



Ömer Berke GÖKTÜRK 131180032

Umut Barış KORKUT 141180752

BM 495 BİLGİSAYAR PROJESİ I

FİNAL RAPORU

T.C

GAZİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ANKARA

2018

1. ÖZET	1
2. Giriş	2
2.1. PROJENİN AMACI	2
2.2. PROJENİN KAPSAMI	2
2.3. LİTERATÜR TARAMASI	3
3. KULLANILACAK ARAÇ VE YÖNTEMLER	8
3.1. KULLANILAN ALGORİTMALAR.....	8
3.1.1. KNN.....	8
3.1.2. Random Forest.....	9
3.1.3. Naive Bayes	10
3.1.4. SVM	10
3.1.5. Derin Öğrenme	11
3.1.6. Lineer Regresyon	12
3.2. ÖZELLİK ÇIKARIMI	13
3.2.1. Makine Öğrenmesi Aktarımı.....	13
3.3. KULLANILAN ARAÇLAR.....	14
3.3.1. Orange	14
3.3.2. İşletim sistemi.....	14
3.3.3. Programlama dili	14
3.3.4. Kullanılan kütüphaneler.....	14
3.3.5. Geliştirme ortamı.....	15
3.3.6. Veri seti temin aracı.....	15
4. GERÇEKLEŞTİRİLEN UYGULAMA.....	16
4.1. GÖRÜNTÜLERİN TOPLANMASI.....	16
4.2. GÖRÜNTÜLERİN VEKTÖRLERE DÖNÜŞTÜRÜLMESİ	21
4.3. ALGORİTMALARIN UYGULANMASI	21
4.4. SONUÇLARIN KARŞILAŞTIRILMASI.....	22
5. SONUÇ VE DEĞERLENDİRME	26
6. KAYNAKLAR.....	27

1. ÖZET

Her yıl yanlış veya geç tespitlerden ve yanlış tespitlere bağlı olarak yapılan yanlış müdahalelerden dolayı, tarım alanında ciddi zayıtlar verilmektedir. Bu durum, tarımda verimliliği düşürmektedir.

Yanlış tespitin birçok farklı sebebi vardır. Öncelikle tarımla ilgilenen herkes, bitki bilimi konusunda uzman değildir. Bu nedenle karşılaşılan bilinmeyen durumlara müdahale mümkün olmamaktadır. Uzman kişilerin tespitte hata yapmalarının nedeni ise, belirtilerin insanlar tarafından tespit edilmesinin mümkün olmadığı durumların bir hayli fazla olmasıdır.

Diğer hastalıklarda olduğu gibi, bitki hastalıklarında da erken ve doğru teşhis çok önemlidir. Zamanında yapılan doğru tespitler, birçok bitkinin hayatını kurtarabilecek sonuçlar doğuracaktır. Bu nedenle tarım faaliyetlerinin verimliliği ve devamlılığı açısından; bitki hastalıkları ciddi sıkıntılar yaratsa da, doğru ve zamanında yapılan tespitler, hastalıkların tarım açısından kayıplar yaşatmasının önüne geçmek için çok önemlidir.

Bitkilerin ayırt edici birçok özelliği vardır. Bu özellikler, hastalık durumlarında da farklı şekillerde, kendilerini göstermektedir. Ancak belirtiler, her zaman açık değildir. Hatta bazı durumlarda gözle görülemeyen farklılıklar, hastalığın çok önemli belirtisi olabilir. Eğer bitkilerde, belirti özelliği taşıyan farklılıklar zamanında tespit edilebilirse, bitkiyi kurtarma ihtimali çok yüksektir.

Bilgisayar sistemleri, insanın ayırt edemeyeceği belirtileri, net bir şekilde ortaya çıkarmaktadır. Bu bağlamda, bitkilere bilgisayarlar aracılığıyla bakılacak olursa, doğru tespit ihtimali artacaktır. Bilgisayarlar tarafından tespit edilen bu farklı özelliklerden, makine öğrenmesi teknikleri kullanılarak modeller oluşturulduğunda, insanlardan daha başarılı tahmin sistemleri oluşturulmuş olur.

Makine öğrenmesi bünyesinde, birçok farklı algoritma vardır. Hastalık tespiti için kullanılan; logistic regression, kNN, random forest, naive bayes, svm learner gibi algoritmalar arasından en yüksek başarıyı, logistic regression algoritmasının sağladığı görülmüştür.

2. GİRİŞ

2.1. Projenin Amacı

Tarım alanında her yıl, tedavi edilemeyen bitki hastalıklarından dolayı ciddi kayıplar yaşanmaktadır. Bu kayıplar bazen yanlış tespitlerden bazen de tespit edilememekten kaynaklanmaktadır. Yapılan yanlış tespitler sonucu, kullanılan yanlış ilaçların oluşturduğu maliyetler ve tedavi edilemediğinden dolayı kaybedilen bitkilerin oluşturduğu maliyetler, tarım faaliyetlerini olumsuz etkilemektedir.

Tarımsal ilaç kullanımı Pesticides use							(Ton - Tonnes)
	İnsektisitler Insecticides	Fungusitler Fungicides	Herbisitler Herbicides	Akarisitler Acaricides	Rodentisitler Rodenticides	Diğer Other	Toplam Total
2006	7 628	19 900	6 956	902	3	9 987	45 376
2007	21 046	16 707	6 669	966	51	3 277	48 716
2008	9 251	16 707	6 177	737	351	5 613	38 836
2009	9 914	17 863	5 961	1 533	78	2 302	37 651
2010	7 176	17 396	7 452	1 040	147	5 344	38 555
2011	6 120	17 546	7 407	1 062	421	6 978	39 534
2012	7 264	18 124	7 351	859	247	8 766	42 611
2013	7 741	16 248	7 336	858	129	7 128	39 440
2014	7 586	16 674	7 794	1 513	149	6 007	39 723
2015	8 117	15 984	7 825	1 576	197	5 327	39 026
2016	10 425	20 485	10 025	2 025	259	6 835	50 054

Kaynak: Gıda, Tarım ve Hayvancılık Bakanlığı

Tablodaki rakamlar, yuvarlamadan dolayı toplamı vermeyebilir.

Source: Ministry of Food, Agriculture and Livestock

Figures in table may not add up to totals due to rounding.

Şekil 2.1. Türkiye’de yıllık tarımsal ilaç kullanımı[1.]

Şekilde görülen ilaç tüketimlerinin bir kısmı gerekliyen, büyük bir çoğunluğu gereksiz tüketimi oluşturmaktadır. Bunun bir numaralı sebebi, bitki hastalıklarını tespit sistemlerinin yetersizliğidir.

