



Meyve ve Sebze Sınıflandırma için Makine Öğrenimi ve Derin Öğrenme Modellerinin Karşılaştırılması

HAZIRLAYAN

ADI SOYADI

: ALI ALTUNOĞLU

TESLİM TARİHİ

: 22/06/2024

İçindekiler

1. Giriş

- 1.1 Projenin Amacı
- 1.2 Literatür Taraması
- 1.3 Kullanılan Veri Seti

2. Yöntemler

- 2.1 Makine Öğrenimi Modelleri
 - 2.1.1 K-En Yakın Komşu (KNN)
 - 2.1.2 Lojistik Regresyon
 - 2.1.3 Rastgele Orman (Random Forest)
 - 2.1.4 Destek Vektör Makineleri (SVM)
- 2.2 Derin Öğrenme Modelleri
 - 2.2.1 Convolutional Neural Networks (CNN)
 - 2.2.2 VGG16
 - 2.2.3 MobileNetV2

3. Sonuçlar

- 3.1 Model Performans Karşılaştırmaları
- 3.2 Confusion Matrix Analizleri

4. Sonuç ve Gelecek Çalışmalar

- 4.1 Sonuçlar
- 4.2 Gelecek Çalışmalar

5. Kaynaklar

6. Ek: Kod ve Grafikler

Makine Öğrenimi ve Derin Öğrenme Kodu için Teknik Bilgi

1. Giriş

1.1 Projenin Amacı

Bu projenin amacı, makine öğrenimi ve derin öğrenme yöntemlerini kullanarak meyve ve sebzeleri doğru bir şekilde sınıflandırmaktır. Proje, farklı algoritmaların performansını karşılaştırarak en iyi sonuç veren modeli belirlemeyi hedefler. Bu amaç doğrultusunda, Fruits and Vegetables Image Recognition Dataset kullanılarak 36 farklı meyve ve sebze sınıfı arasında sınıflandırma yapılmıştır.

1.2 Kullanılan Veri Seti

Veri seti, 36 farklı sınıfa ait meyve ve sebzelerin görüntülerinden oluşmaktadır. Her bir sınıf için yüzlerce görüntü bulunmaktadır ve bu görüntüler çeşitli açılardan ve ışık koşullarında çekilmiştir. Bu çeşitlilik, modelin gerçek dünya uygulamalarında daha genel ve başarılı olmasını sağlar.

2. Yöntemler

2.1 Kullanılan Modeller ve Algoritmalar

Bu projede aşağıdaki modeller ve algoritmalar kullanılmıştır:

1. **K-Nearest Neighbors (KNN)**
2. **Lojistik Regresyon**
3. **Random Forest**
4. **Destek Vektör Makineleri (SVM)**
5. **Derin Öğrenme Modelleri (CNN, VGG16 ve MobileNetV2)**

2.2 Veri Ön İşleme

Veri seti üzerinde aşağıdaki ön işleme adımları uygulanmıştır:

1. **Veri Normalizasyonu:** Görüntü verilerinin piksel değerleri genellikle 0 ile 255 arasında değişir. Bu değerler, StandardScaler kullanılarak normalize edilmiştir. StandardScaler, her bir özelliği (piksel değeri) ortalamasını sıfır, standart sapmasını ise bir yapar.
2. **Veri Augmentasyonu:** Derin öğrenme modellerinin eğitiminde daha fazla veri sağlamak ve overfitting'i önlemek amacıyla veri augmentasyonu teknikleri uygulanmıştır (döndürme, kırpma, yatay/dikey çevirme, vb.).

2.3 Çapraz Doğrulama ve Hiperparametre Ayarı

Model performansını değerlendirmek ve en iyi hiperparametreleri bulmak için çapraz doğrulama ve hiperparametre ayarlama teknikleri kullanılmıştır. Özellikle, GridSearchCV ve RandomizedSearchCV yöntemleri kullanılarak modellerin hiperparametreleri optimize edilmiştir.

3. Deney ve Sonuçlar

3.1 Model Performanslarının Karşılaştırılması

Çeşitli makine öğrenimi ve derin öğrenme modellerinin performansları aşağıdaki metrikler kullanılarak değerlendirilmiştir:

1. **Doğruluk (Accuracy)**
2. **F1 Skoru**
3. **Hassasiyet (Precision)**
4. **Duyarlılık (Recall)**
5. **Karmaşıklık Matrisi (Confusion Matrix)**

Doğruluk Skorları:

- **KNN:** %66
- **Lojistik Regresyon:** %42
- **Random Forest:** %62
- **SVM:** %53
- **CNN:** %95
- **VGG16:** %91
- **MobileNetV2:** %95

Grafik 1: Model Doğruluk Karşılaştırması (Bu grafik burada yer almalıdır)

