

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**IDENTIFICATION OF TEA PLANTATION AREAS
USING GOOGLE CLOUD BASED
RANDOM FOREST AND DEEP LEARNING**



M.Sc. THESIS

Berkay ÖZEN

Department of Geomatics Engineering

Geomatics Engineering Programme

JULY 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**IDENTIFICATION OF TEA PLANTATION AREAS
USING GOOGLE CLOUD BASED
RANDOM FOREST AND DEEP LEARNING**

M.Sc. THESIS

**Berkay ÖZEN
(501171668)**

Department of Geomatics Engineering

Geomatics Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Esra ERTEN

JULY 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**GOOGLE BULUT SERVİSE DAYALI
RASTGELE ORMAN VE DERİN ÖĞRENME İLE
ÇAY TARIM ALANLARININ BELİRLENMESİ**

YÜKSEK LİSANS TEZİ

**Berkay ÖZEN
(501171668)**

Geomatik Mühendisliği Anabilim Dalı

Geomatik Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Esra ERTEN

TEMMUZ 2020

Berkay ÖZEN, a M.Sc. student of ITU **Graduate School of Science Engineering and Technology** student ID 501171668, successfully defended the **thesis** entitled “**IDENTIFICATION OF TEA PLANTATION AREAS USING GOOGLE CLOUD BASED RANDOM FOREST AND DEEP LEARNING**”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Assoc. Prof. Dr. Esra ERTEN**
Istanbul Technical University

Jury Members : **Prof. Dr. Nebiye MUSAOĞLU**
Istanbul Technical University

Assoc. Prof. Dr. Koray KAYABOL
Gebze Technical University

.....

Date of Submission : 15 June 2020

Date of Defense : 13 July 2020





To my family,



FOREWORD

First of all, I would like to thank my family, who has given all kinds of support throughout my entire life, my valuable advisor Doç Dr Esra Erten, who shared his valuable knowledge during undergraduate thesis and master thesis studies and always guided in doing different and high quality works and Emi Group Inc., Turkey for sharing the label data of the tea fields used in the study.

July 2020

Berkay ÖZEN
(GIS Specialist)

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD.....	ix
TABLE OF CONTENTS.....	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xix
SUMMARY	xxi
ÖZET	xxv
1. INTRODUCTION	1
1.1 Motivation.....	1
1.2 Literature Review	2
1.3 Purpose of Thesis	8
2. STUDY AREA AND MATERIALS	9
2.1 Study Area	9
2.2 Airborne Data	9
2.3 Satellite Imagery.....	11
3. METHODOLOGY	17
3.1 Cloud Environment For Remote Sensing Imagery.....	17
3.1.1 Cloud computing.....	17
3.1.1.1 Cloud service models.....	17
3.1.1.2 Cloud deployment models	18
3.1.2 Google cloud.....	19
3.1.3 Google earth engine	20
3.1.4 Google colaboratory.....	20
3.2 Machine Learning For Tea Monitoring	22
3.2.1 Random forest classification	23
3.3 Deep Learning For Tea Monitoring.....	26
3.3.1 Artificial neural networks	27
3.3.2 Deep convolutional neural networks.....	29
3.4 Work Flow	32
4. APPLICATION AND RESULTS	35
4.1 Preprocessing.....	35
4.1.1 Sentinel-2 imagery	37
4.1.2 Training and testing data.....	38
4.2 Random Forest Classification for Detecting Tea Plantation Areas	39
4.3 Deep Learning Classification for Detecting Tea Plantation Areas	49
5. CONCLUSIONS AND RECOMMENDATIONS.....	57
REFERENCES.....	61

APPENDICES	69
CURRICULUM VITAE.....	81



ABBREVIATIONS

ANN	: Artificial Neural Network
App	: Appendix
CNN	: Convolutional Neural Network
ESS	: Error sum-of-squares
GEE	: Google Earth Engine
GIS	: Geographic Information Systems
NDVI	: Normalized Difference Vegetation Index
RS	: Remote Sensing
TF	: TensorFlow
RF	: Random Forest
DL	: Deep Learning
ML	: Machine Learning
EVI	: Enhanced Vegetation Index
EVI2	: Enhanced Vegetation Index II
SAVI	: Soil Adjust Vegetation Index
CVI	: Chlorophyll Vegetation Index
IPVI	: Infrared Percentage Vegetation Index
GNDVI	: Green Normalized Difference Vegetation Index
BNDVI	: Blue Normalized Difference Vegetation Index
GBNDVI	: Green Blue Normalized Difference Vegetation Index
RBNDVI	: Red Blue Normalized Difference Vegetation Index
GRNDVI	: Green Red Normalized Difference Vegetation Index
NDBG	: Normalized Difference Blue Green
NDBR	: Normalized Difference Blue Red
NDGR	: Normalized Difference Green Red
GPU	: Graphic Processor Unit
CPU	: Central Processor Unit
TPU	: Tensor Processor Unit
OA	: Overall Accuracy
OOB	: Out-of-Bag
GSR	: Ground Sampling Resolution
EPSG	: European Petroleum Survey Group
WGS84	: World Geodetic System 1984
GDAL	: Geospatial Data Abstraction Library
CRS	: Coordinate Reference System
S2	: Sentinel 2
API	: Application Programming Interface
SVM	: Support Vector Machines
SAM	: Spectral Angle Mapper
LAI	: Leaf Area Index
ESA	: European Space Agency

USGS	: United States Geological Survey
NASA	: National Aeronautics and Space Administration
GLCM	: Gray Level Co-occurrence Matrix
NIR	: Near Infrared
RGB	: Red-Green-Blue
NirRG	: Near Infrared-Red-Green
NDVIRG	: NDVI-Red-Green



SYMBOLS

P	: Probability
d	: Distance
μ	: Mean
σ	: Standard Deviation
θ_K	: Random Vector





LIST OF TABLES

	<u>Page</u>
Table 2.1 : Sentinel-2 MSI: Multispectral instrument, level-1c bands.	12
Table 4.1 : General statistics of the data. [1].....	38
Table 4.2 : Statistics of training and testing data.	39
Table 4.3 : RF parameters and model accuracies.....	41
Table 4.4 : Confusion matrix of RF classification.	42
Table 4.5 : Validation error matrix of RF classification.....	43
Table 4.6 : Training OOB errors of all cases.	44
Table 4.7 : Training & validation results of RF classification for all cases.	45
Table 4.8 : Hiperparameters of GEE team.	53
Table 4.9 : Hiperparameters of best fit.....	53
Table 4.10 : Test results of model.	54
Table 4.11 : Training and testing results of models.	56
Table C.1 : Layers of DL Method.....	77



LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : Study area.....	10
Figure 2.2 : Samples of orthophoto data (left), Sentinel-2 data (middle) and label data (right).....	11
Figure 2.3 : GLCM [2].	14
Figure 3.1 : Tree model of random forest technique. [3]	26
Figure 3.2 : Model of an artificial neuron [4].	28
Figure 3.3 : Convolution process a)input matrix b)filter c)output matrix.	30
Figure 3.4 : Maxpool-pooling process.....	31
Figure 3.5 : Full connected layer.	31
Figure 3.6 : Relu activation function.	32
Figure 3.7 : Standard neural network vs. neural network using dropout.	32
Figure 3.8 : Work flow of deep learning.....	33
Figure 3.9 : Work flow of random forest.....	34
Figure 4.1 : RF variable importance for DN + Indices + GLCM(5x5).	47
Figure 4.2 : RF classification for DN.	48
Figure 4.3 : RF classification for DN + Indices.	48
Figure 4.4 : RF classification for DN + GLCM(5x5).	48
Figure 4.5 : RF classification for DN + Indices + GLCM(5x5).	49
Figure 4.6 : U-NET model [5].	52
Figure 4.7 : Calculation speed of CPU, GPU and TPU.....	54
Figure 4.8 : Learning and loss graph.	54
Figure 4.9 : Classified Sentinel-2 imagery with U-NET.	55
Figure A.1 : RF Variable Importance for DN.....	71
Figure A.2 : RF Variable Importance for DN + Indices.....	71
Figure A.3 : RF Variable Importance for DN + GLCM(3x3).	72
Figure A.4 : RF Variable Importance for DN + GLCM(5x5).	72
Figure A.5 : Random Forest Variable Importance for DN + GLCM(7x7).	73
Figure A.6 : RF Variable Importance for DN + Indices + GLCM(3x3).	73
Figure A.7 : RF Variable Importance for DN + Indices + GLCM(7x7).	74
Figure B.1 : Learning and Loss Graph of RGB.	75
Figure B.2 : Learning and Loss Graph of NirRG.....	75
Figure B.3 : Learning and Loss Graph of NDVIRG.	76



IDENTIFICATION OF TEA PLANTATION AREAS USING GOOGLE CLOUD BASED RANDOM FOREST AND DEEP LEARNING

SUMMARY

With the increasing population, food has become the most important parameter that determines international relations. Agricultural areas, which are the source of livelihood of the farmer, turn into urban areas with increasing population and face the danger of decrease with the pressure of construction. In addition to population growth, climate change threatens sustainable agriculture. Turkey is located in the areas affected by climate change. Moreover, as a result of the wrong and unconscious agricultural field management, increasing population and industrialization seen in recent years, the danger is much higher. If this condition continues, the environmental problems facing Turkey, will reach much more serious. Change of agricultural land, and these changes can not be managed correctly, it is just one of the most important issues in Turkey but all over the world in recent years. In this case, considering the spatial and temporal features, satellite images become a unique tool for monitoring the change in agricultural areas. Monitoring with satellite images allows analysis of agricultural activities of all sizes (from farm scale to basin scale), taking into account differences in environmental conditions, and even variations in the field. In this study, the livelihoods of more than 1 million people in Turkey, which ranks fifth in worldwide production of black tea plantation monitoring is investigated using remote sensing technologies. The main aim of the thesis is to identify the cultivated tea plantations with a satellite image obtained at time t .

Although experience has been gained in monitoring of agricultural lands with satellite images, there is not much study on the detection of tea plantations. The main reason for this is tea plants has very similar spectral reflections with other agricultural products in the Eastern Black Sea region, where it grows. This classification problem is tried to be solved by advanced machine learning methods. However, especially deep learning methods require hardware with high computing power. In addition, the difficulty is further increased when the size of remote sensing images is added. Google has offered researchers the opportunity to solve these challenges with cloud computing. GEE is a cloud system that contains satellite data in petabyte scale and users can process this data using the JavaScript and Python programming languages. It makes possible to process satellite images without downloading and by using parallel calculation method using Google infrastructure. In addition, GEE can connect with Google's other cloud services (Drive, Cloud Storage) and the TensorFlow open source library for machine and deep learning can easily read and process data on it. In line with these developments, the thesis aims to evaluate the prediction of tea fields by using GEE and other services and the future role of GEE in this sense. As a method, both machine learning and deep learning were evaluated. For the Eastern Black Sea region, the study area, there is only access to satellite images that are free of charge

via GEE. For this reason, Sentinel-2 data in GEE with the best spatial resolution was preferred for the study. Training and test data are needed for both mentioned classification methods. For this reason, binary annotated data previously created from high resolution orthophoto images were used as the ground truth data of the study area. These data were transferred to the cloud after they were converted to the appropriate format and then used. A total of 10 thousand points were selected from these location accuracy data and labeled via GEE. Afterwards, 50% of these data were randomly allocated for training and 50% for testing and exported to the cloud. Random Forest (RF) classification algorithm was chosen as machine learning. First, the best model evaluation for RF was made and the result was determined as 500 trees and 1 split in each tree. Vegetation Indices (VI) have been taken as additional features since tea fields in the study area will be difficult to classify using only spectral features in RF. These indices are Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI, EVI2), Infrared Percentage Vegetation Index (IPVI), Chlorophyll Vegetation Index (CVI), Soil Adjusted Vegetation Index (SAVI). In addition to the spectral properties, the texture properties of the tea plants that were monitored were used to increase the performance of the RF classification method with the Gray Level Co-occurrence Matrix (GLCM). 3 different matrix has been selected for this, 3x3, 5x5, 7x7. RF algorithm was prepared in GEE environment by using the training points data set previously prepared with these parameters, and then output results were obtained in GEE environment. According to the data obtained, when using RF classification using only spectral bands (DN), overall accuracy was achieved at 74.45%. Similarly, when using DN + indices, the result did not change and 74.46% overall accuracy was obtained. In addition, when GLMC values are added to these data, it has been observed that the accuracy is increased for both DN and DN + Indices data. This increase was accomplished by all GLCM matrices. However, the highest accuracy value was achieved with Indices, DN, and GLCM created with a 5x5 matrix and has an accuracy rate of 76.43%. Also, as a result of this classification, the feature ranking was obtained and when this ranking was examined, it was observed that the Green band was the highest. Due to the prevalence of green in the region, the change in values in the reflection will affect the result of the classification.

On the other hand, U-NET artificial neural network was used for the deep learning method and Colab application was used together with GEE. Then, the same training and test data generated randomly for the RF algorithm were used. In the following processes, region of interest obtained from the satellite image and training / test data were transferred to the Drive environment in *tffrecord* format and the data structure was determined by re-reading in the colab environment. As an additional feature, the previously specified indices are also added to the data. The best parameters are determined for the most suitable model after some tests. As a result of this parameter selection, the number of epoch is 150, the number of steps per epoch is 20, the number of batch is 20, the learning rate is 0.001 and Adam Optimizer is chosen as an optimizer. The Tensor Process Unit (TPU) provided by Colab for faster reading and processing of *tffrecord* data has been identified as the working environment. The model created with these parameters was trained and the model tested as a result of the training provided a high accuracy of 94.69%. The biggest reason for this is the U-NET architecture, which can provide success even with low resolution and a small amount of training data. Especially designed for the segmentation of biomedical images, this network architecture is quite successful in classifying partially low resolution images such as

Sentinel-2. Research has revealed that the colab and GEE cloud infrastructure used in deep learning has greatly accelerated the training and prediction processes and revealed that it has great potential for operational studies. As high resolution satellite images, which are freely available, increase, it is foreseen that the method proposed in this thesis study will provide an idea and provide current cadastral information in a very short time.





GOOGLE BULUT SERVİSE DAYALI RASTGELE ORMAN VE DERİN ÖĞRENME İLE ÇAY TARIM ALANLARININ BELİRLENMESİ

ÖZET

Artan nüfus ile birlikte gıda uluslararası ilişkileri belirleyen en önemli parametre haline gelmiştir. Çiftçinin geçim kaynağı olan tarım alanları, artan nüfus ile barınma alanlarına dönüşmekte, yapılaşmanın baskısı ile azalma tehlikesi ile karşı karşıyadır. Nüfus artışına ek olarak, iklim değişikliği de sürdürülebilir tarımı tehdit etmektedir. Türkiye bulunduğu konum sebebiyle iklim değişikliğinden etkilenen bölgede yer almaktadır. Ayrıca son yıllarda görülen yanlış ve bilinçsiz tarım alan yönetimi, artan nüfus ve sanayileşme sonucunda artık tehlike çok daha büyüktür. Bu durum devam ettiği takdirde Türkiye'yi bekleyen çevresel sorunlar çok daha ciddi boyutlara ulaşacaktır. Tarım alanlarının değişimi ve bu değişimlerin doğru bir şekilde yönetilememesi son yıllarda sadece Türkiye'de değil tüm dünyadaki en önemli konulardan biridir. Bu durumda konumsal ve zamansal özellikleri düşünüldüğünde, uydu görüntüleri tarım alanlardaki değişimin izlenmesinde benzersiz bir araç olmaktadır. Uydu görüntüleri ile izleme, her ölçekteki tarımsal faaliyetin (çiftlik ölçeğinden havza ölçeğine), toprak/çevre koşullarındaki farklılıkları hatta tarla içindeki değişkenlikleri de göz önüne alarak analiz edilmesine olanak sağlamaktadır. Bu çalışmada, dünya üretiminde beşinci sırada yer aldığımız ve Türkiye'de 1 milyondan fazla kişinin geçim kaynağı olan siyah çay arazilerinin izlenmesi araştırılmıştır. Tezin temel amacı, t anında elde edilen bir uydu görüntüsü ile ekili çay arsalarının tespiti.

Uydu görüntüleri ile tarım arazilerinin izlenmesinde deneyim kazanılmasına rağmen, çay tarlalarının tespiti ile ilgili çok fazla çalışma bulunmamaktadır. Bunun başlıca nedeni, çay bitkisinin yetiştiği bölge olan Doğu Karadeniz'deki diğer tarım ürünleri ile çok benzer spektral yansıtımının olmasıdır. Bu sınıflandırma problemi ileri seviye makine öğrenme yöntemleri ile giderilmeye çalışılmaktadır. Ancak özellikle derin öğrenme yöntemleri yüksek hesaplama gücü olan bilgisayarlara ihtiyaç duymaktadır. Bunun yanında uzaktan algılama görüntülerinin boyutu eklenince zorluk daha da artmaktadır. Google, bu zorlukları bulut bilişim ile çözmeye olanaklarını araştırmacılara sunmuştur. Google Earth Engine (GEE) petabyte boyutlarında uydu verilerini içinde barındıran ve kullanıcıların JavaScript ve Python programlama dillerini kullanarak bu verileri işleyebileceği bir bulut sistemidir. Google altyapısını kullanarak paralel hesaplama yöntemiyle uydu görüntülerini çok hızlı bir şekilde ve indirmeden işlenmesini mümkün kılmaktadır. Bunun yanı sıra Google'ın diğer bulut servisleri (Drive, Cloud Storage) ile bağlantı kurabilmektedir. Aynı zamanda yine Google tarafından geliştirilen ve makine öğrenmesi ve derin öğrenme için kullanılan TensorFlow açık kaynak kütüphanesi kolaylıkla GEE üzerindeki verileri okuyup işleyebilmektedir. Bu gelişmeler doğrultusunda, tez, GEE ve diğer servisleri kullanarak çay alanlarının tahmini ve bu anlamda GEE'nin gelecekteki rolünü değerlendirmeyi hedeflemektedir. Yöntem olarak hem makine öğrenmesi hem de

derin öğrenme değerlendirilmiştir. Çalışma bölgesi olan Doğu Karadeniz için GEE üzerinden sadece ücretsiz olan uydu görüntülerine erişim bulunmaktadır. Bu sebeple GEE içinde bulunan ve en iyi mekansal çözünürlüğe sahip olan Sentinel-2 verileri çalışma için tercih edilmiştir. Bahsi geçen her iki sınıflandırma yöntemi için de eğitim ve test verilerine ihtiyaç duyulmaktadır. Bu sebeple çalışma alanına ait yer gerçeği verileri için yüksek çözünürlüklü ortofoto görüntülerinden oluşturulmuş olan ikili etiket verileri kullanılmıştır. Bu veriler uygun formata getirildikten sonra bulut ortamına aktarılmış ve sonrasında kullanılmıştır. Bu yer doğruluğu verileri içerisinde toplamda 10 bin nokta rastgele olacak şekilde GEE üzerinden seçilmiş ve etiketlenmiştir. Sonrasında bu verilerin %50'si eğitim %50's, de test için rastgele olarak ayrılmıştır. Her iki sınıflandırma için daha adil bir karşılaştırma yapılması amacıyla seçilen bu rastgele noktalar bulut ortama kaydedilmiştir. Makine öğrenmesi olarak Random Forest (RF) sınıflandırma algoritması seçilmiştir. İlk olarak RF ile sadece spectral bandlar kullanarak en iyi model bulunması amacı ile farklı parametreler ile model değerlendirilmesi yapılmıştır. Bu değerlendirme sonucu en iyi model 500 ağaç ve her ağaçta 2 düğüm noktası olacak şekilde belirlenmiştir. Çalışma alanının RF kullanarak sadece spektral özelliklerle ayırt etmesi zor olacağından ötürü Vejetasyon İndisleri (VI) ek özellik olarak alınmıştır. Sentinel-2 uydu görüntülerinden hesaplanabilen Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI, EVI2), Infrared Percentage Vegetation Index (IPVI), Chlorophyll Vegetation Index (CVI) ve Soil Adjusted Vegetation Index (SAVI) vejetasyon indisleri sınıflandırma işleminde kullanılmıştır. Spektral özelliklere ek olarak izlenen çay arsalarının doku özellikleri Gri Seviye Birlikte Oluşma Matrisi (GLCM) ile RF sınıflandırma yönteminin başarımını arttırmak amacıyla kullanılmıştır. Belirlenen bu parametrelerle daha önceden hazırlanan eğitim verisi nokta seti kullanılarak RF algoritması GEE ortamında hazırlandı ve sonra yine GEE ortamında çıktı sonuçları elde edildi. Elde edilen verilere göre RF ile sadece spektral bandlar kullanarak sınıflandırma yapıldığında %72.88 genel doğruluğa erişebilmektedir. Benzer şekilde spektral bandlar + indisler kullanıldığında da sonuç değişmemekete ve %73.71 genel doğruluk elde edilmiştir. Bunun yanı sıra, bu verilere GLMC değerleri eklendiğinde, hem RGBNir hem de RGBNir + İndis verileri için doğruluğun arttığı gözlemlenmiştir. Bu artış bütün GLCM matrisleri tarafında gerçekleştirilmiştir. Ancak en yüksek doğruluk değeri 5x5 matris ile oluşturulan GLCM, İndisler ve RGBNir ile oluşturulan veri ile gerçekleştirilmiştir ve 76.43% doğruluk oranına sahiptir. Aynı zamanda bu sınıflandırma sonucu özellik sıralaması çıkarılmıştır ve bu sıralama incelendiğinde Yeşil bandının en yüksek olduğu gözlemlenmiştir. Bölgedeki yeşilin hakimliği sebebi ile yansıtımdaki değerlerin değişimi sınıflandırma sonucunu etkileyecek niteliktedir.

Öte yandan, derin öğrenme yöntemi için yapay sinir ağı kullanılmış olup bu yöntem için GEE ile beraber Colab uygulaması kullanılmıştır. Colab, web ortamında ve yine Google altyapısını kullanarak Python programlama dili kod yazmanızı sağlayan bir Google servisidir. İlk etapta burada gerekli bağlantılar (GEE ve Drive) yapıp kullanılacak olan kütüphaneler eklenmiştir. Sonrasında RF algoritması için rastgele oluşturulan aynı eğitim ve test verisi kullanılmıştır. Sonraki süreçlerde TensorFlow kullanılacağından ve bu kütüphane tfrecord dosya formatını çok daha efektif kullanabildiğinden çalışma alanı uydu görüntüsü ve eğitim/test verileri bu formatta Drive ortamına aktarılmıştır. Aktarılan veriler tekrar colab ortamında okunmuş ve veri yapısı belirlenmiştir. Veri yapısı belirlendikten sonra ek özellik olarak daha önce belirtilen indisler burada da kullanılarak veriye işlenmiştir. Veriler

hazır hale getirildikten sonra model oluřturma iřlemi bařlamıř ve en uygun model iin en iyi parametreler belirlenmiřtir. Bu parametre seimi sonucu epoch sayısı 150, epoch bařı adım sayısı 20, para sayısı 20, ğrenme oranı 0.001 ve optimize olarak da Adam Optimizer seilmiřtir. Bunun yanı sıra tfrecord verilerinin daha hızlı okunması ve iřlenmesi aısından Colab’ın saėlamıř olduėu Central Process Unit (CPU) ve Graphic Process Unit (GPU) haricinde Tensor Process Unit (TPU) alıřma ortamı olarak belirlenmiřtir. Bu parametreler ile oluřturulan model eėitilmiřtir ve eėitim sonucu test edilen model 94.69% oranında yksek bir doėruluk saėlamıřtır. Bunun en byk sebebi dřk öznrlk ve az sayıda eėitim verisiyle bile bařarım saėlayabilen U-NET mimarisidir. zellikle biyomedikal grntlerin segmentasyonu iin tasarlanan bu aė mimarisi Sentinel-2 gibi kısmen dřk öznrlėe sahip grntlerin sınıflandırılmasında olduka bařarılıdır. Yapılan arařtırma, derin ğrenmede kullanılan colab ve GEE bulut altyapısının eėitim ve tahmin srelerini olduka hızlandırmıř ve operasyonel alıřmalar iin byk bir potansiyele sahip olduėunu ortaya ıkarmıřtır. cretsiz eriřilebilen yksek öznrlkl uydu grntleri artıka, bu tez alıřmasında nerilen yntemin, izlenen ay arazileri hakkında ok kısa srede fikir verebileceėi ve gncel kadastral bilgi saėlayacaėı n grlmřtir.