2.2. Projenin Kapsamı

Bitki hastalıkları tespit edilirken, bitkilerde karşılaştırılacak birçok farklı özellik bulunmaktadır. Hasta olan ve sağlıklı bitkiler arasında, bazı özel farklılıklar görülmektedir. Hastalık tespiti için; bitkilerin yaprak şekilleri, kök şekilleri, gövde, çeşitli kısımların renkleri gibi özellikler incelenerek, farklılıklar tespit edilir. Bu yöntemi yürütecek uzman kişi, gözle algılayabileceği farklılıkları kolayca tespit edebildiği durumda, hastalık için net bir ifadede bulunabilir. Ancak duyu organlarıyla algılayamayacak farkları, insanların tespit etmesi mümkün değildir. Hatta farklı bir koku yaymayan ve görüntü olarak farklı algılanmayan bitkilerde

hastalık tespiti mümkün olmamaktadır. Olumsuz bir durumdan şüphelenilemeyecek bu gibi anlarda, bitki ölümleri gerçekleşmeden, sorun algılamak imkânsızdır.

Sensörler, görüntü işleme cihazları, boyut ölçüm yöntemleri gibi araçlar, insanın fark edemeyeceği hususları, kolaylıkla tespit edebilir. Bahsedilen, insanların ölçüm yapamayacağı durumlar için, bu tarz sistemler geliştirilebilir. Ancak ölçüm sonuçlarının insanlar tarafından yorumlanması, yine bir takım hatalar doğuracaktır. Ölçüm ve değerlendirme aşaması bir bütün olarak ele alınıp, bilgisayar sistemleri aracılığıyla gerçekleştirilirse, tespit ve çözüm işlemlerinin başarı yüzdesi artırılabilir. Ölçüm için kullanılacak mekanik veya elektronik sistemlere yorum mekanizması eklemek, makine öğrenmesi alanının sınırları içindedir. Yani makine öğrenmesi, bitkilerle ilgili çeşitli ölçüm değerlerini karşılaştırıp, problemin türüne göre – hastalıklı veya hastaliksız yorum yapabilen bir daldır.

2.3. Literatür Taraması

Makine öğrenmesi yöntemleriyle; bitki türlerinin, hastalıklarının tespit edilmesiyle ilgili daha önce yapılan bir takım çalışmalar vardır.

Dünyada bulunan birçok üzüm türü vardır. Bu üzüm türlerinin yaprakları incelenerek, üzüm türlerini tespit etmeye yönelik bir çalışma yapılmıştır. Bu tespit işleminde yapay sinir ağları kullanılmıştır[2.]

Hindistan’da üzüm yetiştiriciliği, Hindistan tarım faaliyetlerinin önemli bir kısmını oluşturmaktadır. Avrupa’dan ve Batı Asya’dan gelen üzümün yetiştiriciliği için, üzümün yetiştirme şartlarının bilinmesi gerekir. Üzümün meyve verebilecek elverişlilikte yetiştirilmesi için, sıcak ve kuru hava gereklidir. Üzüm sağlığını etkileyen en önemli faktörlerden ikisi; toz ve küftür. Onun dışında bitki hastalıkları da üzüm yetiştiriciliğini ciddi oranda etkilemektedir. Sağlıklı üzüm yetiştirebilmek için; elverişli hava koşullarının sağlanması, toz ve küfün uzaklaştırılması, bitkinin doğru zamanda budanması ve bitki hastalıklarının zamanında tespit edilmesi son derece önemlidir[2.]

Hindistan’daki üzüm yetiştiriciliği için yapılan literatür çalışmasında, sağlıklı koşulların düzeltilmesi için kullanılabilecek teknolojik sistemler; desen tanıma, resim

işleme şeklinde belirlenmiştir. Bu tarz teknolojik bir sistemden beklenti; zararlı otların tespit edilmesi, meyve ve sebzelerin ayırt edilmesi ve bitki hastalıklarının tespit edilmesidir.

Bitki hastalıklarının otomatik olarak saptanması, önemli bir araştırma konusudur. Böyle bir çalışma başarıyla yürütüldüğü durumda, oluşturulan bitki tarlalarının verimliliği artabilir ve bitki yapraklarında belirtiler olduğu anda, bitki hastalıkları algılanabilir. Bu nedenle, bitki hastalık vakaları, hızlı ve doğru bir şekilde zamanında tespit edilebilmelidir. Daha önce bahsedildiği üzere, yanlış tespitler kadar zamanında yapılamayan tespitler de verimlilik açısından sıkıntılar doğurmaktadır. Bu durum, en büyük sorunlardan biri olan, bitki ilaçlarının gereksiz kullanımını beraberinde getirmektedir. Bitki hastalıkları için kullanılan böcek ilaçları, eğer gereğinden fazla kullanılırsa hem maliyeti artırır, hem de toksik kalıntılar, sağlık açısından sorunlar yaratır. Bu nedenle, bitki hastalıklarının tespiti için, düzgün ve etkili modeller geliştirmek gerekir.

Al-Bashish, D., M. Braik ve S. Bani-Ahmad, yapraktaki hastalıkların tespiti ve sınıflandırılması için K-means algoritmasını önermişlerdir. Aynı ekip, yaprak doku özelliklerinin sınıflandırılması için CCM' yi, hastalıkları 6 kategoride sınıflandırmak için de BPNN'yi önermiştir[2.]

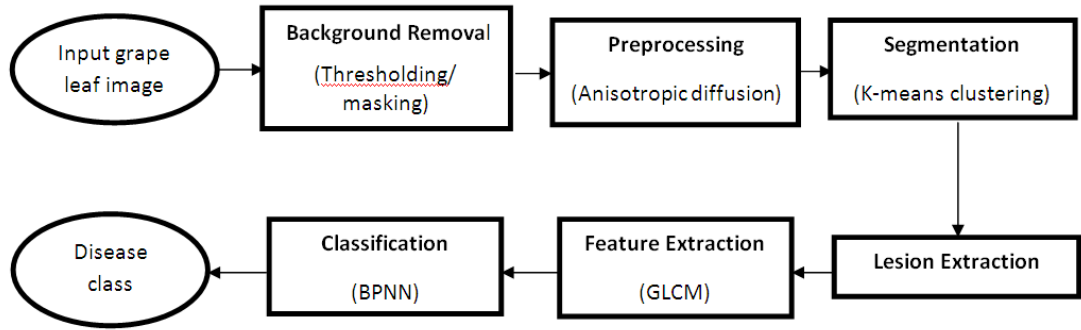
A. Camargo, J.S. Smith, bitkilerin ya da bitkinin yapraklarının RGB renklerini; H, I3a ve I3b' ye dönüştürmeyi tartışmışlardır. Dönüştürülen görüntü, daha sonra histogram kullanılarak analiz edilen yoğunluk dağılımlarına göre bölünür. Çıkarılan bölge; hedef bölgenin bir parçası olmayan piksel bölgesi kaldırılarak işlenir. Daha sonra her düğümün ve pikselin bütün komşuları incelenir[2.]

