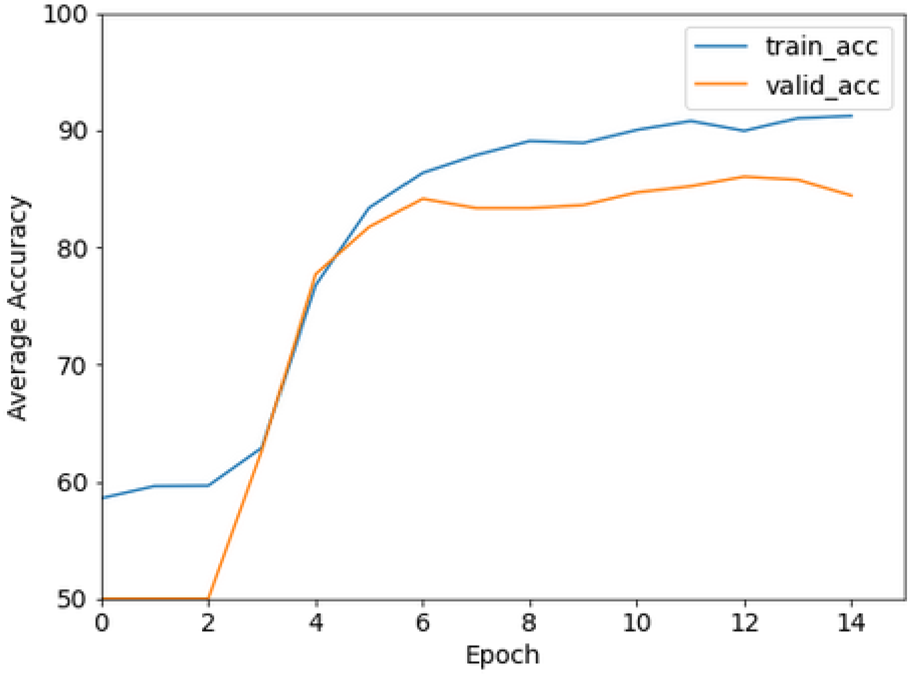


Figure 4.35: Accuracy vs. Epoch Graph of EfficientNet B2.

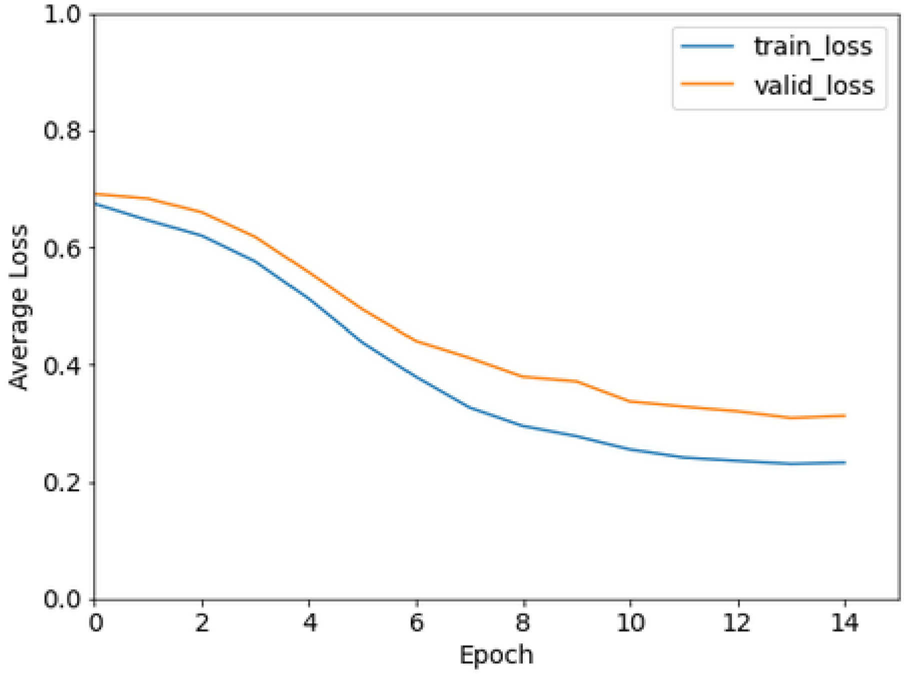


Figure 4.36: Loss vs. Epoch Graph of EfficientNet B2.

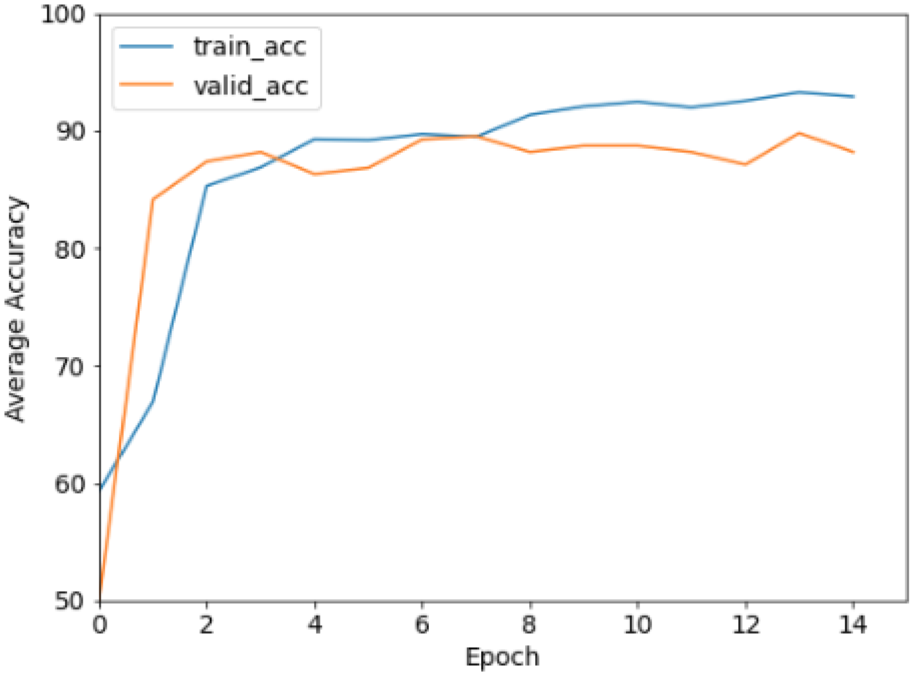


Figure 4.37: Accuracy vs. Epoch Graph of EfficientNet B3.

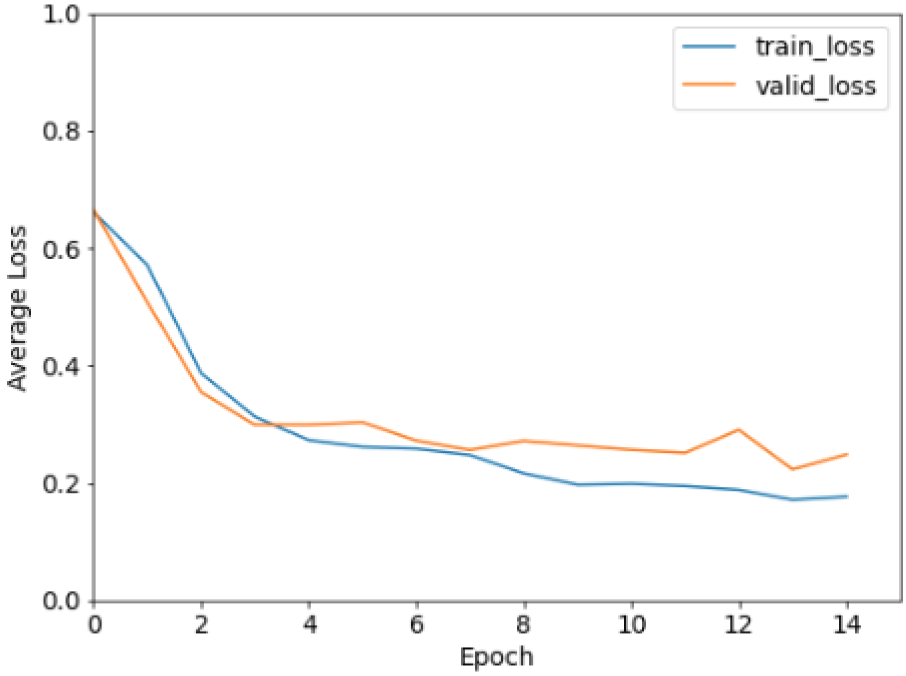


Figure 4.38: Loss vs. Epoch Graph of EfficientNet B3.