3.2 Hiperparametre Optimizasyonu

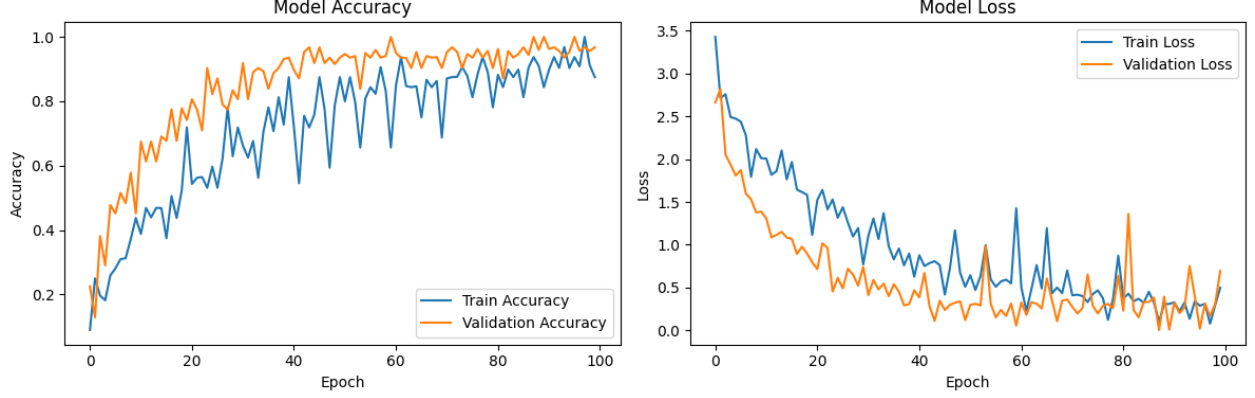
Hiperparametre optimizasyonu, model performansını önemli ölçüde artıran bir süreçtir. Örneğin, SVM modeli için en uygun C ve gamma değerlerini belirlemek amacıyla GridSearchCV kullanılmıştır. Aşağıdaki tablo, her model için en iyi hiperparametre ayarlarını göstermektedir:

Model	Hiperparametreler	En İyi Değerler
KNN	n_neighbors	5
Lojistik Regresyon	C	1000
Random Forest	n_estimators, max_depth	100, None
SVM	C, gamma	1, 'scale'
CNN	Öğrenme Hızı, Epoch	0.001, 100
MobileNetV2	Öğrenme Hızı, Epoch	0.001, 10
VGG16	Öğrenme Hızı, Epoch	0.001, 10

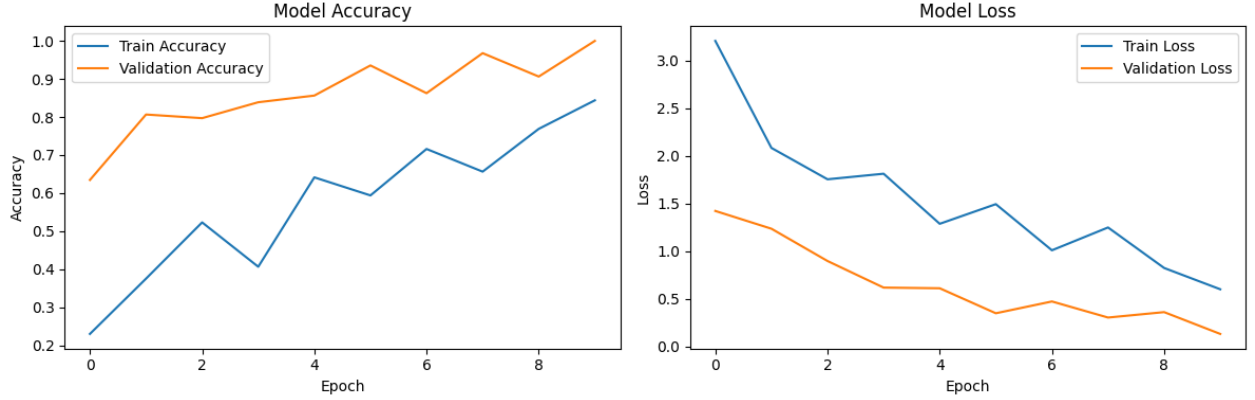
3.3 Derin Öğrenme Modellerinin Analizi

Derin öğrenme modelleri, özellikle büyük ve karmaşık veri setlerinde üstün performans göstermektedir. Bu projede kullanılan CNN modeli, diğer makine öğrenimi modellerine göre daha yüksek doğruluk ve F1 skoru elde etmiştir. Aşağıda, CNN modelinin eğitim ve doğrulama doğruluk ve kayıp grafikleri gösterilmiştir:

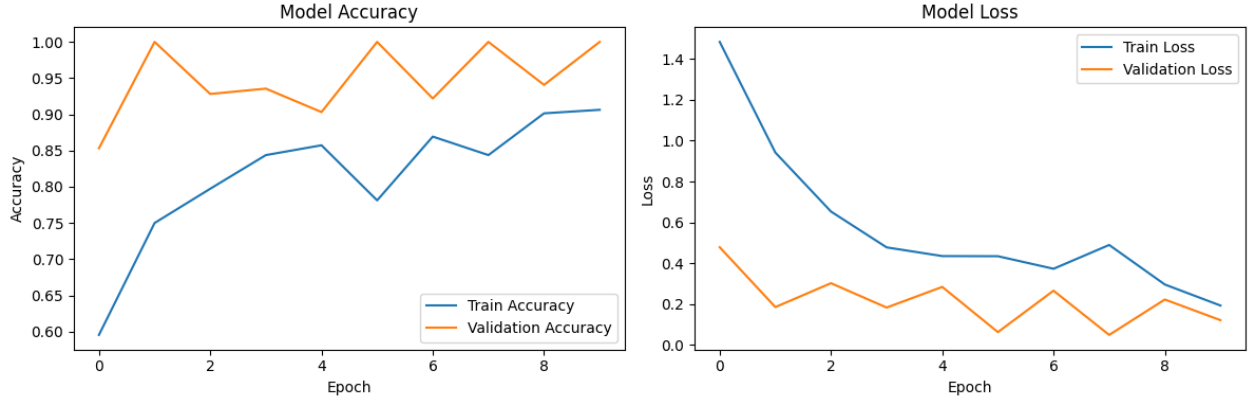
Grafik 2: CNN Modeli Eğitim ve Doğrulama Doğruluk ve Kayıp Grafiği



