



Classification of Cabbage and Broccoli with Deep Learning Method for Robotic Harvesting Systems

Erhan Kahya^{1,a,*}, Fatma Funda Özdüven^{2,b}

¹Tekirdağ Namık Kemal University, Vocational College of Technical Sciences, Department of Electronic and Automation, Control and Automation Technology Program, Tekirdağ, Türkiye

²Tekirdağ Namık Kemal University, Vocational College of Technical Sciences, Department of Plant and Livestock Production, Greenhousing Program, Tekirdağ, Türkiye

*Corresponding author

| ARTICLE INFO | ABSTRACT |
|---|--|
| <p>Research Article</p> <p>Received : 10-06-2023 Accepted : 05-09-2023</p> <p>Keywords: Cabbage Broccoli Deep learning Classification Description</p> | <p>Classification of cabbage and broccoli using deep learning is very important in robotic harvesting systems. Deep learning is a machine learning method that allows learning complex models using artificial neural networks and large data sets. With the help of this method, it can be used effectively in plant classification and visual recognition problems. In order to classify plants such as cabbage and broccoli, a deep learning model must first be created. For this reason, Inception_v3 image recognition and classification modelling, which is one of the deep learning methods, was used in the study. The study was carried out over 2 classes. The created classes are cabbage and broccoli. The tpu hardware accelerator provided by Google Colab was used for training the model. The number of training cycles (epoch) is 10. The learning rate of 0.001 was determined as training parameters. According to these results, it was concluded that the Inception_v3 model was successful for training the broccoli and cabbage data set. During the training process, the loss value of the model gradually decreased and the accuracy value increased. In the validation phase, which is the last phase, the loss value was 0.0005 and the accuracy value was 1.0000.</p> |

Türk Tarım – Gıda Bilim ve Teknoloji Dergisi, 11(9): 1639-1647, 2023

Robotik Hasat Sistemlerinde Kullanılmak Amacıyla Lahana ve Brokolinin Derin Öğrenme Metodu ile Sınıflandırılması

| MAKALE BİLGİSİ | ÖZ |
|---|---|
| <p>Araştırma Makalesi</p> <p>Geliş : 10-06-2023 Kabul : 05-09-2023</p> <p>Anahtar Kelimeler: Lahana Brokoli Derin öğrenme Sınıflandırma Tanımlama</p> | <p>Robotik hasat sistemlerinde lahana ve brokolinin derin öğrenme kullanılarak sınıflandırılması oldukça önemlidir. Derin öğrenme, yapay sinir ağları ve büyük veri setleri kullanılarak karmaşık modellerin öğrenilmesine olanak sağlayan bir makine öğrenme yöntemidir. Bu yöntem yardımıyla bitki sınıflandırmasında ve görsel tanıma problemlerinde etkili bir şekilde kullanılabilir. Lahana ve brokoli gibi bitkilerin sınıflandırılması için öncelikle bir derin öğrenme modeli oluşturulması gerekmektedir. Bu nedenle yapılan çalışmada derin öğrenme yöntemlerinden olan Inception_v3 görüntü tanıma ve sınıflandırma modellemesi kullanılmıştır. Çalışma oluşturulan 2 sınıf üzerinden yürütülmüştür. Oluşturulan sınıflar lahana ve brokoli'dir. Modelin eğitimi için Google Colab'ın sağladığı tpu donanım hızlandırıcısı kullanılmıştır. Eğitim döngüsü (epoch) sayısı 10'dur. Eğitim parametreleri olarak öğrenme hızı 0,001 tespit edilmiştir. Bu sonuçlara göre brokoli ve lahana data setin eğitimi için Inception_v3 modelinin başarılı olduğu sonucuna varılmıştır. Eğitim sürecinde modelin kayıp değeri giderek düşmüş ve doğruluk değeri artmıştır. Son aşama olan doğrulama aşamasında kayıp değeri 0,0005, doğruluk değeri 1,0000 olarak gözlenmiştir.</p> |

^a ekahya@nku.edu.tr

^b <https://orcid.org/0000-0001-7768-9190>

^b fozduven@nku.edu.tr

^b <https://orcid.org/0000-0003-4286-8943>



Giriş

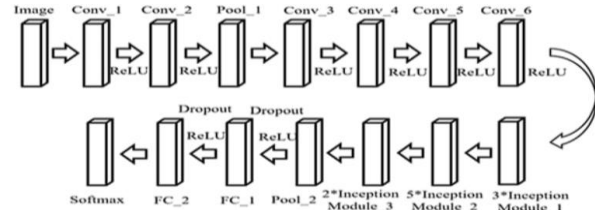
Gıda güvenliği, iklim değişikliği ve sürdürülebilir kalkınma, 21. yüzyılın en büyük zorlukları arasında yer almaktadır. Bu zorluklara yanıt olarak, bilim insanları ve mühendisler, tarım sektörünün verimliliğini ve sürdürülebilirliğini artırmak için yenilikçi çözümler bulmak için çaba sarf etmektedirler. Bu çözümler arasında, özellikle derin öğrenme teknolojisi dikkat çekmektedir. Çoklu soyutlama seviyelerinde yer alan verilerin öğrenilmesi için işleme katmanından oluşan modellere derin öğrenme izin vermektedir. Derin öğrenme, geri yayılım algoritmalarını kullanarak, karmaşık veri yapısını keşfetmek için dâhili parametrelerin nasıl kullanılması gerektiğini göstermektedir (Anonim1). Derin öğrenme, çok katmanlı yapay sinir ağları kullanılarak karmaşık özellikleri öğrenebilen bir makine öğrenme alt dalıdır.

Tarım, geniş veri kaynaklarına sahip bir sektördür. Toprak özellikleri, iklim koşulları, bitki büyüme evreleri ve hasat verimleri, tarımın çeşitli yönlerini anlamak için değerli veri sağlar. Son yıllarda, bu veri türlerinin toplanması ve analizi, uydu görüntüleri, hava durumu izleme istasyonları ve IoT cihazları gibi teknolojik gelişmelerle önemli ölçüde kolaylaşmıştır. Derin öğrenme, bu geniş ve çeşitli veri kaynaklarından anlamlı bilgiler çıkararak tarımsal süreçlerin ve özellikle hasat uygulamalarının iyileştirilmesine yardımcı olabilir. Hasat süreci, bir bitkinin veya ürünün tarladan toplanması ve işlenmesi aşamalarını içerir. Bu süreç, bitki türüne, iklim koşullarına ve tarımsal uygulamalara bağlı olarak önemli ölçüde değişebilir. Derin öğrenme, hasat sürecinin çeşitli yönlerini iyileştirmek için kullanılabilir. Ayrıca bitki hastalıklarını ve zararlıları tespit etmek içinde kullanılabilir. Oluşturulan model yardımıyla, hastalıklı bitkilerin veya zararlıların varlığını belirlemek için bitki görüntüleri analiz edebilir. Bu, hastalıkların veya zararlıların erken tespitini ve kontrolünü sağlayabilir, bu da hasat verimini ve kalitesini artırabilir. Derin öğrenme, hasat sürecini daha da otomatikleştirmek için kullanılabilir. Bu amaçla oluşturulan sistem ile bir hasat robotunun kontrolü yapılabilir. Model, bir bitkinin konumunu ve olgunluk derecesini belirleyebilir ve bu bilgiyi kullanarak robotun bitkiyi en etkili şekilde toplamasını sağlayabilir. Derin öğrenme modelleri ile zararlılarla mücadele için gereken pestisit ve diğer kimyasal maddelerin kullanımını azaltabilir ve aynı zamanda verim kayıplarını önleyebilir.