S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund ve diğer ekip üyeleri, hastalık tespiti ve derecelendirmesi için, K-means kümeleme algoritmasını önermişlerdir.

H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, özellik çıkarımından önce yeşil piksellerin maskelenmesini önermiş ve daha doğru bir sınıflandırma elde etmişlerdir.

Bütün bu araştırmacılar, yaprak örneklerine ait görüntüleri; sabit bir mesafeden, düzgün bir arka plana sahip olan ve güzel ışıklandırılmalı olarak çekilmiş resimlerden toplamışlardır.

A. Meunkaewjinda, P. Kumsawat ve ekibi; üzüm yaprağı rengini tanımak için SOFM ve BPNN'yi, segmentasyon için MSOFM'yi, sınıflandırma için GA ve SVM algoritmalarını ve karmaşık birçok hesap için başka algoritmaları kullanmayı önermiştir.



Şekil 2.2. Üzüm yaprağıyla ilgili sınıflandırmalar için yapılan işlemler[2.]

Yine Hindistan'da yapılan bir başka çalışma, pamuk yetiştiriciliğinde verimi artırmaya yöneliktir. Hindistan bir tarım ülkesidir ve nüfusunun büyük çoğunluğu tarımla geçimini sağlamaktadır. Bu nedenle tarım verimliliği, ülke ekonomisi için son derece önemlidir. Ancak verimliliği artırmaya yönelik hamleler, teknik bilgi gerektirmektedir. Bu nedenle Hindistan tarımı için, teknolojik bilgisayar sistemlerinin etkin kullanılabilmesi önemlidir. Beyaz altın olarak nitelendirilen pamuk, Hindistan'da 60 milyon insana geçim kaynağı oluşturmaktadır. Bu açılardan ele alınan pamuk, verimliliğin artırılması gereken önemli sorunlardan birini oluşturmaktadır[4.]

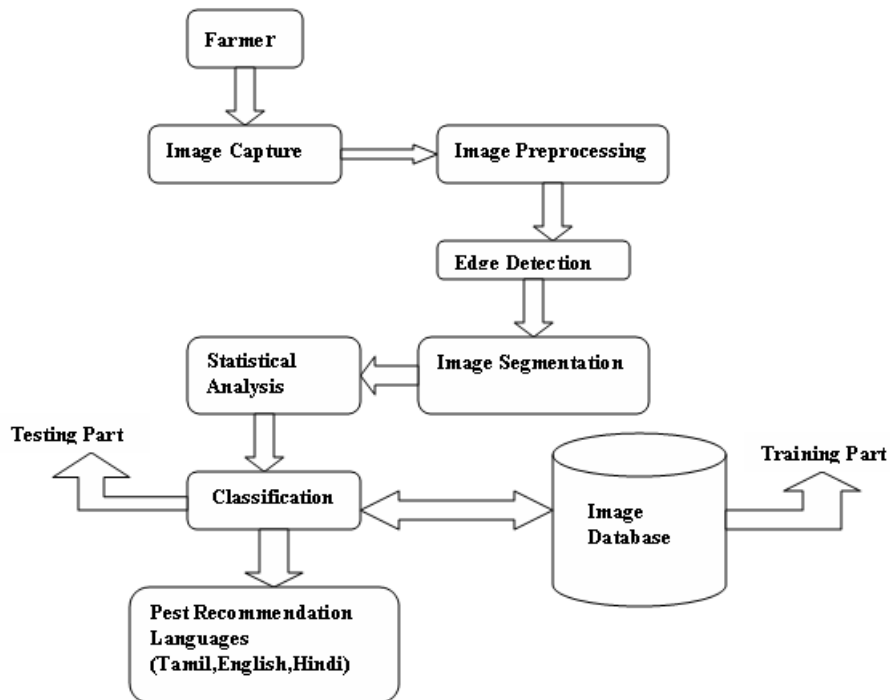
Yapılan çalışmada, pamuk yetiştiriciliği için tarım alanında görüntü işleme yöntemlerinin kullanılması düşünülmüştür. Görüntü RGB özelliği ile piksel sayma tekniği, tarım alanında yoğun bir şekilde uygulanmaktadır. Görüntü işlemenin tarım alanındaki bir takım amaçları şu şekildedir:

1- Hastalıklı kök, yaprak, meyve tespiti

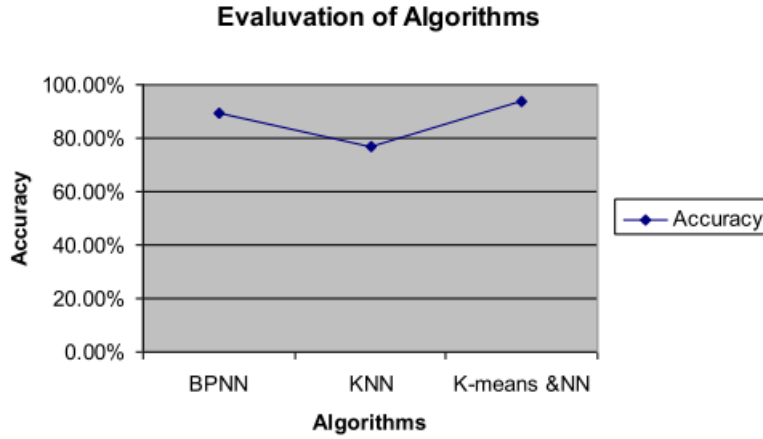
- 2- Etkilenen bölgenin hastalığa göre ölçülmesi
- 3- Etkilenen bölgenin sınırlarının tespiti
- 4- Etkilenen bölgenin renginin belirlenmesi
- 5- Meyvelerin boyut ve şeklinin belirlenmesi
- 6- Nesnenin doğru tespiti

Literatür taramasında bulunan, bu amaçlara yönelik geliştirilmiş olan çalışmada, görüntüler dijital kamera tarafından elde edilir, işlenir ve daha sonra algoritmalar uygulanarak başarı oranı tespit edilir[4.]

Daha önce yapılmış olan bu çalışmada, bilgisayar sistemleri tarafından tespit edilmesi beklenen hastalıklar; Fusarium solgunluğu, Verticillium solgunluğu, kök çürüklüğü, kemik çürüklüğü, gri küf, mor lekeler, yaprak dökme, bakteriyel bulantı, yaprak kıvrıcıklığıdır.