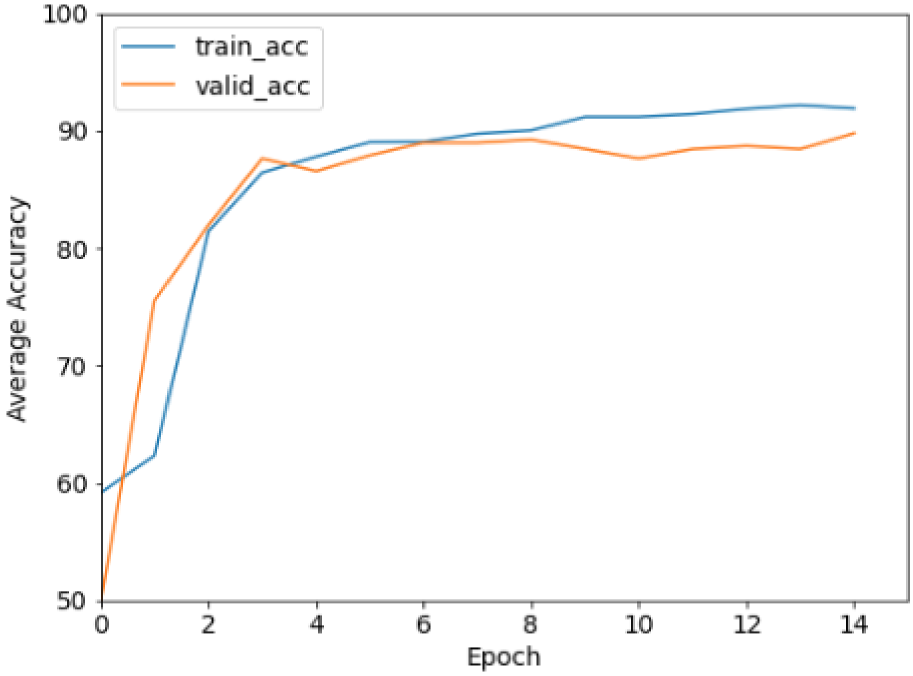


Figure 4.39: Accuracy vs. Epoch Graph of EfficientNet B4.

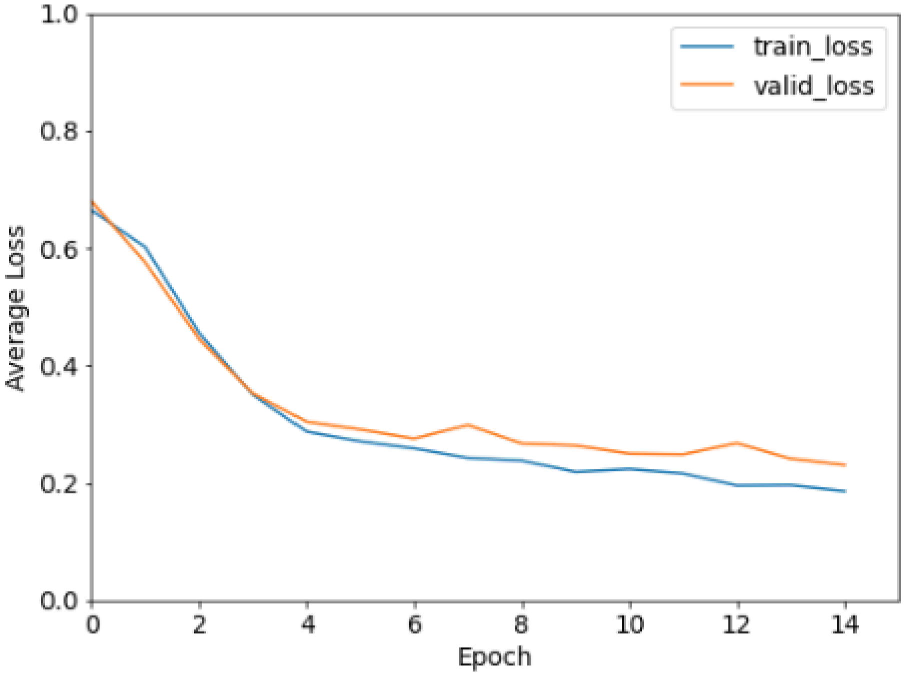


Figure 4.40: Loss vs. Epoch Graph of EfficientNet B4.

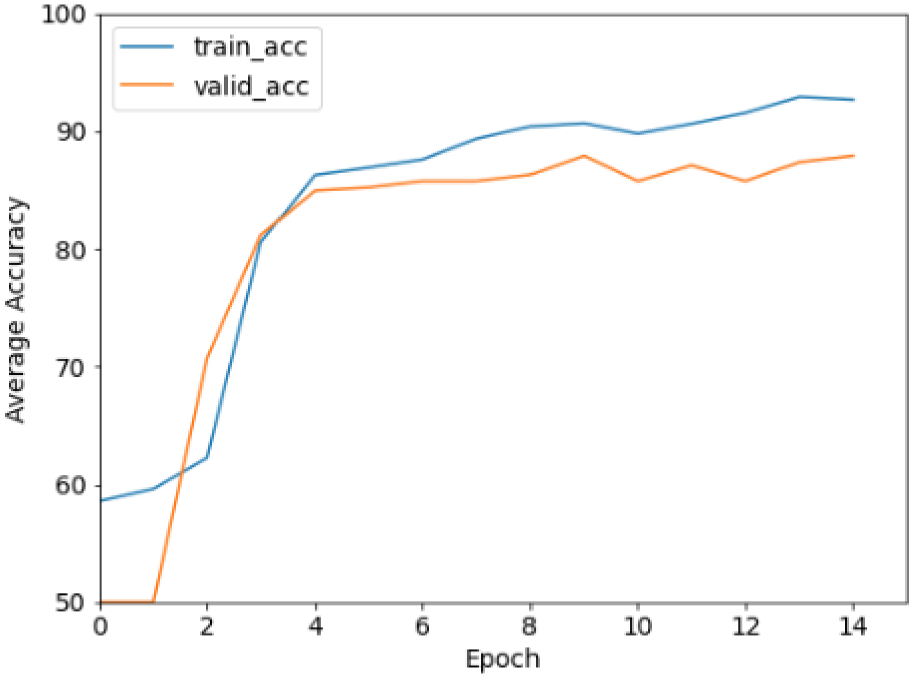


Figure 4.41: Accuracy vs. Epoch Graph of EfficientNet B5.

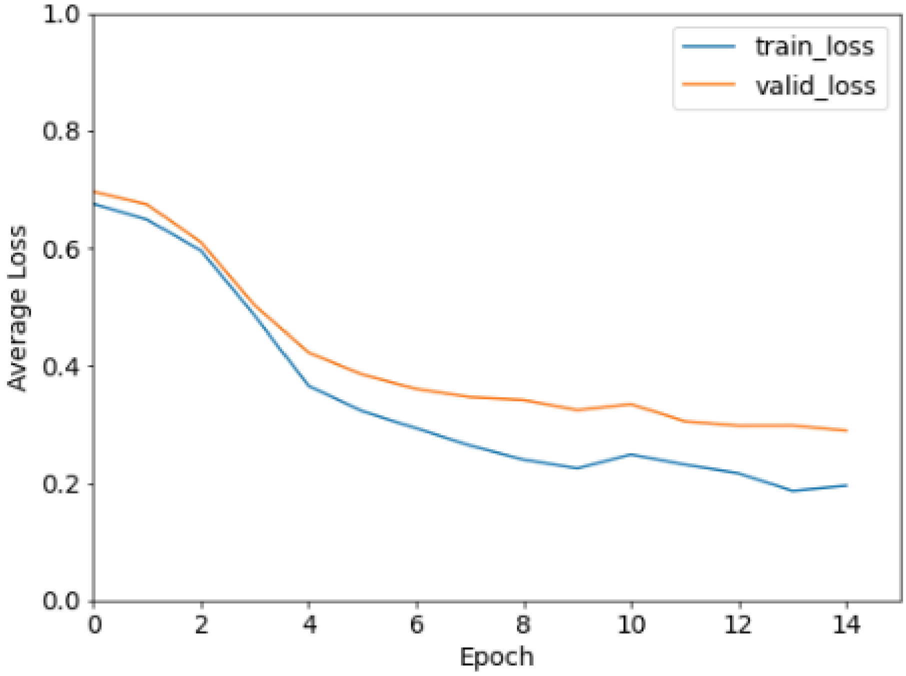


Figure 4.42: Loss vs. Epoch Graph of EfficientNet B5.