Grafik 3: VGG16 Modeli Eğitim ve Doğrulama Doğruluk ve Kayıp Grafiği



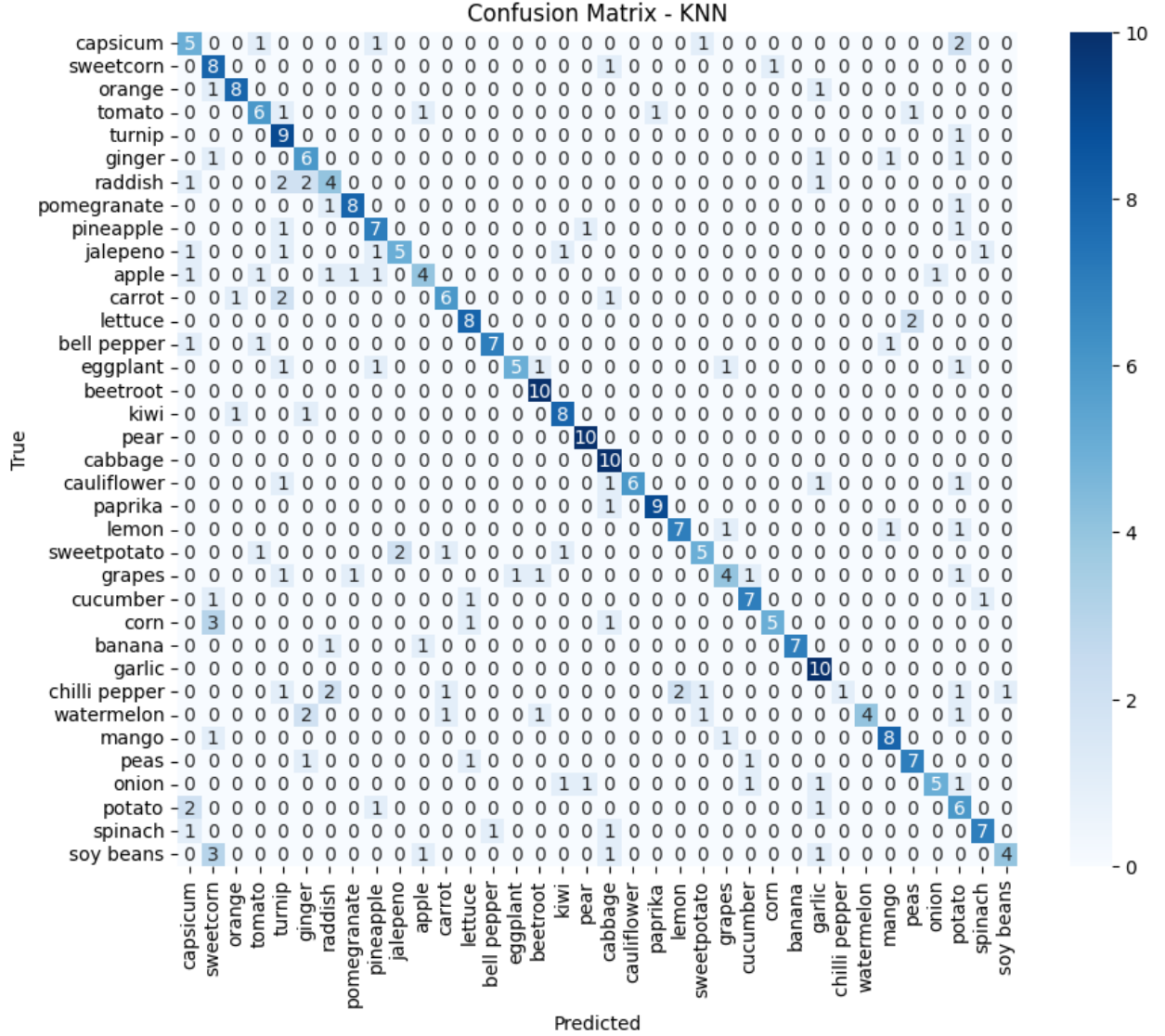
Grafik 4: MobileNetV2 Modeli Eğitim ve Doğrulama Doğruluk ve Kayıp Grafiği