1. INTRODUCTION

1.1 Motivation

Remote sensing technologies are one of the most common uses of precise agriculture, providing valuable information to farmers and other agriculture professionals on classification of crop types, soil properties, mapping of crop yield and etc. Specifically, identification of crop type from satellite imagery increases spatial and temporal knowledge for monitoring and evaluation in agriculture. It provides modeling of agricultural ecosystem variables such as agricultural crop detection, crop yield, crop water stress, crop diseases, harvest amount as well as planning of future agricultural lands and evaluation of factors affecting the crop through remote sensing methods. For this reason, remote sensing data for governments are gaining importance for agricultural planning. Remote sensing data, especially obtained for the purpose of carrying out government-supported agricultural activities, provides a more transparent flow of information between the farmers and the government that carry out these activities. In this way, both product planning and efficiency and better evaluation of the wages spent for the activity can be realized quickly.

With the developing technology, the quality of both space-based and airborne-based remote sensing devices and the sensitivity of multispectral band sensors have increased. However, the sizes and data accumulation of the produced data have also accordingly increased. Remote sensing data should be processed in order to play a role in decision support systems. It takes quite a while to make this data with standard methods and personal computers due to its size and multiplicity. For this reason, the concepts of machine learning (ML) and cloud computing are seen as a careful development for remote sensing science, especially in recent years.

Although many studies have demonstrated the benefits of using remote sensing technology in agriculture, there are difficulties in classifying plants with similar reflective properties. In Turkey, especially in this case it is available for tea plantations.

Tea is grown in the Eastern Black Sea region of Turkey, ranks first in the country in tea production, the worldwide ranks of the 5th. Therefore, tea production is also given importance by the government. For this reason, it is important to follow the tea agricultural lands in the region. However, when it comes to monitoring operational tea plantation, the implementation of the remote sensing method is limited to airborne and unmanned aerial vehicle and their costs are very high. For this reason, it should be approached with a different method.

1.2 Literature Review

Since the analysis and planning of agricultural areas are of great importance, many studies have been carried out for this purpose. One of the main tools used in the effective management and monitoring of agriculture is agricultural thematic mapping, which provides statistics and information on agricultural products and other agricultural resources. In order to manage and decide on the products to be grown, it is necessary to have knowledge about the types of crops and the land area being planted. In addition, data on the quality and health of the crops produced are important in terms of sustainable agricultural production. One of the important technologies that provide an economical and feasible approach to land cover information and thus agricultural mapping to collect the necessary data is remote sensing [6]. Classification of land cover and land cover from satellite images increases spatial knowledge for monitoring and evaluation in agriculture. Agricultural product pattern detection by remote sensing methods also enables modeling of agricultural ecosystem variables such as biomass, crop yield, evapotranspiration and crop productivity [7]. The use of satellites for agriculture was launched in 1973 by Bauer et al. [8]. The first satellite image classification process by Bauer began with the launch of the first Landsat (Landsat 1) satellite by the USA. In this study, Bauer classified the corn and soybean fields in the midwestern region of the USA [9]. Later, the SPOT 1 satellite was launched by France in 1986 and the IRS-1A satellite was launched by India in 1988. The wide adoption of agricultural classification and the inclusion of remote sensing as a useful technology have led to the launch of newer and better satellite systems.

In recent years, high resolution satellite systems such as IKONOS, QuickBird and SPOT 5, 6, 7 have also been used in agricultural product pattern detection. High

resolution satellite images give higher accuracy results than low positional resolution satellite images for more accurate classification of agricultural products and area estimation. In the study conducted by Yang et al, parametric and nonparametric classification algorithms were compared to determine the agricultural product pattern using SPOT-5 satellite images. In this study, Support Vector Machines (SVM), Mahalanobis Distance, Nearest Neighborhood, Highest Probability and Spectral Angle Mapper (SAM) methods were used. As a result of the study, the highest accuracy for SPOT-5 was achieved with the Highest Probability Classifier and Support Vector Machines methods [10]. The classification performances of SPOT4, SPOT5, IKONOS, QuickBird XS and QuickBird PS satellites were compared in the parcel based classification study conducted by Turker et al. As a result of the study, the highest accuracy was achieved with IKONOS satellite images (85.5%) [11].

Although the resolutions of the world observation satellites increase, this type of high resolution remote sensing data is mostly produced with commercial companies and is chargeable. Therefore, end-users have difficulties to access, except for academic studies. In addition, there are remote sensing data available to all types of users that are freely accessible. The most widely known of these are the National Aeronautics and Space Administration (NASA) and the Landsat program under the management of the United States Geological Survey (USGS). However, the spatial resolution of Landsat 8, which was placed in the last orbit within the scope of the program, is 30m [12]. On the other hand, another program open to access, such as Landsat, is Sentinel. The use of Sentinel-2 optical satellite images in agricultural land classification, where images can be obtained from the European Space Agency (ESA), has become widespread in the recent years and attracted the attention of researchers. In the study carried out by Vuolo et al., 9 different agricultural products were classified using Sentinel-2 satellite images [13].

The most commonly used approaches for determining agricultural areas in the literature are pixel-based and object-based classification processes. E. O. Makinde et al, observed that the pixel-based and object-based image classification performed with RapidEye satellite images with a spatial resolution of 6.5 m, the accuracy rate of the object-based classification [14]. In addition, non-optical sensors such as SAR sensors have also been widely used in agriculture classification. For example, Küçük at al.

focused on the classification of rice fields using Co-Polar X-Band Sar Images in their work [15]. On the other hand, as an alternative to classical pixel and object-based approaches, various learning-based algorithms have been developed to obtain more accurate and reliable information from satellite images [16]. These algorithms are divided into two as supervised learning and unsupervised learning. Automatic unsupervised classification was one of the most preferred classification methods until 10 years ago. Unsupervised classification does not require a pre-built training data [17]. In addition, supervised learning methods have increased its popularity in recent years. The most commonly used are Random Forest (RF), Bagging, Boosting, Decision Tree, Neural Networks, SVM and K-Nearest Neighborhood. At the same time, these algorithms are also known as ML methods [2]. These methods are data-based methods and learn the relationships between the predictor and the corresponding responses [18]. ML methods aim to find the most suitable model for data by using the decision rules obtained from the input data by making use of sufficient data sets and parameters. In contrast, some articles in recent years have reported that a combination of multiple classifiers has been used in the remote sensing literature to produce a single classification [19]. The resulting classifier, called a community classifier, is usually more accurate than any classifier that makes up the community [20]. Two well-known methods are the strengthening of the classification trees [21] and the bagging of Breiman [22].

However, Breiman [18] proposed random forests that add an additional random layer to bagging. In addition to building each tree using a different boot instance of the data, random forests change how classification or regression trees are structured. In standard trees, each node is split using the best split between all variables. In a random forest, each node is split using the best among a subset of tokens randomly selected on that node. This somewhat unreasonable strategy performs very well and is robust against excessive overfitting compared to many other classifiers, including discriminant analysis, support vector machines and neural networks [18].

Using the RF classifier for many years, it has yielded many successful results. To give an example to these; Majid et al. [23] focused on the Forest classification capacity of Sentinel-2 in their work and as a result completed their work with the RF classifier with an accuracy of 87%. In addition, Shah et al. [24] compared the regression analysis

with the RF classifier using Hyperspectral data to understand the amount of chlorophyll in the leaf. As a result of this study, they showed the accuracy and robustness of the RF model.

Another important issue in the classification of agricultural areas is the indices used. Vegetative Indices (VI) provide ecological information from satellite and drone data through analysis of multiple or hyperspectral image bands. The reflection of light varies with chlorophyll content, plant type, sugar content, water content in tissues and other factors. Therefore, spectral reflection responses captured by satellite images may reflect the interaction and fusion of carbon, nitrogen and water cycles [25] [26]. A wide variety of plant characteristics can be derived through various indices. These indices are also used to increase the accuracy of classification algorithms. Indices increase spectral information and increase the separability of related classes. All these factors cause an increase in the quality of the Land Use Land Cover (LULC) mapping produced [27]. Indices have double functions; Besides helping to classify different soil structures, giving information about plant growth and health. Different combinations of vegetation indices increase the spectral properties of some crops while suppressing some plants [28].

Although there are many vegetation indices, such as Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Enhanced Vegetation Index II (EVI2), Infrared Percentage Vegetation Index (IPVI), Chlorophyll Vegetation Index (CVI), Soil Adjust Vegetation Index (SAVI) were used in the study. The NDVI index, originally developed by NASA, is widely used in the remote sensing classification. In addition, Gitelson et al. [29] analyzed the sensitivity of visible reflectivity to chlorophyll, and stated that the green channel is more sensitive to chlorophyll than the red channel, thus improving GNDVI and achieving a satisfactory sensitivity [29]. Based on this study, Wan et al. [30] plant indices made by replacing NDVI's red band reflection with all combinations of visible band reflections were used to estimate the Leaf Area Index (LAI) of rice. On the other hand, Crippen [31] has developed IPVI, which is the same functionality as NDVI but runs faster. The EVI is the most common alternative vegetation index for some issues related to NDVI (soil and atmospheric limitations). In a 1995 study, Liu and Huete [32] analyzed multiple soil types and atmospherically strengthened vegetative indices; they concluded that

reducing one might increase the other as a result of the interaction between the earth and the atmosphere. In 2008, Jiang et al. [33] showed that EVI2 index can reveal different vegetation dynamics, especially when red reflection is low and NDVI becomes saturated.

In addition, texture extraction from remote sensing data is also widely used for classification. Classification based image segmentation algorithms, texture similarity, brightness similarity, outline energy, curvilinear continuity etc. Haralick proposed 14 different measurements as Gray Level Co-occurrence Matrix (GLCM) for texture extraction from the image [34]. The most important of these are Similarity, Entropy, Contrast, Homogeneity, and Correlation. In the studies of Akar and Güngör [2], using these GLCM filters with the WorldView-2 data for the extraction of tea and hazelnut fields, the general accuracy of the classification was to be increased 8%. This shows that texture extraction is an important parameter for classification.

Another approach in the classification of agricultural areas is deep learning methods. Deep learning (DL) algorithms have seen a huge increase in popularity for remote sensing image analysis over the past few years [35]. Artificial neural networks are inspired by the models of the biological brain and try to reproduce some functions using neurons, which are simple but largely interconnected processing units. A typical neural network architecture contains several layers of neurons that feed each other, and the "deep" part comes from it. DL has produced impressive results in object recognition. Recently, land use has been implemented in remote sensing studies, including image classification, and always shows great potential [35]. Silva et al. [36] used a neural network to classify tea leaves and reached hit percentages higher than 89%. Doğan and Türkoğlu [37] have achieved an accuracy rate of 92.14% with the AID (Aerial Image Dataset) dataset used in the study, which has made it possible to classify satellite images with artificial neural networks. Hang et al. [38] compared the pixel-based and object-based neural networks in their study, achieving accuracy rates of over 85% for both types.

Although classification with DL methods gives very good results, it takes a long time to process the data and create a model. Especially, considering the size of the remote sensing data, this process is not easy. Currently, data collected by a single satellite center accumulates at the rate of terabytes per day [39], and remote sensing data has

been universally accepted as “big data” [40]. According to Wang et al. [41] remote sensing has created big challenges for big data, data integration and processing. At the same time, in their books [41], they explained how it should be by suggesting cloud computing for these challenges. Chen et al. [40] shared the architecture and testing of a system for storing and processing remote sensing data with a similar approach. There are currently some cloud-based services that include satellite images and provide them to users. To give an example to these; EarthExplorer by USGS and SentinelHub by ESA. However, EarthExplorer acts as a platform that only allows downloading data. SentinelHub, on the other hand, allows to process the data by writing scripts with JavaScript (JS) programming language in the cloud environment, and currently only performs simple classification and analysis operations.

On the other hand, in 2015, Google and Gorelick and others [42] announced the Google Earth Engine (GEE). In the first place, simple analysis and classification operations such as SentinelHub could be done with JS and Python programming languages. However, thanks to the connection between Google Services, Colab Jupyter took the classification process of satellite images to the next level by using DL in the cloud with its notebook environment and TensorFlow DL library integrated there. Liss et al. [43] used the RF classifier on GEE to identify the archaeological sites in Faynan, Jordan region and concluded that GEE is a useful and highly computable system. On the other hand, Hu et al. [44] used GEE to study land use and land cover change and included time series analysis. Teluguntla et al. [45] used GEE to identify all cultivated areas of China and Australia by RF. Compared to traditional computing platforms, HuE cloud computing platform had a higher speed and greater advantages in terms of processing large-scale remote sensing data, according to Hu et al. [44]. They concluded that the CVAPS algorithm, coupled with NDVI time series analysis and supported by GEE, requires less time than traditional methods supported by ordinary PCs or computing workstations to update the black maps of western China [44]. Wang et al. [46] used GEE with DL. However, they included offline (local pc) model training and online prediction.

1.3 Purpose of Thesis

ML and cloud computing are of course not a new concept for remote sensing. ML for classification of remote sensing data has been a method used for years. Storing remote sensing data in the cloud has also been a matter of years. However, with the development of cloud computing, end users have been able to access these data and perform transactions in the cloud environment. Although DL algorithms prove themselves to classify, but it is not a method preferred by end users in remote sensing because it consumes a lot of hardware resources. Thanks to cloud computing, users can classify remote sensing data using DL algorithms using the resources of the cloud environment.

Classification of agricultural lands with the help of these technologies will positively affect decision support mechanisms. In particular, an important agricultural product grown in the Eastern Black Sea region of Turkey tea, is a government-funded agricultural products. However, due to the climate of the region, because of the hazelnuts grown in the same place, they have similar reflective properties, so remote sensing and classification processes are quite difficult. This study aims to determine tea growing areas by using RF classification algorithm and artificial neural networks in cloud environment. In addition, the effects of vegetation indices and tissue extraction methods on RF classification method and classification performance of RF and artificial neural networks were investigated.

2. STUDY AREA AND MATERIALS

In this section, the study area, the data and materials used are explained in line with the presented thesis scope.

2.1 Study Area

The study area is located in the central of Blacksea surrounding the city of Trabzon, approximately at $40^{\circ}54'48.9''N, 40^{\circ}16'32.9''E$ (shown in Figure 2.1). The elevation of the study area ranges from 0m to 1153m above mean sea level.

Tea is one of the agricultural plants used as row materials in industry. In Turkey, one of the leading provinces in tea cultivation is in Trabzon. As a matter of fact, Trabzon province ranks second after Rize in terms of both tea planted areas and production quantities. In Trabzon, 14.7% (15832 hectares) of arable land (107417.5 hectares) is devoted to tea farming. Turkey tea gardens (76 631 hectares) approximately 21% (15 832 hectares) is located up in Trabzon. Trabzon in tea cultivation, first it began in the eastern district of neighboring Rize. Of 1945 started tea cultivation, then spread to other districts. Today, Hayrat is another district in Trabzon where tea farming gained importance after Of, the center of tea farming. [47]

In addition to tea plantations, various land cover types exist including hazelnut deciduous forest and coniferous forest. Hazelnuts plantations at different successional stages can be found in the study area.

The visual figure of the study area is as in Figure 2.1.

2.2 Airborne Data

This study aimed to explore the possibility of taking advantage of free-available Sentinel-2 data for the simultaneous detection of tea plantation areas with ML and DL techniques to compare its performance with very high resolution airborne data. Within the scope of the study, an airborne multispectral images with 30cm spatial

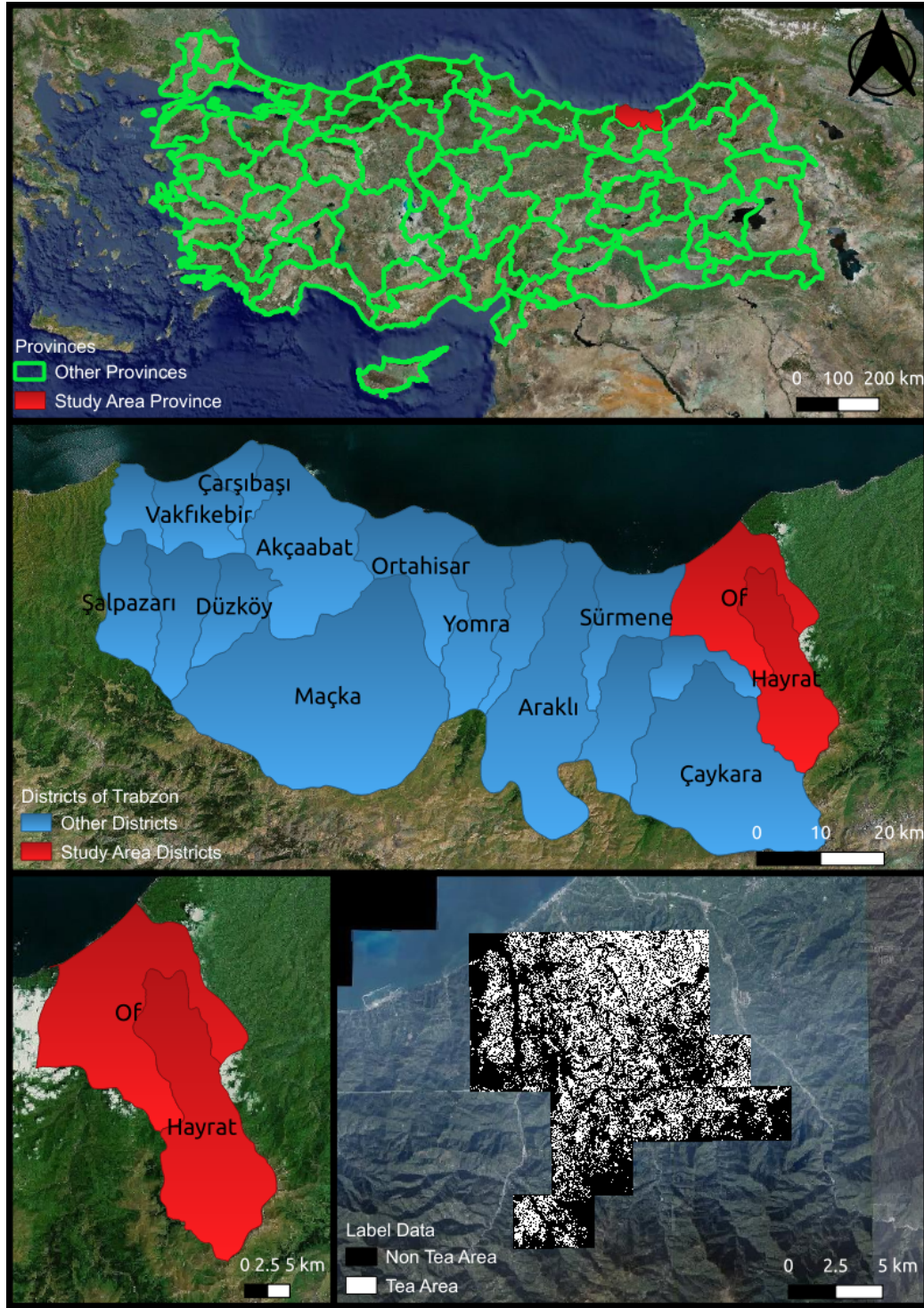


Figure 2.1 : Study area.

resolution were used as references. This reference data is orthophoto data and consists of 29 frames located in Of and Hayrat districts of Trabzon. Each frame have 4 spectral bands, red, green, blue and near infrared (R-G-B-NIR), and each band has an 8-bit radiometric resolution.

In addition to these orthophoto data, there are label data produced from the same data and consisting of 29 frames. The produced label data is in raster data format and has the same sampling GSR as orthophoto data. These label data, which are in binary data type, are "Tea Area" and "Not Tea Area". The area covered by the tea fields on the earth by calculating from the label data used corresponds to approximately 6 409 hectares. One sample of the label data in the binary data type used in the study is presented in Figure 2.2.

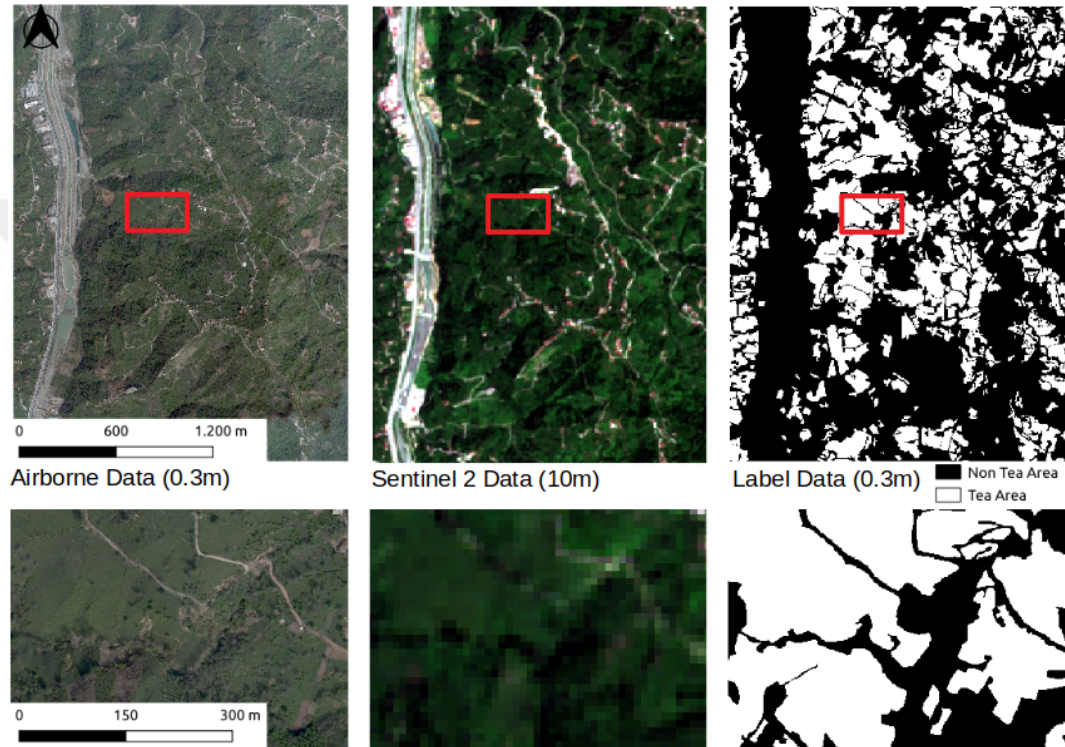


Figure 2.2 : Samples of orthophoto data (left), Sentinel-2 data (middle) and label data (right)

The orthophoto images and the labels were provided by the EMI Group Inc. (Turkish Surveying Company) and this reference orthophoto data was produced in May 2013 [48]. The area covered by 29 orthophoto images used on earth meets a total area of 17 095 hectares.

2.3 Satellite Imagery

GEE's public data archive contains over 40 years of historical imagery and scientific data and is updated and expanded daily. It contains more than two petabytes of geographical data available for analysis immediately.

The GEE data catalog contains standard raster data. The sources of these data are world observation platforms such as Landsat, Sentinel, MODIS and others which are shared with the public free of charge.

Usage is limited due to the high resolution satellite images are paid. It is seen that the usage of Sentinel-2 optical satellite images in agricultural land classification, which images can be obtained free of charge from the ESA, has become widespread in recent years. [13]

Combining high spatial resolution, wide coverage and fast revisiting time, the Sentinel-2 offers opportunities for a good distinction of land classes. The Sentinel-2 has a 13-band multispectral sensor from 0.443 to 2.190 μ m. Visible R, G, B and NIR band have 10m spatial resolution. In addition, four red bands were also available in 20m resolution [49]. Details on Sentinel-2 bands are as in Table 2.1.

Table 2.1 : Sentinel-2 MSI: Multispectral instrument, level-1c bands.