Derin öğrenme alanında Inception_v3 özellikle görüntü tanıma ve sınıflandırma problemleri için geliştirilmiş, oldukça etkili ve popüler bir sinir ağı modelidir. Inception_v3, öncülü olan Inception v1 ve v2 modellerinin bir devamıdır ve bu modellerin başarısını daha da ileriye taşımayı amaçlamaktadır. Inception ailesinin temel felsefesi, çeşitli ölçeklerde özellikleri yakalayabilme yeteneğine sahip bir ağ yapısı oluşturmaktır. Bu, modelin farklı ölçeklerdeki özellikleri öğrenme yeteneğini geliştirir ve modelin daha geniş bir bağlamı anlamasına yardımcı olur. Inception_v3, Google tarafından geliştirilmiş bir derin öğrenme mimarisidir. İlk olarak 2015 yılında ImageNet Large Scale Visual Recognition Challenge yarışmasında birinci olmuştur. Inception_v3, özellikle görüntü sınıflandırma problemlerinde başarılı sonuçlar veren bir sinir ağıdır. Bu mimari, evrişimli sinir ağı (CNN) tabanlıdır ve birçok katmandan oluşur. Girdi olarak bir

görüntü alır ve çıktı olarak görüntüdeki nesnelere sınıflandıran bir olasılık dağılımı üretir.

Inception_v3, özellikle büyük ölçekli görüntü tanıma ve sınıflandırma problemleri için etkili olan bir modeldir. Model, ImageNet veri setindeki görevlerde yüksek performans göstermiştir, bu da modelin geniş bir dizi görsel görevde etkili olduğunu göstermektedir. Şekil 1,2 ve 3'de Inception_v3 çalışma prensibi gösterilmiştir. Inception_v3, ayrıca kendi başına kullanılabilir gibi, daha karmaşık sistemlerin bir parçası olarak da kullanılabilir. Bu, modelin geniş bir dizi uygulamada kullanılabilmesini sağlar.



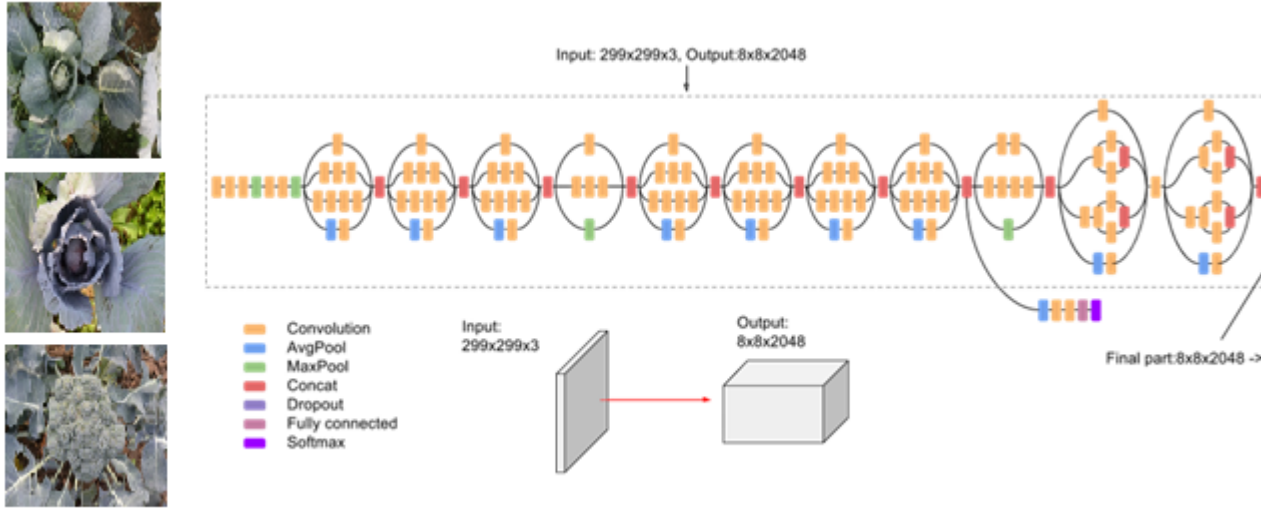
Şekil 1. Inception_v3 prensibi (Jiangchuan ve ark., 2019)
Figure 1. Inception_v3 principle (Jiangchuan et al., 2019)

| Type | Patch size/stride or remarks | Input size |
|-------------|------------------------------|------------|
| conv | 3×3/2 | 299×299×3 |
| conv | 3×3/1 | 149×149×32 |
| conv padded | 3×3/1 | 147×147×32 |
| pool | 3×3/2 | 147×147×64 |
| conv | 3×3/1 | 73×73×64 |
| conv | 3×3/2 | 71×71×80 |
| conv | 3×3/1 | 35×35×192 |
| 3×Inception | 5×5 | 35×35×288 |
| 5×Inception | 17×17 | 17×17×768 |
| 2×Inception | 8×8 | 8×8×1280 |
| pool | 8×8 | 8×8×2048 |
| linear | logits | 1×1×2048 |
| softmax | classifier | 1×1×1000 |

Şekil 2. Inception_v3 çalışma prensibi (Anonim 2)
Figure 2. Inception_v3 working principle (Anonymous 2)

Inception_v3 'te, ağız genişlik ve derinliği eniyenerek ağız maksimum bilgi akışı hedeflenmiştir. Derinlik arttıkça ağız genişliği de sistematik olarak artmaktadır. 5×5 ve 7×7 boyutlarında filtre yerine onları karşılayacak iki ya da üç tane 3×3 boyutlu filtreler kullanılmaktadır (Anonim 4). Özellikle, modelin tarım, tıbbi görüntüleme ve otonom sürüş gibi alanlarda başarılı bir şekilde uygulandığı görülmüştür. Bu, Inception_v3 'ün sadece bilgisayar görüşü problemlerini çözmek için değil, aynı zamanda geniş bir dizi alan ve uygulamada derin öğrenme çözümlerini etkin bir şekilde uygulamak için de kullanılabilirliğini göstermektedir.

Birçok alanda kullanılan Inception_v3, tarım alanında verimliliği artırmak, hastalıkları tespit etmek ve hasat süreçlerini optimize etmek gibi çeşitli uygulamalar için kullanılmaktadır. Özellikle Inception_v3, görüntü tanıma ve sınıflandırma yetenekleri sayesinde bitki hastalıkları ve zararlılarını tespiti, toprak ve su kalitesi değerlendirmesi ve verim tahmini gibi bir dizi görevde başarıyla uygulanmıştır.