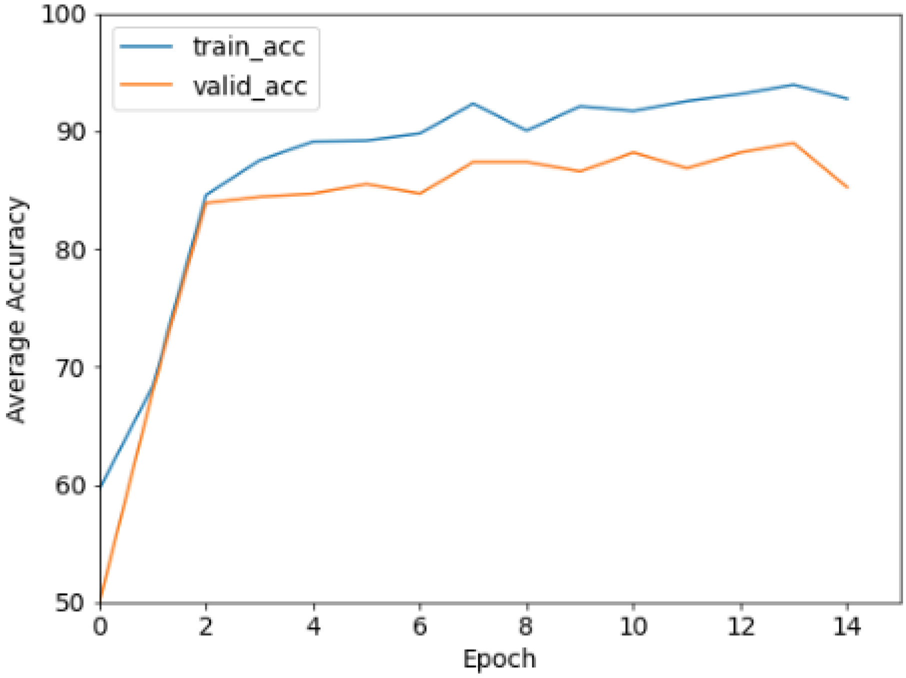


Figure 4.43: Accuracy vs. Epoch Graph of EfficientNet B6.

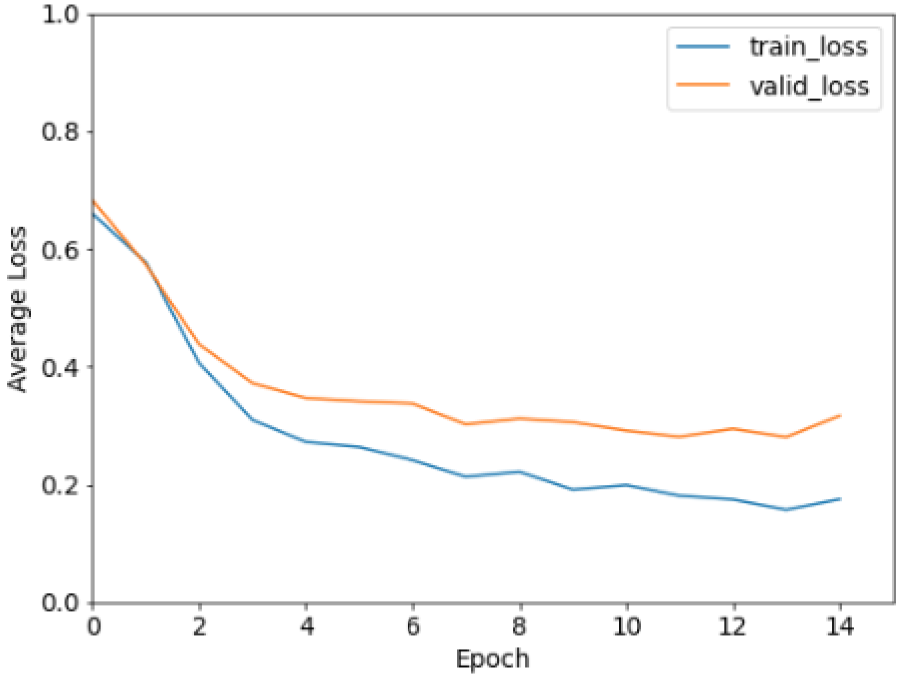


Figure 4.44: Loss vs. Epoch Graph of EfficientNet B6.

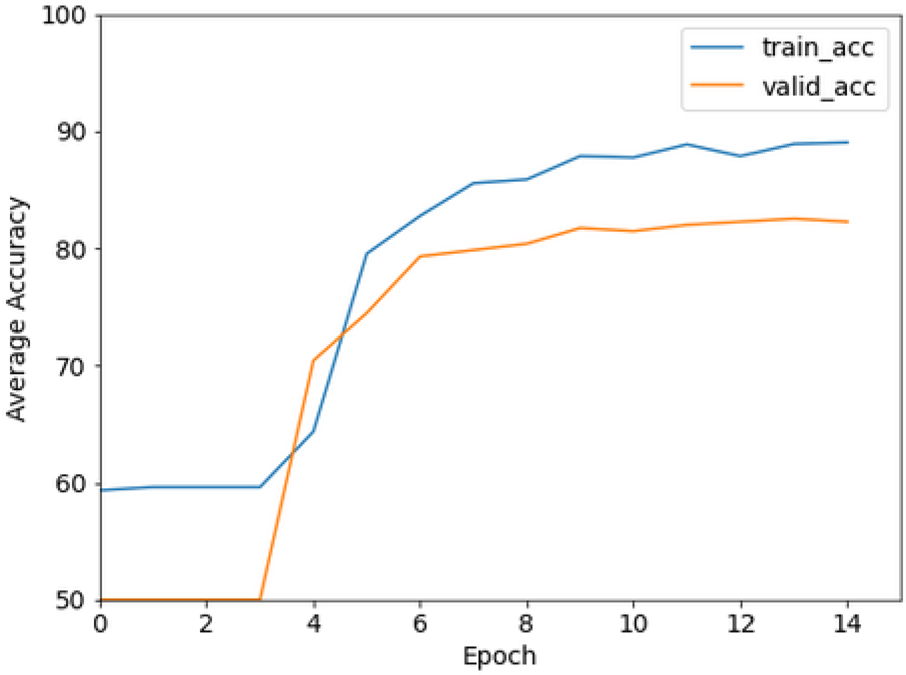


Figure 4.45: Accuracy vs. Epoch Graph of EfficientNet B7.

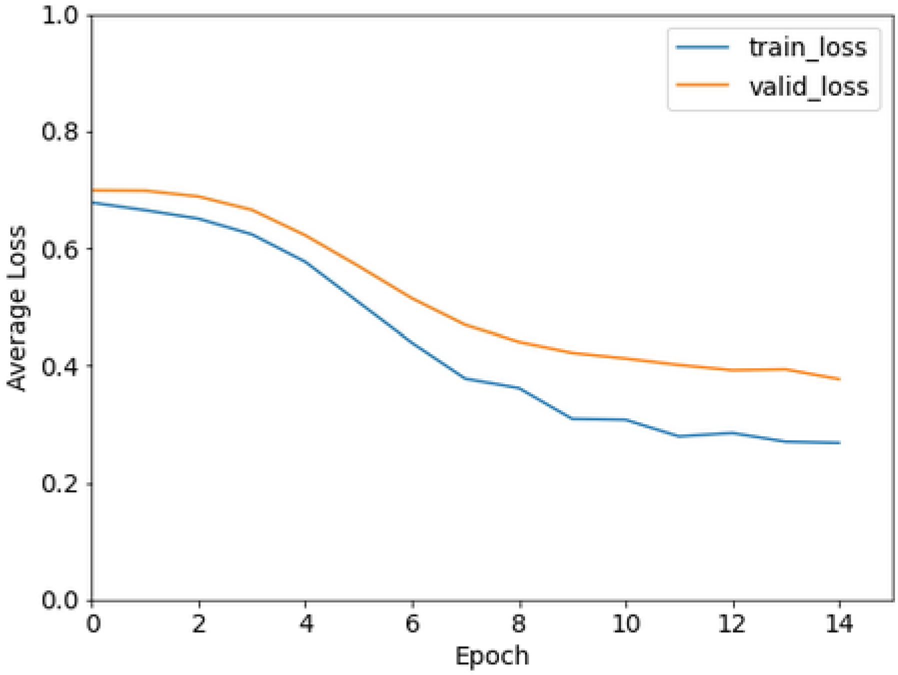


Figure 4.46: Loss vs. Epoch Graph of EfficientNet B7.