Name	Scale	Resolution	Wavelength	Description
B1	0.0001	60 meters	443.9nm (S2A) / 442.3nm (S2B)	Aerosols
B2	0.0001	10 meters	496.6nm (S2A) / 492.1nm (S2B)	Blue
B3	0.0001	10 meters	560nm (S2A) / 559nm (S2B)	Green
B4	0.0001	10 meters	664.5nm (S2A) / 665nm (S2B)	Red
B5	0.0001	20 meters	703.9nm (S2A) / 703.8nm (S2B)	Red Edge 1
B6	0.0001	20 meters	740.2nm (S2A) / 739.1nm (S2B)	Red Edge 2
B7	0.0001	20 meters	782.5nm (S2A) / 779.7nm (S2B)	Red Edge 3
B8	0.0001	10 meters	835.1nm (S2A) / 833nm (S2B)	NIR
B8A	0.0001	20 meters	864.8nm (S2A) / 864nm (S2B)	Red Edge 4
B9	0.0001	60 meters	945nm (S2A) / 943.2nm (S2B)	Water vapor
B10	0.0001	60 meters	1373.5nm (S2A) / 1376.9nm (S2B)	Cirrus
B11	0.0001	20 meters	1613.7nm (S2A) / 1610.4nm (S2B)	SWIR 1
B12	0.0001	20 meters	2202.4nm (S2A) / 2185.7nm (S2B)	SWIR 2
QA10		10 meters		Always empty
QA20		20 meters		Always empty
QA60		60 meters		Cloud mask

Besides, different indices obtained from Sentinel-2 data and used especially for the extraction of vegetation areas are also used. These indices; Normalized Difference of 4 Bands (Blue, Green, Red, NIR) combinations, NDVI, EVI, SAVI, EVI2, IPVI, CVI. The equations of these indices are given respectively.

$$NDBG = \frac{Blue - Green}{Blue + Green} \quad (2.1)$$

$$NDBR = \frac{Blue - Red}{Blue + Red} \quad (2.2)$$

$$NDGR = \frac{Green - Red}{Green + Red} \quad (2.3)$$

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (2.4)$$

$$GNDVI = \frac{NIR - Green}{NIR + Green} \quad (2.5)$$

$$BNDVI = \frac{NIR - Blue}{NIR + Blue} \quad (2.6)$$

$$GRNDVI = \frac{NIR - (Green + Red)}{NIR + (Green + Red)} \quad (2.7)$$

$$GBNDVI = \frac{NIR - (Green + Blue)}{NIR + (Green + Blue)} \quad (2.8)$$

$$RBNDVI = \frac{NIR - (Red + Blue)}{NIR + (Red + Blue)} \quad (2.9)$$

$$PNDVI = \frac{NIR - (Green + Red + Blue)}{NIR + (Green + Red + Blue)} \quad (2.10)$$

$$EVI = 2.5 * \frac{NIR - Red}{NIR + 6 * Red - 7.5 * Blue + 1} \quad (2.11)$$

$$EVI2 = 2.4 * \frac{NIR - Red}{NIR + Red + 1} \quad (2.12)$$

$$IPVI = \frac{NIR}{\frac{NIR + Red}{2}} \quad (2.13)$$

$$CVI = NIR * \frac{Red}{Green^2} \quad (2.14)$$

$$SAVI = \frac{(NIR - RED) * (1 + 0.5)}{(NIR + RED + 0.5)} \quad (2.15)$$

On the other hand, texture is the repetition of a pattern or patterns over a region. These patterns can be of different qualities such as fine, rough, smooth, random or striped in terms of quality. Texture properties are a second-order type of statistics. The gray level differences between two different pixels in different places are compared. Different textures can be found by revealing the texture properties. Tissue properties can be found in a variety of ways, such as GLCM, first order slope distributions, edge matrix matrices.

In this study, texture properties were revealed using GLCM defined by Haralick [34]. GLCM is based on the prediction of quadratic composite state probability density function, $P(i, j|d, \theta)$. G matrix represents the GLCM matrix and shows the probability of switching from gray level i to gray level j when the distance between pixels is d and the angle is θ . The square matrix is used and is examined in four directions, usually $\theta = 0^\circ, 45^\circ, 90^\circ$ and 135° , to ensure that the pattern properties do not change with rotation (Figure 2.3). Haralick proposed to average the properties obtained from the GLCM for these four directions. As a distance, generally $d = 1, 2$ values are preferred [50].

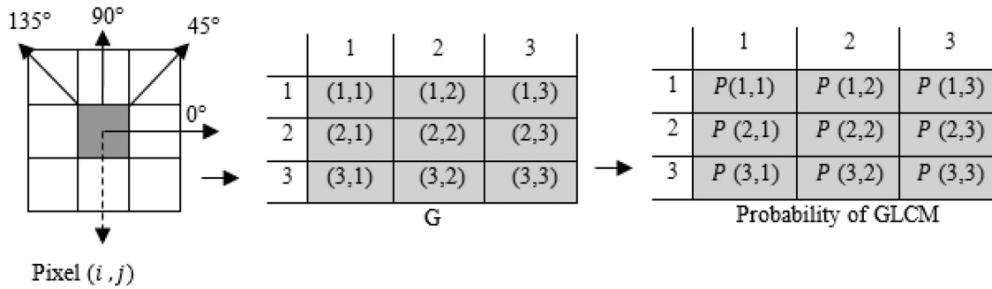


Figure 2.3 : GLCM [2].

GLCMs with $d = 1$ and $\theta = 0^\circ, 45^\circ, 90^\circ$ and 135° values are created by hovering over the square windows. Texture properties such as mean, variance, homogeneity, entropy, contrast and correlation are extracted from these matrices.

$$Similarity; \sum_i \sum_j P_{ij}^2 \quad (2.16)$$

$$Entropy; \sum_i \sum_j P_{ij} \log(P_{ij}) \quad (2.17)$$

$$Contrast; \sum_i \sum_j (i - j) P_{ij} \quad (2.18)$$

$$Homogeneity; \sum_i \sum_j \frac{P_{ij}}{1 + |i - j|} \quad (2.19)$$

$$Correlation; \sum_i \sum_j \frac{(i - \mu)(j - \mu)P_{ij}}{\sigma^2} \quad (2.20)$$

Each P_{ij} is the probability of the corresponding occurrence in G ; as shown in Figure 2.3, and μ refers to the mean and σ is the standard deviation.

The energy property obtained from GLCM is a measure of the homogeneity in the image. Energy is 1 for stabil images. The contrast feature shows the amount of local changes in the image and it is between a pixel and its neighbor. Contrast is 0 when images are stabil. The correlation feature is a measure of the gray level linear dependencies in the image and shows how a pixel and its neighbor are related. The correlation feature is 1 for fully positive correlated images and -1 for completely negative correlated images. The homogeneity property is a measure of the proximity of the elements in the GLCM to the distribution of the elements in the diagonal GLCM. The entropy feature shows the property of the spatial irregularity of the gray levels in the image [50].

In this thesis, the use of Sentinel-2 (Optical) satellite data, which is provided free of charge by the ESA and included in the GEE catalog, and related index and texture data to determine tea areas is investigated.



3. METHODOLOGY

Within the scope of the presented thesis, the performance of DL and ML techniques based on cloud computing in determining tea areas were evaluated. In this section, cloud technology, DL and ML algorithms and their basis are mentioned.

3.1 Cloud Environment For Remote Sensing Imagery

In this section, the cloud environment used in the studies conducted within the scope of the thesis presented and some basic definitions related to them are explained.

3.1.1 Cloud computing

Cloud computing has emerged as a hot topic among IT industries, academics and individual users due to its abilities of offering flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services. According to the definition of the National Institute of Standards and Technology (NIST), cloud computing is regarded as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. Cloud computing has five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. Based on the definition of NIST, cloud computing has three service models and four deployment models. [41]

3.1.1.1 Cloud service models

Cloud computing employs a service-driven model; services of clouds can be grouped into three categories:

- software as a service (SaaS),
- platform as a service (PaaS),

- infrastructure as a service (IaaS).

(1) Infrastructure as a Service: IaaS contains the essential structure for cloud IT. It ordinarily gives access to systems, (virtual or on dedicated computers), and data storage. IaaS gives the most significant level of adaptability and the executives command over IT assets. It is generally like the current IT resources with which numerous IT departments are familiar [51]. Companies providing IaaS include Amazon, Google, Rackspace, Microsoft, etc.

(2) Platform as a Service: PaaS generally aims to eliminate the need for management of basic infrastructure requirements such as hardware and operating systems. In this way, users can directly deal with their projects instead of dealing with jobs such as maintenance management [51].

(3) Software as a Service: SaaS is an entire product managed by the service provider. Users usually do not interfere with the application in any way and use it only. These are mostly applications such as web-based e-mail service. [51].

3.1.1.2 Cloud deployment models

Cloud deployment refers to a cloud that is designed to provide specific services based on demands of users. A deployment model may embrace diversified parameters such as storage size, accessibility and proprietorship, etc. There are four common cloud deployment models that differ significantly:

- Public Clouds,
- Private Clouds,
- Hybrid Clouds.
- Multi Clouds,

(1) Public Cloud: In a public cloud, a cloud service provider makes cloud-based services running on its own infrastructure available to users. Therefore, users do not have to buy their own hardware and software. These are usually paid and users only pay a rental fee for the service they use. Public cloud offers instant access to SaaS

business applications, IaaS computing and storage resources, and PaaS for application deployment and development [52].

(2) Private Cloud: Private cloud are models that only one company operates and all management is in one company. It can be located in its own facility as well as in a third party location. Since all management belongs to a single company, he is also responsible for its safety and security. The efficiency of this cloud is high because it does not have the ability to share it with other users. [52].

(3) Hybrid Cloud: Hybrid cloud, as the name suggests, is a mixture of public and private cloud. These clouds are generally preferred when a company has data or programs that cannot be easily moved. SaaS applications are provided in the public cloud for fast deployment of new applications, and PaaS for optional additional storage or computing capacity, while unmovable data and programs are kept in the company's private cloud. [52].

(4) Multi Cloud: Multicloud refers to infrastructure comprising multiple vendors' public clouds — the use of services from two or more major cloud providers (e.g., IBM Cloud and Google), or services from a major cloud provider and at least one SaaS software vendor. [52].

3.1.2 Google cloud

Google Cloud Platform is a suite of public cloud computing services offered by Google. The platform includes a range of hosted services for compute, storage and application development that run on Google hardware. Google Cloud Platform services can be accessed by software developers, cloud administrators and other enterprise IT professionals over the public internet or through a dedicated network connection [53].

Google Cloud Platform offers services for compute, storage, networking, big data, ML and the internet of things (IoT), as well as cloud management, security and developer tools.

Within the scope of the presented thesis, Google Cloud Platform was used. One of the biggest reasons for this is the PaaS it provides and the relationships that can be established between each other. GEE for people interested in earth science, data and

their analysis, and Colaboratory for people interested in python and data science are the main services used in this thesis.

3.1.3 Google earth engine

GEE is a planetary-scale platform by Google which brings together satellite imagery from all around the globe and makes it available online to a wide gamut of people enabling them to make use of this vast collection of datasets and carry out interesting research and studies. It contains various datasets and tools such as Landsat Imagery, NDVI, surface reflectance which dates back to almost 40 years.

In particular, time series approaches for monitoring our environmental development and change has become much more important for our world these days. The prospect of time series is that it is the key to a more sustainable future planning by interpreting the past. Producing time series with satellite images takes a lot of time with commercial software. Moreover, the operations performed with the commercial software are also software and hardware dependent. However, thanks to GEE, large satellite images can be processed in a short time and it is also very easy and fast to produce, interpret and visualize time series.

GEE's public data archive contains over 30 years of historical imagery and scientific data and is updated and expanded daily. It contains more than two petabytes of geographical data available for analysis immediately. The GEE data catalog contains standard raster data. The sources of these data are world observation platforms such as Landsat, Sentinel, MODIS and others which are shared with the public free of charge.

GEE can be used in two ways. One of The Earth Engine (EE) Code Editor which is web-based Integrated Development Environment (IDE) for the Earth Engine JavaScript(JS) Application Programming Interface (API). The other is The Earth Engine Python API which is Python Programming Language package.

3.1.4 Google colaboratory

ML has become the boon for the IT industry. Now, AI is not a science fiction idea as it has evolved to reality. AI helps in doing the work, which is impossible to do manually. DL is one of the main parts of AI to observe and provide accurate results of massive data. This technology has become a necessity in the business sector in recent years.

Also, Developers are expanding the capabilities of Artificial Intelligence. But the projects of Artificial Intelligence are different from traditional ones. The technology stacks and skills play a vital role in handling these projects. To implement the plans in AI and DL, The developers need to use suitable programming language. Python is used for DL by almost all programmers for their work.

Python is the programming language that is stable, flexible, and provides various tools to developers. All these properties of Python make it the first choice for DL. From development to implementation and maintenance, Python is helping developers to be productive and confident about the software they are developing. There are many benefits of using Python in DL. These benefits are increasing the popularity of Python. That is why you see Python mostly in DL Software [54].

Python is a simple language offering reliable code. DL is all about complicated algorithms and versatile workflows. So the simplicity of Python helps the developers to deal with the complex algorithms. Also, it saves the time of developers as they only require concentrating on solving the ML problems rather than focusing on the technicality of language.

Python is known as the most flexible language in DL. It provides various options for users. The flexibility factor reduces the possibility of errors. It let the programmers take the situation completely under control, and work on it comfortably. Also, developers require a well-structured and well-tested environment to develop the best coding solutions. The DL algorithms are very complex, but Python is the rescue with an extensive range of libraries and frameworks.

For all these reasons, python was chosen as the programming language to be used within the scope of the thesis.

Python is a platform-independent programming language. For this reason, python can be installed any platform. One of these platform is web. The Jupyter Notebook App is a server-client application that allows editing and running notebook documents, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...), via a web browser. Notebook documents are both human-readable documents containing the analysis description and the results (figures,

tables, etc..) as well as executable documents which can be run to perform data analysis.

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to ML, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs. It also contains a DL library for many python software languages [55].

Colab also contains so many DL library for many python. For example, TensorFlow(TF) or TF Keras Library. Keras is a minimalist Python library for DL that can run on top of Theano or TensorFlow. It was developed to make implementing DL models as fast and easy as possible for research and development.

In 2018, GEE developers announced that GEE can now be processed via DL methods by communicating with TF via Colab. After the development and success of the keras library, TF has made keras available in its own structure as tf.keras [56].

For all these reason, all calculations and codings used in this thesis were carried out in the cloud environment and Google Colab and GEE were used for this.

3.2 Machine Learning For Tea Monitoring

ML is a to programming computers to optimize performance criteria by learn and improve from experience without being explicitly programmed [57]. ML areas focus on how machines learn using certain rules. The purpose of ML is to create and learn a mathematical model and then predict new data [58]. Supervised ML divided by Kotsiantis [59] into 5 groups: logic based algorithms, perceptron based techniques, statistical learning algorithms, instance-based learning, and support vector machines.

Image classification, also known as extracting information, is the process of converting the data into meaningful land cover information by using pixel values in the image. In other words, image classification is the categorization of pixels in the image according to land cover classes by using logical decision rules in spatial area or statistical decision rules in multi-band areas [60]. Image classification algorithms in remote sensing are developed according to application variety. Pixel-based approaches

such as Minimum Distance, Maximum Likelihood Estimation (MLE) algorithms are widely used. In such approaches, spatial components are ignored while considering the spectral properties of the image. Therefore, as an alternative to this approach, object-oriented approaches that use spatial information as well as spectral information have been developed. Object-based classification approaches break the object into parts and then perform the classification according to the statistical properties of each object [61]. In recent years, different learning-based algorithms for classification have been developed in order to reach the most accurate and reliable information on satellite images quickly. Commonly used learning based classifiers include RF classification method, Bagging, Boosting, Decision Trees, Artificial Neural Networks, Support Vector Machines, K Nearest Neighborhood classifiers. These algorithms are also called ML methods. ML methods can automatically remove rules and restrictions that users cannot see by using sufficient data and parameters. These methods try to find the most suitable model for the new data with the decision rules created using the input data. [3]

The method of RF, which is a learning based classifier, provides high accuracy, and is a fast classifier, and RF algorithm was used in the study.

3.2.1 Random forest classification

In the past 20 years, the RF classifier has received a great deal of attention due to the successful classification results and processing speed achieved [62] [63] [8]. RF classifier gives reliable classifications using predictions from a range of decision trees [18]. In addition, this classifier can be easily used to select and rank features that have the highest ability to distinguish between target classes. This is an important asset because it takes time to select the most relevant variables due to the high size of the remote sensed data [64] [65].

RF is a collection of many tree-type classifiers, it can be considered as an advanced bagging method [18]. Breiman and Cutler [66] described RF as a classifier whose accuracy is quite high among existing algorithms. In addition, because it does not have a specific pattern, it states that it can work with more trees if it is desired [3].

In the RF method, wood type classifiers are used as formula will come. Here, x represents input data; θ_K represents the random vector [18] [3].

In RF method of Breiman [18], bagging is handled with random feature selection. There are two reasons why the bagging method is preferred in RF; First, increased accuracy is used as a random feature in the bagging process; The second is the calculation of generalized errors called Out-of-Bag (OOB) [18]. For random feature selection, a new training dataset is created by changing the main dataset. Then a tree is developed from the new training set using the random feature selection, and these newly developed trees are not pruned [67] [18]. Pal [62] states that the selection of pruning method and the lack of feature selection criteria affect the performance of tree-based classifiers. The lack of pruning makes RF more advantageous than other decision tree methods.

To create a tree with the RF classifier, 2 user-defined parameters are required. These parameters are the number of variables (m) and number of trees to be developed (N) in each node to determine the best split. [68].

The initial value is randomly selected by the user, and subsequent m s are increased or decreased according to generalized errors (ADF). Correlation and power decreases when m decreasing, correlation and power increases when m increases. In this way, it is found optimal m and classification sensitivity increases, error decreases. According to Breiman, [69], when the m variable value is selected, taking m value equal to the square root of the variable M (total number of variables) generally gives optimum results. Not all input variables are used in RF. A m value selection is made to simplify the computational complexity of the RF algorithm and reduce the correlation between trees. This m value represents the number of variables that are important in classification. Important variables are determined by variable significance measurements. There are 3 parameters obtained from RF. These parameters are generalized error, variable importance and proximity analysis. [70].

Generalized error data helps to understand classification accuracy. New T_k train data with displacement from the T train data is produced. The $h(x, T_k)$ classifier is created using the new training data set. Voting is made from the predictions put in the bag with the classifier. Voting takes place with this classifier only for each x , y training data. Does not include, T_k , x and y [18].

The RF algorithm calculates the significance and effects of variables [71], by examining how much the prediction error is when the locations of the data in the generalized error data are changed. Variable significance measurements, that is, determining how important the variables are, are made by changing the locations of the variables. Errors resulting from changes reveal the importance of that variable in the process. There are 4 types of variable significance measurement methods. These; 1: error increase; 2: average margin increase (average margin increase); 3: differential of margin increases and 4: Gini decrease [72]. In the study of Furlanello [72], method 2 and 4 gave more consistent results than others.

RF uses the CART (Classification and Regression Tree) algorithm to develop the largest size tree without pruning [18]. In the CART algorithm, splitting is performed by applying a certain criterion in a node. To do this, first consider the values with all the qualities, and after all matches, two divisions are obtained. Selection process is applied on these divisions [73]. Knots with homogeneous class distribution are preferred in the division processes. In the measurement of knot homogeneity; Gini Index, Entropy, Misclassification Error, Gain Ratio Criteria are used. The RF method uses the Gini index. Select a random sample (pixel) for a given training data set T and belong to this sample class C_i . Accordingly, the Gini index is expressed as follows (Equation 3.1);

$$\sum_{i \neq 0} \sum (f(C_i, T)/|T|)(f(C_i, T)) \quad (3.1)$$

In equation 3.1, $f(C_i, T)/|T|$ indicates the probability that the selected C_i sample belongs to its class. [62].

With the Gini measurements, the split position, which has the smallest Gini index, is determined [74]. As the Gini index grows, class heterogeneity increases, while the Gini Index decreases, class homogeneity increases. When a Gini index of a lower node is less than the Gini index of a parent node, that branch is successful. When the Gini index reaches zero, that is, when there is a class in each leaf node, the tree branching process ends [75]. No matter how many trees are wanted to produce, the best branch is determined for each node, and so many trees are produced [71]. Nodes are divided into branches and tree structures are formed according to the division criteria determined by

using the training data created in brief. In Figure 3.1, an example of the tree structure created according to the most suitable division positions determined in the RF classifier is shown. In figure, the x_{ij} represents the input data.

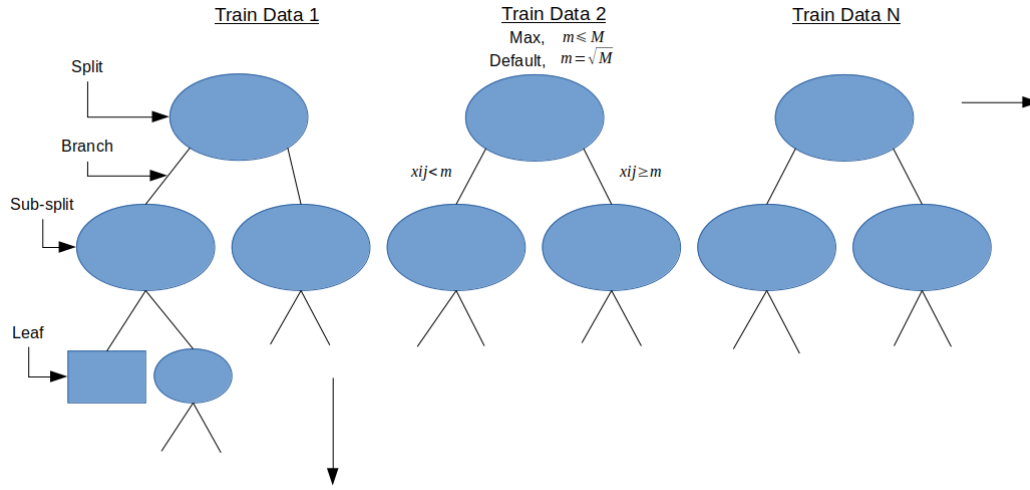


Figure 3.1 : Tree model of random forest technique. [3]

The stage after tree development is the process of classifying the input data. In this process, RF places the input data on each tree in the forest. Voting is made among the determined trees and the tree with the most votes is assigned to a class [71].

3.3 Deep Learning For Tea Monitoring

There are different approaches to classification. Especially in recent years, DL approaches have come to the fore for the classification process. In the previous classification processes, image pre-processes, feature inferences, classification and decision processes, and image processing techniques were used. After the concept of DL, many studies have been done on this subject in the past few years. Especially its success in classification and recognition brought more tendency to this issue. Satellite images cover a large area. For this reason, it is very difficult to process the data. Due to its DL structure, it requires a high amount of parameters during operations. Classification of satellite images is a difficult process. A high amount of processing is required for each classification process. As the classification item increases, these operations become more difficult. In order to make the operations on this situation faster, developing a fast algorithm and detecting objects automatically will provide many benefits.

DL has brought a new momentum to artificial intelligence studies. DL is a method developed by inspiring the structure of neurons and developed to train multilayer artificial neural networks [37]. Geoffrey Hinton and Ruslan Salakhutdinov [76] In their work, they called this artificial neural network DL, which was created by showing how multi-layered nerve structures work and how to define features that have not yet been defined [76]. It is an training process with a multi-layered neural network that reveals hidden features of DL images by some methods. The methods used are Convolution Neural Networks (CNN), Pooling, Relu, Dropout, Normalization, Softmax Layer, fully bonded Layer. Each process aims to achieve a result by carrying out a different process in the DL step. DL has a multi-layered structure and performs the task of each layer and transfers it to a lower layer. The transmitted data is transmitted to the other substrate. Thus, it creates a neural network structure transmitted from the top layer to the bottom layer and the result is obtained [37].

3.3.1 Artificial neural networks

The basis of today's neural networks was laid by the neurologist Donald Hebb. By studying the structure of nerve cells in the human brain, Hebb has formed the basis of artificial neural networks. Dendrit, soma, axon synapses are examined. Later, based on this theory, the neurons in the human brain were modeled mathematically and used to be used in computer systems in the form of artificial neural networks [37].