Şekil 3. Inception_v3 çalışma prensibi(Anonim 3)
Figure 3. Inception_v3 working principle(Anonymous 3)

Bitki hastalıkları ve zararlılarının erken tespiti Inception_v3 için örnek uygulama alanları olarak verilebilir. Ayrıca, Inception_v3, tarım alanında uydu ve drone görüntülerinin analizinde de kullanılabilir. Model, bu tür görüntülerdeki özellikleri öğrenme ve tanıma yeteneğine sahiptir. Bu da modelin toprak ve su kalitesi değerlendirmesi, verim tahmini ve hasat planlaması gibi görevlerde kullanılabileceği anlamına gelir.

Tarımsal alanda derin öğrenme konusunda birçok araştırma yapılmış ve yapılmaya devam edilmektedir. Zhang ve ark. (2018)'nin yaptıkları çalışmada derin öğrenme tabanlı domates hasat robotu çalışması buna bir örnek olarak verilebilir. Yaptıkları çalışmada deneysel sonuçları, 0,01 saniyeden daha kısa bir tahmin süresiyle ortalama %91,9 doğruluk oranına ulaşmışlardır. Bir diğer çalışmada De Luna ve ark. (2019) derin öğrenme ile domates meyvesinin boyut sınıflandırması çalışması yapmışlardır. Oluşturdukları derin öğrenme yaklaşımı ile VGG16 için %82,31-%78,21-%55,97 eğitim doğrulama-test doğruluğu, VGG16 için %48,17-%41,44-%37,64 ile düşük performansları bulmuşlardır. Mu ve ark. (2020) yaptıkları çalışmada derin öğrenme yardımıyla domatesin fide üzerinde bulunmasını ve olgunluk derecelerinin tespitini yapmışlardır. Afonso ve ark. (2020) yapmış oldukları çalışmada sera için domates meyvesi için tespit sisteminde derin öğrenme metodunu kullanmışlardır.

Bu araştırmalar değerlendirildiğinde Inception_v3, tarım alanında çeşitli uygulamalar için kullanılabilir bir araç olabileceği düşünülmektedir. Çünkü model, karmaşık görüntü tanıma ve sınıflandırma problemlerini çözme yeteneğine sahiptir. Bu çalışmanın amacı, lahana ve brokoli bitkilerinin görüntülerini Inception_v3 mimarisini kullanarak robotik hasat sistemlerinde kullanılmak üzere sınıflandırma yapmaktır. Sınıflandırma, bir veri kümesini belirli kategorilere ayırma işlemidir. Bu çalışmada, lahana ve brokoli bitkileri iki farklı sınıf olarak kabul edilmiştir. Makine öğrenmesi, verilerden öğrenerek belirli görevleri otomatik olarak gerçekleştirmeyi sağlayan bir yapay zeka dalıdır. Yapılan çalışmada sınıflandırma için makine öğrenmesi algoritmalarından yararlanılmıştır.

Materyal ve Yöntem

Materyal

Bu projede, lahana ve brokoli bitkilerinin Python programlama dili ile görüntü işleme yaparak sınıflandırılması yapılmıştır. Brokoli (*Brassica oleracea var.italica*) Brassicaceae familyası sebzeleri arasında yer alan Avrupa ve Amerika'da geniş alanlarda yetiştirilen ve sevilerek tüketilen bir serin iklim sebzesidir. Brokkolinin üretim miktarı son yıllarda ülkemizde hızla artmaktadır (Eşiyok ve ark., 2010; Vural ve ark., 2000). Vitamin, protein ve mineral madde bakımından oldukça zengin olan brokkoli beslenme ve sağlık açısından önemlidir ve sahip olduğu düşük kalori nedeniyle iyi bir diyet sebzesi olarak tanımlanmaktadır. Ülkemizde brokoli Ege ve Marmara Bölgeleri'nde yaygın olarak üretilmekte taze, kurutulmuş, konserve ve dondurulmuş olarak değerlendirilmektedir (Bozokalfa ve ark., 2004). Brokoli yetiştiriciliğinde üretim tekniği kadar hasadın doğru zamanda yapılması taze-sofralık tüketimde özellikle sanayi sebzeciliğinde önemlidir. Hasadın gecikmesi, olgunlaşmamış çiçek taslaklarının açılması ve taçlardaki yeşil rengin sarıya dönmesi pazar kalitesini olumsuz yönde etkiler. Brokoli taçlarında yüksek miktarda klorofil bulunmaktadır ve yeşil rengi oluşturan bu pigment hasat sonrası dönemde parçalanarak yeşil sebzelerde bozulmanın ilk belirtisi olan yeşil rengin azalmasına ve sararmalara yol açmaktadır (Bozkalfa ve ark. 2010, Funamoto ve ark 2002). Beyaz baş lahana (*Brassica oleracea var. capitata*) Brassicaceae familyasında yer almaktadır. Bu lahana türünün sebze olarak değerlendirilen kısmı beyaz renkli başıdır. Beyaz baş sarmalık, turşuluk ve salata olarak sıkça tüketilmektedir (Tavalı ve ark.2014). Yetiştirilen farklı lahana türleri, yaprakların boyutu, şekli ve rengi ile başın boyutu, şekli, rengi ve dokusu bakımından büyük karakteristik farklılıklar gösterir. Bu da çeşitli iklim koşullarında yetiştirme potansiyeline sahip çok sayıda lahana çeşidinin ortaya çıkmasına neden olur. Hasat, fide dikiminde 2-5 ay sonra başlar (Kara ve Baktemür 2013). Lahana başlarının hasat olgunluğuna gelip gelmedikleri çok kolay anlaşılır. En önemli hasat göstergesi baş iriliği ve başın sıklığıdır. Çeşidin özelliğine göre baş belirli bir iriliğe

ulaştığı zaman hasat zamanı gelmiş demektir. Başlar tek tek kesilerek hasat genelde elle yapılır. Piyasaya sunulacak başlar bütün, sağlam, renk ve baş şekli bakımından homojen olmalı üzerinde herhangi bir hastalık ve zararlı belirtisi bulunmamalıdır (Şalk ve ark., 2008).

Görüntü işleme, fotoğraf ve video gibi görsel veriler üzerinde çeşitli işlemler yaparak bilgi elde etmeyi amaçlayan bir bilim dalıdır. Görüntü işleme teknikleri, yapay zeka, veri bilimi ve bilgisayar görüşü gibi alanlarda yaygın olarak kullanılmaktadır.