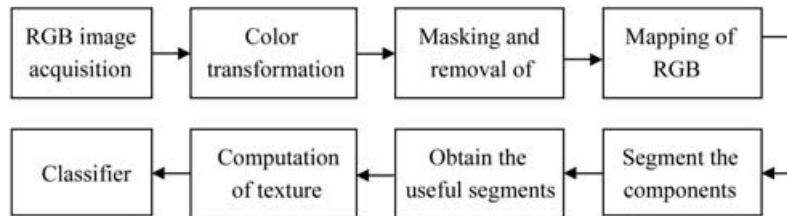


Şekil 2.3. Pamuk yaprağındaki hastalıkların tespiti için oluşturulan işlem basamakları[4.]



Şekil 2.4. Pamuk yaprağındaki hastalıkların tespiti için uygulanan algoritmalar ve başarı oranları

Bitki yapraklarının sağlıklı bölgelerinin saptanması ve hastalıkların doku özelliklerine göre sınıflandırılması üzerine yapılan bir başka çalışmada, dijital kamera vasıtasıyla çeşitli yaprak görüntüleri toplanmıştır. Toplanan resimlere algoritmalar uygulamak için öncelikle, tespit için yararlı olabilecek yaprak özellikleri, görüntü işleme teknikleriyle belirlenmiştir[3.]



Şekil 2.5. Önerilen yaklaşım için işlem basamakları[3]

3. KULLANILACAK ARAÇ VE YÖNTEMLER

Raporun bu bölümünde hangi araçların ve yöntemlerin kullanıldığı sebebiyle birlikte açıklanıp kullanılan araç ve yöntemler hakkında kısaca bilgiler verilmiştir.

3.1. Kullanılan Algoritmalar

Görsel sınıflandırma için yöntem olarak makine öğrenmesinin dalları olan KNN (K Nearest Neighbours, En yakın K komşu), Random Forest (Rastsal Orman), Naive Bayes, SVM (Support Vector Machine, Destek Vektör Makinesi), Derin Öğrenme (Deep Learning) ve Lineer Regresyon gibi bir çok yöntem uyguladık. Derin öğrenme yöntemi sınıflandırmanın doğruluğu konusunda en başarılı yöntem oldu fakat daha çok bilgisayar gücü gerektirmektedir, lineer regresyon ise daha az bilgisayar gücü gerektirmekteyken sınıflandırma doğruluğu konusunda daha az başarılıdır.

3.1.1. KNN

KNN algoritması her veriyi uzayda bir nokta olarak düşünür ve her özelliği (feature) uzayın bir boyutu olarak düşünür. Verilerin birbirlerine uzaklığını çeşitli uzaklık fonksiyonları ile hesaplayarak sınıflandırma yapar.

Devamlı (continious) verilerde kullanılan üç temel uzaklık fonksiyonu vardır. Bunlardan en yaygın olarak kullanılanları Euclidean (Öklidian) ve Manhattan (Manhattan) uzaklık fonksiyonlarıdır. Devamlı olmayan yani kategorisel verilerde ise Hamming (Hemming) uzaklık fonksiyonu kullanılır.

Distance functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Hamming Distance

$$D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

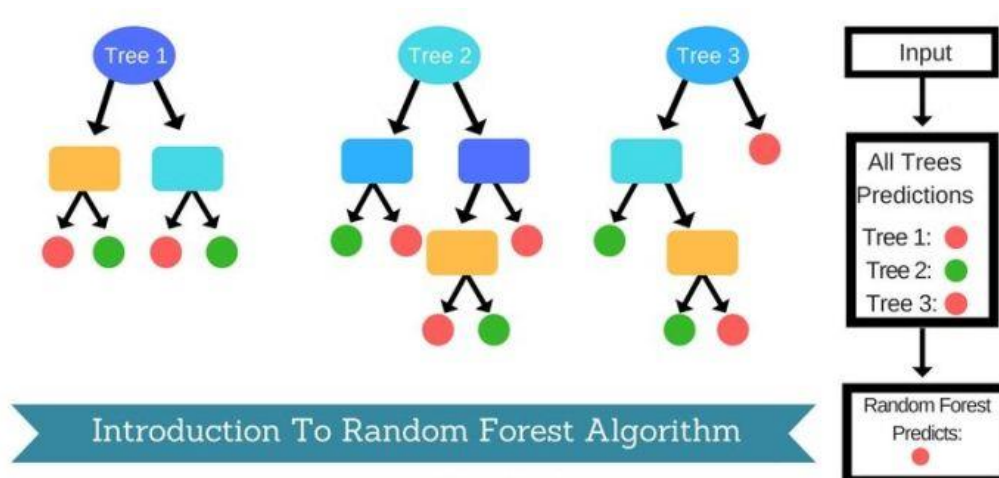
X	Y	Distance
Male	Male	0
Male	Female	1

Şekil 3.1. KNN algoritmasında kullanılan fonksiyonlar

3.1.2. Random Forest

Rastsal orman (Random Forrest) bir diğer deyişle Rastsal Karar Ormanları (Random Decision Forrests) makine öğrenimi modelinin eğitimi sırasında verilerin özelliklerine bakarak çok sayıda karar ağacı oluşturur. Karar ağaçlarının ortalamasını alarak verinin hangi sınıfa ait olduğuna belirler.

Karar ağacı derken, bir ağaç yapısı oluşturularak ağacın yaprakları seviyesinde sınıf etiketleri ve bu yapraklara giden ve başlangıçtan çıkan kollar ile de özellikler üzerindeki işlemler ifade edilmektedir.



Şekil 3.2. Rastsal orman algoritmasının görselleştirilmiş hali

3.1.3. Naive Bayes

Naive Bayes sınıflandırma yöntemi, özellikler (features) arasındaki bağıntılara bakarak Bayes Teoreminin uygulanmasına dayanır. Bayes teoremi koşullu olasılık hesaplaması yapar. Naive Bayes yönteminde girilen verinin olasılığı en yüksek olan sınıfa ait olduğu tahmin edilir. Veri setinin boyutunun çok olduğu durumlarda etkilidir.

The diagram shows the Naive Bayes formula: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Arrows point from the labels to the corresponding parts of the formula: 'Likelihood' points to $P(x | c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c | x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Şekil 3.3. Naive bayes algoritması için fonksiyon

Şekil 3.3. de görülen teorem, bir rastsal değişken için koşullu olasılıklar ile önsel(marjinal) olasılıklar arasındaki ilişkiyi göstermektedir.