CHAPTER V

DISCUSSION

In this chapter, the models trained and tested in the previous chapter were subjected to an additional testing phase. The purpose of this test phase, which consists of satellite images of different difficulty levels, is to measure the classification performance of all models in this thesis under different conditions. 6 satellite images, 3 of which are construction machines and 3 of which are non-construction machines, at easy, medium, and hard levels, which the models have never seen before, were examined. At the end of this chapter, the performances of the models in these satellite images were compared and the most successful model was selected.

5.1 MODEL TESTING WITH IMAGES OF DIFFERENT DIFFICULTY LEVELS

When the results obtained in this thesis are compared with the results in the literature, some points draw attention. When the study conducted by *Arabi et al.* [50] is examined, it is seen that there are 3271 images in the dataset they used. This number is 21 more than the number of images used in this thesis. In the study, which includes street view images, MobileNet was used as the base model. The minimum precision value is 83.70% for the excavator, while the maximum precision value is 96.94%, which belongs to the mixer truck. When the results of the MobileNet model in *Arabi et al.*'s study and the MobileNet architectures in this thesis are examined, it is seen that the lowest precision among the MobileNet architectures in this thesis belongs to MobileNetV2 with 86.16%.

The total number of images of the dataset used by *Guo et al.* [33] in their construction machinery study with drone images is 240. The authors used 216 images of this dataset for training and 24 for testing. Although *Guo et al.*'s VGG16 model achieves 98.8% precision and surpasses the VGG16 model in this thesis, this value, which was tested with only 24 images, is not very healthy.

Looking at the literature, the dataset used in *Li et al.*'s [38] study is at an important point. The aim of the study, which performs binary classification with satellite images, just like in this thesis, is to detect cloud and non-cloud images. *Li et al.* used a dataset of more than 200.000 satellite images for their study. As a result of the study, the test accuracy value was 0.96 and the F1 value was 0.88. Although the highest test accuracy value in this thesis is 0.98 and the highest F1 value is 0.92, it is not interesting that these values are higher than *Li et al.* because the dataset used in this thesis is smaller than the dataset used in *Li et al.*'s study.

When the general model results up to this point are examined, it is seen that some models attract more attention than others and are one step ahead. When the total training time data is examined, it is seen that AlexNet is the fastest trained model with 357.82 seconds, while EfficientNet B5 is the slowest trained model with 1179.78 seconds. In addition to this information, AlexNet's calculation time for an epoch during training gave the fastest result with 32.53 seconds, while EfficientNet B5 gave the slowest result with 84.27 seconds.

When the training loss values are examined, it is seen that VGG19 is the model with the least loss value with 0.1. The highest loss values with 0.23 belong to EfficientNet B1, EfficientNet B2, and EfficientNet B7 models. VGG19 has not only the lowest loss but also the highest training accuracy at 96.33%. In contrast, ResNet34 has the lowest training accuracy of 90.17%.

Considering the test results, the remarkable speed of AlexNet during the training process continued to be effective when examining the test data. AlexNet was again the fastest model by examining the test dataset in a total of 7.10 seconds. On the other hand, DenseNet169 analyzed the test dataset in 11.08 seconds and was the slowest model.

Considering the test accuracy, test loss, and precision values, it can be easily said that the EfficientNet B3 and DenseNet121 models are at completely opposite poles. This is because the test accuracy, test loss, and precision values of EfficientNet B3 have the highest values with 98.38%, 0.08, and 0.97, respectively, while the values provided by DenseNet121 have the lowest values with 75.80%, 0.47, and 0.79, respectively. When the recall values are examined, the difference between DenseNet169 and ResNet152 models is very large. While DenseNet169 gave the best results with a recall value of 0.97, ResNet152 gave the worst result with a recall value of 0.77. In addition to these comparisons, on the basis of the F1 score, VGG19 gave the highest result with 0.92, while ResNet152 gave the lowest result with 0.83.

The purpose of this evaluation phase is to see the responses of the models trained for this thesis to images of different difficulty levels. The evaluation results obtained at the end of the training processes of deep learning algorithms are important to measure the success of the model. Unfortunately, it is not entirely correct to make an inference just by examining the training accuracy of a model. It is a very accurate way to give some images that the model has not seen before as input and see its reaction to those images.

To put this thesis on a fairer ground, a total of 6 satellite images, 3 of which are construction machines and 3 of which are non-construction machines, were taken from Google Earth. These images, which were determined as easy, medium, and difficult to classify, were placed in each difficulty level as 1 construction machine and 1 non-construction machine. Images can be viewed in detail in Figure 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6.

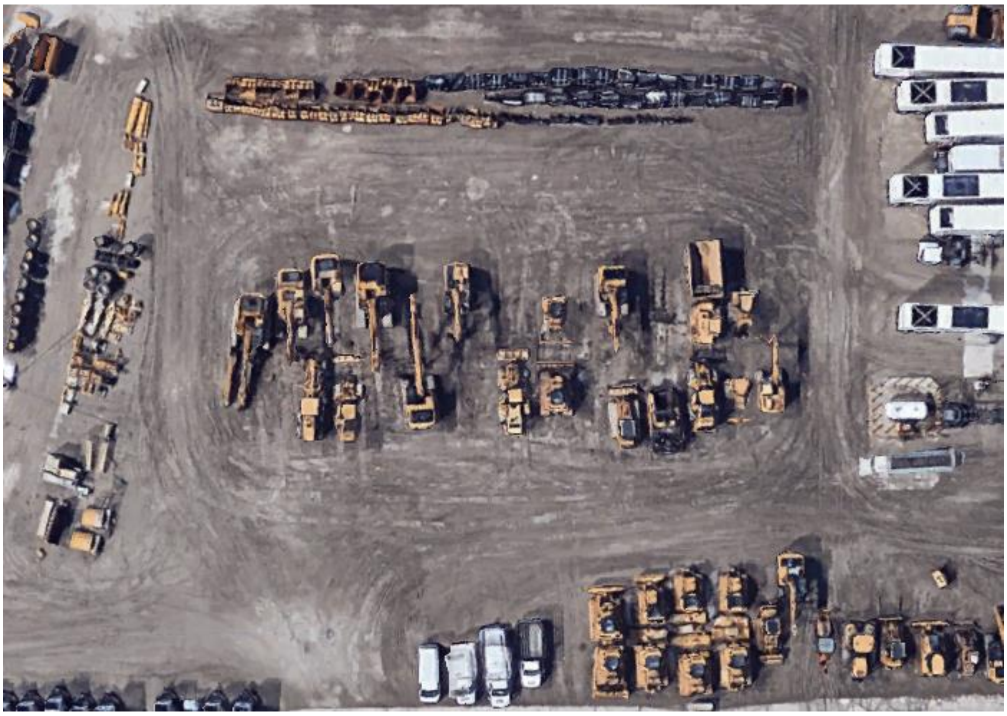


Figure 5.1: Easy-to-Classify Construction Machine Satellite Image.