By looking at the communication between neurons, how learning took place was examined. Input information ($x_1..x_n$) collects information from other cells in the nucleus (Net) based on their weight ($w_1..w_n$). The axons transmit this information as it is and the result is produced by the total value activation function. Synapses evaluate this result and give the output value. Thus, a learning takes place between incoming information and output. By looking at this situation, it was tried to be shown that the working principle between brain cell and artificial neural network is similar [37] [77]. (Figure 3.2)

In the following years, artificial neural networks such as feed forward neural networks, feedback artificial neural networks, multi-layer neural networks, Boltzman machine have been developed [37]. Towards the end of the 90s, artificial intelligence studies came to a halt and a new approach could not be demonstrated. In 2006, Geoffrey

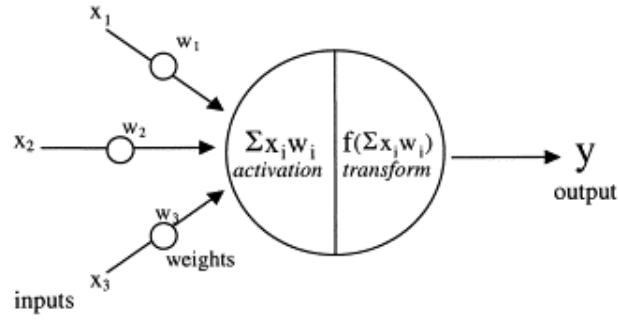


Figure 3.2 : Model of an artificial neuron [4].

Hinton and Ruslan Salakhutdin [78] introduced the concept of DL by giving a new perspective to artificial intelligence.

The learning ability of a network in artificial neural networks is related to the learning of the network by weight values in the network [37]. The weight of all data changes during the training of the network. When weights are fixed, the net is considered learned. After the training process is completed, the network is tested with test data. There are 3 different training processes in ANN, and these are supervised learning, unsupervised learning, and reinforced learning.

Supervised Learning: In the supervised learning method, the training data includes “label” information. In other words, in the development of a model for the solution, the data whose results are known are used. Thus, based on the model created, it is aimed to estimate the results of data that do not have label information in the data set. While the network is being trained, it is ensured that the error occurred by looking at the result produced by the data entered during the output. Thus, weights in the network are checked and rearranged. The same data is entered into the network many times to reduce the occurrence of errors and to learn the network healthily [78].

Unsupervised Learning: There is no label information in unsupervised learning method. It is aimed to reveal hidden relationships or groups based on the components in the data set. The network receives and categorizes this data. Weights are arranged according to the incoming data. And, according to the input data, it keeps the data with similar features in a region of close valence. Input data is grouped and categorized. The best example of this network is the Kohonen network [78] [79].

Reinforced learning: This type of learning strategy helps you learn. The input data are not said to be the result outputs on the network. However, it can be said that the result of the network during the training is true or false. [37] [79].

3.3.2 Deep convolutional neural networks

CNNs are the most popular artificial intelligence approach among today's artificial neural networks. Image classification produces the most efficient results for object detection. CNNs were introduced by Fukushima [80]. Convolution takes advantage of the spatial structure of the picture, preserving the spatial relationships of the pixels in the image. [80]

The convolution process in an image with an $N \times N$ matrix occurs by shifting a $M \times M$ -shaped matrix on the image. This matrix, which is hovered over the image, is also called a filter. With this matrix, prominent points in the image come to the fore and the attributes of the image are determined [37].

Convolution neural network works similarly to the human vision system. When looking at an image, the corner, edge and curves are primarily seen by the brain [81] [82]. Then the shapes and structures in the image become evident. Convolution neural network, which has a layered structure, has a multi-layered architecture that carries out the process of revealing the features of the image such as edge, fold, shape. Each layer determines its own attribute properties and leaves it to the next layer. Thus, attributes are taught and taught to the network. The last layer in the network often allows data entered as classifier to be included in a class [83] [84] [82].

Unlike standard learning strategies, deep convolution neural networks (DCNN) are one of the best performance and accuracy architectures for object classification and detection [85]. Attributes extracted from the image with DL work much better than traditional feature extractions [86]. In the DCNN, the model's architecture, learning task, and training procedure are similar to the working principle of the human brain [87]. The layers in the CNN are generally as follows

Convolution: This layer is the main building block of CNN. Responsible for perceiving the features of the picture. This layer applies some filters to the image to remove low and high level features in the image. For example, this filter may be a filter to detect edges. These filters are usually multidimensional and contain pixel values. First, the

filter is positioned in the upper left corner of the image. Here, the indices between the two matrices (image and filter) are multiplied by each other and all the results are summed, then the result is stored in the output matrix. Then move this filter to the right by 1 pixel (also known as “digit”) and repeat the process. After the 1st line is over, 2 lines are passed and the transactions are repeated. The output matrix is created after all operations are finished. An example of the convolution process is shown in Figure 3.3. Different filters can be applied to the convolution layer and provide different input to the other layer. Convolution processing is an essential function of image processing. Convolution process formulas are as follows. [37]

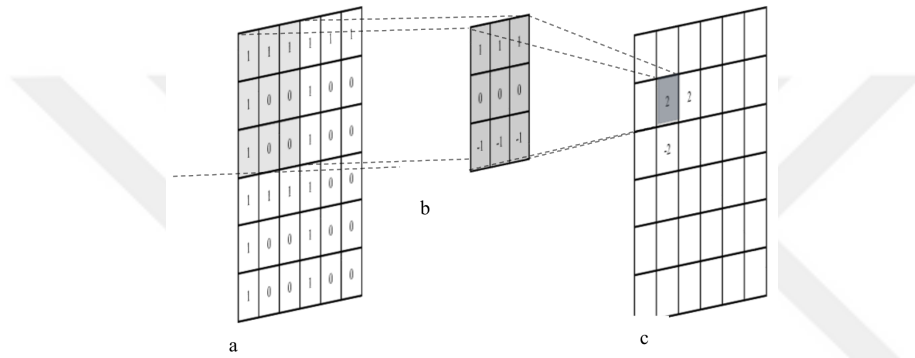


Figure 3.3 : Convolution process a)input matrix b)filter c)output matrix.

$$y_{mn} = f\left(\sum_{j=0}^{j-1} \sum_{i=0}^{i-1} X_{m+i} w_{ij} + b\right) \quad (3.2)$$

X is the matrix of two-dimensional data of the input image. W filter, b bias, and y output represent activation function [37] [88].

Pooling: This layer is a layer that is frequently added between consecutive convolutional layers in CNN. The task of this layer is to reduce the shift size of the impression and the parameters and number of calculations within the network. In this way, incompatibility in the network is checked. There are many Pooling operations, but the most popular is max pooling. There are also average pooling, and L2-norm pooling algorithms working on the same principle. The amount of data in the layer is also reduced [89]. In the example in Figure 3.4, a 4x4 matrix is given. A 2x2 maxpool has been applied to this matrix. It is assigned to the new matrix created by taking the largest value in the 2x2 matrices. The created new matrix is half the size of the previous matrix.

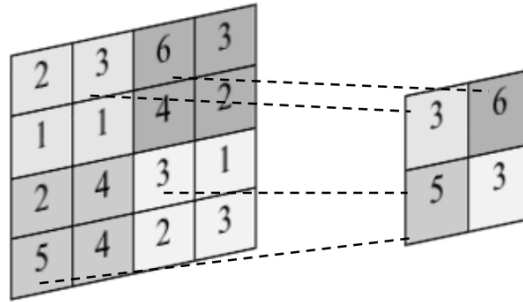


Figure 3.4 : Maxpool-pooling process.

Full-Connected: neurons from previous layers can be displayed here as an array. These neurons are located exactly (tightly) depending on the neurons in the next layer. Each neuron binds to all neurons that follow it [89]. This situation can be seen more clearly in Figure 3.5. [37]

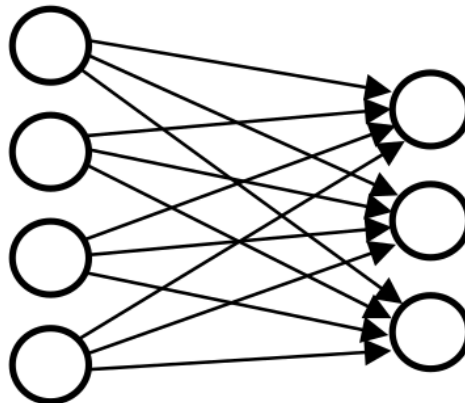


Figure 3.5 : Full connected layer.

Relu (Rectified Linear Unit):After all Convolutional layers, usually the Non-Linearity layer comes. Because all layers can be a linear function, the Neural Network acts as a single perception, so the result can be calculated as a linear combination of outputs. This layer is called the activation layer because one of the activation functions is used. In the past, nonlinear functions such as sigmoid and tahn have been used, but this function is now being used as the Rectifier (ReLu) function gives the best result on the speed of Neural Network training. Simoid, hyperbolic tangent, sinus, step, threshold value functions are used. The activation function $f(x) = \max(0, x)$ in the relu layer used in deep convolution networks [90] [91] is given in Figure 3.6. [37]

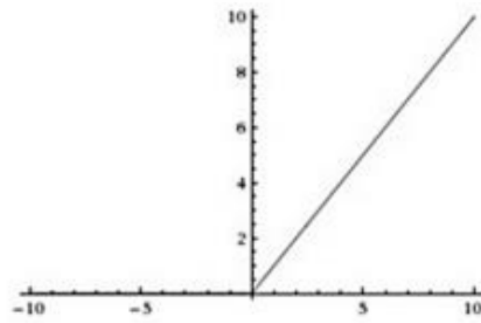


Figure 3.6 : Relu activation function.

Dropout: Used to forget some neurons to prevent overfitting during training in the artificial neural network. With Dropout, some connections within the network are removed. Thus, the memorization of the network is tried to be prevented. It also improves network performance [92]. Figure 3.7 shows the standard neural network view and the neural network view using the dropout. [37]

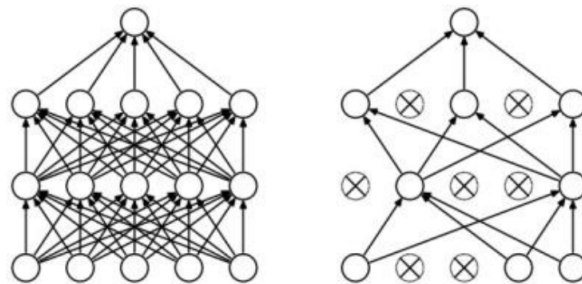


Figure 3.7 : Standard neural network vs. neural network using dropout.

3.4 Work Flow

Flow charts of the works to be done within the scope of the thesis are like Figure 3.8 for ANN and Figure 3.9 for RF.

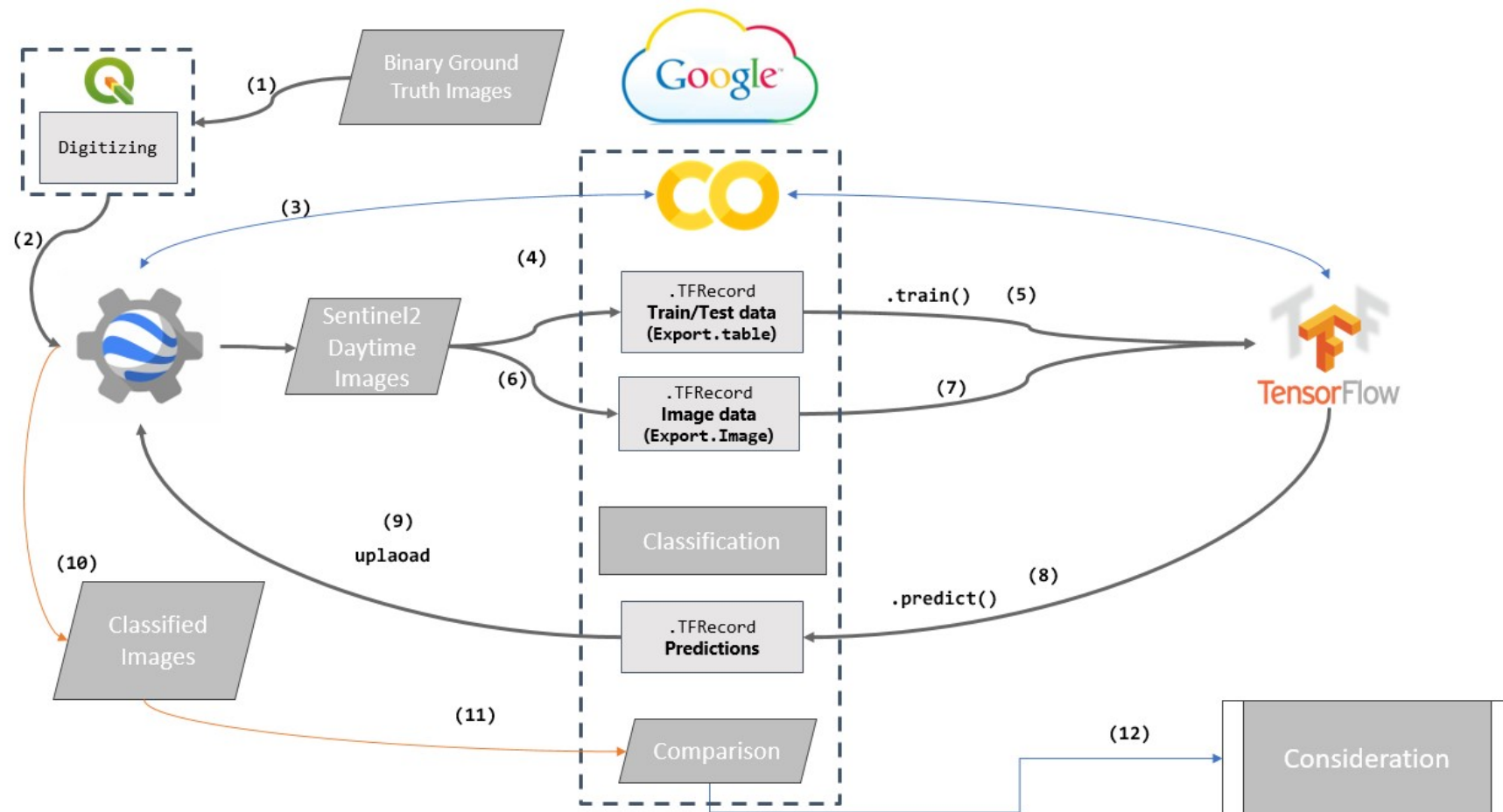


Figure 3.8 : Work flow of deep learning.

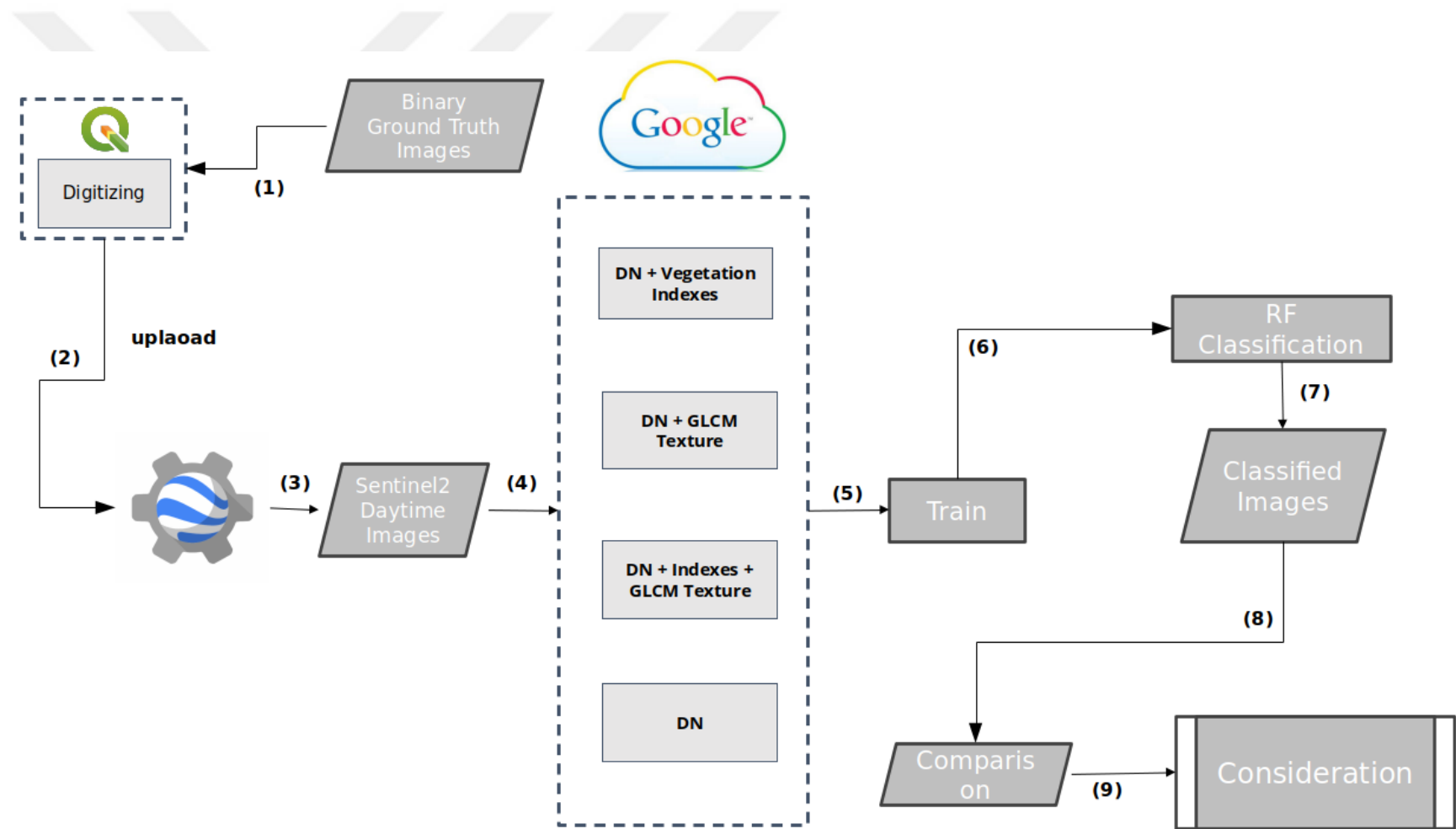


Figure 3.9 : Work flow of random forest.

4. APPLICATION AND RESULTS

In this section, tea plantation areas were identified with Google Cloud based RF and DL classification algorithms. Specifically, preprocessing details conducted in GEE and Colab is introduced in Section 4.1 and RF and DL classification results with different feature combinations are represented in Section 4.2 and 4.3 respectively.

4.1 Preprocessing

In order to create a reference map that locates the tea plantations in the region of interest, label images from airborne orthophoto which is actually mask data to identify tea fields, must be transferred to the cloud environment, see in Figure 2.2. However, since a standard CRS (Coordinate Reference System) was not defined in these data, the conversion process was performed first. As a result of the conversion process, all data are defined with EPSG 4326 code WGS84 datum. Later, 29 label images were combined to create one binary image showing the identified tea fields mask. Afterwards, these binary images were converted to vector data format (*.shp) using GDAL (Geospatial Data Abstraction Library) in order to use it in the Cloud environment and reduce the file size. Thus, a vector data was obtained indicating tea fields and non-tea fields. This vector data was uploaded directly to the Assets section of GEE.

In addition, there is a resolution difference between the label data and the Sentinel-2 data. As mentioned in Section 2, the label data has a 0.3m spatial resolution. However, Sentinel-2 data has a 10m spatial resolution. The difference between these needs to be eliminated. For this, gradual operations were carried out in the cloud environment. First of all, the label data in GEE environment in vector format has been converted in raster format. Then, with the "clipToBoundsAndScale" method in GEE, this raster data resolution was re-created to be 10m and recorded in the GEE Assets section. With this recorded data, a new vector data was created again and in the rest of the study, these data were used.

In order to work more easily on the cloud and evaluate the data separately, the data in vector format was also divided into 3 and uploaded to the GEE Assets section. These;

- Tea area vector data,
- Non-tea area vector data,
- Region of Interest (ROI) which is the merged area of label frames

After the data transfer to GEE is completed, the Colab environment must be prepared. For this, we had to define the libraries and modules we will use in colab. These;

- **Authentication Module:** One goal of Colab is to develop software in the cloud and share it easily with others. However, for both personal security and code security, running or modifying shared codes directly depends on an authorization, and this module checks whether the user has these privileges.
- **Earth Engine API:** It must be added to communicate with GEE via Colab. Also, authorization must be defined in order to be able to access your own data via GEE.
- **TensorFlow:** The library required for DL algorithms.
- **Folium:** Python library for map visualization on the web just like the Leaflet library
- **Drive Module:** The output data that will occur during the process should be saved somewhere in the cloud. There are 2 possible options for geographic data and 3 possible options for other data types.

For geographical data;

- GEE Assets
- Google Cloud Storage
- Google Drive

For other data;

- Google Cloud Storage
- Google Drive

After adding the required module and library, general variables that will be used throughout the project are defined. The most important of these are labels and bands to be used. Binary system was used as a label and the number of classes was determined as 2. Also, the output location for this study was determined as personal drive. Therefore, the drive module has been added. Also, other files in this drive can be directly accessed via colab.

4.1.1 Sentinel-2 imagery

The orthophoto images where the label data were created belong to March 2013. However, Sentinel-2 data is available to users as of June 2015, and data from 2015-06-23 are available in GEE. For this reason, the closest date range (2015-06-23 - 2015-09-01) to the date on which orthophoto images were created was selected and the frames covering the study region were determined by filtering from the Sentinel-2 collection. The study area is located in a single frame of Sentinel-2 and hence a collection of frames was obtained as a result of filtering. In addition to this, Turkey's northern regions, has a four seasons climate is more cloudiness and rainfall. For this reason, the study area is cloudy in most of the frames in the frame collection obtained. For this, using the QA60 (Cloud Mask) band of Sentinel-2, the frame collection was filtered with an auxiliary function, and the collection was reconstructed with frames with a cloudiness ratio below 10%.

After the collection was created, the first frame was selected by placing the frames in the date order. Thus, the frame in which the study will be conducted has been determined. Afterwards, this frame was clipped using the region of interest vector data added to the GEE Assets section, and only the relevant bands (Red, Green, Blue, NIR) were selected and the main data to be used later in the study was created. Here it should be noted that only these 4 preprocessed 10m resolution Sentinel-2 bands out of the original 13 bands were used in the thesis in order to make a fair comparison with airborne data.

Another useful feature of GEE is the user-defined band feature that can be added to the frame. For example, when it is desired to create an index, instead of creating an extra frame, the created index can be processed as a band to the relevant frame like NDVI as a band. Thus, all desired data can be accessed with a single frame. The indices specified

in Chapter 2 are important for vegetation extraction and will be used frequently for the further processes of the study. For this reason, these indices are processed into the main data to be used with the help of an additional function (get usage of addBands link). The general statistics of these indices processed and the existing bands in the region of interest are as in Table 4.1.

Table 4.1 : General statistics of the data. [1]

Band Name	Total Pixel in ROI	Mean	Max	Min	StdDev
B2 (Blue)	1708312	0.081	0.973	0.061	0.014
B3 (Green)	1708312	0.081	0.990	0.042	0.017
B4 (Red)	1708312	0.049	1.138	0.000	0.025
B8 (NIR)	1708312	0.377	1.134	0.013	0.069
BNDVI	1708312	0.628	0.786	-0.765	0.147
CVI	1708312	2.691	9.733	0.003	0.446
EVI	1708312	0.769	22.087	-0.272	0.180
EVI2	1708312	0.548	0.880	-0.143	0.125
GBNDVI	1708312	0.385	0.596	-0.864	0.160
GNDVI	1708312	0.631	0.782	-0.726	0.145
GRNDVI	1708312	0.477	0.683	-0.796	0.181
IPVI	1708312	1.757	1.998	0.487	0.160
NDBG	1708312	0.005	0.306	-0.242	0.036
NDBR	1708312	0.275	0.997	-0.384	0.102
NDGR	1708312	0.270	0.997	-0.458	0.092
NDVI	1708312	0.757	0.998	-0.512	0.160
PNDVI	1708312	0.275	0.521	-0.884	0.177
RBNDVI	1708312	0.474	0.696	-0.819	0.181
SAVI	1708312	0.526	0.777	-0.138	0.115

4.1.2 Training and testing data

One of the aims of the presented study is to compare RF classification with CNN classification. For this reason, the training and test data required for classification must be exactly the same pixels. For this purpose, train and test pixels should be determined first. To do this, instead of exporting the pixel data directly, GEE's randomPoints method creates points that determine these pixels and these point data are exported. In this process, a data set is generated by creating random points within the specified areas. The fields used, the tea fields previously transferred to the Assets section of GEE and the non-tea fields are vector data. Using these data, 5000 point data were generated randomly for each area. Binary label data (0 or 1) has been added to the produced data under the label 'isTea' for the relevant field. The randomColumn method of GEE

was used to further increase randomness. Thanks to this method, an extra column is created on the relevant data and deterministic pseudorandom numbers are added. The numbers are double-precision floating point numbers in the range 0.0 (inclusive) to 1.0 (exclusive) [42].