Bu çalışmada, veri setinin hazırlanmasında kullanılan lahana ve brokoli Tekirdağ Naip köyünde çekilmiş 176 lahana fotoğrafı ve 83 brokoli fotoğrafı üzerinde görüntü

işleme teknikleri uygulanmıştır. Ayrıca, OpenCV kütüphanesi kullanılmıştır. OpenCV, açık kaynaklı bir bilgisayar görüşü kütüphanesidir ve Python programlama dili ile kolayca entegre edilebilir. OpenCV ile resim dosyaları üzerinde renk dönüşümü, filtreleme, eşikleme, morfolojik işlemler, kenar tespiti, histogram oluşturma gibi işlemler yapılabilir.

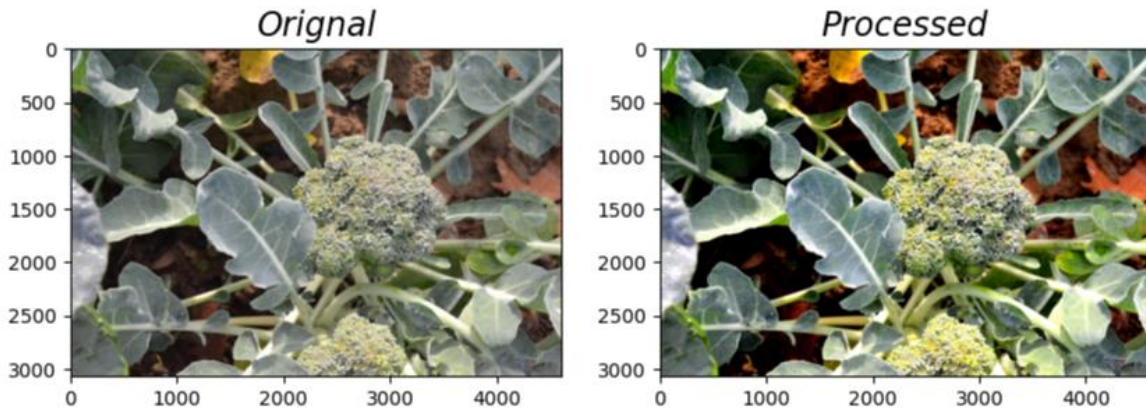
Yöntem

Veri kümesinin hazırlanması

Naip Köyünde lahana ve brokoli fotoğrafları çekilmiş ve etiketlenmiştir. Şekil 4'de Naip köyünde çekilmiş fotoğraflardan örnekler verilmiştir.



Şekil 4. Naip Köyünde tarlada çekilmiş fotoğraf örnekleri
Figure 4. Examples of photographs taken in the field in Naip Village



Şekil 5. Orjinal ve işlenmiş fotoğraf örneği
Figure 5. Example of original and processed image

Veri ön işleme

Fotoğrafların boyutları küçültülmüş, gri tonlamaya dönüştürülmüş ve histogram eşitleme yapılmıştır.

Kullanılan Dil ve Kütüphaneler

Bu çalışma Python programlama dilinde yazılmıştır. Veri analizi, görüntü işleme ve makine öğrenimi gibi alanlarda kullanılan kütüphaneleri içe aktararak ilerlenmiştir. Ayrıca, TensorFlow kütüphanesi ile bir

Inception_v3 modeli oluşturmak ve eğitmek için gerekli olan sınıfları ve fonksiyonları da içe aktarılmıştır.

Veri Seti

Lahana ve brokoli bitkilerinin farklı açılardan ve ışık koşullarından çekilmiş fotoğrafları çekilmiştir. Toplamda 176 lahana fotoğrafı ve 83 brokoli fotoğrafı elde edilmiştir. Fotoğrafların boyutları 512×512 piksel olarak ayarlanmıştır.

Derin Öğrenme Modeli

Bu projede Inception_v3 derin öğrenme mimarisi kullanılmıştır. Resim format olarak jpg kullanılmıştır.

Veri ön işleme

Çalışmada fotoğrafların renk kanalları RGB olarak belirlenmiştir. Fotoğrafların %80'i eğitim verisi, %10'u doğrulama verisi ve %10'u test verisi olarak ayrılmıştır. Eğitim verisi üzerinde veri artırma (data augmentation) yöntemleri uygulanmıştır. Bu yöntemler yatay çevirme (horizontal flip), dikey çevirme (vertical flip), döndürme (rotation), yakınlaştırma (zoom) ve kaydırma (shift) dir. Şekil 5'de brokoliye ait orijinal ve işlenmiş resim gösterilmiştir.

Model oluşturma

Eğitim aşamasında, Inception_v3 modelinin son katmanı ile Dense, Flatten, GlobalAverage Pooling2D, Dense ve Dropout katmanları eklenmiştir. Bu katmanların amacı, modelin öznetelik haritalarını düzleştirmek, ortalamak, sınıflandırmak ve aşırı uyumu önlemektir.

Eğitimin yapıldığı ortam ve özellikleri

Modelin eğitimi için Google Colab'ın sağladığı tpu donanım hızlandırıcısı kullanıldı. Bu sayede modelin eğitimi daha hızlı ve verimli gerçekleştirildi.

- Eğitim döngüsü (epoch) sayısı olarak 10
- Küçük grup (batch) boyutu olarak 8
- Optimizasyon algoritması olarak Adam
- Kayıp fonksiyonu olarak kategorik çapraz entropi (categorical cross entropy)
- Başarı metriği olarak doğruluk (accuracy)

Model eğitimi

Model eğitimi dört kısımdan oluşmaktadır. Kodun birinci kısmı Inception_v3 modelini yüklemekte ve giriş olarak 512×512×3 boyutunda bir görüntü alacak şekilde ayarlanmaktadır. Ayrıca, modelin üst kısmını (son tam bağlantılı katman) dahil etmemekte ve modelin ağırlıklarını ImageNet üzerinde eğitilmiş olanlara ayarlanmaktadır. Modelin katmanlarının eğitilebilirliği ise False olarak belirlenmektedir. Yani modelin ağırlıkları değiştirilmemektedir. Kodun ikinci kısmı, Inception_v3 modelinin çıkışına bazı ek katmanlar ekler. Önce GlobalAveragePooling2D katmanı ile modelin çıkışındaki öznetelik haritalarının ortalaması alınır ve 2 boyutlu bir tensöre indirgenir. Sonra Dense katmanı ile bu tensör 128 nöronlu bir tam bağlantılı katmana bağlanır ve ReLU aktivasyon fonksiyonu uygulanır. Sonraki adımda Dropout katmanı ile bu katmandaki nöronların %20'si rastgele devre dışı bırakılır ve aşırı uyum önlenir. Son olarak, Dense katmanı ile bu katmandaki nöronlar 2 nöronlu bir tam bağlantılı katmana bağlanır ve softmax aktivasyon fonksiyonu uygulanır. Bu son katman, görüntünün iki sınıftan birine ait olma olasılıklarını verir.