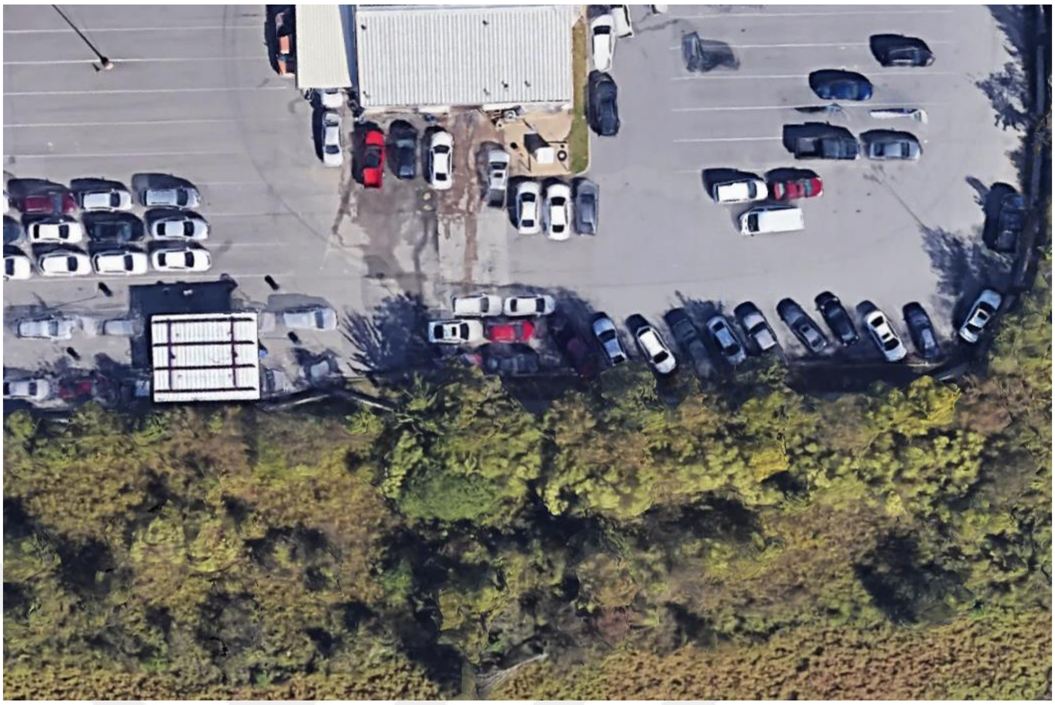


Figure 5.2: Easy-to-Classify Non-Construction Machine Satellite Image.

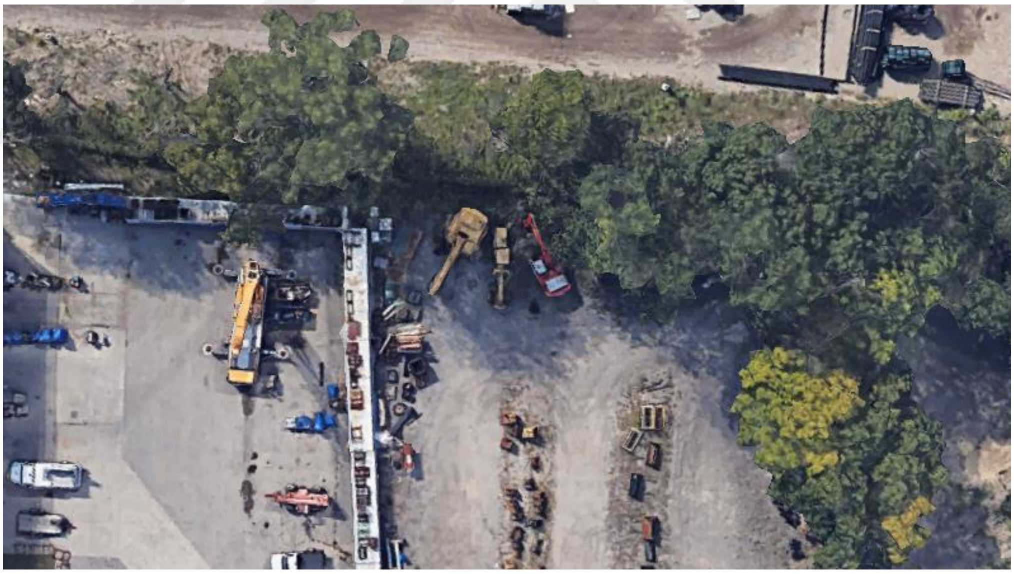


Figure 5.3: Medium-to-Classify Construction Machine Satellite Image.

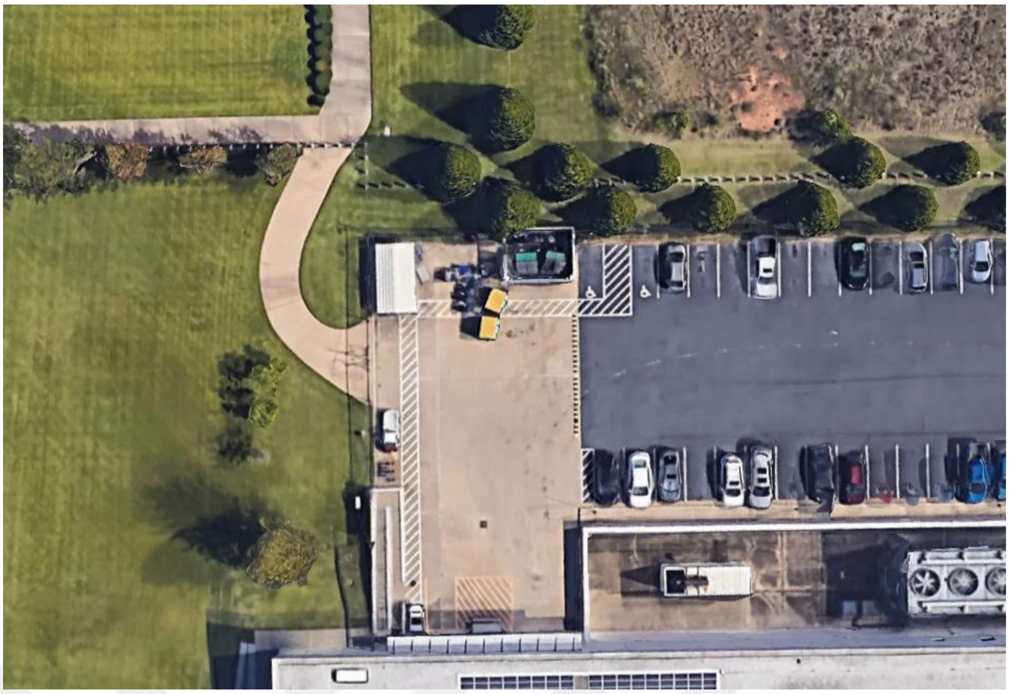


Figure 5.4: Medium-to-Classify Non-Construction Machine Satellite Image.

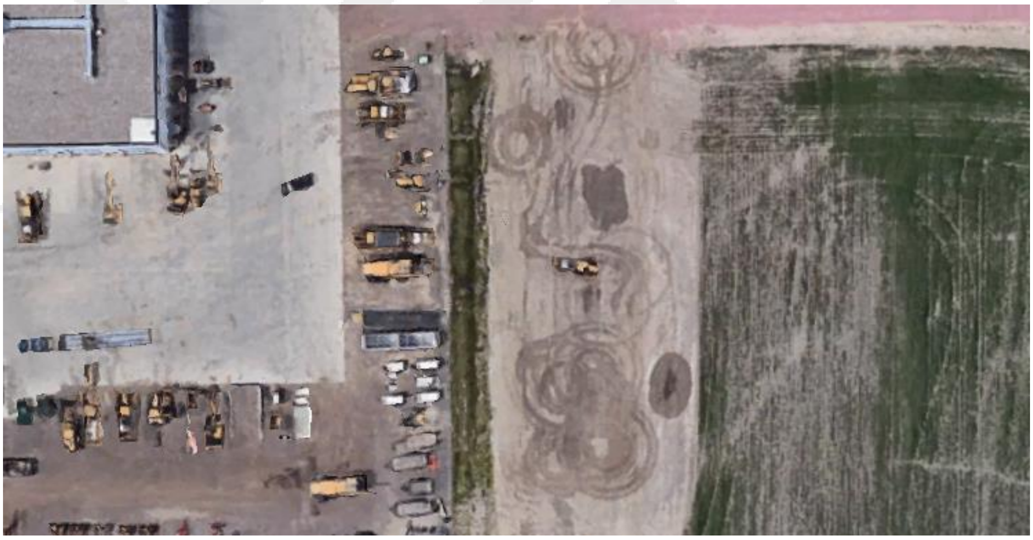


Figure 5.5: Hard-to-Classify Construction Machine Satellite Image.