2 different point data created for tea fields and non-tea fields are primarily divided into 3 within itself. The first group in which the random column is less than 0.5, the second group between 0.5 and 0.8, and the third group greater than 0.8. Each group was combined into each other and mixed and almost evenly distributed 3 different groups were obtained from tea fields and non-tea fields. These groups allow to obtain 3 different training and test data by combinations. These data are 70% training-30% test data, 50% training-50% test version and 30% training-70% test data. As a result of the process, data as in Table 4.2 were obtained.

Table 4.2 : Statistics of training and testing data.

Point Data	Number of Points		
	(I)	(II)	(III)
trainingTeaPoints	3474	2498	1526
testingTeaPoints	1526	2502	3474
trainingNonTeaPoints	3532	2475	1468
testingNonTeaPoints	1468	2525	3532
Total Training Points	7006	4973	2994
Total Testing Points	2994	5027	7006
Total Points	10000	10000	10000

After completing the creation of point data, these data were then exported for use in both GEE and Colab. The export was done in 2 different ways. In order to access the data directly through the GEE code editor, this data was first transferred to the Assets section of GEE. In order to be used in Colab, it was transferred to the Drive environment in GeoJSON format.

Thus, the preliminary preparation of the general data has been completed.

4.2 Random Forest Classification for Detecting Tea Plantation Areas

For the RF classification, training and testing data sets must be created first. For this, examples were made from the region of interest determined using the data of the training and test points that were previously created. This sampling was first made

over Sentinel-2's own bands (Blue, Green, Red, NIR) and datasets was obtained. These data sets are 3 different training and test data prepared before.

Before deciding on the model for the RF algorithm, it is necessary to determine the 2 parameters that are important for the algorithm and which training and test data to proceed with. The parameters are the number of trees required for growth and the number of variables required to divide each node. There are many approaches to determine these variables. In this study, the best model was determined by determining the number of trees in different numbers and variables per division and evaluating their test results. The numbers and test results and overall accuracies (OA) determined are as in Table 4.3.

According to the results in Table 4.3, the best-fitting model is 500 trees and 1 split per node. In addition, the most optimal training and test data was determined as the 50% train - 50% test data set. The purpose of creating this model was to determine the best model using only the digital number (DN) values of Sentinel-2 which are its own bands, R-G-B-NIR. The same model needs to be retrained with index and texture data in order to get better results.

It is known that the results will change as the size of the matrix used to create the texture data changes. For this reason, 3 different matrixes are used for GLCM texture (3x3, 5x5, 7x7). Accordingly, 8 different cases occurred in total. These are; DN, DN + Indices, DN + GLCM (3x3), DN + GLCM (5x5), DN + GLCM (7x7), DN + Indices + GLCM (3x3), DN + Indices + GLCM (5x5), DN + Indices + GLCM (7x7). The training data were prepared again using the determined model, using the indices and texture data and classified with RF algorithm. Examples of 6 cases of confusion matrix and error matrix obtained as a result of classification are given in Table 4.4 and Table 4.5.

Table 4.3 : RF parameters and model accuracies.

N	m	30% Train - 70% Test			50% Train - 50% Test			70% Train - 30% Test		
		OOB Error	OA(%)	Test OA(%)	OOB Error	OA(%)	Test OA(%)	OOB Error	OA(%)	Test OA(%)
1000	1	0.257515	0.976	0.742	0.254977	0.972	0.745	0.254924	0.964	0.754
1000	2	0.263527	0.999	0.740	0.259803	0.998	0.740	0.257636	0.997	0.746
1000	3	0.268203	0.999	0.740	0.257792	0.998	0.739	0.259349	0.999	0.741
1000	4	0.268203	0.999	0.740	0.261814	0.998	0.734	0.258778	0.998	0.742
500	1	0.258517	0.974	0.745	0.257390	0.972	0.745	0.257922	0.964	0.754
500	2	0.261189	0.997	0.742	0.257189	0.995	0.742	0.256923	0.996	0.743
500	3	0.263861	0.998	0.741	0.258194	0.998	0.738	0.259920	0.997	0.741
500	4	0.266199	0.998	0.740	0.260808	0.998	0.735	0.258635	0.996	0.740
250	1	0.261857	0.973	0.741	0.257189	0.970	0.746	0.256494	0.962	0.754
250	2	0.262191	0.995	0.741	0.259803	0.993	0.739	0.259635	0.993	0.741
250	3	0.266199	0.997	0.740	0.263020	0.996	0.737	0.260206	0.995	0.740
250	4	0.267535	0.997	0.740	0.264227	0.996	0.735	0.262632	0.994	0.738
100	1	0.260855	0.971	0.743	0.258395	0.968	0.746	0.260634	0.962	0.756
100	2	0.262191	0.992	0.734	0.264026	0.990	0.738	0.261633	0.989	0.752
100	3	0.264529	0.993	0.740	0.259401	0.991	0.736	0.261347	0.991	0.741
100	4	0.272879	0.993	0.740	0.265433	0.991	0.735	0.272338	0.990	0.742
50	1	0.265197	0.965	0.739	0.261211	0.966	0.745	0.268627	0.959	0.753
50	2	0.267201	0.985	0.734	0.263624	0.986	0.734	0.267628	0.986	0.747
50	3	0.273881	0.985	0.735	0.263221	0.985	0.731	0.271767	0.986	0.739
50	4	0.277889	0.985	0.736	0.266841	0.985	0.732	0.274051	0.987	0.742
10	1	0.302473	0.945	0.716	0.288791	0.938	0.726	0.294799	0.934	0.725
10	2	0.301136	0.952	0.717	0.295633	0.953	0.726	0.289654	0.949	0.713
10	3	0.311832	0.951	0.712	0.303482	0.950	0.712	0.301229	0.948	0.718
10	4	0.304479	0.953	0.720	0.304689	0.951	0.720	0.292655	0.949	0.715

Table 4.4 : Confusion matrix of RF classification.

	Tea	NonTea	Total Row	PA(%)	UA(%)	OA	κ	OOB Error
(a) DN						(0.972451)	(0.944898)	(0.257792)
Tea	(2388)	(87)	(2475)	(0.964848)	(0.979491)			
NonTea	(50)	(2448)	(2498)	(0.979984)	(0.965680)			
Total	(2438)	(2535)	(4973)					
(b) DN + Indices						(0.996582)	(0.993163)	(0.266439)
Tea	(2469)	(6)	(2475)	(0.997576)	(0.995565)			
NonTea	(11)	(2487)	(2498)	(0.995596)	(0.997593)			
Total	(2480)	(2493)	(4973)					
(c1) DN + GLCM(3x3)						(0.956163)	(0.912338)	(0.261613)
Tea	(2406)	(69)	(2475)	(0.972121)	(0.941683)			
NonTea	(149)	(2349)	(2498)	(0.940352)	(0.971464)			
Total	(2555)	(2418)	(4973)					
(c2) DN + GLCM(5x5)						(0.995777)	(0.991554)	(0.248743)
Tea	(2472)	(3)	(2475)	(0.998788)	(0.992771)			
NonTea	(18)	(2480)	(2498)	(0.992794)	(0.998792)			
Total	(2490)	(2483)	(4973)					
(c3) DN + GLCM(7x7)						(0.998391)	(0.996783)	(0.248542)
Tea	(2472)	(3)	(2475)	(0.998788)	(0.997981)			
NonTea	(5)	(2493)	(2498)	(0.997998)	(0.998798)			
Total	(2477)	(2496)	(4973)					
(d1) DN + Indices + GLCM(7x7)						(0.998190)	(0.996380)	(0.239493)
Tea	(2472)	(3)	(2475)	(0.998788)	(0.997579)			
NonTea	(6)	(2492)	(2498)	(0.997598)	(0.998798)			
Total	(2478)	(2495)	(4973)					

Table 4.5 : Validation error matrix of RF classification.

		Tea	NonTea	Total Row	PA(%)	UA(%)	OA	κ
(a)	DN						(0.745176)	(0.490426)
	Tea	(1839)	(686)	(2525)	(0.735442)	(0.755546)		
	NonTea	(595)	(1907)	(2502)	(0.76219)	(0.735442)		
	Total	(2434)	(2593)	(5027)				
(b)	DN + Indices						(0.741794)	(0.483664)
	Tea	(1830)	(695)	(2525)	(0.724752)	(0.752158)		
	NonTea	(603)	(1899)	(2502)	(0.758993)	(0.732074)		
	Total	(2433)	(2594)	(5027)				
(c1)	DN + GLCM (3x3)						(0.749353)	(0.498715)
	Tea	(1885)	(640)	(2525)	(0.746535)	(0.752495)		
	NonTea	(620)	(1882)	(2502)	(0.752198)	(0.746233)		
	Total	(2505)	(2522)	(5027)				
(c2)	DN + GLCM (5x5)						(0.75373)	(0.507466)
	Tea	(1897)	(628)	(2525)	(0.751287)	(0.756681)		
	NonTea	(610)	(1892)	(2502)	(0.756195)	(0.750794)		
	Total	(2507)	(2520)	(5027)				
(c3)	DN + GLCM (7x7)						(0.758106)	(0.516252)
	Tea	(1889)	(636)	(2525)	(0.748119)	(0.765087)		
	NonTea	(580)	(1922)	(2502)	(0.768185)	(0.751368)		
	Total	(2469)	(2558)	(5027)				
(d)	DN + Indices + GLCM (7x7)						(0.763278)	(0.526617)
	Tea	(1889)	(636)	(2525)	(0.748119)	(0.77323)		
	NonTea	(554)	(1948)	(2502)	(0.778577)	(0.75387)		
	Total	(2443)	(2584)	(5027)				

Each case was repeated 50 times in order to evaluate the classification performance of the RF classification algorithm and to calculate its accuracy. The values obtained as a result of this process are as in Table 4.6 and Table 4.7. When the values are examined, the training accuracy for each case is above 95% and a similar situation is observed in κ values. When the test data are examined, the accuracy is about 75%. However, the κ values of the test data are around 50% and remain low. The reason for the big difference between Kappa and general accuracy is thought to be due to the classification with 2 classes and the almost homogeneous distribution of the classes. Having the general formula $\kappa = (OA - e)/(1 - e)$ where e is the overall probability of random agreement, κ , by its nature, indicates the probability of a random sample to which class it belongs. In this case, the results were considered reasonable due to the homogeneity and 2 classes. When all cases were examined, the case with the highest accuracy was observed as a classification when all data were used at the same time. Among them, the case using 5x5 matrix stands out. Immediately after, a case using 7x7 matrix was seen. The biggest reason for this is that it is a 2-class structure and the classes have a homogeneous structure within themselves.

Table 4.6 : Training OOB errors of all cases.

Training	
Case	OOB Error (%)
Case I – DN	25.639 ± 0.15
Case II – DN +Indices	26.439 ± 0.15
Case III – DN + GLCM(3x3)	25.977 ± 0.26
Case IV – DN + GLCM(5x5)	24.915 ± 0.22
Case V – DN + GLCM(7x7)	25.092 ± 0.2
Case VI – DN + Indices + GLCM(3x3)	24.804 ± 0.25
Case VII – DN + Indices + GLCM(5x5)	24.00 ± 0.21
Case VIII – DN + Indices + GLCM(7x7)	23.919 ± 0.2

Table 4.7 : Training & validation results of RF classification for all cases.

Cases	Training					
	OA (%)	κ (%)	User Accuracy		Producer Accuracy	
			Tea (%)	NonTea (%)	Tea (%)	NonTea (%)
Case I – DN	96.963 $\pm$ 0.09	93.926 $\pm$ 0.17	97.830 $\pm$ 0.11	96.136 $\pm$ 0.14	96.028 $\pm$ 0.15	97.889 $\pm$ 0.11
Case II – DN +Indices	99.718 $\pm$ 0.05	99.437 $\pm$ 0.11	99.682 $\pm$ 0.09	99.755 $\pm$ 0.08	99.753 $\pm$ 0.09	99.685 $\pm$ 0.09
Case III – DN + GLCM(3x3)	95.669 $\pm$ 0.21	91.34 $\pm$ 0.43	93.915 $\pm$ 0.34	97.552 $\pm$ 0.22	97.626 $\pm$ 0.22	93.731 $\pm$ 0.37
Case IV – DN + GLCM(5x5)	99.545 $\pm$ 0.07	99.089 $\pm$ 0.14	99.158 $\pm$ 0.12	99.934 $\pm$ 0.05	99.934 $\pm$ 0.05	99.159 $\pm$ 0.12
Case V – DN + GLCM(7x7)	99.858 $\pm$ 0.03	99.715 $\pm$ 0.07	99.755 $\pm$ 0.07	99.96 $\pm$ 0.04	99.96 $\pm$ 0.04	99.757 $\pm$ 0.07
Case VI – DN + Indices + GLCM(3x3)	97.73 $\pm$ 0.15	95.46 $\pm$ 0.29	97.167 $\pm$ 0.21	98.302 $\pm$ 0.18	98.306 $\pm$ 0.19	97.159 $\pm$ 0.21
Case VII – DN + Indices + GLCM(5x5)	99.613 $\pm$ 0.05	99.226 $\pm$ 0.1	99.314 $\pm$ 0.09	99.913 $\pm$ 0.05	99.913 $\pm$ 0.05	99.316 $\pm$ 0.09
Case VIII – DN + Indices + GLCM(7x7)	99.832 $\pm$ 0.04	99.665 $\pm$ 0.08	99.722 $\pm$ 0.06	99.942 $\pm$ 0.05	99.942 $\pm$ 0.05	99.724 $\pm$ 0.06
	Validation					
	OA (%)	κ (%)	User Accuracy		Producer Accuracy	
			Tea (%)	NonTea (%)	Tea (%)	NonTea (%)
Case I – DN	74.455 $\pm$ 0.15	48.918 $\pm$ 0.29	75.628 $\pm$ 0.18	73.366 $\pm$ 0.16	72.510 $\pm$ 0.21	76.417 $\pm$ 0.23
Case II – DN +Indices	74.463 $\pm$ 0.17	48.937 $\pm$ 0.34	75.752 $\pm$ 0.19	73.277 $\pm$ 0.18	72.304 $\pm$ 0.24	76.642 $\pm$ 0.23
Case III – DN + GLCM(3x3)	74.902 $\pm$ 0.16	49.801 $\pm$ 0.32	74.922 $\pm$ 0.25	74.883 $\pm$ 0.21	75.206 $\pm$ 0.35	74.594 $\pm$ 0.41
Case IV – DN + GLCM(5x5)	75.408 $\pm$ 0.2	50.816 $\pm$ 0.39	75.731 $\pm$ 0.22	75.088 $\pm$ 0.25	75.11 $\pm$ 0.35	75.708 $\pm$ 0.3
Case V – DN + GLCM(7x7)	75.875 $\pm$ 0.2	51.756 $\pm$ 0.39	76.891 $\pm$ 0.26	74.918 $\pm$ 0.22	74.301 $\pm$ 0.3	77.463 $\pm$ 0.34
Case VI – DN + Indices + GLCM(3x3)	75.587 $\pm$ 0.15	51.176 $\pm$ 0.31	76.1 $\pm$ 0.2	75.086 $\pm$ 0.22	74.929 $\pm$ 0.34	76.251 $\pm$ 0.31
Case VII – DN + Indices + GLCM(5x5)	76.429 $\pm$ 0.15	52.863 $\pm$ 0.29	77.197 $\pm$ 0.19	75.692 $\pm$ 0.18	75.323 $\pm$ 0.27	77.546 $\pm$ 0.26
Case VIII – DN + Indices + GLCM(7x7)	76.142 $\pm$ 0.16	52.292 $\pm$ 0.32	77.317 $\pm$ 0.21	75.047 $\pm$ 0.16	74.301 $\pm$ 0.22	78.001 $\pm$ 0.27

In addition, the importance of the features were determined for each RF classification (DN, DN + Texture, DN + Indices, DN + Texture + Indices). Figure 4.1 gives the feature ranking results of the classification with optimum input features, such as NDBD, NDBR, Green NDVI. The other feature ranking results for different input data design, which are in line with Figure 4.1 are give in the Appendix A.1, A.2, A.3, A.4, A.5, A.6, and A.7. In addition, satellite images classified according to RF algorithm is as in Figure 4.2, 4.3, 4.4, 4.5.



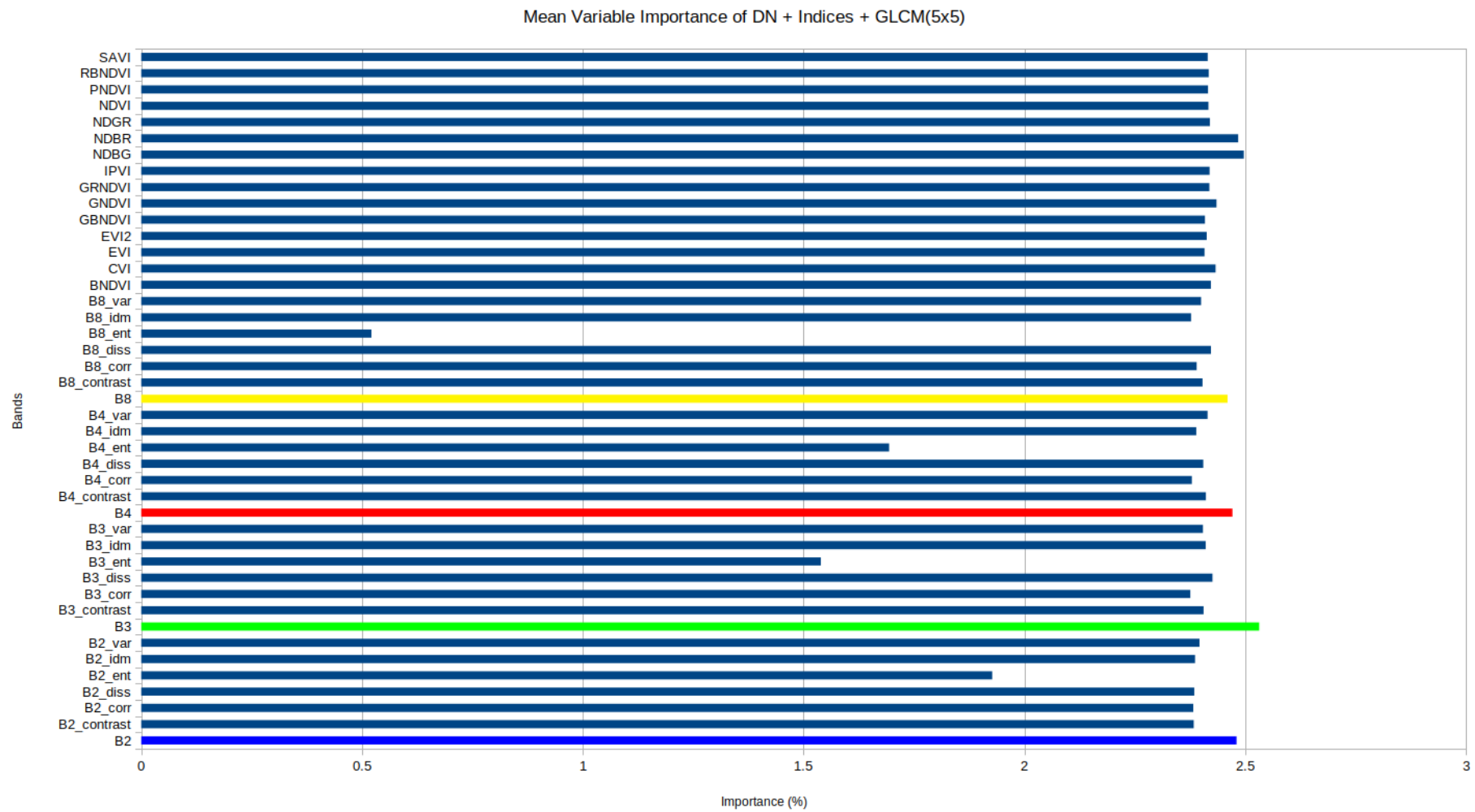


Figure 4.1 : RF variable importance for DN + Indices + GLCM(5x5).

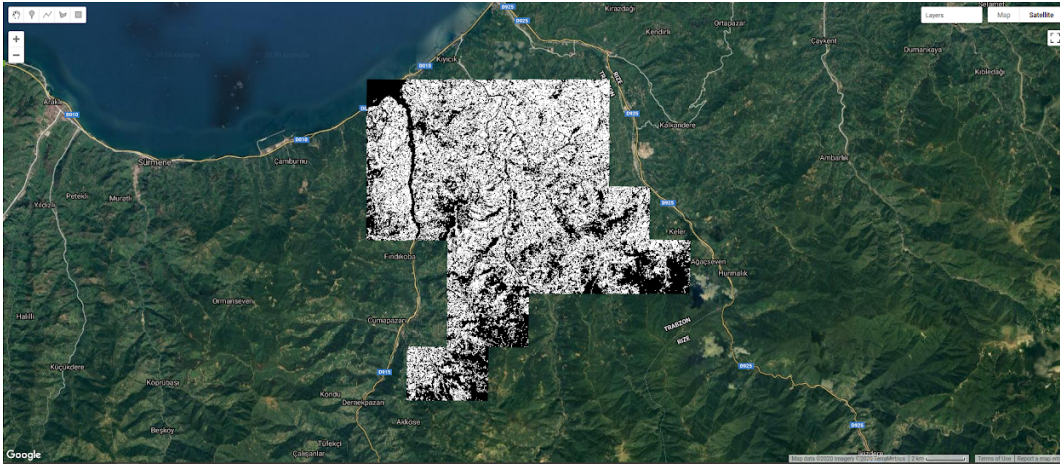


Figure 4.2 : RF classification for DN.

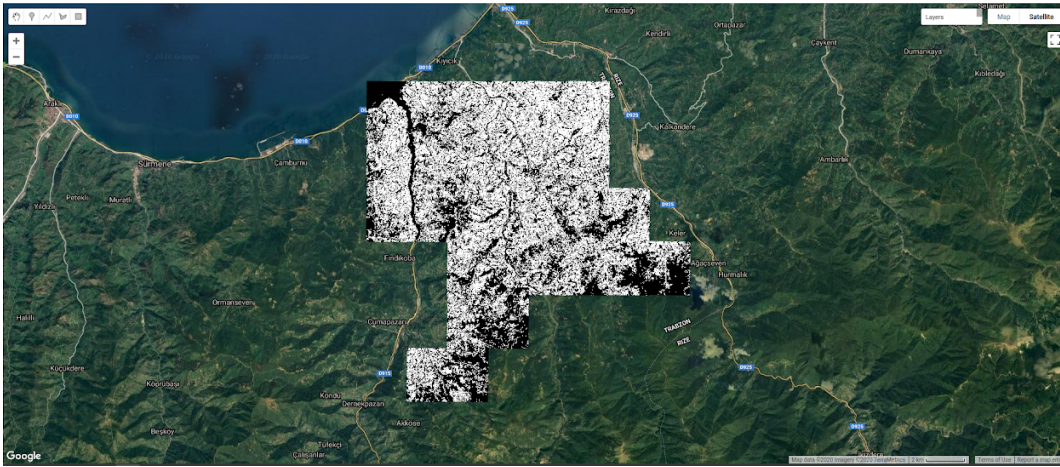


Figure 4.3 : RF classification for DN + Indices.

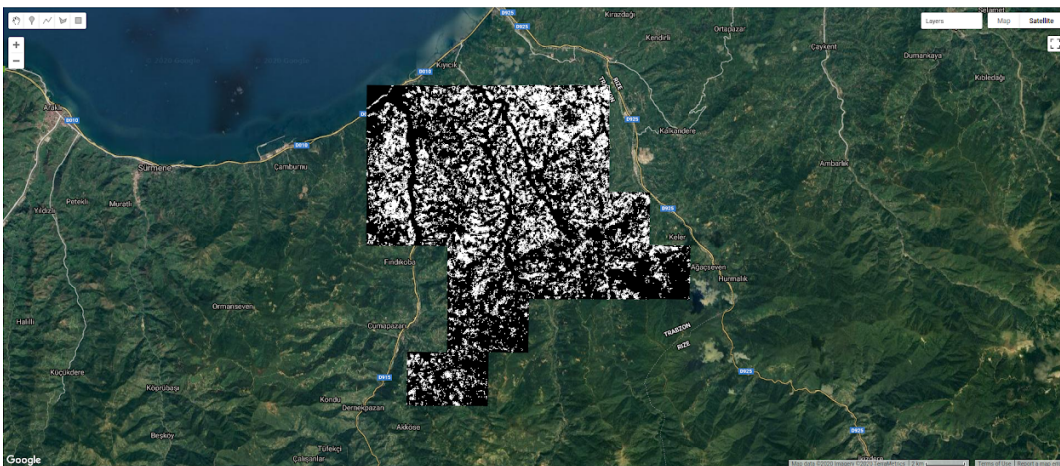


Figure 4.4 : RF classification for DN + GLCM(5x5).