Kodun üçüncü kısmı, yeni oluşturulan modeli derler ve kayıp fonksiyonu olarak kategorik çapraz entropi, optimizasyon algoritması olarak adam ve metrik olarak doğruluk kullanır. Kodun dördüncü kısmı, eğitim ve test verilerini yükler ve ön işleme yapar. ImageDataGenerator sınıfını kullanarak görüntülerin piksel değerlerini 0-1 arasına ölçeklendirir ve eğitim verilerine bazı veri artırma teknikleri uygular (kesme, yakınlaştırma, yatay çevirme). flow_from_directory metodu ile eğitim ve test klasörlerindeki görüntüler okunur ve hedef boyut olarak (512, 512) belirlenir. Ayrıca sınıf modu olarak kategorik

seçilir, yani her görüntünün hangi sınıfa ait olduğunu belirten bir vektör oluşturulur.

Eğitim parametreleri olarak öğrenme hızı 0,001, kayıp fonksiyonu categorical_cross entropy ve optimizasyon algoritması olarak Adam seçilmiştir. Eğitim süresi 10 epoch'dur.

```
Python'a eşit eğitim için gerekli olan kütüphaneler
import numpy as np
import pandas as pd
import seaborn as sns
from PIL import Image
from PIL import ImageEnhance
from skimage.io import imread
import matplotlib.pyplot as plt

import os, random, pathlib, warnings, itertools, math
warnings.filterwarnings("ignore")

import tensorflow as tf
import tensorflow.keras.backend as K
from sklearn.metrics import confusion_matrix

from tensorflow.keras import models
from tensorflow.keras.models import Model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import InceptionV3, preprocess_input
from tensorflow.keras.layers import Dense, Flatten, GlobalAveragePooling2D, Dense, Dropout

K.clear_session()
```

```
dataset = './archive'

train_folder = os.path.join(dataset, "train")
test_folder = os.path.join(dataset, "validation")
validation_folder = os.path.join(dataset, "test")
```

```
def count_files(rootdir):
    '''counts the number of files in each subfolder in a directory'''
    for path in pathlib.Path(rootdir).iterdir():
        if path.is_dir():
            print("There are " + str(len([name for name in os.listdir(path) \
                if os.path.isfile(os.path.join(path, name))])) + " files in " \
                + str(path.name))

count_files(os.path.join(train_folder))
```

```
İşlemlere bazı işlemler yaparak (keskinleştirme, renk ayarı vb.) görüntü
bu işlemler için gerekli kod hücreleri
image_folder = "Broccoli" # Gösterilmesini istenen data
number_of_images = 2 # Gösterilecek Adet

def Preprocess():
    j=1
    for i in range(number_of_images):
        folder = os.path.join(test_folder, image_folder)
        a = random.choice(os.listdir(folder))

        image = Image.open(os.path.join(folder, a))
        image_duplicate = image.copy()
        plt.figure(figsize=(10,10))

        plt.subplot(number_of_images, 2, j)
        plt.title(label="Original", size=17, pad="7.0", loc="center", fontstyle="italic")
        plt.imshow(image)
        j+=1

        image1 = ImageEnhance.Color(image_duplicate).enhance(1.35)
        image1 = ImageEnhance.Contrast(image1).enhance(1.45)
        image1 = ImageEnhance.Sharpness(image1).enhance(2.5)

        plt.subplot(number_of_images, 2, j)
        plt.title(label="Processed", size=17, pad="7.0", loc="center", fontstyle="italic")
        plt.imshow(image1)
        j+=1

Preprocess()
```

Ağın Eğitilmesi

Model eğitim dosyaları tek tek program içine eklenmiştir. Oluşturulan yeni kod hücreleri yardımıyla ağın eğitilmesi yapılmıştır. Yeni kod eklemesini program ekranının orta sekmesinde bulunan "+Kod" sekmesi seçilerek tek tek girilmiştir. Her kod program ekranında bulunan ok işareti ile sırası ile çalıştırılmıştır. Oluşturulan model Python'un bir kütüphanesi olan Tensorflow ile Keras Modülü kullanarak eğitilmiştir.

Yazılmış olan kod satırları eğitim aşamasında her bir materyal (lahana ve brokoli) için tek tek çalıştırılmıştır. Modellerin eğitiminde alınan değerler bilgisayar ortamında kaydedildikten sonra eğitimin test aşaması için "Prediction(tahmin)" değerleri oluşturulmuştur.

Araştırma Bulguları ve Tartışma

Inception_v3 modeli, tarım sektöründe görüntü analizi ve veri işleme konularında önemli bir araç olarak kullanılabilecek potansiyele sahiptir. Hastalık teşhisi, verim tahmini, zirai ilaç kullanımı ve hasat zamanı belirleme gibi alanlarda, tarımın sürdürülebilirliği ve verimliliği için önemli katkılar sağlayabilir. Milioto ve ark. (2018) yapmış oldukları bir çalışmada oluşturdukları sistemlerini Almanya ve İsviçre'deki farklı tarlalarda çalışan bir tarım robotunda uygulamış ve değerlendirmişlerdir. Inception_v3 modelinin gerçek zamanlı olarak tarım ürünlerini ve yabancı otları etkili bir şekilde sınıflandırabildiğini göstermişlerdir. Sistem, yaklaşık 20 Hz hızında çalışmakta ve iyi bir genelleme yeteneği sergilemektedir. Sladojevic ve ark. (2016) yılında Inception_v3 modeli ile bitki hastalıklarının tespiti ve teşhisini yapmışlardır. Benzer bir çalışma Ferentinos (2018) yapmıştır. Zhou ve ark. (2020) yaptıkları derin öğrenme ile Inception_v3 çalışmalarında, akıllı telefonlarda kullanılmak üzere gerçek zamanlı kiwi tespit çalışmalarında Inception_v3 modellemesini kullanmışlardır. Inception_v3 hayvan hastalıklarının tespitinde de kullanılan bir modeldir. Zhuang ve ark. (2019) yaptıkları çalışmada bir sürüdeki hasta piliçleri otomatik olarak oluşturdukları Inception_v3 modeli ile başarıyla tespit etmişlerdir. Bu konuda birçok çalışma yapılmaya devam etmektedir. Ancak, modelin uygulamada kullanılması, hassas veri toplama ve modelin yerleştirilmesi gibi zorlukları da beraberinde getirebilir. Bu nedenle, Inception_v3 gibi derin öğrenme modellerinin tarım alanında kullanımı, dikkatli planlama ve uygulama gerektiren bir süreç içerir.

Yaptığımız çalışma sonunda araştırma bulguları için ilk olarak, Preprocess fonksiyonu veri ön işleme adımları gerçekleştirir. Belirli bir sebze (brokoli) için belirli bir sayıda (2) rastgele görüntü seçer ve her bir görüntünün orijinalini ve işlenmiş halini yan yana gösterir. Select vegetable değişkeni "lahana" olarak tanımlanır ve belirli bir sebze için birkaç örnek gösteren bir grafik oluşturmak için display_folder adlı bir klasör tanımlanır. Train_folder adlı klasördeki tüm sebzelerin rastgele bir örneğini seçip bunları images adlı bir listede depolar.