3.4 Confusion Matrix Analizleri

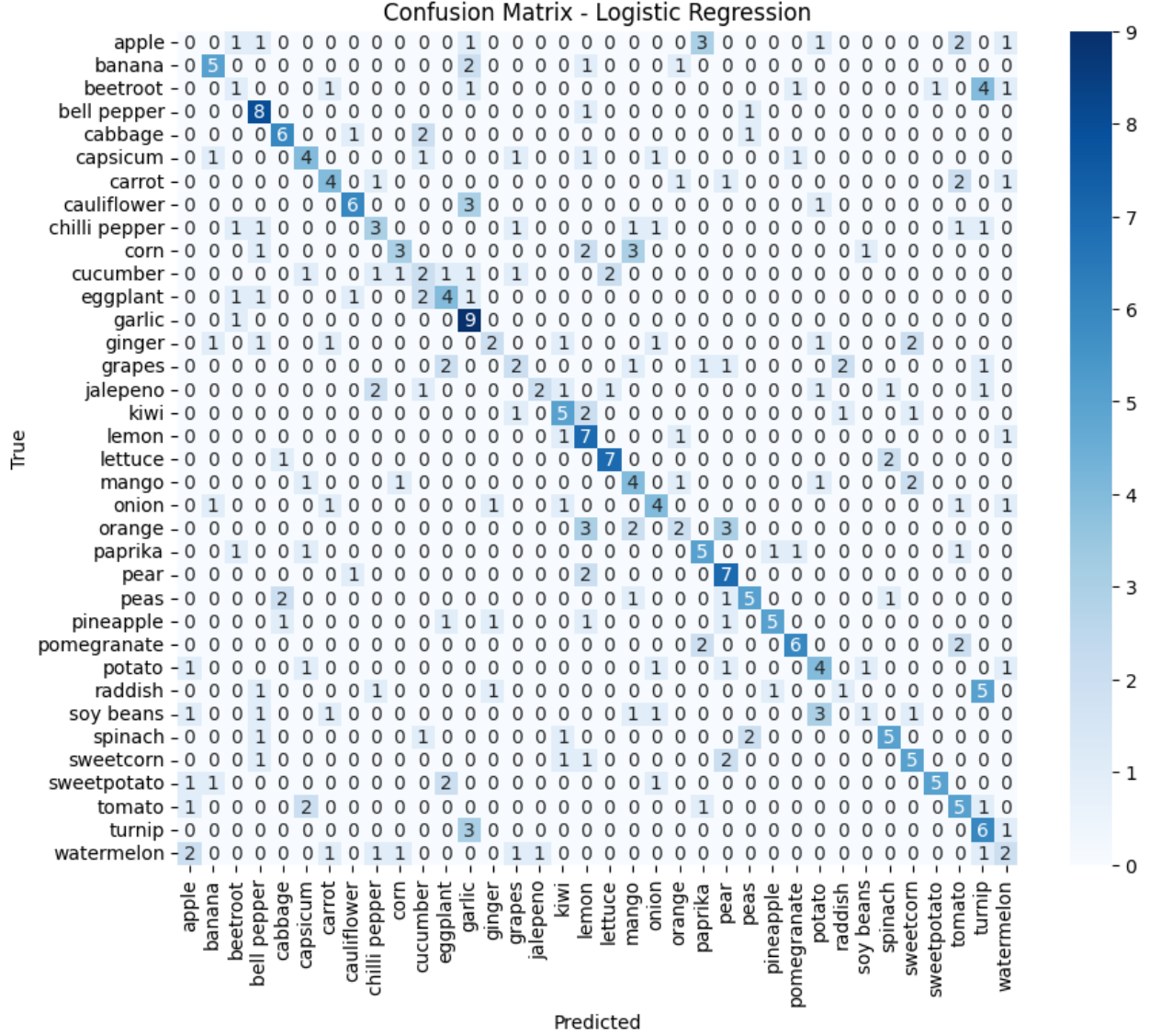
Confusion Matrix, modelin hangi sınıflarda iyi performans gösterdiğini ve hangi sınıflarda zorlandığını anlamamıza yardımcı olur. Aşağıda, her model için confusion matrix görselleri yer almaktadır:

Grafik 5: KNN Confusion Matrix



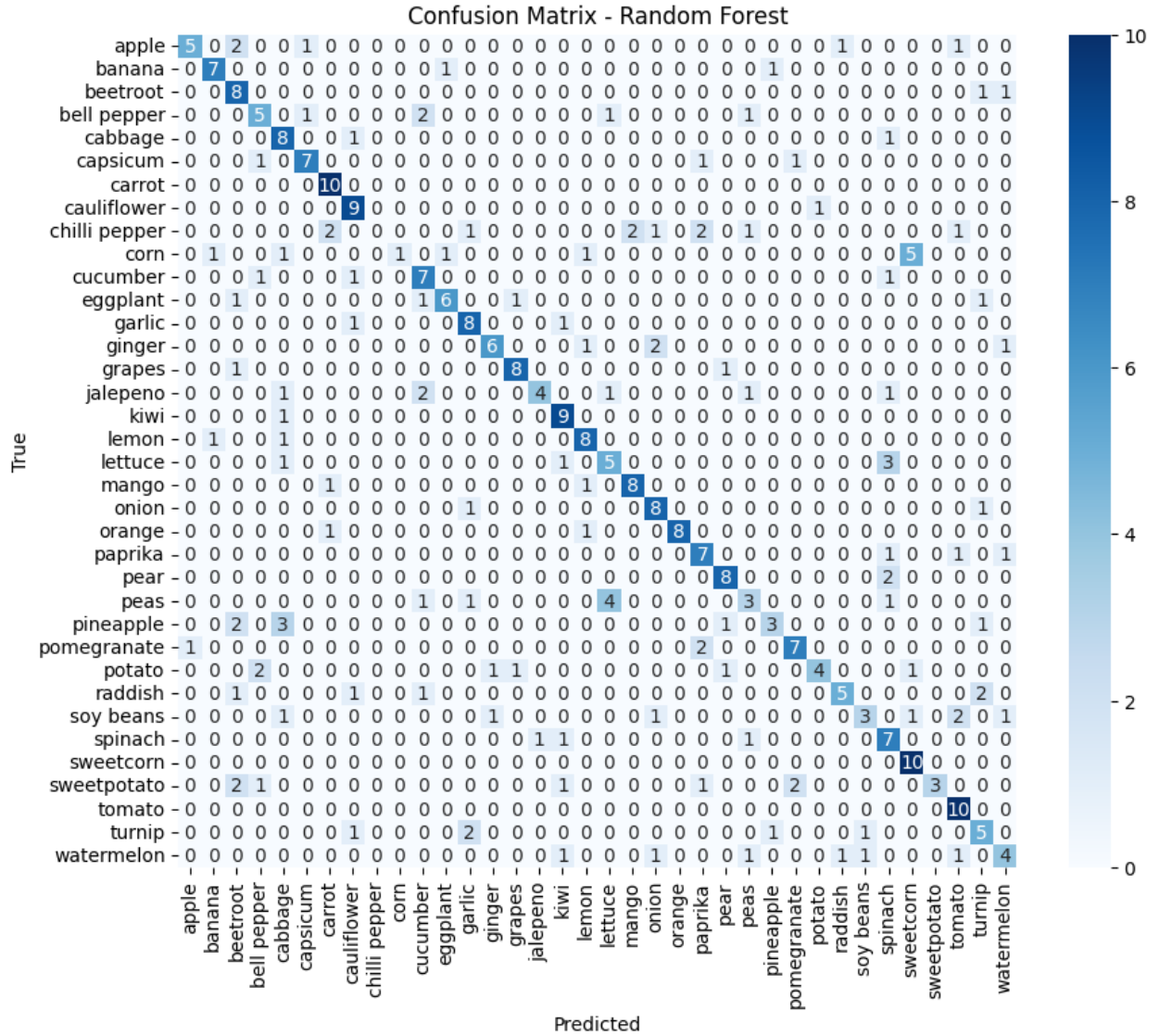
- **Genel Performans:** KNN modeli, bazı sınıflarda oldukça başarılı iken, bazı sınıflarda ciddi karışıklıklar yaşamakta. Genelde daha çok hata yapıyor ve yanlış sınıflandırmalar fazla.
- **Dikkat Çeken Hatalar:**
 - o capsicum, sweetcorn, tomato gibi sınıflarda diğer birçok sınıfla karışıklık var.
 - o orange ve ginger gibi sınıflarda oldukça fazla hata yapmış.
 - o carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda ciddi karışıklıklar var.
 - o sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı düşük.

Grafik 6: Lojistik Regresyon Confusion Matrix



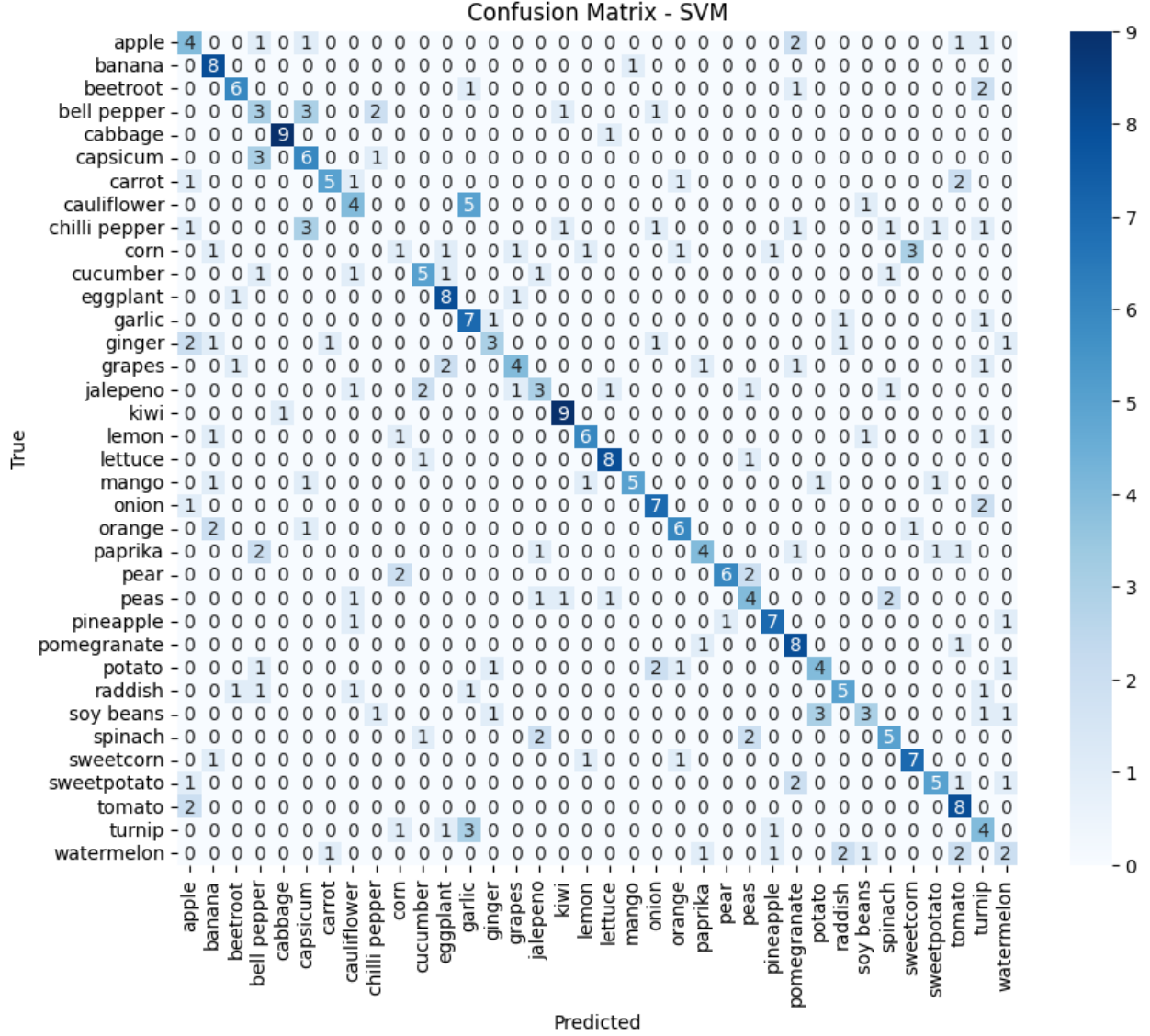
- **Genel Performans:** Logistic Regression modeli, genel olarak daha fazla hata yapıyor gibi görünüyor. Doğru tahminlerin sayısı KNN modeline göre daha az.
- **Dikkat Çeken Hatalar:**
 - o capsicum, sweetcorn, tomato gibi sınıflar yine birçok diğer sınıfla karışmış.
 - o orange ve ginger sınıflarında belirgin hatalar mevcut.
 - o carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda ciddi karışıklıklar gözlemleniyor.
 - o sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı oldukça düşük.