$P(c|x)$: B olayı gerçekleştiği durumda x olayının meydana gelme olasılığı

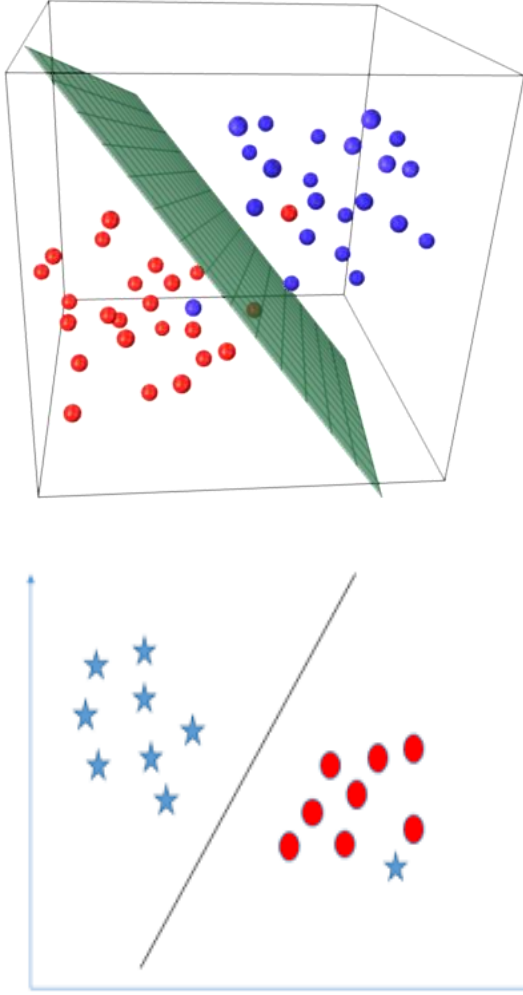
$P(x|c)$; A olayı gerçekleştiği durumda x olayının meydana gelme olasılığı

$P(c)$ ve $P(x)$: c ve x olaylarının önsel olasılıklarıdır.

3.1.4. SVM

SVM (Support Vector Machine) sınıflandırma için uzayda noktalar haline düşünülen verileri ikiye bölerek ayırır. Uzayın her boyutu bir özellik olarak düşünülür. (İki özellik için düzlem doğru ile ayrılır, üç özellik için uzay düzlem ile ayrılır vb.) Uzayı ikiye bölerken farklı sınıfların birbirine olan en uzak noktaları esas alınır.

Örneğin en uzak doğrunun bulunabilmesi için yapılan yöntem şöyledir: iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu çizgilerin aralarındaki orta noktalardan geçen doğru ayrım doğrusu olarak bulunur.



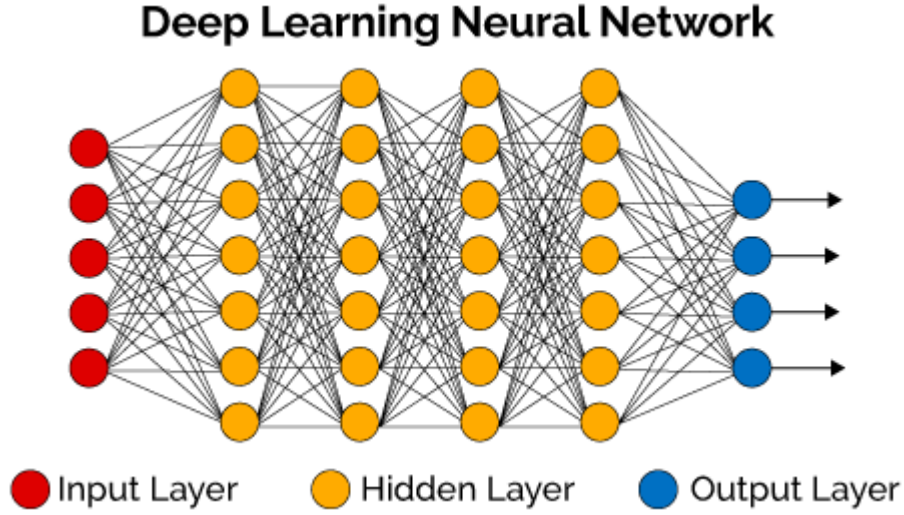
Şekil 3.4. SVM uzay bölmesi örneklendirmesi

3.1.5. Derin Öğrenme

Derin öğrenme (deep learning) çok sayıda gizli katman içeren yapay sinir ağları makine öğrenme yöntemidir. İnsan beynini taklit ederek yapılan bir yöntemdir. Yapay sinir ağları tıpkı insan beynindeki gibi birbirlerine çeşitli şekillerde bağlanmışlardır. Bu ağlar öğrenme, hafızaya alma ve veriler arasında ilişki kurma

yetilerine sahiptirler. Öğrenme verilerin özelliklerinin bağlantı ağırlıklarını tekrar tekrar ayarlayarak olur.

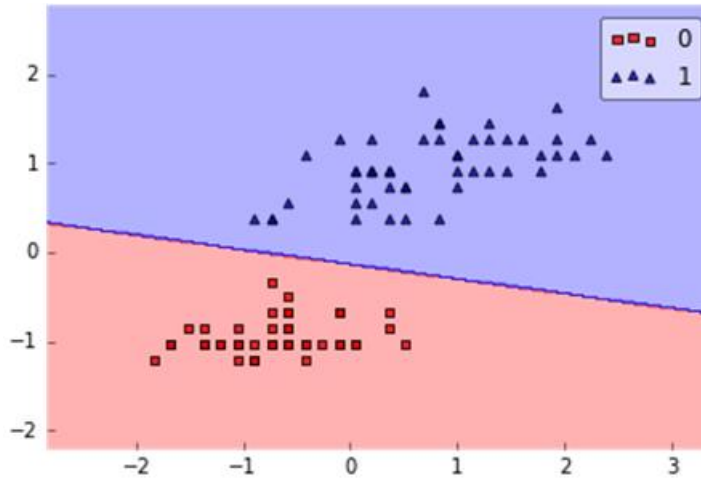
Derin öğrenme algoritmaları çok sayıda veri ile çok başarılı sonuçlar verirler.



Şekil 3.5. Derin öğrenme katmanlarının görselleştirilmiş hali

3.1.6. Logistic Regresyon

Logistic regression, SVM gibi lineer bir metottur. Sınıflandırma için uzayda noktalar haline düşünülen verileri ikiye bölerek ayırır. Uzayın her boyutu bir özellik olarak düşünülür (İki özellik için düzlem doğru ile ayrılır, üç özellik için uzay düzlem ile ayrılır vb.). Uzayı ikiye bölerken her sınıfa ait olma olasılığını hesaplar.