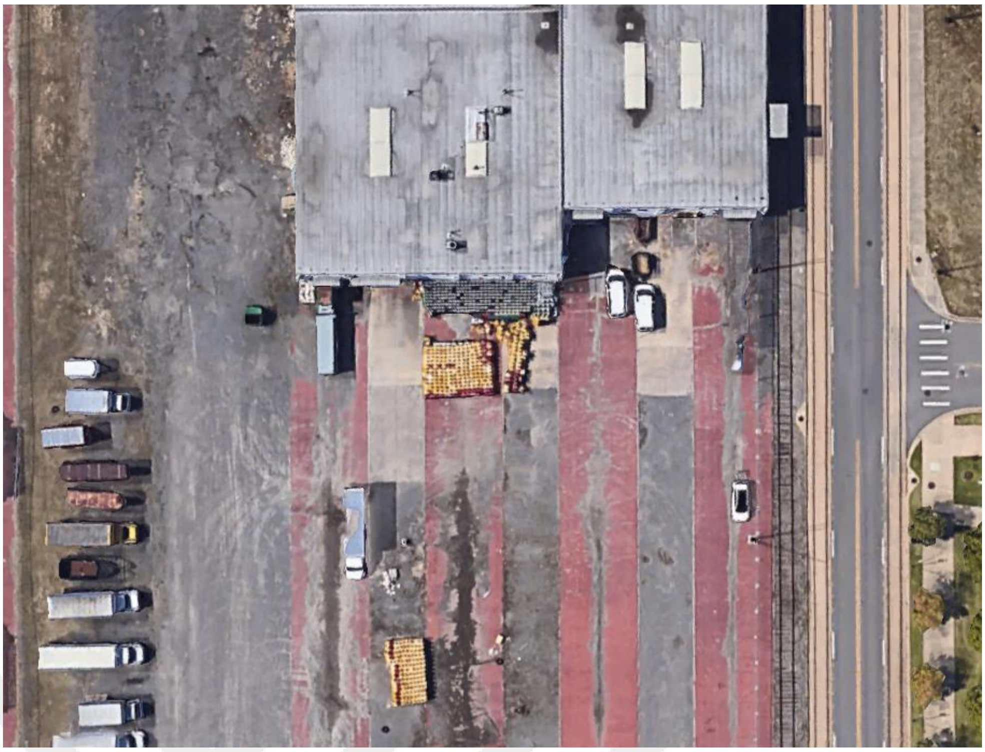


Figure 5.6: Hard-to-Classify Non-Construction Machine Satellite Image.

As a piece of detailed information, satellite images from Google Earth cannot always provide the same quality resolution as images taken with a digital camera because Google Earth works with different satellite image providers.

External factors such as shadow, an interlacing of multiple objects, and objects that are very similar to a construction machine are some of the factors that affect the difficulty of the images. The reason why Figure 5.1 and Figure 5.2 are easy-to-classify satellite images is that there is nothing to cause confusion. Especially when Figure 5.1 is examined, all the construction machines captured in the satellite image can be seen clearly.

Figure 5.3 is a medium-to-classify satellite image because the shadow from the trees around the construction machine is worthy of attention. The reason why Figure 5.4 is a medium-to-classify satellite image is that the yellow rectangular object in the middle of the image resembles a work machine. A model should not be confused by this similarity.

The construction machines in Figure 5.5 look smaller and have lower resolution compared to other satellite images. In fact, some parts of construction machines are not fully visible due to satellite image providers. This makes Figure 5.5 a hard-to-classify satellite image. Figure 5.6 is a hard-to-classify non-construction machine image, as Figure 5.6 has more of a construction machine-like image than Figure 5.4, and the yellow vehicle on the left side of the satellite image is likely to cause confusion.

Table 5.1: Classification Performances of All Models at Different Difficulty Levels.

Models	Easy to classify		Medium to classify		Hard to classify	
	CM	NCM	CM	NCM	CM	NCM
AlexNet	✓	✓	✓	✓	✗	✗
VGG16	✓	✓	✓	✓	✗	✓
VGG19	✓	✓	✓	✓	✗	✓
ResNet-18	✓	✓	✓	✓	✗	✗
ResNet-34	✓	✓	✗	✓	✓	✓
ResNet-50	✓	✓	✓	✓	✗	✓
ResNet-101	✓	✓	✓	✓	✓	✓
ResNet-152	✓	✓	✓	✓	✓	✓
MobileNet V2	✓	✓	✓	✓	✓	✓
MobileNet V3 Small	✓	✓	✓	✓	✗	✓
MobileNet V3 Large	✓	✓	✓	✓	✓	✓
DenseNet-121	✓	✓	✓	✓	✗	✓
DenseNet-161	✓	✗	✓	✓	✗	✓
DenseNet-169	✓	✓	✗	✓	✗	✓
DenseNet-201	✓	✓	✓	✓	✗	✓
EfficientNet B0	✓	✓	✓	✓	✓	✓
EfficientNet B1	✓	✓	✓	✓	✓	✓
EfficientNet B2	✓	✓	✓	✓	✗	✓
EfficientNet B3	✓	✓	✓	✓	✓	✓
EfficientNet B4	✓	✓	✓	✓	✓	✓
EfficientNet B5	✓	✓	✓	✓	✗	✓
EfficientNet B6	✗	✓	✓	✓	✓	✓
EfficientNet B7	✓	✓	✓	✓	✓	✓

In Table 5.1, classification performances of all models trained in this thesis on satellite images at different difficulty levels are shown. If a model classified an image correctly, it was ticked, and box was marked with a cross, if it was classified incorrectly.

When the predictions for Figure 5.1, which is an easy to classify image, are examined, it is seen that all models make correct predictions except for EfficientNet

B6. On the other hand, only the DenseNet-161 model misclassified Figure 5.2, which is an easy to classify non construction machine image.

Only ResNet-34 and DenseNet-169 misclassified Figure 5.3, which is a medium-to-classify construction machine image, while other models predicted correctly. In Figure 5.4, which is a medium to classify construction machine image, all models have been classified correctly.

When it came to hard-to-classify satellite imagery, things started to get messy. Among the 23 models, 11 models correctly classified Figure 5.5, which is the image of a hard-to-classify construction machine, while the remaining 12 models misclassified Figure 5.5.

When the predictions of the models for the hard to classify non construction machine satellite image in Figure 5.6 are examined, all other models have correctly classified Figure 5.6, except for AlexNet and ResNet-18 models.

When Table 5.1is examined, it is seen that 9 models out of 23 correctly classify all satellite images at all difficulty levels. These models are ResNet-101, ResNet-152, MobileNet V2, MobileNet V3 Large, EfficientNet B0, EfficientNet B1, EfficientNet B3, EfficientNet B4, and EfficientNet B7.

When the overall model results of the architectures that predict all images correctly are evaluated, EfficientNet B3's test accuracy value of 98.38%, test loss value of 0.08 and precision of 0.97 on test images distinguish EfficientNet B3 from other models. EfficientNet B3's outstanding success in both the test dataset consisting of 10% of the training dataset and its success on images with different difficulty levels made EfficientNet B3 the most successful model in this thesis.

CHAPTER VI

CONCLUSION

The ever-changing and developing technology does not limit the use of construction machinery only to the works carried out in legal ways, but it can also continue its activities in areas where fixed-location cameras and drone flights are likely to be inaccessible. Both for this point and in situations where drone activities may be limited, the importance of easy and global access of satellite images emerges. With construction machinery detection applications performed using satellite images, both manpower and time can be saved. This technology can be used in many large-scale construction projects as well as in any sector other than the construction industry.

Each construction project has its specific purpose and responsibilities to fulfill. The level of complexity and size of the projects vary according to the purpose they address. In addition, the size of a project depends on how complex it is. According to *Vital et al.* [60] the more complex a project, the larger the construction site.