When Figure 4.1 is examined, the importance of 4 features is very low. These are the entropy values of Sentinel-2 bands. In addition, although the importance of other

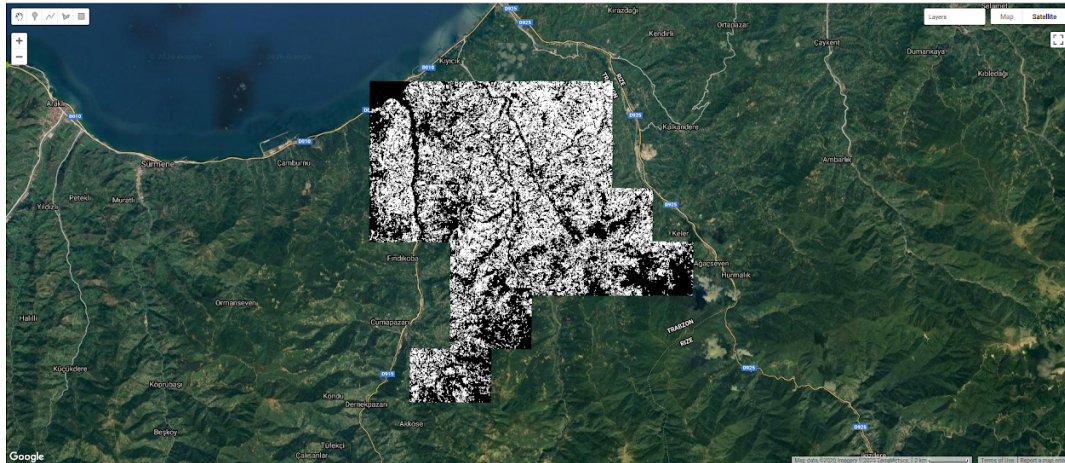


Figure 4.5 : RF classification for DN + Indices + GLCM(5x5).

features are very close to each other, 7 more features stand out. These are NDBG, NDBR and Green NDVI, as well as Sentinel-2's own bands. In the light of all these, the RF classification process was performed again using the same parameters using these 7 bands. However, contrary to the expected accuracy, it was found to be lower than all the features and 74.895% due to the increased number of correlated features causing misleading classification.

4.3 Deep Learning Classification for Detecting Tea Plantation Areas

Classification with DL algorithms cannot be done through the code editor of GEE. For this reason, it is necessary to perform operations over colab with python using the GEE API. Before classifying with the DL Algorithm, training and test data sets should be determined first, as in RF classification. To do this, in addition to Sentinel-2's own bands (Red, Green, Blue, NIR), the label data should also be processed as a band(isTea). However, since the label data is at 0.3m resolution and Sentinel-2 is at 10m resolution, there is a scale problem. This problem was addressed by rescaling the label data. Afterwards, this label image was combined with Sentinel-2 bands and a labeled data was obtained. In addition, it is known that increasing the number of features in the DL classification has positive effects on training and results. For this reason, the indices used in RF classification is added to the image with an auxiliary function via the colab as an extra feature.

The satellite image, which is within the region of interest and with all the features added, has an irregular area and is relatively large to be processed. Also, since the colab with tensorflow is not yet able to receive visual data directly from GEE, this data needs to be saved in order to read it from an environment that the colab can access. Since TensorFlow will be used for DL classification, all data in "*.tfrecord" format will increase performance. This is because tfRecord can help to read the data efficiently, serialize the data and store it in a series of files (100-200MB each), each of which can be read linearly. The TFRecord format is a simple format for storing a series of binary records [93]. GEE also has the option to export in tfrecord format. In the light of all this, the data set to be used was again divided into patches of 256x256 size with an auxiliary function and saved to the cloud environment (personal drive folder) in tfrecord format. As a result of this process, 766 equal sized image data were obtained in total. To separate this data into train and test, it was first mixed in the series. Afterwards, 70% are defined for the train and 30% for the test.

After the export process is finished, the file paths of the data in tfrecord format in drive are defined in an array. This series was mixed in itself in order to generate training and test data randomly. Of the scrambled series, 70% of the data was devoted to train data and the remaining 30% to test data. As a result of this process, a total of 536 training images and 330 test images were obtained.

For parsing TFRecord files, featuresDict is a mapping between feature names and float32 objects. This mapping is necessary for telling TensorFlow how to read data in a TFRecord file into tensors. Because, features in the TensorFlow context (i.e. tf.train.Feature) are not to be confused with Earth Engine features. Tf.feature is a protocol message type for serialized data input to the model and ee.Feature is a geometry-based geographic data structure. Specifically, all numeric data exported from Earth Engine is exported as float32.

After this process, a parsing function should be done for the data in TFRecord files. The data comes in flattened 2D arrays per record and should be used to use the first part of the array to the model and the last item of the array as a class tag. The parsing function reads data from a serialized Example proto into a dictionary in which the keys are the feature names and the values are the tensors storing the value of the features for that example. After the parsing, the first element of the tuple is a dictionary with

bands for keys and the numeric value of the bands for values. The second element of the tuple is a class label.

After determining the data structure, the model must be created. Before creating the model, there's still a wee bit of pre-processing to get the data into the right input shape and a format that can be used with cross-entropy loss. Specifically, Keras expects a list of inputs and a one-hot vector for the class. The purpose of loss functions is to compute the quantity that a model should seek to minimize during training.

While creating the model, the parameters and model structure used by the GEE Team in U-NET study were used. U-Net, a kind of CNN, was first announced in 2015 by Olaf Ronneberger, Phillip Fischer, and Thomas Brox, with the proposal to perform better segmentation on biomedical images [5]. U-Net is more successful than traditional models in terms of a different architecture and pixel based image segmentation created from convolutional neural network layers. Moreover, even if there are few training images.

U-Net takes its name from its architecture similar to the letter U, as can be seen from the figure. Input images are obtained as segmented exit map at exit. The most special aspect of its architecture is its second half. The network does not have a fully connected layer. Only the convolution layer is used. Every standard convolution process is activated with ReLU.

The U-net consists of two parts: the encoder and the decoder. The size of the input information is reduced along the layers starting and progressing with the first steps in the classical autocoder architecture. The encoder part of the architecture is completed and the decoder part begins. Linear feature representation is learned in this section and the size increases gradually. At the end of the architecture, the output size is equal to the input size. This architecture is ideal for maintaining the output size, but one problem is that it compresses the input linearly, and a bottleneck occurs where it cannot convey all the attributes. At this point, the difference from U-Net emerges. U-Net performs reverse convolution on the decoder side, and in addition, it can overcome this 'bottleneck' problem, which causes attributes to be lost due to the links from the encoder of the architecture.

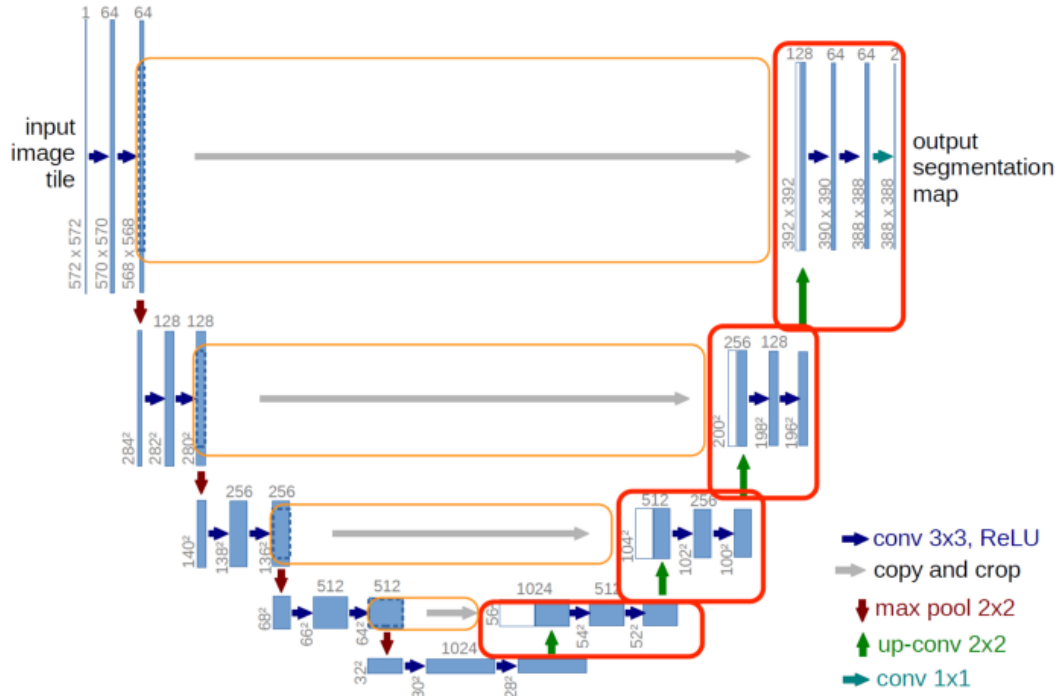


Figure 4.6 : U-NET model [5].

In this study, the model consists of many layers such as 2 dimensional convolutional, batch normalization and activation. The encoder and decoder blocks forming the layer structure have been applied 5 times. Each encoder block consists of 2D convolutional, batch normalization, relu activator, 2D convolutional, batch normalization, relu activator, 2D max pooling layers, respectively. Each decoder block consists of 2D convolutional transpose, concatenate, batch normalization, relu activator, 2D convolutional, batch normalization, relu activator, 2D convolutional, batch normalization, relu activator layers, respectively. At the center of these two blocks is the center with 2D convolutional, batch normalization, relu activator, 2D convolutional, batch normalization, relu activator layers. The table for 98 layers in total is included in Appendix C.1. Besides all the flour, hyperparameters are determined as in the Table 4.8.

The network was trained from scratch using all these parameters. However, when the model was tested, it was observed that the parameters of the GEE Team, which were taken as reference, were insufficient for the model. The accuracy of the test data did not exceed 0.65. Therefore, different models were created by increasing the number of Epoch and Batch and it was determined that the most suitable model was provided with the parameters in Table 4.9.

Table 4.8 : Hiperparameters of GEE team.

Parameters	Values
Epoch	10
Batch	16
Steps Per Epoch	10
Optimizer	SGD
Learning Rate	0.001
Working Area	GPU

Table 4.9 : Hiperparameters of best fit.

Parameters	Values
Epoch	150
Steps Per Epoch	20
Batch	20
Optimizer	Adam
Learning Rate	0.001
Working Area	TPU

Specified in the table, the Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. According to Kingma et al [94], the method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters". On the other hand, TPU, which is designated as the working area, only supports TensorFlow. In addition, it is much more effective in performance than CPU and GPUs and is used for models with matrix calculations density. In this way, models and predictions can be created more quickly.

4.7

The learning and loss graph of the best fit model is as in figure 4.8. When the graph is analyzed, it is observed that learning slows down after the 100th epoch. In addition, model accuracy was observed as approximately 0.98 and loss was observed as approximately 0.09. Since it is the model that gives the best results compared to other models created, it has been continued with this.

The network used after these results was tested with test data and the results are as in Table 4.10.

Using the specified model, the satellite image previously exported in tfRecords format will be classified. Since the target file is large, it was divided into many patches during export. There will also be a JSON basket file called "mixer", which describes the

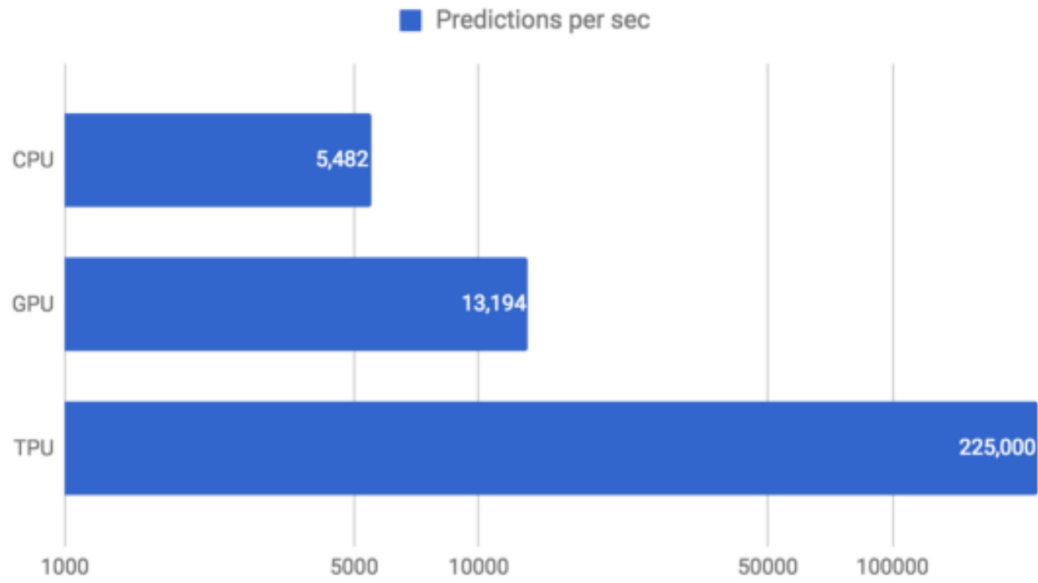


Figure 4.7 : Calculation speed of CPU, GPU and TPU.

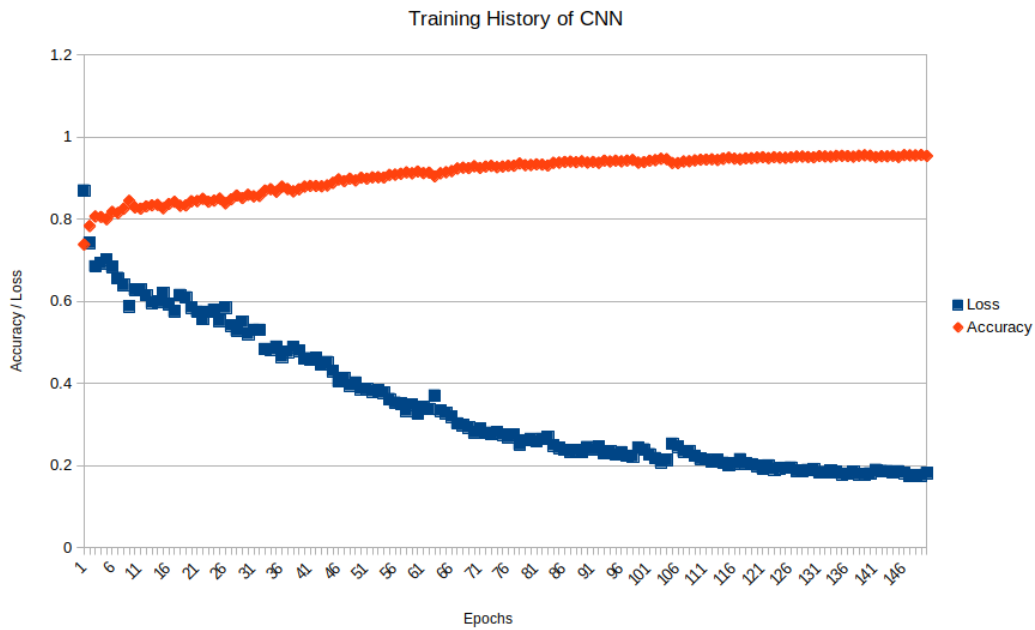


Figure 4.8 : Learning and loss graph.

Table 4.10 : Test results of model.

Parameters	Values
Loss	0.2355
Test Accuracy	0.9469

format and geographic reference of the image. Firstly, the image files and the blender file are accessed via drive, some information from the blender that will be useful during

model extraction has been obtained through these files. Examples of this information are patch sizes, projection information, and the total number of patches. Thanks to this information, the data in patch form has been made suitable for aggregation classification in a single data.

After preparing the satellite image to be classified, the model trained with the predict method was applied on this data. As a result of the application, a probability value is obtained for each pixel value as much as the number of classes (2 - tea area and non-tea area) and classification is made according to this value.

The data obtained after the classification process is completed is a list of class possibilities. These are written to the file containing a class label, which is the index of maximum probability. Afterwards, the data obtained was directly recorded in the cloud. In order to access this recorded visual data via GEE, it was later transferred to the Assets section. The image resulted from the operations is as in Figure 4.9.

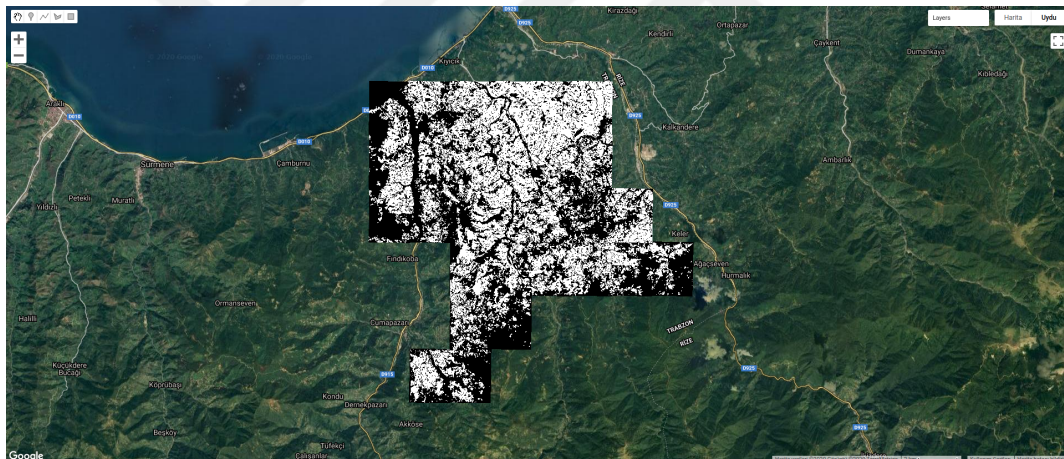


Figure 4.9 : Classified Sentinel-2 imagery with U-NET.

In addition to this, in order to examine the effect of indices on classification, a new training dataset was trained from scratch using the same architecture, using Red-Green-Blue (RGB), Near Infrared-Red-Green (NirRG) and NDVI-Red-Green (NDVIRG) band combinations. For this, a data set consisting of 766 patches, which were previously exported and containing DN + indices, was used. New training and test data were created by selecting the band combinations specified in this data separately. The data sets consist of 536 training and 230 test images. The training and test accuracy results of these data sets are as in Table 4.11.

Table 4.11 : Training and testing results of models.

Dataset	Training Accuracy	Training Loss	Test Accuracy
RGB	0.9755	0.0989	0.5563
NirRG	0.9750	0.1015	0.9285
NDVIRG	0.9748	0.1012	0.9072

As seen in Table 4.11, the accuracy of the models trained with RGB, NirRG and NDVIRG datasets created without using all the indices remains low. Extra features are observed to strengthen DL and UNET architecture. The learning graphics of the related datasets are included in Appendix B.1, B.2, B.3.

5. CONCLUSIONS AND RECOMMENDATIONS

Within the scope of the presented thesis, the tea areas located in the Eastern Black Sea Region in Turken were identified in the cloud environment with RF and U-Net -a convolutional neural network- classification methods using free-available satellite imagery. Specifically, the capability of Sentinel-2 imagery for detecting tea plantation areas was analyzed based on airborne imagery in four wavelength channels (Red, Green, Blue and Near Infrared). Different combinations of 4 Sentinel-2 bands (R-G-B-NIR), 15 spectral indices and 24 texture bands, as independent variables, were trained in order to produce spatial distribution of tea plantation.

When RF classification is examined, it is seen that the results do not change much when using the spectral bands namely DN and classification process using DN + indices. General accuracies of both train and test data are very close to each other due to the features' correlation each other. However, when GLCM texture data is added to these data, it is observed that the accuracy increases. According to the test results, when classifying with DN only, 74.455% general accuracy was obtained. If the indices are added, this accuracy could only reach 74.463%. However, when classification was made with 3x3 GLCM data, the general accuracy of the DN + GLCM classification was 74.902%, while this accuracy increased to 75.408% with 5x5 GLCM 75.875% and 7x7 GLCM. The main reason for this situation is that the determined areas have a homogeneous distribution and the low resolution of Sentinel data. Different GLCM matrixes are used for DN + GLCM + Indices classification. While the overall accuracy was 75.587% when using a 3x3 matrix, this accuracy was 76.429% in the 5x5 matrix, and the 7x7 matrix increased to 76.142%. Thus, the highest accuracy level was achieved with the data set using the 5x5 GLCM matrix and using all other features. At this point, the feature importance of this data set was examined, and among the findings, it was observed that all other GLCM properties and Index properties were highly correlated, except for the entropy values obtained by GLCM. In addition, it was determined that Sentinel-2 has its own bands among the highest values in terms

of properties. In this context, the added indices and the texture features revealed by GLCM made the DNs stand out more. The biggest effect of this is that the spectral reflectance values in the ROI are very close to each other. The biggest proof of this is that the most important band is the Green band. The changes in the green band due to the general dominance of the green in the region have had a great impact on the classification result. This situation also negatively affected the vegetation indices in the study area. These indices were ineffective due to the very close reflection values of the vegetation. In addition, the performance of different approaches and created indices for determining vegetation areas was found to be high. As can be understood from the results, texture extraction positively affected the classification results. In this context, the extraction of texture plays a major role in determining agricultural areas. Especially with a method such as RF as in this study, its success increases even more.

In addition, when the classification process was tested using artificial neural networks, an overall accuracy of 94.69% was achieved. This rate is considerably higher than the RF classification results. The most important reason for this reason from the U-Net architecture due to its structure. The origin of the U-Net architecture is to segment low-resolution biomedical images based on pixels and to do this with a small amount of training data. For this reason, this artificial neural network increases the resolution by applying size reduction and size increase in the images. Therefore, it works very well with low resolution data and provides a better segmentation by capturing spectral transition differences. In addition, compared to other architectures, it does not require different image reproduction techniques because training can be provided with a small data set. As can be seen in the U-Net results, the classification process has been completed with high success. However, in order to understand that this high performance with the rate of 94.69% comes with indices, which are additional features, training was also carried out and tested with different data sets (RGB, NirRG, NDVIRG). The lowest accuracy of these datasets tested belongs to the RGB dataset and has a 55.63% accuracy rate. Besides, NirRG has 92.85% accuracy and NDVIRG has 90.72% accuracy rates. In this context, it is seen that extra features positively affect the result. However, Nir is directly or indirectly present in all data sets except for RGB data set. The specific absorption of organic compounds in the NIR region is one of the most important reasons for this success. In addition, all other indices will be examined

in more detail in future studies. In general, the U-NET artificial neural network showed a very high performance with low resolution data with 2-class classification such as tea plantation area detection.

In addition to the findings of all these classifications, the performance of cloud computing and cloud environment were also evaluated. GEE is an important development for the science of remote sensing. Thanks to the cloud service it provides, processing large data such as satellite images by performing parallel operations on the server side. With short scripts, a processed data can be obtained within seconds. Although it has not completed its development, GEE has communicated with other services and transferred the deep learning methods to the cloud for remote sensing thanks to its Google infrastructure. This is a great advantage for users / researchers. Because deep learning methods consume a lot of resources. However, Colab eliminates this problem with its CPU, GPU and TPU support. In this way, projects on deep learning can gain great momentum.

As a result of the research conducted in general, the colab and GEE cloud infrastructure used in deep learning has accelerated the training and forecasting processes and revealed that it has a great potential for operational studies. As high resolution satellite images, which are freely available, are proposed, the method proposed in this thesis study is expected to provide insight and updated cadastral information in a very short time. Accordingly, much more free-access high resolutional data will be available in the near future, which will have a significantly influence obtaining high-quality cadastral information data.