Grafik 7: Random Forest Confusion Matrix



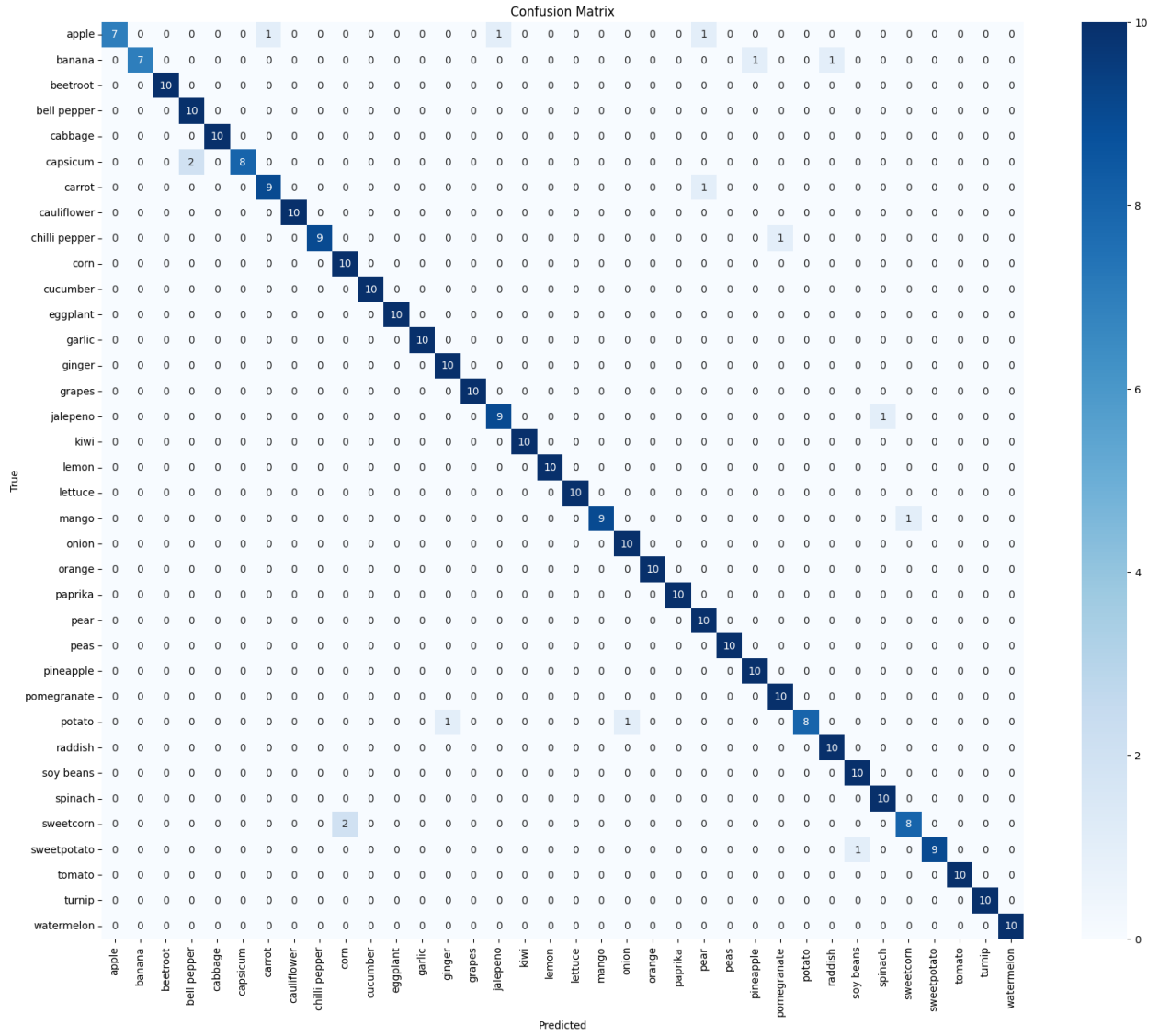
- **Genel Performans:** Random Forest modeli, KNN ve lojistik regresyon modellerine göre daha iyi performans gösteriyor. Yanlış sınıflandırmalar azalmış ve doğru tahmin oranı daha yüksek.
- **Dikkat Çeken Hatalar:**
 - o capsicum, sweetcorn, tomato gibi sınıflarda hala karışıklık mevcut, ancak önceki modellere göre daha az.
 - o orange ve ginger sınıflarında hata oranı düşmüştü.
 - o carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda karışıklıklar azalmış, ancak hala mevcut.
 - o sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı artmış.