Construction projects can be examined under four headings in terms of complexity and size of construction sites according to the needs of the society they serve. The first and most common of these are residential projects. The level of complexity of residential construction sites is not high considering the whole construction industry. Therefore, it is difficult to say that residential projects are large construction sites. Secondly, when commercial projects are examined, they have a more complex structure than residential projects in terms of purpose. Since commercial buildings do not have to be as close to the center of the city as residential buildings, the area covered by the construction site tends to be wider. When industrial structures are considered as the third title, it is seen that the construction site size and complexity have increased even more. Especially considering the size and complexity of industrial mass production factories, it is seen that it is generally spread over large areas in parts far from the city. Fourthly, if infrastructure projects are examined, it is not even sincere to see how large and complex these projects are. Especially when the road projects are taken into account, the size of the construction site spreads kilometers.

Drones are mostly used in relatively small or medium-sized construction sites for construction activity monitoring. Although drones perform great work, they can be inefficient when it comes to very large construction sites. When the issue is considered in terms of manpower, time consumption and cost, using satellite images instead of drones for construction activity monitoring in large sites can be a promising solution.

In this thesis, construction machinery detection is performed using satellite images. Satellite images from various states of The United States of America were used to train 23 pre-trained deep learning models with the transfer learning method. The custom construction machinery dataset consisting of 3250 satellite images with different kind of construction machinery placed on different ground types is created from scratch using Google Earth. To the best of the authors' knowledge, no study detects construction machinery using satellite images although few studies in the literature perform construction machinery detection using drone images.

For the sake of this thesis, 23 different models, each with different parameter numbers and depths, were trained using the Python programming language. After the training phase, the results of all models were analyzed and EfficientNet B3 was the model that gave the best results.

6 new satellite images at 3 different difficulty levels were taken using Google Earth in order to obtain fairer, accurate, and healthy results without relying only on the training results. Satellite imagery of 1 construction machinery and 1 non-construction machinery at each difficulty level was used to test all models. As a result of this test phase, the EfficientNet B3 model, which correctly classifies 6 images at easy, medium, and hard difficulty levels, was chosen as the most efficient and successful model in this thesis.

Although the success of EfficientNet B3 in detecting construction machinery using satellite images is an indisputable fact, more efficient models can be obtained with more types of construction machinery, datasets containing more satellite images, and more different hyper parameter research.

As future work, danger zones of construction machineries can also be detected in addition to the construction machinery classification. In the future, this technology can be used in construction sites that can be viewed with real-time live satellite images, and can be combined with a system that warns before occupational accidents occur. Construction sites inherently contain too many dangerous items. These items, which directly affect the lives of both workers and employers, can cause serious injuries or even death. Like any issue concerning human life, construction sites must have high priorities that control the dangerous workflow. The concept of construction activity monitoring comes to the fore in this regard. Every construction machinery has a hypothetical danger zone that can lead to fatal accidents. The zone, which is almost impossible to measure with the human eye at a single glance, can be detected using satellite imagery and artificial intelligence. There are some blind spots where construction machine operators cannot interfere. Any worker in the vicinity of a construction machine during operation may not be noticed and have a fatal accident due to these blind spots. The safety zone feature can be used to analyze the entire construction site in large projects that have access to live satellite imagery. Additionally, detecting construction machinery from satellite imageries can be used for detecting illegal logging or unlicensed constructions from satellite surveillance systems.

REFERENCES

[1] ATANGANA NJOCK Pierre Guy, SHEN Shui-Long, ZHOU Annan and MODONI Giuseppe (2021), "Artificial neural network optimized by differential evolution for predicting diameters of jet grouted columns", *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 13, no. 6, pp. 1500-1512.

[2] CONGRO Marcello, DE ALENCAR MONTEIRO Vitor Moreira, BRANDÃO Amanda, DOS SANTOS, Brunno, ROEHL Deane and DE ANDRADE SILVA Flávio (2021), "Prediction of the residual flexural strength of fiber reinforced concrete using artificial neural networks", *Construction and Building Materials*, vol. 303, p. 124502.

[3] ADESANYA Elijah, ALADEJARE Adeyemi, ADEDIRAN Adeolu, LAWAL Abiodun and ILLIKAINEN Mirja (2021), "Predicting shrinkage of alkali-activated blast furnace-fly ash mortars using artificial neural network (ANN)", *Cement and Concrete Composites*, vol. 124, p. 104265.

[4] TANG Shuai, ROBERTS Dominic and GOLPARVAR-FARD Mani (2020), "Human-object interaction recognition for automatic construction site safety inspection", *Automation in Construction*, vol. 120, p. 103356.

[5] LIU Jian and HU Chuan-zheng (2017), "Application of Information Technology in Active Safety Control for Construction Machine", *Procedia Engineering*, vol. 174, pp. 1182-1189.

[6] PANERU Suman and JEELANI Idris (2021), "Computer vision applications in construction: Current state, opportunities & challenges", *Automation in Construction*, vol. 132, p. 103940.

[7] KRIZHEVSKY Alex, SUTSKEVER Ilya and HINTON Geoffrey Everest (2012), "ImageNet Classification with Deep Convolutional Neural Networks", *NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA.

[8] POE Edgar Allan (1836), "Maelzel's Chess-Player", *Southern Literary Journal*, pp. 318-326.

[9] LOWE David (1999), "Object recognition from local scale-invariant features", *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece.

[10] LOWE David (2004), "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, no. 60, p. 91–110.

[11] NABIZADEH-SHAHRE-BABAK Zahra, KARIMI Nader, KHADIVI Pejman, ROSHANDEL Roshanak, EMAMI Ali and SAMAVI Shadrokh (2021), "Detection of COVID-19 in X-ray images by classification of bag of visual words using neural networks", *Biomedical Signal Processing and Control*, vol. 68, p. 102750.

[12] SANCHEZ Jorge and PERRONNIN Florent (2011), "High-dimensional signature compression for large-scale image classification", *CVPR 2011*, Colorado Springs, CO, USA.

[13] ZEILER Matthew and FERGUS Rob (2014), "Visualizing and Understanding Convolutional Networks", *ECCV 2014 - 13th European Conference*, Zurich, Switzerland.

[14] HOWARD Andrew (2013), "Some Improvements on Deep Convolutional Neural Network Based Image Classification".

[15] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, SUN Jian (2015), "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916.

[16] SIMONYAN Karen and ZISSERMAN Andrew (2015), "Very Deep Convolutional Networks for Large-Scale Image Recognition", *ICLR 2015*.

[17] IOFFE Sergey and SZEGEDY Christian (2015), "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift",

Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. Volume 37, p. 448–456.

[18] XIE Saining, GIRSHICK Ross, DOLLÁR Piotr, TU Zhuowen and HE Kaiming (2017), "Aggregated Residual Transformations for Deep Neural Networks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492-1500.

[19] CHEN Yunpeng, LI Jianan, XIAO Huaxin, JIN Xiaojie, YAN Shuicheng and FENG Jiashi (2017), "Dual Path Networks", *arXiv:1707.01629*.

[20] ZOPH Barret, VASUDEVAN Vijay, SHLENS Jonathon and LE Quoc (2018), "Learning transferable architectures for scalable image recognition", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697-8710.

[21] LIU Chenxi, ZOPH Barret, NEUMANN Maxim, SHLENS Jonathon, HUA Wei, LI Li-Jia, FEI-FEI Li, YUILLE Alan, HUANG Jonathan and MURPHY Kevin (2018), "Progressive Neural Architecture Search", *Proceedings of the European conference on computer vision (ECCV)*, pp. 19-34.

[22] REAL Esteban, AGGARWAL Alok, HUANG Yanping and LE Quoc (2019), "Regularized Evolution for Image Classifier Architecture Search", *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, pp. 4780-4789.

[23] MAHAJAN, Dhruv, GIRSHICK Ross, RAMANATHAN Vignesh, HE Kaiming, PALURI Manohar, LI Yixuan, BHARAMBE Ashwin and VAN DER MAATEN Laurens (2018), "Exploring the limits of weakly supervised pretraining", *Proceedings of the European conference on computer vision (ECCV)*, pp. 181-196.

[24] TOUVRON Hugo, VEDALDI Andrea, DOUZE Matthijs and JÉGOU Hervé (2019), "Fixing the train-test resolution discrepancy", *arXiv preprint arXiv:1906.06423*.

[25] XIE Qizhe, LUONG Minh-Thang, HOVY Eduard and LE Quoc (2020), "Self-training with Noisy Student improves ImageNet classification", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687-10698.