Şekil 3.6. Logistic regresyon uzay bölme örneği

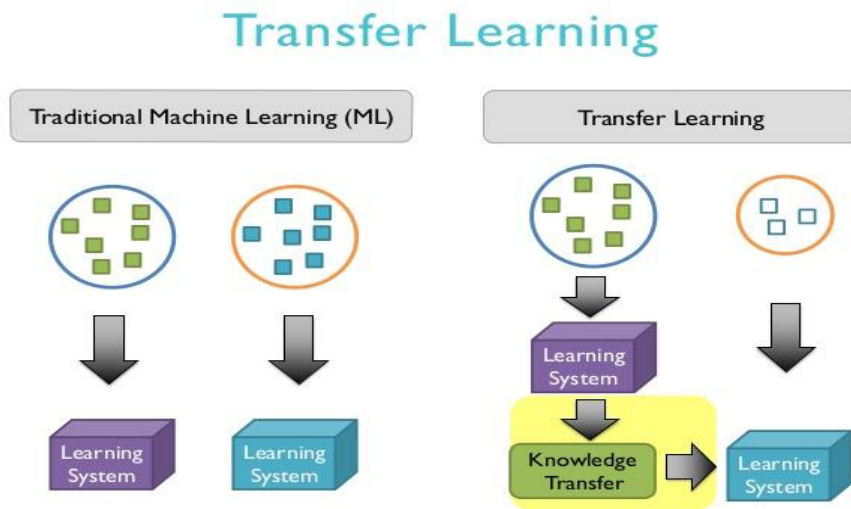
3.2. Özellik Çıkarımı

Görsellerden özellik çıkarımı (feature extraction) yöntemi olarak öğrenme aktarımı (transfer learning) kullanmayı tercih ettik. Öğrenme aktarımı daha önceden eğitilmiş bir yapay sinir ağının (neural network) çıkarttığı özellikleri kullanma yöntemine verilen addır. Önceden eğitilmiş model olarak VGG16, VGG19, ResNet50, Inception V3, Xception gibi modelleri araştırdık. Google'ın sunduğu Inception v3 modelini ve kullanmaya karar verdik. Bu modeli kullanmaya karar vermemizin sebebi olarak Inception v3 modelinin daha yüksek sınıflandırma doğruluk başarısına sahip olması söylenebilir.

3.2.1. Makine Öğrenmesi Aktarımı

Görsellerden manuel olarak kullanışlı özellikler (features) çıkarmak çok zordur. Örneğin bir yüzeyin dokusunun (pürüzlü, sert vb.) nasıl olduğunu kod yazarak ayırt etmek çok zordur. Bu yöntemi manuel olarak yapmaktansa derin öğrenme kullanmak çok kullanışlıdır çünkü derin öğrenme ham verilerden özellik çıkarımı yapabilen bir yöntemdir.

Makine öğrenmesi aktarımı (Transfer Learning) yöntemi daha önceden eğitilmiş bir makine öğrenmesinin parametrelerini yeniden kullanmaya yarar. Bu sayede çok daha az veri ile çok daha başarılı sonuçlar elde edebiliriz.



Şekil 3.7. Öğrenme aktarımı örnek şeması

Makine öğrenmesi aktarımını gerçekleştirmek için Orange3 aracını kullandık. Orange3 aracını kullanarak yüklenen görselleri inceptionv3 makine öğrenim aktarmasına tabi tutup, görsellerden elde edilen 2048 adet özelliği vektör halinde csv formatında kaydettik.

3.3. Kullanılan Araçlar

3.3.1. Orange

Orange3 veri analizi ve makine öğrenmesi uygulamaları için görsel arayüzlü bir araçtır. İçeriğinde makine öğrenimi ve veri analizini kolaylaştıran çeşitli algoritmalar bulunmaktadır.

3.3.2. İşletim sistemi

İşletim sistemi: Windows 10

Geliştirdiğimiz uygulama için işletim sistemi olarak Windows 10'u tercih etme sebebimiz olarak önceden Windows kullanıcısı olmamız, Windows'a aşina olmamız, windowsun Python dilini desteklemesi söylenebilir.

3.3.3. Programlama dili

Geliştirme dili: Python 3.6

Geliştirdiğimiz uygulama için dil olarak Python'ın şu anki en son sürümü olan 3.6 yı kullanmayı tercih ettik. Python tercih etme sebebimiz olarak güncel makine öğrenmesi kütüphanelerinin Python desteğinin fazla olması söylenebilir. Python 3.6 sürümünü tercih etme sebebimiz olarak en güncel sürüm olması ve güncel makine öğrenmesi kütüphaneleri ile uyumluluk sorunu içermemesi söylenebilir.

3.3.4. Kullanılan kütüphaneler

Kullanılan kütüphaneler: Scikit-Learn

Görsel sınıflandırma için derin öğrenme yöntemi kullanan kütüphaneleri araştırdık. Caffe, Caffe2, Chainer, CNTK(Microsoft Cognitive Toolkit), Deeplearning4j, Keras, MATLAB, MxNet, TensorFlow, Theano, Torch/PyTorch kütüphanelerini araştırdık.

Projenin başlarında uygulamada Google şirketinin geliştirdiği TensorFlow makine öğrenmesi kütüphanesini temel olarak kullanmayı tercih ettik.

Daha sonra makine öğrenmesi kütüphanelerinden Scikit-Image, Scikit-Learn gibi kütüphaneleri araştırdık. Scikit-Learn kütüphanesini kullanmaya karar verdik. Bu kütüphaneyi kullanmayı tercih etme sebebimiz olarak Scikit-Learn kütüphanesine aşina olmamız, proje kapsamında yapacağımız işler için yeterli olması ve göreceli olarak kolay olması söylenebilir.

3.3.5. Geliştirme ortamı

IDE (Bütünleşik geliştirme ortamı): Jupyter-Notebook

Bütünleşik geliştirme ortamı olarak Notepad++, Jupyter Notebook, atom, PyCharm gibi bazı uygulamaları araştırdık. Geliştirme ortamı olarak başlarda Microsoft Command Line (Komut İstemi) ortamını kullandık. Sonradan proje büyüdükçe Jupyter-Notebook kullanmaya karar verdik. Bu kararı vermemize sebep olarak Jupyter-Notebook'a aşina olmamız ve kullanıcı dostu arayüzü olması söylenebilir.

3.3.6. Veri seti temin aracı

Sınıflandırma tanımları ile ilgili veri setlerinin temini için Google Görseller sitesinden gerekli görselleri Chrome web tarayıcısının bir eklentisi olan Fatkun Batch Image Downloader kullanarak indirdik.