Son adım olarak, Inception_v3 modeli yüklenmiştir. Eğitim ve test veri kümeleri tanımlanmıştır. Eğitim veri seti, train_folder içindeki klasörlere göre sınıflandırılmış resimleri kullanarak modeli eğitir. Test veri seti, test_folder içindeki klasörlere göre sınıflandırılmış resimlerle modelin performansını değerlendirir. Doğrulama klasöründeki resimleri modelin tahminleriyle karşılaştırarak gerçek ve tahmin edilen etiketleri döndürür. Bu fonksiyonda, doğrulama klasöründeki alt klasörlerin isimleri lahana ve brokoli türlerini temsil eder ve bunlara sayısal bir eşleme atanır. Ardından her alt klasördeki resimler model tarafından sınıflandırılır ve gerçek ve tahmin edilen etiketler listelere eklenir. Gerçek ve tahmin edilen etiketleri kullanarak bir karışıklık matrisi oluşturur ve bunu görselleştirir. Karışıklık matrisi, modelin her sınıfı ne kadar doğru tahmin ettiğini gösterir. Bu fonksiyonda, toplam çıktı etiketi sayısı 2 olarak belirlenir. Oluşturulan sınıflar lahana ve brokoli olarak iki sınıftır. Daha sonra renk haritası ve çizim boyutu gibi parametreler ayarlanır. Son adım, karışıklık matrisi kaydedilir ve ekranda gösterilir. Şekil 6'da karışık matris sonuç ekranı gösterilmiştir. Şekil 7'de ürün tahmin ekran örnekleri verilmiştir.

```
select_vegetable="Broccoli"
rows,columns = 1,5

display_folder=os.path.join(train_folder,select_vegetable)
total_images=rows*columns
fig=plt.figure(1,figsize=(20,10))

for i,j in enumerate(os.listdir(display_folder)):

    img = plt.imread(os.path.join(train_folder,select_vegetable,j))
    fig=plt.subplot(rows,columns,i+1)
    fig.set_title(select_vegetable,pad=11,size=20)
    plt.imshow(img)

    if i==total_images-1:
        break
```

```
images = []

for food_folder in sorted(os.listdir(train_folder)):
    food_items = os.listdir(train_folder+'/'+food_folder)
    food_selected = np.random.choice(food_items)
    images.append(os.path.join(train_folder,food_folder,food_selected))

fig=plt.figure(1,figsize=(15,10))

for subplot,image in enumerate(images):
    category=image.split('/')[-2]
    imgs = plt.imread(image)
    a,b,c=imgs.shape
    fig=plt.subplot(3,5,subplot+1)
    fig.set_title(category,pad=10,size=18)
    plt.imshow(imgs)

plt.tight_layout()
```

```
IMAGE_SIZE = (224, 224)

inception = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet',
                        include_top=False)

for layer in inception.layers:
    layer.trainable = False

x = inception.output
x = GlobalAveragePooling2D()(x)
x = Dense(128,activation='relu')(x)
x = Dropout(0.2)(x)

prediction = Dense(8, activation='softmax')(x)

model = Model(inputs=inception.input, outputs=prediction)

model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

```
train_datagen = image.ImageDataGenerator(rescale = 1./255,
                                          shear_range = 0.2,
                                          zoom_range = 0.2,
                                          horizontal_flip = True)

test_datagen = image.ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory(
    train_folder,
    target_size = (224, 224),
    batch_size = 32,
    class_mode = 'categorical')

test_set = test_datagen.flow_from_directory(
    test_folder,
    target_size = (224, 224),
    batch_size = 32,
    class_mode = 'categorical')
```

```
class_map = training_set.class_indices
class_map
```

```
#Burada ki "epochs" dotaları tekrar tekrar devir ederek daha sağlıklı sonuçlar elde etmek için kullanılan ayar. İsteğe göre ayarlanabilir.
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=20,
    steps_per_epoch=len(training_set),
    validation_steps=len(test_set))
```

```
#Modeli kaydetmek için kullanışlı komut
model.save('drive/MyDrive/Proje/python/modelson.h5')
```

```
K.clear_session()
path_to_model='drive/MyDrive/Proje/python/model.h5'
print("Loading the model..")
model = load_model(path_to_model)
print("Done!")
```

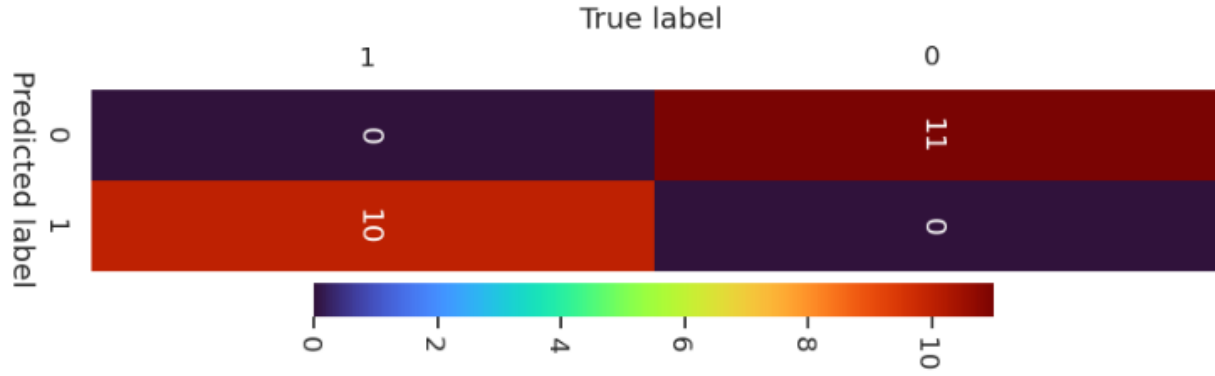
```
category={0: 'Bean', 1: 'Broccoli', 2: 'Cabbage', 3: 'Carrot', 4: 'Cauliflower', 5: 'Cucumber', 6: 'Potato', 7: 'Tomato'}

def predict_image(filename,model):
    img = image.load_img(filename, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_processed = np.expand_dims(img_array, axis=0)
    img_processed /= 255.