Grafik 8: SVM Confusion Matrix



- **Genel Performans:** SVM modeli, Random Forest'tan biraz daha düşük performans gösteriyor. Ancak, doğru tahmin oranı lojistik regresyon ve KNN modellerine göre daha yüksek.
- **Dikkat Çeken Hatalar:**
 - o capsicum, sweetcorn, tomato gibi sınıflarda karışıklık mevcut, ancak önceki modellere göre daha az.
 - o orange ve ginger sınıflarında hata oranı düşmüş.
 - o carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda karışıklıklar azalmış, ancak hala mevcut.
 - o sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı artmış.

Grafik 9: CNN Confusion Matrix

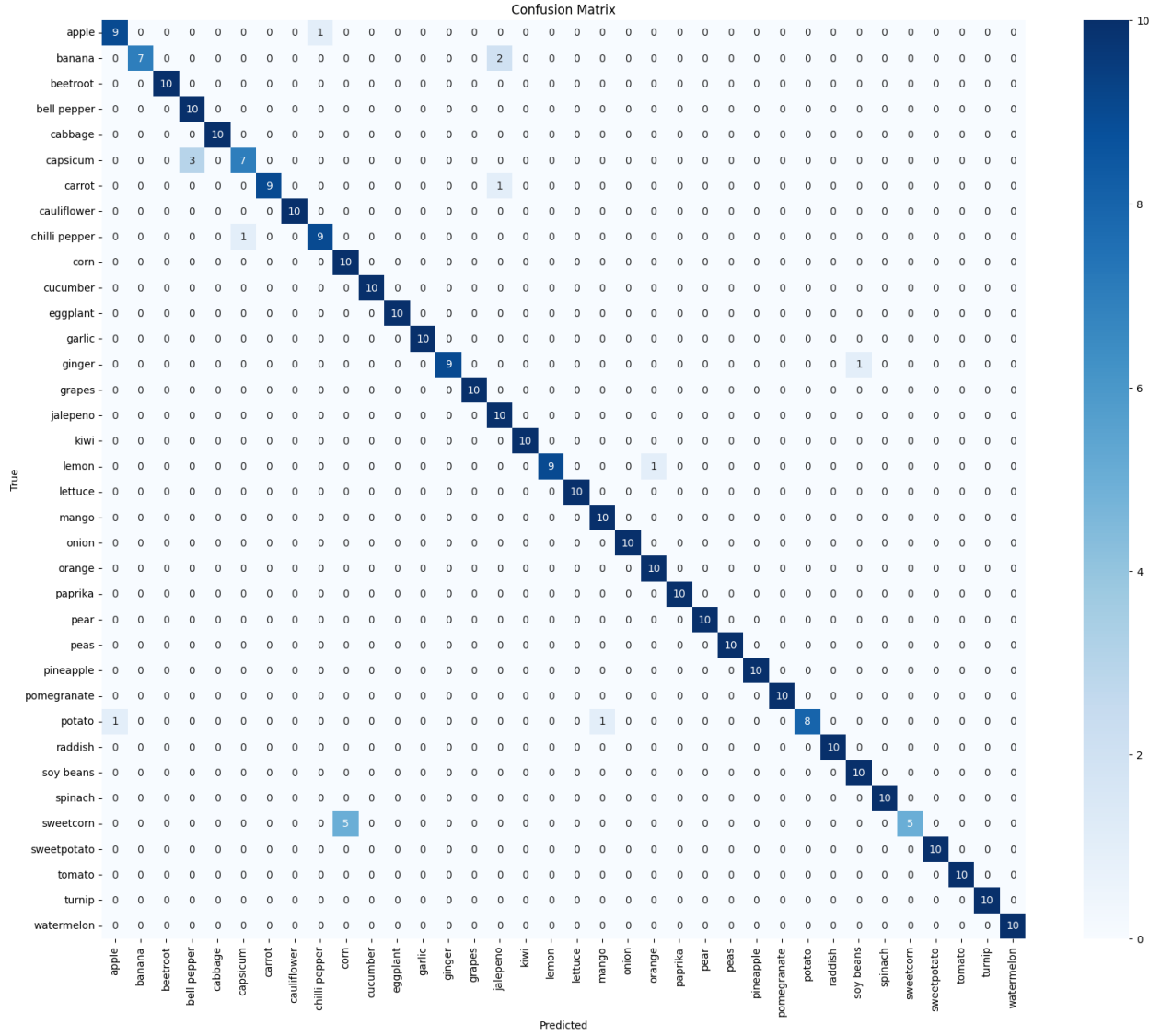


- **Genel Performans:** CNN modeli, makine öğrenimi modellerine kıyasla en yüksek doğruluk oranına sahip. Yanlış sınıflandırmalar oldukça az ve doğru tahmin oranı çok yüksek.
- **Dikkat Çeken Hatalar:**
 - capsicum, sweetcorn, tomato gibi sınıflarda karışıklık neredeyse hiç yok.
 - orange ve ginger sınıflarında hata oranı çok düşük.
 - carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda karışıklıklar minimum seviyede.
 - sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı çok yüksek.