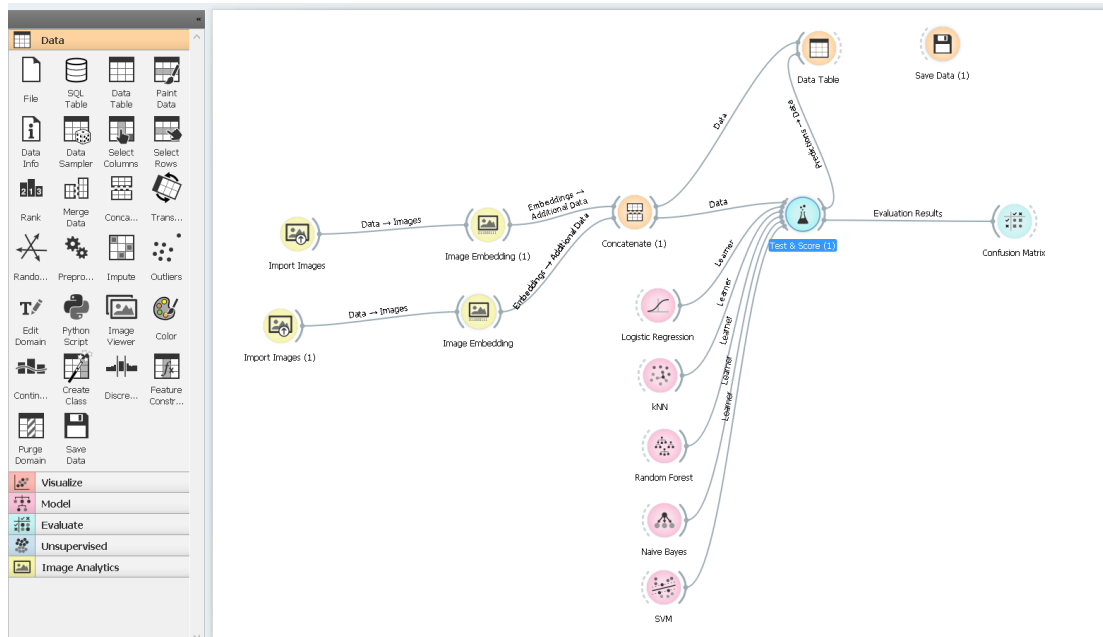
Fatkun Batch Image Downloader, Google görsellerden aranan fotoğrafları toplu olarak indirmeye yarayan bir programdır. İstenen fotoğraf kriterleri arama motoruna yazıldıktan sonra toplu indirme butonuna basılır ve indirilmesi istenmeyen fotoğraflar elenir. Elenmeyen fotoğraflar indirilir.

4. GERÇEKLEŞTİRİLEN UYGULAMA

Literatür taramasında incelenen ve bahsedilen üç farklı örnekteki aşamalara benzer aşamalar içeren bir sistem geliştirildi. Bu sistem; görüntülerin toplanması, görüntü işleme yöntemleriyle analizi sonucunda özellik çıkarımı, oluşturulan modellere yönelik çeşitli algoritmaların uygulanması ve test sonuçlarının analizini içermektedir.

4.1. Görüntülerin Toplanması

Sistemin ilk aşamasını oluşturan görüntü toplama işlemi, daha önceden çekilmiş yaprak fotoğraflarının internet ortamından temini şeklinde gerçekleşmiştir. Belirlenen modele uyan 131 resim, sistemin geliştirileceği bilgisayarlara indirilmiştir. Daha önceki örnek çalışmalarda edinilen tecrübelerle göre, verimli bir görüntü işleme için, tutarlı ve belirlenen kriterlere uygun resim seçmek son derece önemlidir. Belirlenen bu kriterler; düzgün arka plan, düzgün ve yeterli ışıklandırma, yeterli çözünürlük, uygun fotoğraf mesafesi şeklindedir. Bu kriterlere uygun resimler, tek tek incelenerek, veri seti için toplanmıştır. Toplanan veri setindeki resimlerin vektörlere dönüştürülmesi, özellik çıkarımı yapılması için Orange 3 kullanılmıştır.



Şekil 4.1. Orange üzerinde örnek şema

Şekil 4.1. de görülen şemada; “import image” butonu ile, toplanan veri seti programa eklenmiştir. İki tane “import image” kullanılmıştır. Bu aşamada, hastalıklı ve sağlıklı bitkilere ait veri tabanları ayrılmıştır. “Image embadding” butonu, eklenen veri setlerindeki resimlerden vektörler oluşturmaya yarar. İki ayrı veri setinden oluşturulan vektörler “concatenate” butonu ile birleştirilir. “Data table” butonu verileri tablolar halinde görmek için kullanılmaktadır. Makine öğrenmesi algoritmaları da kendi isimleriyle adlandırılan butonlar vasıtasıyla programda uygulanmaktadır. “Test and score” butonu, uygulanan algoritmalar sonucunda elde edilen çeşitli test sonuçlarını barındırmaktadır. En sondaki “confusion matrix” butonuysa, adından anlaşılacağı üzere, test sonuçlarının etkinliğini gösteren confusion matrixi içermektedir. Confusion matrix elde edilen test sonucunun; doğru ve yanlış tahminlere göre başarı oranlarını gösteren ve makine öğrenmesi tekniklerinin gerçek başarısını ortaya çıkaran bir yöntemdir.

Veri setindeki yaprak fotoğraflarını seçerken yaprak görsellerinin sahip olması gereken kriterler:

- İyi aydınlatılmış ortamda çekilmesi
- Görselde yalnızca tek yaprağın odakta olması
- Arka planın mümkün olduğunca sade ve beyaz olması,

gerekmektedir.



Şekil 4.2. Kriterlere uymayan yaprak görseli.

Şekil 4.2.’deki görsel makine öğrenmesi için kötü bir veridir. Bunun sebebi odakta birden çok yaprak bulunması ve arka planın karışık olmasıdır.



Şekil 4.3. Kriterlere uymayan yaprak görseli.

Şekil 4.3.'deki görsel makine öğrenmesi için kötü bir veridir. Bunun sebebi odakta birden çok yaprak bulunması ve arka planın bulanık olmasıdır.



Şekil 4.4. Kriterlere uymayan yaprak görseli.

Şekil 4.4.'deki görsel makine öğrenmesi için kötü bir veridir. Bunun sebebi arka planın bulanık olmasıdır.