REFERENCES

- [1] **Url-1** Sentinel-2 Mission, <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>, date retrieved 29.03.2020.
- [2] **Akar, O. and Güngör, O.** (2015). Integrating multiple texture methods and NDVI to the Random Forest classification algorithm to detect tea and hazelnut plantation areas in northeast Turkey, *International Journal of Remote Sensing*, 36(2), 442–464.
- [3] **Ok, A., Akar, O. and Gungor, O.** (2011). Rastgele Orman Sınıflandırma Yöntemi Yardımıyla Tarım Alanlarındaki ürün Çeşitliliğinin Sınıflandırılması, *TUFUAB 2011 VI. Teknik Sempozyumu*, Antalya, Turkey.
- [4] **Agatonovic-Kustrin, S. and Beresford, R.** (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, *Journal of Pharmaceutical and Biomedical Analysis*, 22(5), 717 – 727.
- [5] **Ronneberger, O., Fischer, P. and Brox, T.**, (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation, 1505.04597.
- [6] **Townshend, J., Justice, C., Li, W., Gurney, C. and McManus, J.** (1991). Global land cover classification by remote sensing: present capabilities and future possibilities, *Remote Sensing of Environment*, 35(2), 243 – 255.
- [7] **Wesseling, J. and Feddes, R.** (2006). Assessing crop water productivity from field to regional scale, *Agricultural Water Management*, 86(1), 30 – 39.
- [8] **Rodriguez-Galiano, V., Ghimire, B., Rogan, J., Chica-Olmo, M. and Rigol-Sanchez, J.** (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification, *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 93 – 104.
- [9] **Chappelle, E.W., Kim, M.S. and McMurtrey, J.E.** (1992). Ratio analysis of reflectance spectra (RARS): An algorithm for the remote estimation of the concentrations of chlorophyll A, chlorophyll B, and carotenoids in soybean leaves, *Remote Sensing of Environment*, 39(3), 239 – 247.
- [10] **Yang, C., Everitt, J.H. and Murden, D.** (2011). Evaluating high resolution SPOT 5 satellite imagery for crop identification, *Computers and Electronics in Agriculture*, 75(2), 347 – 354.
- [11] **Turker, M. and Ok, A.** (2011). Field-based crop classification using SPOT4, SPOT5, IKONOS and QuickBird imagery for agricultural areas:

A comparison study, *International Journal of Remote Sensing*, 32, 9735–9768.

- [12] **Url-2** Landsat Missions, <https://www.usgs.gov/land-resources/nli/landsat>, date retrieved 29.03.2020.
- [13] **Vuolo, F., Neuwirth, M., Immitzer, M., Atzberger, C. and Ng, W.T.** (2018). How much does multi-temporal Sentinel-2 data improve crop type classification?, *International Journal of Applied Earth Observation and Geoinformation*, 72, 122 – 130.
- [14] **O., M., Salami, A., Olaleye, J. and Okewusi, O.** (2016). Object Based and Pixel Based Classification Using RapidEye Satellite Imagery of ETI-OSA, Lagos, Nigeria, *Geoinformatics FCE CTU*, 15, 59–69.
- [15] **Küçük, , Taşkın, G. and Erten, E.** (2016). Paddy-Rice Phenology Classification Based on Machine-Learning Methods Using Multitemporal Co-Polar X-Band SAR Images, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6), 2509–2519.
- [16] **Akar, O. and Güngör, O.** (2012). Classification of multispectral images using Random Forest algorithm, *UCTEA Chamber of Surveying and Cadastre Engineer Journal of Geodesy and Geoinformation*, 1(106), 105–112.
- [17] **Dizdaroğlu, T.**, (2019), Sentinel-1 Ve Sentinel-2 Verilerinden Tarımsal Ürün Sınıflandırması İçin Makine Öğrenme Algoritmalarının Karşılaştırılması.
- [18] **Breiman, L.** (2001). Random Forests, *Machine learning*, 45(1), 5–32.
- [19] **Giacinto, G. and Roli, F.** (1997). Ensembles of Neural Networks for Soft Classification of Remote Sensing Images.
- [20] **Dietterich, T.G. et al.** (2002). Ensemble learning, *The handbook of brain theory and neural networks*, 2, 110–125.
- [21] **Schapire, R., Freund, Y., Barlett, P. and Lee, W.** (1997). Boosting the margin: A new explanation for the effectiveness of voting methods., volume 26, pp.322–330.
- [22] **Breiman, L.** (1996). Bagging predictors, *Machine learning*, 24(2), 123–140.
- [23] **Aghlmand, M., Kalkan, K. and Onur, M.** (2019). Investigation of Forest Classification Capability of Sentinel-2 Satellite.
- [24] **Shah, S.H., Angel, Y., Houborg, R., Ali, S. and McCabe, M.F.** (2019). A Random Forest Machine Learning Approach for the Retrieval of Leaf Chlorophyll Content in Wheat, *Remote Sensing*, 11(8).
- [25] **Chang, L., Peng-Sen, S. and Shi-Rong, a.** (2016). A review of plant spectral reflectance response to water physiological changes, *Chinese Journal of Plant Ecology*, 40, 80–91.

- [26] **Jinru, X. and Su, B.** (2017). Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications, *Journal of Sensors*, 2017, 1–17.
- [27] **Üstüner, M., Balik Sanli, F., Abdikan, S., Esetlili, M. and Kurucu, Y.** (2014). Crop Type Classification Using Vegetation Indices of RapidEye Imagery, volume XL-7.
- [28] **Kuzucu, A. and Balcik, F.** (2017). Testing The Potential Of Vegetation Indices For Land Use/Cover Classification Using High Resolution Data, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W4, 279–283.
- [29] **Gitelson, A.A., Kaufman, Y.J., Stark, R. and Rundquist, D.** (2002). Novel algorithms for remote estimation of vegetation fraction, *Remote Sensing of Environment*, 80(1), 76 – 87.
- [30] **min Wang, F., feng Huang, J., lin Tang, Y. and Wang, X.Z.** (2007). New Vegetation Index and Its Application in Estimating Leaf Area Index of Rice, *Rice Science*, 14(3), 195 – 203.
- [31] **Crippen, R.E.** (1990). Calculating The Vegetation Index Faster, *Remote Sensing of Environment*, 34(1), 71 – 73.
- [32] **Liu, H.Q. and Huete, A.** (1995). A feedback based modification of the NDVI to minimize canopy background and atmospheric noise, *IEEE Transactions on Geoscience and Remote Sensing*, 33(2), 457–465.
- [33] **Jiang, Z., Huete, A.R., Didan, K. and Miura, T.** (2008). Development of a two-band enhanced vegetation index without a blue band, *Remote Sensing of Environment*, 112(10), 3833 – 3845.
- [34] **Haralick, R.M., Shanmugam, K. and Dinstein, I.** (1973). Textural Features for Image Classification, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), 610–621.
- [35] **Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G. and Johnson, B.A.** (2019). Deep learning in remote sensing applications: A meta-analysis and review, *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166 – 177.
- [36] **Silva, J., Palma, H.H., Núñez, W.N., Ruiz-Lazaro, A. and Varela, N.** (2020). Neural Networks for Tea Leaf Classification, *Journal of Physics: Conference Series*, 1432, 012075.
- [37] **Doğan, F. and Türkoğlu,** (2019). Derin Öğrenme İle Uydu Görüntülerinin Sınıflandırılması.
- [38] **Hang, T., Raghavan, V. and Yonezawa, G.** Pixel-and Object-based Deep Neural Network LULC Classification using Remote Sensing Data and R Software, <http://www.jsgis-map.org/geoinforum2018/14.pdf>.

- [39] **Gamba, P., Du, P., Juergens, C. and Maktav, D.** (2011). Foreword to the Special Issue on “Human Settlements: A Global Remote Sensing Challenge”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(1), 5–7.
- [40] **Xu, C., Du, X., Yan, Z. and Fan, X.** (2020). ScienceEarth: A Big Data Platform for Remote Sensing Data Processing, *Remote Sensing*, 12(4).
- [41] **Wang, L., Yan, J. and Ma, Y.** (2019). *Cloud Computing in Remote Sensing*.
- [42] **Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D. and Moore, R.** (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone, *Remote Sensing of Environment*.
- [43] **Liss, B., Howland, M.D. and Levy, T.E.** (2017). Testing Google Earth Engine for the automatic identification and vectorization of archaeological features: A case study from Faynan, Jordan, *Journal of Archaeological Science: Reports*, 15, 299 – 304.
- [44] **Hu, Y., Dong, Y. and Batunacun** (2018). An automatic approach for land-change detection and land updates based on integrated NDVI timing analysis and the CVAPS method with GEE support, *ISPRS Journal of Photogrammetry and Remote Sensing*, 146, 347 – 359.
- [45] **Teluguntla, P., Thenkabail, P.S., Oliphant, A., Xiong, J., Gumma, M.K., Congalton, R.G., Yadav, K. and Huete, A.** (2018). A 30-m landsat-derived cropland extent product of Australia and China using random forest machine learning algorithm on Google Earth Engine cloud computing platform, *ISPRS Journal of Photogrammetry and Remote Sensing*, 144, 325 – 340.
- [46] **Wang, Y., Li, Z., Zeng, C., Xia, G. and Shen, H.** (2020). An Urban Water Extraction Method Combining Deep Learning and Google Earth Engine, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 768–781.
- [47] **Doğanay, S.** (2006). Geographical Features of Tea Plantation in Trabzon, *Doğu Coğrafya Dergisi*, 11(16), 89–111.
- [48] **Jamil, A. and Bayram, B.** (2018). Tree Species Extraction and Land Use/Cover Classification From High-Resolution Digital Orthophoto Maps, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(1), 89–94.
- [49] **Aghlmand, M., Kalkan, K. and Onur, M.** (2019). Investigation of Forest Classification Capability of Sentinel-2 Satellite.
- [50] **Demirhan, A. and Guler, I.** (2010). Image segmentation using self-organizing maps and gray level co-occurrence matrices, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 25, 285–291.
- [51] **Url-3** What is cloud computing?, <https://aws.amazon.com/what-is-cloud-computing/>, date retrieved 19.03.2020.

- [52] **Url-4** Cloud computing: A complete guide, <https://www.ibm.com/cloud/learn/cloud-computing/>, date retrieved 19.03.2020.
- [53] **Url-5** Google Cloud Platform, <https://searchcloudcomputing.techtarget.com/definition/Google-Cloud-Platform>, date retrieved 19.03.2020.
- [54] **Raschka, S., Patterson, J. and Nolet, C.** (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence, *Information*, 11, 193.
- [55] **Url-6** What is Colaboratory?, <https://colab.research.google.com/notebooks/intro.ipynb>, date retrieved 19.03.2020.
- [56] **Clinton, N. and Brown, C.** TensorFlow with EE, <https://sites.google.com/earthoutreach.org/eeus2018/agenda/session-descriptions>, date retrieved 25.03.2020.
- [57] **Alpaydin, E.** (2010). *Introduction to Machine Learning, Second Edition (Adaptive Computation and Machine Learning)*, Adaptive Computation and Machine Learning, The MIT Press, 2 edition.
- [58] **Farda, N.M.** (2017). Multi-temporal Land Use Mapping of Coastal Wetlands Area using Machine Learning in Google Earth Engine, *IOP Conference Series: Earth and Environmental Science*, 98, 012042.
- [59] **Kotsiantis, S.** (2007). Supervised machine learning: A review of classification techniques, *Informatica (Ljubljana)*, 31(3), 249–268.
- [60] **Gao, J.** (2008). *Digital Analysis of Remotely Sensed Imagery*, McGraw-Hill Professional, 1st edition.
- [61] **Yıldırım, I., Ersoy, O.K. and Yazgan, B.** (2005). Improvement of classification accuracy in remote sensing using morphological filter, *Advances in Space Research*, 36(5), 1003 – 1006, atmospheric Remote Sensing: Earth's Surface, Troposphere, Stratosphere and Mesosphere- I.
- [62] **Pal, M.** (2005). Random forest classifier for remote sensing classification, *International Journal of Remote Sensing - INT J REMOTE SENS*, 26, 217–222.
- [63] **Du, P., Samat, A., Waske, B., Liu, S. and Li, Z.** (2015). Random Forest and Rotation Forest for fully polarized SAR image classification using polarimetric and spatial features, *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 38 – 53.
- [64] **Belgiu, M. and Drăguț, L.** (2016). Random forest in remote sensing: A review of applications and future directions, *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24 – 31.
- [65] **Körting, T.S., Fonseca, L.M.G. and Câmara, G.** (2013). GeoDMA—Geographic Data Mining Analyst, *Computers Geosciences*, 57, 133 – 145.

- [66] **Breiman, L. and Cutler, A.**, (2005). Random Forests, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm, date retrieved 05.04.2020.
- [67] **Archer, K.J. and Kimes, R.V.** (2008). Empirical characterization of random forest variable importance measures, *Computational Statistics Data Analysis*, 52(4), 2249 – 2260.
- [68] **Pal, M.** (2003). Random forests for land cover classification, *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, volume 6, pp.3510–3512 vol.6.
- [69] **Breiman, L.**, (2002). Manual On Setting Up, Using, And Understanding Random Forests, https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf, date retrieved 05.04.2020.
- [70] **Chen, C.** (2007). *Image Processing for Remote Sensing*, CRC Press, 1 edition.
- [71] **Liaw, A. and Wiener, M.** (2001). Classification and Regression by RandomForest, *Forest*, 23.
- [72] **Furlanello, C., Neteler, M., Merler, S., Menegon, S., Fontanari, S., Donini, A., Rizzoli, A. and Chemini, C.** (2003). GIS and the Random Forest Predictor: Integration in R for Tick-Borne Disease Risk Assessment, *Distributed Statistical Computing 2003*, Austrian Association for Statistical Computing (AASC) and the R Foundation for Statistical Computing, Vienna University of Technology, Main Building, Karlsplatz 13, 1040 Wien, Austria, <https://www.r-project.org/conferences/DSC-2003/Proceedings/>.
- [73] **Özkan, Y.** (2013). *Veri Madenciliği Yöntemleri*, Papatya Yayıncılık, İstanbul.
- [74] **Takçı, D.H.** (2008). Veri Madenciliği Dersi G Y T E.
- [75] **Watts, J.D., Powell, S.L., Lawrence, R.L. and Hilker, T.** (2011). Improved classification of conservation tillage adoption using high temporal and synthetic satellite imagery, *Remote Sensing of Environment*, 115(1), 66 – 75.
- [76] **Salakhutdinov, R. and Hinton, G.**, (2007), Deep Belief Networks.
- [77] **Fausett L.** (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*.
- [78] **Hinton, G., Osindero, S. and Teh, Y.W.** (2006). A fast learning algorithm for deep belief nets, *Neural Computation*, 18(7), 1527–1554.
- [79] **Kohonen, T.** (1997). *Self-Organizing Maps*, Springer Series in Information Sciences 30, Springer Berlin Heidelberg, 3rd ed edition.
- [80] **Fukushima, K.** (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, 36(4), 193–202.

- [81] **Hubel, D. and Wiesel, T.** (1959). Receptive fields of single neurones in the cat's striate cortex, *The Journal of Physiology*, 148(3), 574–591.
- [82] **Huang, F. and LeCun, Y.** (2006). Large-scale learning with SVM and convolutional nets for generic object categorization, volume 1, pp.284–291.
- [83] **Bengio, Y., Courville, A. and Vincent, P.** (2013). Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.
- [84] **LeCun, Y., Bengio, Y. and Hinton, G.** (2015). Deep Learning, *Nature*, 521, 436–44.
- [85] **He, K., Zhang, X., Ren, S. and Sun, J.** (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, volume 2015 International Conference on Computer Vision, ICCV 2015, pp.1026–1034.
- [86] **Scott, G., Marcum, R., Davis, C. and Nivin, T.** (2017). Fusion of Deep Convolutional Neural Networks for Land Cover Classification of High-Resolution Imagery, *IEEE Geoscience and Remote Sensing Letters*, PP, 1–5.
- [87] **Cichy, R.M., Khosla, A., Pantazis, D., Torralba, A. and Oliva, A.** (2016). Deep Neural Networks predict Hierarchical Spatio-temporal Cortical Dynamics of Human Visual Object Recognition, *CoRR*, *abs/1601.02970*, 1601.02970.
- [88] **Yu, B.** (2017). Design and Implementation of Behavior Recognition System Based on Convolutional Neural Network, *ITM Web of Conferences*, 12, 01025.
- [89] **Castelluccio, M., Poggi, G., Sansone, C. and Verdoliva, L.** (2015). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks.
- [90] **Jarrett, K., Kavukcuoglu, K., Ranzato, M. and Lecun, Y.** (2009). What is the Best Multi-Stage Architecture for Object Recognition?, volume 12, p.2146–2153.
- [91] **Nair, V. and Hinton, G.** (2010). Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, volume 27, pp.807–814.
- [92] **Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.** (2012). Improving neural networks by preventing co-adaptation of feature detectors, *CoRR*, *abs/1207.0580*, 1207.0580.
- [93] **Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R.,**

Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X., (2015), TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, <https://www.tensorflow.org/>.

[94] **Kingma, D.P. and Ba, J., (2014), Adam: A Method for Stochastic Optimization, <https://arxiv.org/abs/1412.6980>, 1412.6980.**



APPENDICES

APPENDIX A : Random Forest Variable Importance Graphichs

APPENDIX B : Learning and Loss Graph of Datasets

APPENDIX C : Layers of DL Method





APPENDIX A

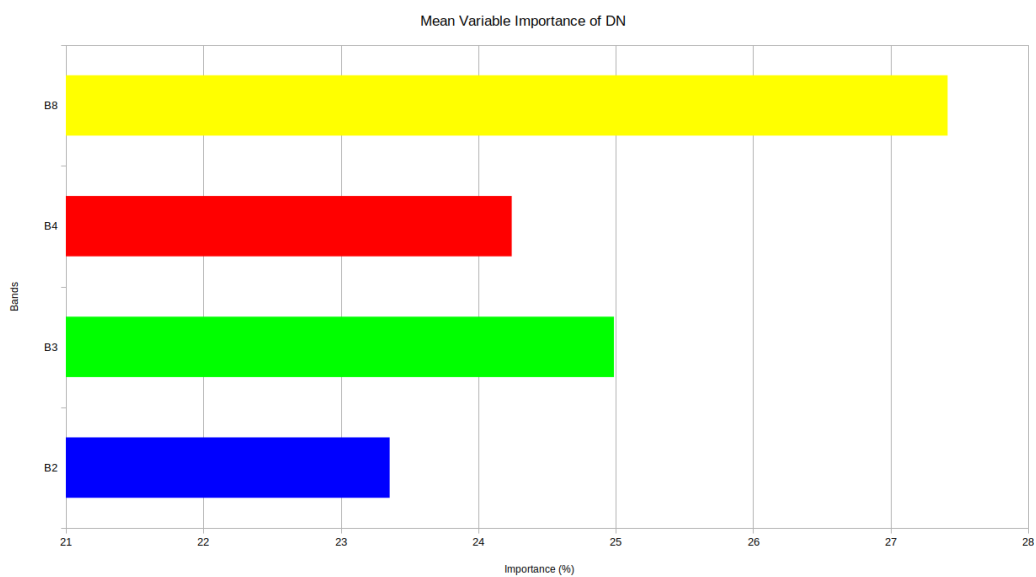


Figure A.1 : RF Variable Importance for DN.

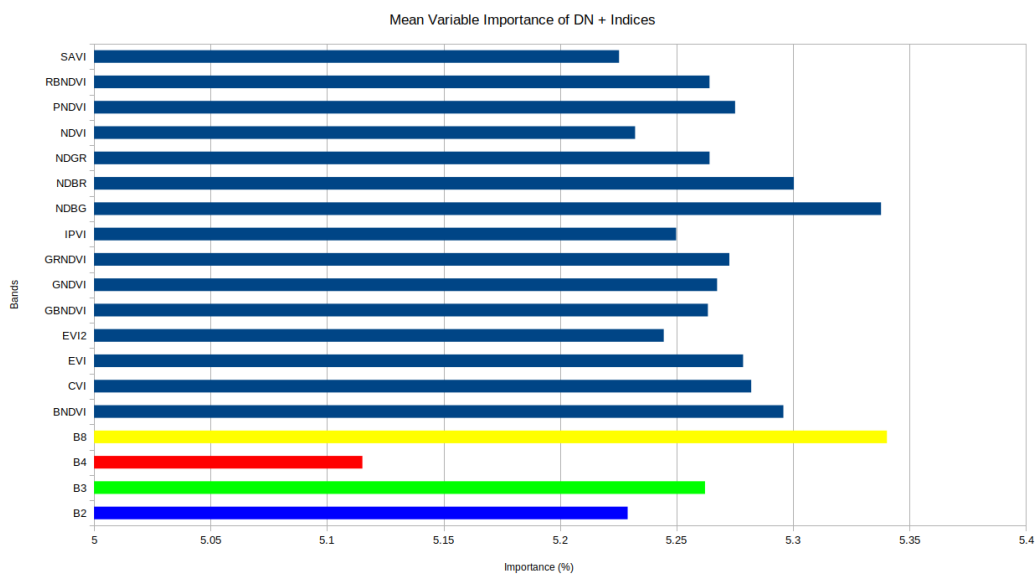


Figure A.2 : RF Variable Importance for DN + Indices.

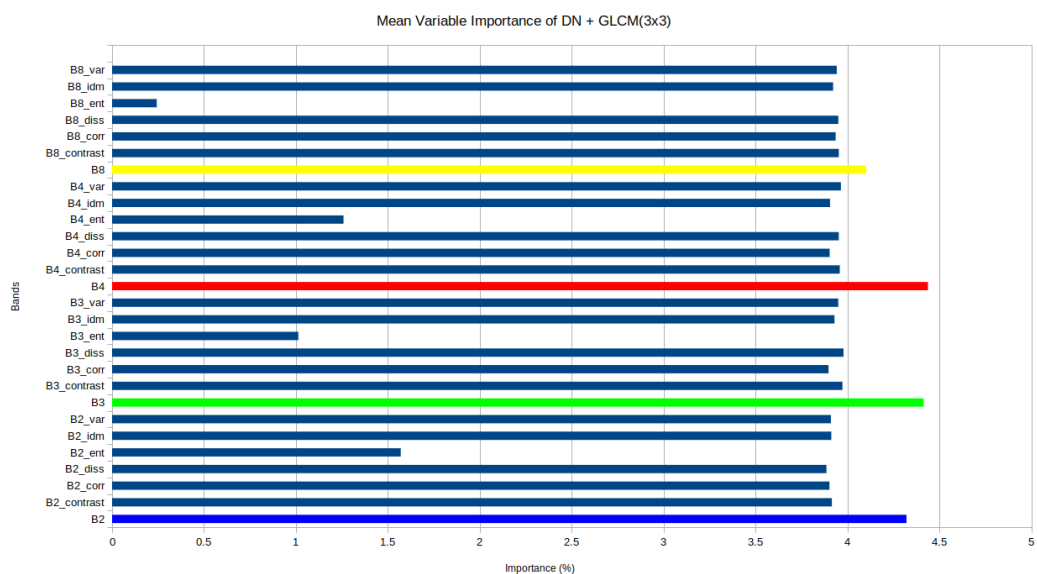


Figure A.3 : RF Variable Importance for DN + GLCM(3x3).

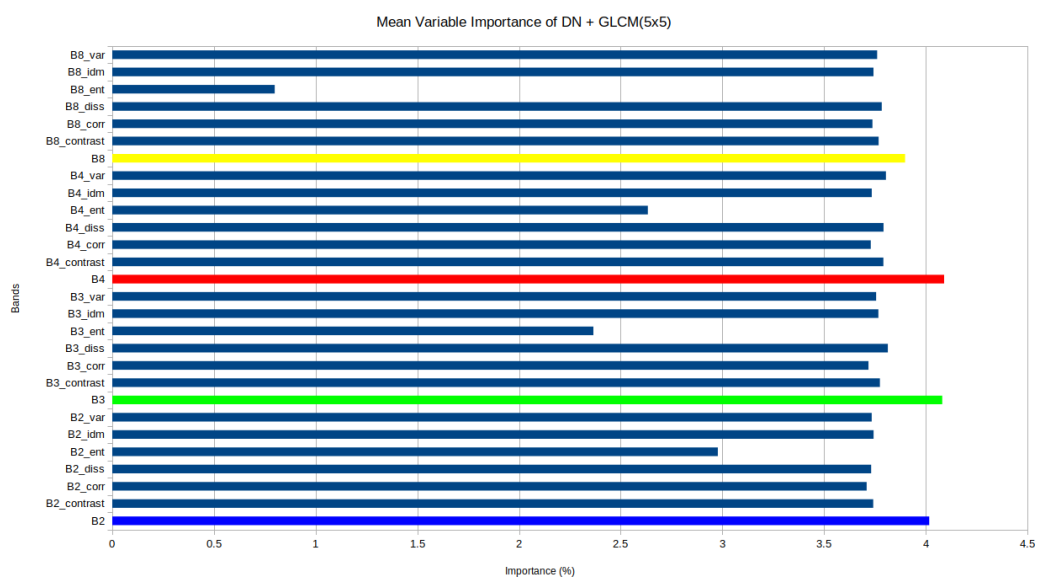


Figure A.4 : RF Variable Importance for DN + GLCM(5x5).