	apple	banana	beetroot	bell pepper	cabbage	capsicum	carrot	cauliflower	chilli pepper	corn	cucumber	eggplant	garlic	ginger	grapes	jalepeno	kiwi	lemon	lettuce	mango	onion	orange	paprika	pear	peas	pineapple	pomegranate	potato	raddish	soy beans	spinach	sweetcorn	sweetpotato	tomato	turnip	watermelon
apple	7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
banana	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
beetroot	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bell pepper	0	0	0	9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cabbage	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
capsicum	0	0	0	2	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
carrot	0	0	0	0	1	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
cauliflower	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chilli pepper	0	0	0	1	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
corn	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
cucumber	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
eggplant	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
garlic	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ginger	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grapes	0	0	0																																	

- **Genel Performans:** VGG16 modeli, CNN modeline benzer şekilde yüksek doğruluk oranına sahip. Yanlış sınıflandırmalar az ve doğru tahmin oranı yüksek.
- **Dikkat Çeken Hatalar:**
 - o capsicum, sweetcorn, tomato gibi sınıflarda karışıklık neredeyse hiç yok.
 - o orange ve ginger sınıflarında hata oranı çok düşük.
 - o carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda karışıklıklar minimum seviyede.
 - o pear, pineapple, jalepeno, ginger gibi sınıflarda doğru tahmin oranı çok yüksek.

Grafik 11: MobileNetV2 Confusion Matrix



- **Genel Performans:** MobileNetV2 modeli, diğer derin öğrenme modelleri gibi yüksek doğruluk oranına sahip. Yanlış sınıflandırmalar çok az ve doğru tahmin oranı çok yüksek.
- **Dikkat Çeken Hatalar:**
 - capsicum, sweetcorn, tomato gibi sınıflarda karışıklık neredeyse hiç yok.
 - orange ve ginger sınıflarında hata oranı çok düşük.
 - carrot, lettuce, bell pepper, apple, cabbage gibi sınıflarda karışıklıklar minimum seviyede.
 - sweetpotato, pineapple, jalepeno gibi sınıflarda doğru tahmin oranı çok yüksek.

4. Sonuç ve Gelecek Çalışmalar

4.1 Sonuçlar

Bu proje, makine öğrenimi ve derin öğrenme modellerini kullanarak meyve ve sebze sınıflandırma problemine çeşitli yaklaşımlar sunmuştur. Yapılan deneyler sonucunda elde edilen bulgular şunlardır:

1. **Makine Öğrenimi Modelleri:** KNN, lojistik regresyon, random forest ve SVM gibi modeller, belirli sınıflarda başarılı olsa da genel olarak derin öğrenme modellerinin gerisinde kalmıştır. Özellikle KNN ve lojistik regresyon modelleri, veri setinin karmaşıklığı nedeniyle düşük performans göstermiştir.
2. **Derin Öğrenme Modelleri:** CNN, VGG16 ve MobileNetV2 modelleri, yüksek doğruluk oranlarıyla dikkat çekmiştir. Bu modeller, özellikle CNN, veri setindeki çeşitli sınıfları başarılı bir şekilde ayırt edebilmiştir. MobileNetV2 modeli, hem hız hem de doğruluk açısından üstün performans göstermiştir.

4.2 Gelecek Çalışmalar

Bu çalışmanın gelecekteki yönleri şunlar olabilir:

1. **Veri Seti Genişletme:** Mevcut veri setine daha fazla görüntü eklenerek modelin genelleme yeteneği artırılabilir. Ayrıca, veri setine daha fazla çeşitlilik katarak modelin farklı koşullarda performansı artırılabilir.
2. **Model İyileştirme:** Derin öğrenme modellerinin mimarileri üzerinde çeşitli optimizasyonlar yapılarak performans artırılabilir. Özellikle, transfer öğrenme teknikleri kullanılarak eğitim süresi kısaltılabilir ve doğruluk oranı artırılabilir.
3. **Gerçek Zamanlı Uygulamalar:** Eğitim ve test edilen modeller, gerçek zamanlı meyve ve sebze sınıflandırma sistemlerine entegre edilebilir. Bu, tarım ve gıda endüstrisinde kullanılabilecek pratik uygulamalar sağlayabilir.
4. **Diğer Görüntü İşleme Teknikleri:** Farklı görüntü işleme ve artırma teknikleri kullanılarak modelin performansı daha da iyileştirilebilir. Örneğin, farklı filtreler ve dönüşümler uygulanarak modelin daha dayanıklı ve genelleme yeteneği yüksek hale getirilebilir.

5. Kaynaklar

- Veri seti: Fruits and Vegetables Image Recognition Dataset
- Python kütüphaneleri: TensorFlow, Keras, Scikit-learn, NumPy, Matplotlib