[26] KOLESNIKOV Alexander, BEYER Lucas, ZHAI Xiaohua, PUIGCERVER Joan, YUNG Jessica, GELLY Sylvain and HOULSBY Neil (2020), "Big transfer (bit): General visual representation learning", *Computer Vision--ECCV 2020: 16th European Conference*, Glasgow, UK, August 23-28, Proceedings, Part V 16.

[27] TOUVRON Hugo, VEDALDI Andrea, DOUZE Matthijs and JÉGOU Hervé (2020), "Fixing the train-test resolution discrepancy: Fixefficientnet", *arXiv preprint arXiv:2003.08237*.

[28] DOSOVITSKIY Alexey, BEYER Lucas, KOLESNIKOV Alexander, WEISSENBORN Dirk, ZHAI Xiaohua, UNTERTHINER Thomas, DEHGHAN Mostafa, MINDERER Matthias, HEIGOLD Georg, GELLY, Sylvain, USZKOREIT Jakob and HOULSBY Neil (2020), "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*.

[29] FORET Pierre, KLEINER Ariel, MOBAHI Hossein and NEYSHABUR Behnam (2020), "Sharpness-Aware Minimization for Efficiently Improving Generalization", *arXiv:2010.01412*.

[30] PHAM Hieu, DAI Zihang, XIE Qizhe, LUONG Minh-Thang and LE Quoc (2021), "Meta Pseudo Labels", *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[31] CASTELLUCCIO Marco, POGGI Giovanni, SANSONE Carlo and VERDOLIVA Luisa (2015), "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks", *arXiv preprint arXiv:1508.00092*.

[32] CHENG Gong, XIE Xingxing, HAN Junwei, GUO Lei and XIA Gui-Song (2020), "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735-3756.

[33] GUO Yapeng, XU Yang and LI Shunlong (2020), "Dense construction vehicle detection based on orientation-aware feature fusion convolutional neural network", *Automation in Construction*, vol. 112, p. 103124.

[34] MÄYRÄ Janne, KESKI-SAARI Sarita, KIVINEN Sonja, TANHUANPÄÄ Topi, HURSKAINEN Pekka, KULLBERG Peter, POIKOLAINEN Laura, VIINIKKA Arto, TUOMINEN Sakari, KUMPULA Timo and VIHERVAARA Petteri (2021), "Tree species classification from airborne hyperspectral and LiDAR data using 3D convolutional neural networks", *Remote Sensing of Environment*, vol. 256, p. 112322.

[35] ABBURU Sunitha and GOLLA Suresh Babu (2015), "Satellite Image Classification Methods and Techniques: A Review", *International Journal of Computer Applications*, vol. 119, pp. 20-25.

[36] SHARMA Atharva, LIU Xiuwen, YANG Xiaojun and SHI Di (2017), "A patch-based convolutional neural network for remote sensing image classification", *Neural Networks*, vol. 95, pp. 19-28.

[37] NOGUEIRA Keiller, PENATTI Otávio and DOS SANTOS Jefersson (2017), "Towards better exploiting convolutional neural networks for remote sensing scene classification", *Pattern Recognition*, vol. 61, pp. 539-556.

[38] LI Yansheng, CHEN Wei, ZHANG Yongjun, TAO Chao, XIAO Rui and TAN Yihua (2020), "Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning", *Remote Sensing of Environment*, vol. 250, p. 112045.

[39] ÖZYURT Fatih (2020), "Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures", *The Journal of Supercomputing*, vol. 76, pp. 8413-8431.

[40] KHAN Shah Nawaz, KHAN Syed Irteza Ali, ABIDEEN Zain UI, KHAN Muhammad Salman and ANWAR Shahzad (2020), "Rapid Aircraft Classification in Satellite Imagery using Fully Convolutional Residual Network", *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Pakistan.

[41] GAO Huajian and LI, X (2020), "Vehicle Detection In High Resolution Image Based On Deep Learning", *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B3-2020, pp. 49-54.

[42] TAN Qulin, LING Juan, HU Jun, QIN Xiaochun and HU Jiping (2020), "Vehicle Detection in High Resolution Satellite Remote Sensing Images Based on Deep Learning", *IEEE Access*, vol. 8, pp. 153394-153402.

[43] CAO Min, JI Hong, GAO Zhi and MEI Tincan (2020), "Vehicle Detection In Remote Sensing Images Using Deep Neural Networks And Multi-Task Learning", *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-2-2020, pp. 797-804.

[44] WU Xin, LI Wei, HONG Danfeng, TIAN Jiaojiao, TAO Ran and DU Qian (2020), "Vehicle detection of multi-source remote sensing data using active fine-tuning network", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 39-53.

[45] OSCO Lucas Prado, DOS SANTOS DE ARRUDA Mauro, GONÇALVES Diogo Nunes, DIAS Alexandre, BATISTOTI Juliana, DE SOUZA Mauricio, GOMES Felipe David Georges, RAMOS Ana Paula Marques, DE CASTRO JORGE Lúcio André, LIESENBERG Veraldo, LI, Jonathan, MA Lingfei, MARCATO JUNIOR José and GONÇALVES Wesley Nunes (2021), "A CNN approach to simultaneously count plants and detect plantation-rows from UAV imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 174, pp. 1-17.

[46] BI Qi, ZHANG Han and QIN Kun (2021), "Multi-scale stacking attention pooling for remote sensing scene classification", *Neurocomputing*, vol. 436, pp. 147-161.

[47] BOULILA Wadii, SELLAMI Mokhtar, DRISS Maha, AL-SAREM Mohammed, SAFAEI Mahmood and GHALEB Fuad (2021), "RS-DCNN- A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification", *Computers and Electronics in Agriculture*, vol. 182, p. 106014.

[48] MA Ailong, WAN Yuting, ZHONG Yanfei, WANG Junjue and ZHANG Liangpei (2021), "SceneNet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 172, pp. 171-188.

[49] JAVADI Saleh, DAHL Mattias and PETTERSSON, Mats (2021), "Vehicle Detection in Aerial Images Based on 3D Depth Maps and Deep Neural Networks", *IEEE Access*, vol. 9, pp. 8381-8391.

[50] ARABI Saeed, HAGHIGHAT Arya Ketabchi and SHARMA Anuj (2020), "A deep-learning-based computer vision solution for construction vehicle detection", *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, pp. 753-767.

[51] FANG Weili, DING Lieyun, ZHONG Botao, LOVE Peter and LUO Hanbin (2018), "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach", *Advanced Engineering Informatics*, vol. 37, pp. 139-149.

[52] KIM Jinwoo (2020), "Visual Analytics for Operation-Level Construction Monitoring and Documentation: State-of-the-Art Technologies, Research Challenges, and Future Directions", *Frontiers in Built Environment*, vol. 6, p. 202.

[53] CHEN Jichi, WANG Hong, WANG Shjie, HE Enqiu, ZHANG Tao and WANG Lin (2021), "Convolutional neural network with transfer learning approach for detection of unfavorable driving state using phase coherence image", *Expert Systems with Applications*, pp. 116016.

[54] HE Kaiming, ZHANG Xiangyu, REN Shaoqing and SUN Jian (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

[55] HOWARD Andrew, ZHU Menglong, CHEN Bo, KALENICHENKO Dmitry, WANG Weijun, WEYAND Tobias, ANDREETTO Marco and ADAM Hartwig (2017), "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv*, vol. abs/1704.04861.

[56] SANDLER Mark, HOWARD Andrew, ZHU Menglong, ZHMOGINOV Andrey and CHEN Liang-Chieh (2018), "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA.

[57] HOWARD Andrew, SANDLER Mark, CHU Grace, CHEN Liang-Chieh, CHEN Bo, TAN Mingxing, WANG Weijun, ZHU Yukun, PANG Ruoming,

VASUDEVAN Vijay, LE Quoc and ADAM Hartwig (2019), “Searching for MobileNetV3”, *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, Korea.

[58] HUANG Gao, LIU Zhuang and VAN DER MAATEN Laurens (2017), "Densely connected convolutional networks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708.

[59] TAN Mingxing and LE Quoc (2019), "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *International Conference on Machine Learning*, pp. 6105-6114.

[60] VIDAL Ludovic-Alexandre and MARLE Franck (2008), "Understanding project complexity: Implications on project management", *Kybernetes*, pp. 1094-1110.