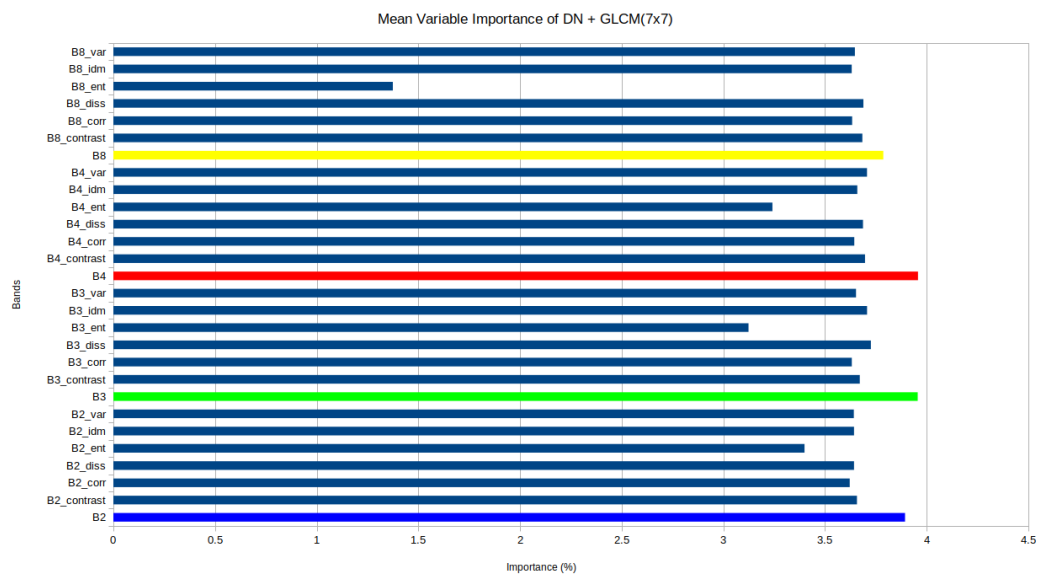


Figure A.5 : Random Forest Variable Importance for DN + GLCM(7x7).

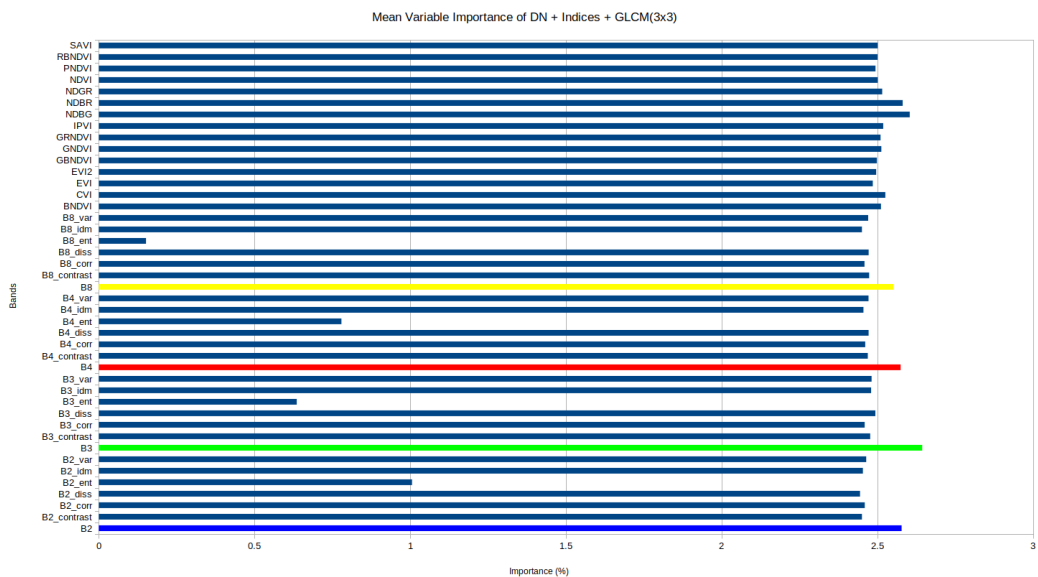


Figure A.6 : RF Variable Importance for DN + Indices + GLCM(3x3).

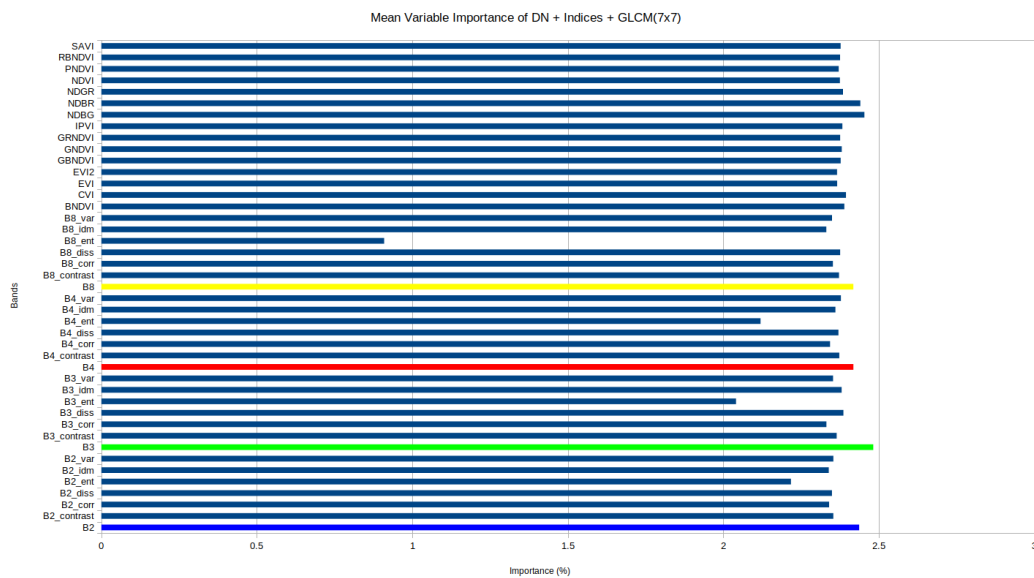


Figure A.7 : RF Variable Importance for DN + Indices + GLCM(7x7).

APPENDIX B

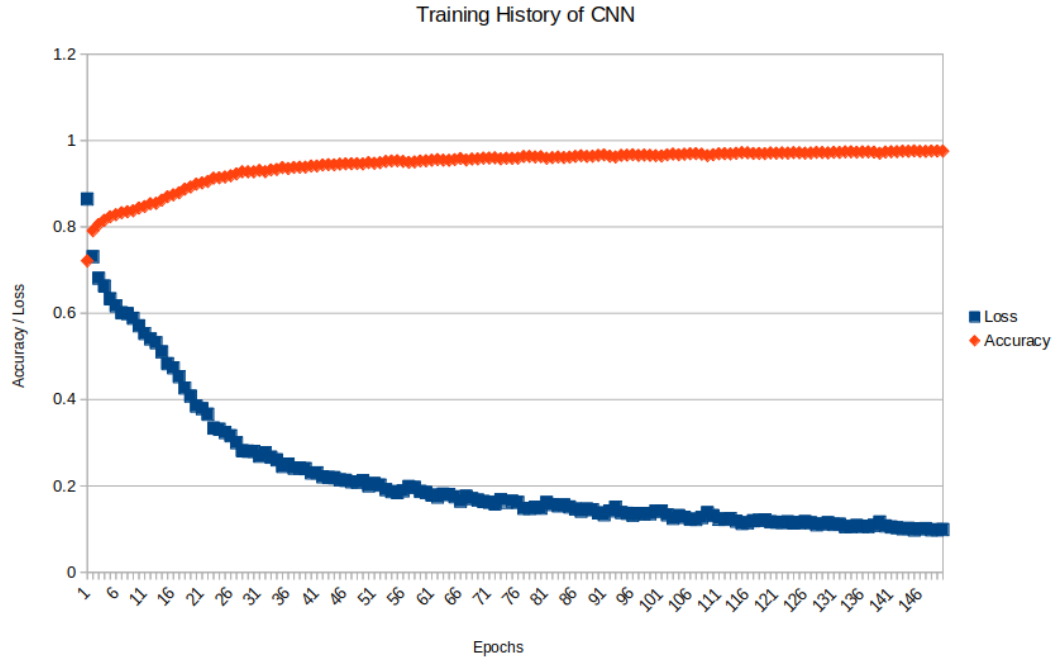


Figure B.1 : Learning and Loss Graph of RGB.

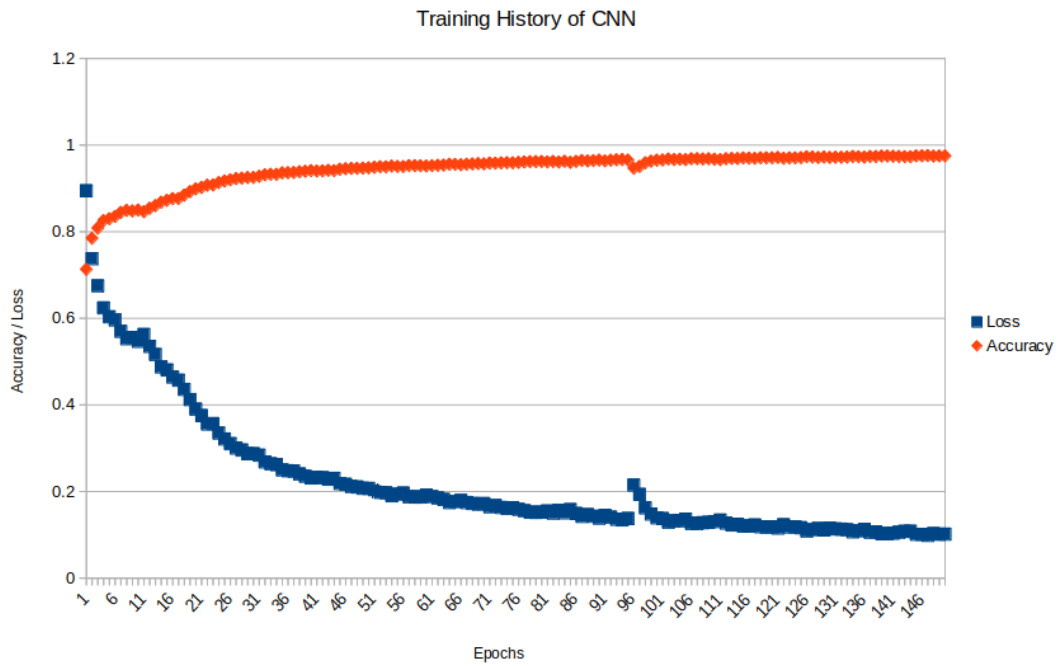


Figure B.2 : Learning and Loss Graph of NirRG.

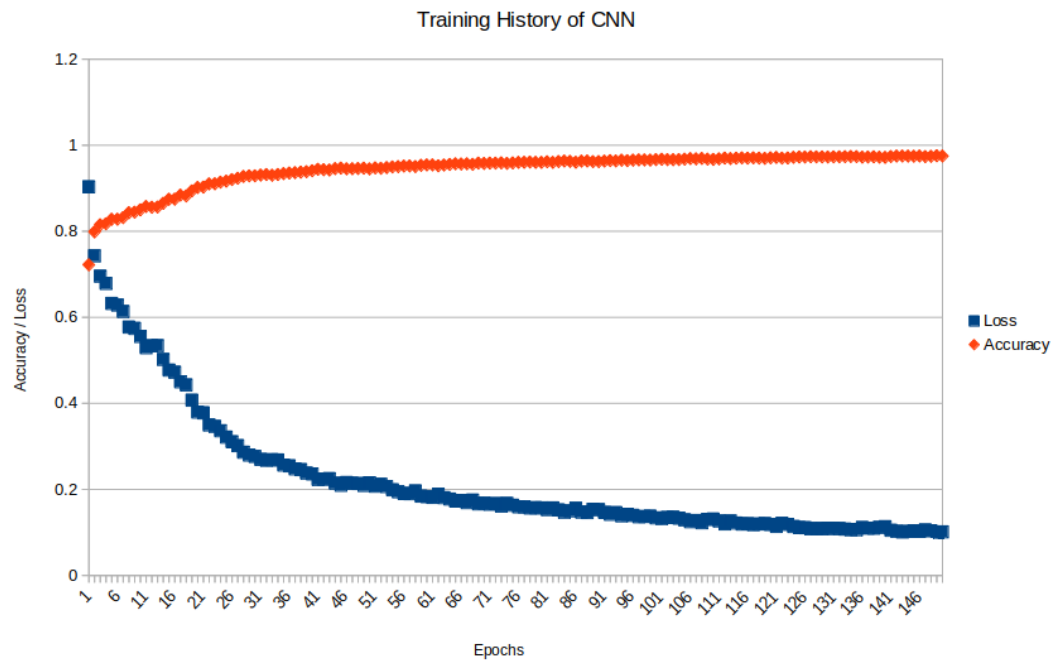


Figure B.3 : Learning and Loss Graph of NDVIRG.

APPENDIX C

Table C.1 : Layers of DL Method.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 256, 256, 19)	0	
conv2d (Conv2D)	(None, 256, 256, 32)	5504	input_1[0][0]
batch_normalization (BatchNormalization)	(None, 256, 256, 32)	128	conv2d[0][0]
activation (Activation)	(None, 256, 256, 32)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 256, 256, 32)	9248	activation[0][0]
batch_normalization_1 (BatchNormalization)	(None, 256, 256, 32)	128	conv2d_1[0][0]
activation_1 (Activation)	(None, 256, 256, 32)	0	batch_normalization_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0	activation_1[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 64)	18496	max_pooling2d[0][0]
batch_normalization_2 (BatchNormalization)	(None, 128, 128, 64)	256	conv2d_2[0][0]
activation_2 (Activation)	(None, 128, 128, 64)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 128, 128, 64)	36928	activation_2[0][0]
batch_normalization_3 (BatchNormalization)	(None, 128, 128, 64)	256	conv2d_3[0][0]
activation_3 (Activation)	(None, 128, 128, 64)	0	batch_normalization_3[0][0]
max_pooling2d_1 (Max-Pooling2D)	(None, 64, 64, 64)	0	activation_3[0][0]
conv2d_4 (Conv2D)	(None, 64, 64, 128)	73856	max_pooling2d_1[0][0]
batch_normalization_4 (BatchNormalization)	(None, 64, 64, 128)	512	conv2d_4[0][0]
activation_4 (Activation)	(None, 64, 64, 128)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 64, 64, 128)	147584	activation_4[0][0]
batch_normalization_5 (BatchNormalization)	(None, 64, 64, 128)	512	conv2d_5[0][0]
activation_5 (Activation)	(None, 64, 64, 128)	0	batch_normalization_5[0][0]

Continued on next page

Continuation of Table C.1			
Layer (type)	Output Shape	Param #	Connected to
max_pooling2d_2 (Max-Pooling2D)	(None, 32, 32, 128)	0	activation_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 256)	295168	max_pooling2d_2[0][0]
batch_normalization_6 (BatchNormalization)	(None, 32, 32, 256)	1024	conv2d_6[0][0]
activation_6 (Activation)	(None, 32, 32, 256)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 256)	590080	activation_6[0][0]
batch_normalization_7 (BatchNormalization)	(None, 32, 32, 256)	1024	conv2d_7[0][0]
activation_7 (Activation)	(None, 32, 32, 256)	0	batch_normalization_7[0][0]
max_pooling2d_3 (Max-Pooling2D)	(None, 16, 16, 256)	0	activation_7[0][0]
conv2d_8 (Conv2D)	(None, 16, 16, 512)	1180160	max_pooling2d_3[0][0]
batch_normalization_8 (BatchNormalization)	(None, 16, 16, 512)	2048	conv2d_8[0][0]
activation_8 (Activation)	(None, 16, 16, 512)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 16, 16, 512)	2359808	activation_8[0][0]
batch_normalization_9 (BatchNormalization)	(None, 16, 16, 512)	2048	conv2d_9[0][0]
activation_9 (Activation)	(None, 16, 16, 512)	0	batch_normalization_9[0][0]
max_pooling2d_4 (Max-Pooling2D)	(None, 8, 8, 512)	0	activation_9[0][0]
conv2d_10 (Conv2D)	(None, 8, 8, 1024)	4719616	max_pooling2d_4[0][0]
batch_normalization_10 (BatchNormalization)	(None, 8, 8, 1024)	4096	conv2d_10[0][0]
activation_10 (Activation)	(None, 8, 8, 1024)	0	batch_normalization_10[0][0]
conv2d_11 (Conv2D)	(None, 8, 8, 1024)	9438208	activation_10[0][0]
batch_normalization_11 (BatchNormalization)	(None, 8, 8, 1024)	4096	conv2d_11[0][0]
activation_11 (Activation)	(None, 8, 8, 1024)	0	batch_normalization_11[0][0]
conv2d_transpose (Conv2DTranspose)	(None, 16, 16, 512)	2097664	activation_11[0][0]
concatenate (Concatenate)	(None, 16, 16, 1024)	0	activation_9[0][0]
			conv2d_transpose[0][0]
batch_normalization_12 (BatchNormalization)	(None, 16, 16, 1024)	4096	concatenate[0][0]
activation_12 (Activation)	(None, 16, 16, 1024)	0	batch_normalization_12[0][0]
conv2d_12 (Conv2D)	(None, 16, 16, 512)	4719104	activation_12[0][0]
batch_normalization_13 (BatchNormalization)	(None, 16, 16, 512)	2048	conv2d_12[0][0]
activation_13 (Activation)	(None, 16, 16, 512)	0	batch_normalization_13[0][0]
conv2d_13 (Conv2D)	(None, 16, 16, 512)	2359808	activation_13[0][0]
Continued on next page			

Continuation of Table C.1			
Layer (type)	Output Shape	Param #	Connected to
batch_normalization_14 (BatchNormalization)	(None, 16, 16, 512)	2048	conv2d_13[0][0]
activation_14 (Activation)	(None, 16, 16, 512)	0	batch_normalization_14[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 32, 32, 256)	524544	activation_14[0][0]
concatenate_1 (Concate- nate)	(None, 32, 32, 512)	0	activation_7[0][0]
			conv2d_transpose_1[0][0]
batch_normalization_15 (BatchNormalization)	(None, 32, 32, 512)	2048	concatenate_1[0][0]
activation_15 (Activation)	(None, 32, 32, 512)	0	batch_normalization_15[0][0]
conv2d_14 (Conv2D)	(None, 32, 32, 256)	1179904	activation_15[0][0]
batch_normalization_16 (BatchNormalization)	(None, 32, 32, 256)	1024	conv2d_14[0][0]
activation_16 (Activation)	(None, 32, 32, 256)	0	batch_normalization_16[0][0]
conv2d_15 (Conv2D)	(None, 32, 32, 256)	590080	activation_16[0][0]
batch_normalization_17 (BatchNormalization)	(None, 32, 32, 256)	1024	conv2d_15[0][0]
activation_17 (Activation)	(None, 32, 32, 256)	0	batch_normalization_17[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 64, 64, 128)	131200	activation_17[0][0]
concatenate_2 (Concate- nate)	(None, 64, 64, 256)	0	activation_5[0][0]
			conv2d_transpose_2[0][0]
batch_normalization_18 (BatchNormalization)	(None, 64, 64, 256)	1024	concatenate_2[0][0]
activation_18 (Activation)	(None, 64, 64, 256)	0	batch_normalization_18[0][0]
conv2d_16 (Conv2D)	(None, 64, 64, 128)	295040	activation_18[0][0]
batch_normalization_19 (BatchNormalization)	(None, 64, 64, 128)	512	conv2d_16[0][0]
activation_19 (Activation)	(None, 64, 64, 128)	0	batch_normalization_19[0][0]
conv2d_17 (Conv2D)	(None, 64, 64, 128)	147584	activation_19[0][0]
batch_normalization_20 (BatchNormalization)	(None, 64, 64, 128)	512	conv2d_17[0][0]
activation_20 (Activation)	(None, 64, 64, 128)	0	batch_normalization_20[0][0]
conv2d_transpose_3 (Conv2DTrans	(None, 128, 128, 64)	32832	activation_20[0][0]
concatenate_3 (Concate- nate)	(None, 128, 128, 128)	0	activation_3[0][0]
			conv2d_transpose_3[0][0]
Continued on next page			

Continuation of Table C.1			
Layer (type)	Output Shape	Param #	Connected to
batch_normalization_21 (BatchNormalization)	(None, 128, 128, 128)	512	concatenate_3[0][0]
activation_21 (Activation)	(None, 128, 128, 128)	0	batch_normalization_21[0][0]
conv2d_18 (Conv2D)	(None, 128, 128, 64)	73792	activation_21[0][0]
batch_normalization_22 (BatchNormalization)	(None, 128, 128, 64)	256	conv2d_18[0][0]
activation_22 (Activation)	(None, 128, 128, 64)	0	batch_normalization_22[0][0]
conv2d_19 (Conv2D)	(None, 128, 128, 64)	36928	activation_22[0][0]
batch_normalization_23 (BatchNormalization)	(None, 128, 128, 64)	256	conv2d_19[0][0]
activation_23 (Activation)	(None, 128, 128, 64)	0	batch_normalization_23[0][0]
conv2d_transpose_4 (Conv2DTrans	(None, 256, 256, 32)	8224	activation_23[0][0]
concatenate_4 (Concate- nate)	(None, 256, 256, 64)	0	activation_1[0][0]
			conv2d_transpose_4[0][0]
batch_normalization_24 (BatchNormalization)	(None, 256, 256, 64)	256	concatenate_4[0][0]
activation_24 (Activation)	(None, 256, 256, 64)	0	batch_normalization_24[0][0]
conv2d_20 (Conv2D)	(None, 256, 256, 32)	18464	activation_24[0][0]
batch_normalization_25 (BatchNormalization)	(None, 256, 256, 32)	128	conv2d_20[0][0]
activation_25 (Activation)	(None, 256, 256, 32)	0	batch_normalization_25[0][0]
conv2d_21 (Conv2D)	(None, 256, 256, 32)	9248	activation_25[0][0]
batch_normalization_26 (BatchNormalization)	(None, 256, 256, 32)	128	conv2d_21[0][0]
activation_26 (Activation)	(None, 256, 256, 32)	0	batch_normalization_26[0][0]
conv2d_22 (Conv2D)	(None, 256, 256, 1)	33	activation_26[0][0]

CURRICULUM VITAE



Name Surname : Berkay Özen

Place and Date of Birth : Soma / 24.02.1992

E-Mail : berkayozen@outlook.com

EDUCATION :

- **B.Sc.** : 2017, Istanbul Technical University, Civil Faculty, Department of Geomatics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 06.2019 - Continuing - Turk Ai - Software Developer.
- 04.2018 - 06.2019 - 4Maps Information Technologies Mapping - GIS Specialist.
- 09.2011 - 01.2014 - Epigra Information Technologies - FrontEnd Developer.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Alkan, Ş., Özen B., Erten, E., Musaoğlu N.,** (2017), "Şehir Gelişimlerinin Bilgi Teknolojileri Kullanarak Analizi: Google Earth Engine Örneği" – Poster Presentation *TUFUAB 2017*, April 27-29, 2017 Afyonkarahisar, Turkey.
- **Özen B., Işık M.S., Erol B.,** (2017) "Web-based Geoid Calculation Tool for Practical Height Transformation in Turkey" – Poster Presentation *ISGGG 2017*, October 18-21, 2017 Çanakkale, Turkey.
- **Özen B., Alkan Ş., Erten E., Musaoğlu N.,** (2017) "Monitoring City Growth Dynamics Using Cloud Computing Google Earth Engine" – Oral Presentation *ISGGG 2017*, October 18-21, 2017 Çanakkale, Turkey.