    prediction = model.predict(img_processed)
    index = np.argmax(prediction)

    plt.title("Prediction - {}".format(category[index]))
    plt.imshow(img_array)

predict_image(os.path.join(validation_folder,'Cauliflower/1050.jpg'),model)
```



Şekil 6. Karışık matris ekran sonucu
Figure 6. Mixed matrix display result



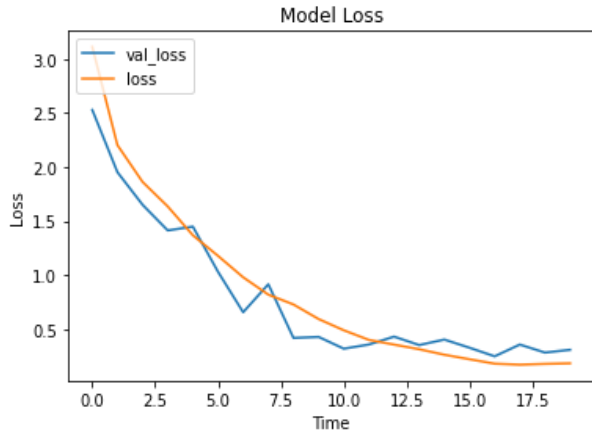
Şekil 7. Tahmin sonuç ekranları
Figure 7. Prediction result screens

Model Loss, bir makine öğrenmesi modelinin ne kadar iyi performans gösterdiğini ölçen bir metriktir. Model Loss, modelin tahminlerinin gerçek değerlerden ne kadar farklı olduğunu gösterir. Model Loss ne kadar düşükse, model o kadar iyi öğrenmiş anlamındadır. Model Loss'un hesaplanması için çeşitli yöntemler vardır. Bu yöntemlerden bazıları regresyon problemlerinde ortalama karesel hata (MSE) veya ortalama mutlak hata (MAE) kullanılabilir. Sınıflandırma problemlerinde ise çapraz entropi (cross-entropy) veya doğruluk (accuracy) gibi metrikler tercih edilebilir. Model Loss'un değerini azaltmak için modelin eğitim sürecinde optimizasyon algoritmaları kullanılır. Bu algoritmalar, modelin ağırlıklarını veya parametrelerini, Loss fonksiyonunun minimum noktasına yaklaştıracak şekilde günceller. Model Loss'un değeri, modelin genelleme yeteneğini de etkiler. Eğer model Loss'unu çok fazla azaltırsa, veriye

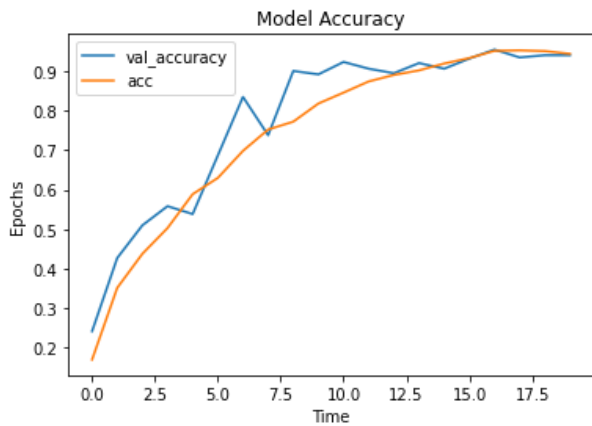
aşırı uyum (overfitting) yapabilir ve yeni verilerde kötü sonuçlar verebilir. Bu nedenle, model Loss'unun hem eğitim hem de test verileri üzerinde uygun bir seviyede olmasına dikkat edilmelidir. Şekil 8'de Model kayıp (Loss) sonuç ekran grafiği gösterilmiştir.

Model Accuracy, bir makine öğrenmesi modelinin verilen bir veri kümesindeki etiketleri ne kadar doğru tahmin ettiğini ölçen bir metriktir. Model Accuracy, modelin doğru tahmin sayısını toplam tahmin sayısına bölerek hesaplanır. Model Accuracy, sınıflandırma problemleri için kullanılan yaygın bir performans ölçütüdür. Şekil 9'da model doğruluk (Accuracy) sonuç grafik ekranı gösterilmiştir.

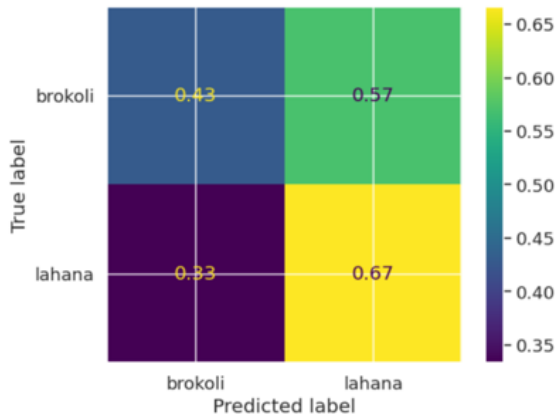
Eğitimin Brokoli, Lahana için yapılmış normalize edilmiş Confusion Matrix değerleri bulunmuştur. Şekil 10'da tahmin sonuç ekranı gösterilmiştir.



Şekil 8. Model kayıp ekran grafiği
Figure 8. Model loss screen graph



Şekil 9. Model doğruluk grafik ekranı
Figure 9. Model accuracy graph screen



Şekil 10. Tahmin ekranı
Figure 10. Forecast screen

Eğitim döngüsü sonuçları;

1. *Epoch*: İlk eğitim aşamasında, kayıp değeri 0,3178, doğruluk değeri 0,8589 olarak bulundu. Doğrulama aşamasında kayıp değeri 0,0607, doğruluk değeri 1,0000 olarak belirlendi.

2. *Epoch*: Kayıp değeri 0,0301, doğruluk değeri 1,0000 oldu. Doğrulama aşamasında kayıp değeri 0,0124, doğruluk değeri 1,0000 olarak gözlemlendi.

3. *Epoch*: Kayıp değeri 0,0122, doğruluk değeri 1,0000 olarak bulundu. Doğrulama aşamasında kayıp değeri 0,0050, doğruluk değeri 1,0000 olarak saptandı.

4. *Epoch*: Kayıp değeri 0,0095, doğruluk değeri 1,0000 olarak tespit edildi. Doğrulama aşamasında kayıp değeri 0,0033, doğruluk değeri 1,0000 olarak belirlendi.

5. *Epoch*: Kayıp değeri 0,0099, doğruluk değeri 0,9959 olarak elde edildi. Doğrulama aşamasında kayıp değeri 0,0027, doğruluk değeri 1,0000 olarak gözlemlendi.

6. *Epoch*: Kayıp değeri 0,0032, doğruluk değeri 1,0000 olarak bulundu. Doğrulama aşamasında kayıp değeri 0,0012, doğruluk değeri 1,0000 olarak saptandı.

7. *Epoch*: Kayıp değeri 0,0021, doğruluk değeri 1,0000 olarak belirlendi. Doğrulama aşamasında kayıp değeri 0,0009, doğruluk değeri 1,0000 olarak gözlemlendi.

8. *Epoch*: Kayıp değeri 0,0019, doğruluk değeri 1,0000 olarak bulundu. Doğrulama aşamasında kayıp değeri 0,0008, doğruluk değeri 1,0000 olarak saptandı.

9. *Epoch*: Kayıp değeri 0,0016, doğruluk değeri 1,0000 olarak tespit edildi. Doğrulama aşamasında kayıp değeri 0,0007, doğruluk değeri 1,0000 olarak belirlendi.

10. *Epoch*: Kayıp değeri 0,0012, doğruluk değeri 1,0000 olarak elde edildi.

Doğrulama aşamasında kayıp değeri 0,0005, doğruluk değeri 1,0000 olarak gözlemlendi. Eğitim sürecinde modelin kayıp değeri giderek düşmüş ve doğruluk değeri artmıştır. Doğrulama aşamasında ise sonuçlar istenilen düzeyde olup, kayıp değeri düşük ve doğruluk değeri yüksektir.

Sonuç

Robotik sistemlerde en önemli parametre ürünlerin doğru sınıflandırılmasıdır. Verilen özelliklere göre hasadı yapılacak ürünün tespitinin başarı oranı artacaktır. Seo ve ark. (2021) yaptıkları derin öğrenme ile sınıflandırma çalışmasında domates serası için sınıflandırma sisteminde tamamen gizlenmiş meyveler yakalanmadığında %88,6 tespit doğruluğuna ulaşmışlardır. Gizlenmiş meyveler hariç tutulduğunda, sistemin doğruluk oranı %90,2 olarak tespit etmişlerdir. Mutha ve ark. (2021) derin öğrenme ile domates olgunluk derecesi çalışmalarında olgunlaşma için doğruluk değerini %99,2, olgunlaşmamış için %94,34 ve hasarlı ürün için %90,23 olarak bulmuşlardır. Yapılan bu benzer araştırmalar göz önüne alındığında yaptığımız araştırmanın sonuçları ile paralellik gösterdiği tespit edilmiştir. Bulunan sonuçlar, brokoli ve lahana içeren data setin eğitimi için Inception_v3 modelinin başarılı olduğunu göstermektedir. Yapılan çalışmada kullanılan modelin lahana ve brokoli bitkilerinin sınıflandırılmasında başarılı sonuçlar verdiği görülmüştür. Ayrıca bu yöntemlerin diğer bitki türleri için de uyarlanabileceği düşünülmektedir. Eğitim setinde kullanılan resimlerin çoğaltılması ve çeşitlendirilmesinin doğruluk değerinin artmasına sebep olacağı anlaşılmıştır.

Kaynaklar

- Afonso M, Fonteijn H, Foirentin FS, Lensink D, Mooij M, Faber N, Polder G, Wehrens R. 2020. Tomato Fruit Detection And Counting in Greenhouses Using Deep Learning. *Frontiers in Plant Science*, 11, doi: 10.3389/fpls.2020.571299
- Anonim 1. 2023. <http://acikerisim.aksaray.edu.tr/xmlui/bitstream/handle/2022.2/0.500.12451/9522/foto-ozgur-2022.pdf?sequence=1>, Erişim tarihi: 08.06.2023
- Anonim 2. 2023. https://pytorch.org/hub/pytorch_vision_inception_v3, Erişim tarihi: 08.06.2023
- Anonim 3. 2023. <https://cloud.google.com/tpu/docs/inception-v3-advanced>, Erişim tarihi: 08.06.2023

- Anonim 4. 2023. <https://openaccess.hacettepe.edu.tr/xmlui/bitstream/handle/11655/4536/10192378.pdf?sequence=1>,Erişim tarihi:08.06.2023
- Bozokalfa MK, Eşiyok D, Yoltaş T, Koçak M. 2004. Bazı Brokkoli Çeşitlerinin Verim Kalite ve Teknolojik Özelliklerinin Belirlenmesi. V. Sebze Tarımı Sempozyumu 21-24 Eylül Çanakkale.
- De Luna RG, Dadios EP, Bandala AA, Vicerra RRP. 2019. Size classification of tomato fruit using thresholding, machine learning and deep learning techniques. *Agrivita*, 41(3), 586–596, doi: 10.17503/agrivita.v41i3.2435
- Eşiyok D, Salman MH, Bozokalfa MK, Şen F, Aşçıoğlu Kaygısız T. 2010. Bazı Brokkoli Çeşitlerinde Raf Ömrü Süresince Kalite Değişimlerinin Belirlenmesi, *Ege Üniv. Ziraat Fak. Derg.*, 2010, 47 (1): 79-86 ISSN 1018 – 8851
- Funamoto Y, Yamauchi N, Shigenaga T, Shigyo M. 2002. Effects of heat treatment on chlorophyll degrading enzymes in stored broccoli (*Brassica oleracea* L.) *Postharvest Biology and Technology* 24: 163-170.
- Jiangchuan L, Mantao W, Lie B, Xiaofan L, Jun S, Yue M. 2019. Classification and recognition of turtle images based on convolutional neural network, *IOP Conference Series: Materials Science and Engineering*, Volume 782, 4. Sustainability and Environmental Protection, Citation Jiangchuan Liu et al 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* 782 052044,doi: 10.1088/1757-899X/782/5/052044
- Kara E, Baktömür G. 2023. Sebze yetiştiriciliği. Iahana (*Brassica oleracea* L.) Yetiştiriciliği S:203-217, İksad Yayınları., ISBN: 978-625-6404-83-0
- Konstantinos P, Ferentinos KP. 2018. Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture*, Vol: 145,Pages:311-318. doi:10.1016/j.compag.2018.01.009
- Milioto A, Lottes P, Stachiniss C. 2018. Real-time semantic segmentation of crop and weed for precision, *Agriculture Robots Leveraging Background Knowledge in CNNs.*, doi:10.1109/icra.2018.8460962
- Mu Y, Chen TS, Ninomiya S, Guo W. 2020. Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors*, 20(10), 2984, doi:10.3390/s20102984
- Mutha SA, Shah AM, Ahmed MZ. 2021.Maturity detection of tomatoes using deep learning. *SN Computer Science*, 2(6), 441, doi: 10.1007/s42979-021-00837-9
- Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence And Neuroscience*, doi:10.1155/2016/3289801
- Seo D, Cho BH, Kim K. 2021. Development of monitoring robot system for tomato fruits in hydroponic greenhouses. *Agronomy*, 11(11), 2211, doi: 10.3390/agronomy11112211
- Şalk A, Arın L, Deveci M, Polat S. 2008. Özel Sebzeçilik, Namık Kemal Üniversitesi Ziraat Fakültesi Bahçe Bitkileri Bölümü, s:488, Tekirdağ.
- Tavalı İE, Maltaş AŞ, Uz İ, Kaplan M. 2014. Vermikompostun beyaz baş lahananın (*brassica oleracea* var. *Alba*) verim, kalite ve mineral beslenme durumu üzerine etkisi. *Akdeniz Univ. Ziraat Fak. Derg.* 27(1): 61-67
- Vural H, Eşiyok D, Duman İ. 2000.Kültür sebzeleri (Sebze Yetiştirme) 440s.ISBN:975-90790-0-2.
- Zhou Z, Song Z, Fu L, Gao F, Li R, Cui Y. 2020. Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. *Computers and Electronics in Agriculture*, Vol:179, doi:10.1016/j.compag.2020.105856
- Zhuang X, ZhangT. 2020. Detection of sick broilers by digital image processing and deep learning, *Biosystems Engineering*,Volume 179, March 2019, Pages 106-116,doi: 10.1016/j.biosystemseng.2019.01.003
- Zhang L, Jia J, Gui G, Hao X, Gao W, Wang M. 2018. Deep learning based improved classification system for designing tomato harvesting robot. *IEEE Access*, 6, 67940-67950, doi: 10.1109/access.2018.2879324