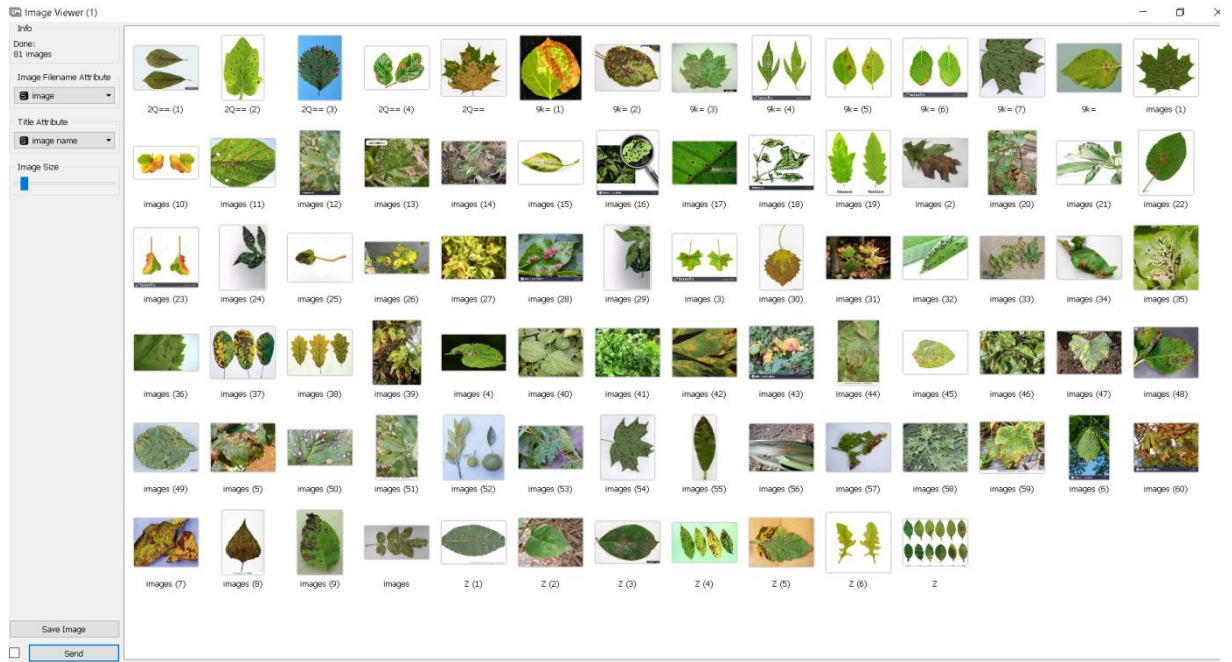
Şekil 4.5. Kriterlere uymayan yaprak görseli.

Şekil 4.5.'deki görsel makine öğrenmesi için iyi bir veridir bunun sebebi odakta yalnızca bir yaprak vardır ve arka plan sade, beyazdır.

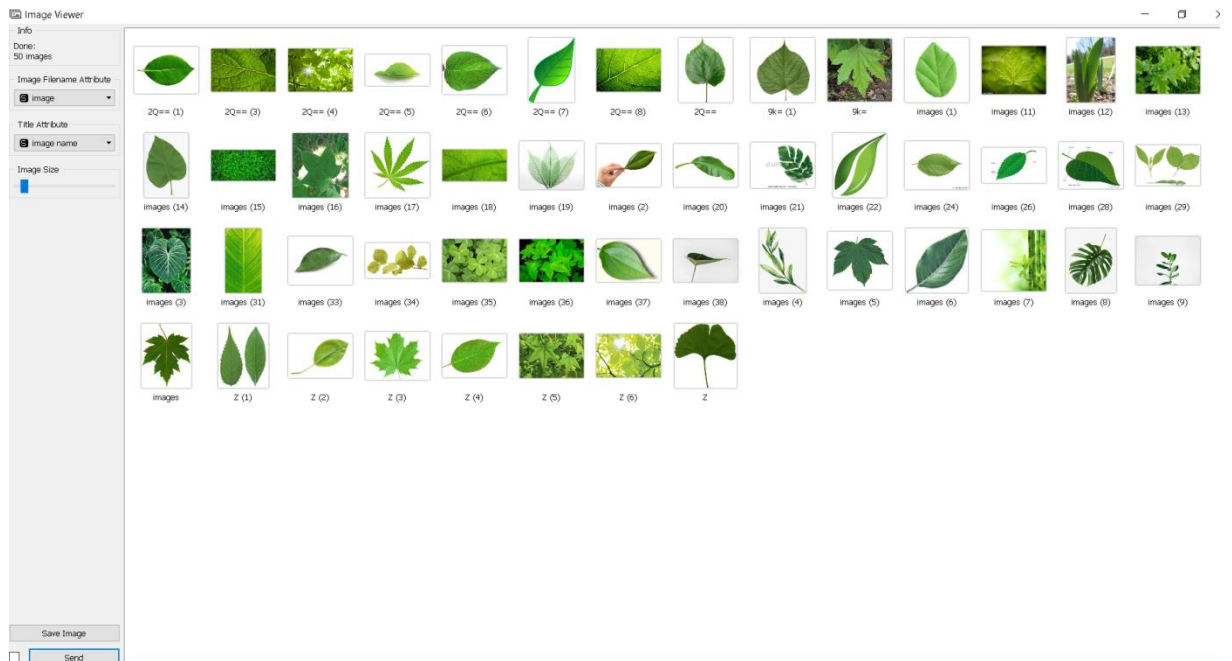
Kulancının uygulamaya girdi olarak vereceği yaprak görsellerini çekerken yaprağın arkasına kağıt ve benzeri bir materyal tutarak başarı oranını artırabilecektir.

Şekil 4.6'da hastalıklı bitkiler sınıfına ait görsel verileri görülmektedir. Veriler çok çeşitli bitki türlerinden ve çok çeşitli hastalık türlerinden seçilmiştir. Görsellerin Kriterlere uygun olmasına dikkat edilmiştir.

Şekil 4.7'de sağlıklı bitkiler sınıfına ait görsel verileri görülmektedir. Veriler çok çeşitli bitki türlerinden seçilmiştir. Görsellerin Kriterlere uygun olmasına dikkat edilmiştir.



Şekil 4.6. Hastalıklı Bitki Resimleri



Şekil 4.7. Sağlıklı Bitki Resimleri

4.2. Görüntülerin Vektörlere Dönüştürülmesi

untitled	untitled	untitled	Source ID	n0	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11
1	0			0.212	0.115	0.055	0.002	0.767	0.076	0.117	0.117	0.294	0.622	0.159	0.18
2	0			0.340	0.000	1.474	0.495	1.216	0.035	0.502	0.073	0.001	1.395	0.053	0.73
3	0			0.353	0.602	0.703	0.000	1.512	0.114	0.269	1.236	0.512	1.115	0.197	0.10
4	0			0.182	0.527	1.173	0.019	0.307	0.206	0.417	0.650	0.008	0.626	0.035	0.33
5	0			0.069	0.497	0.163	0.260	0.513	0.072	0.237	1.121	0.340	0.864	0.175	0.12
6	0			0.100	0.048	0.382	0.352	0.470	0.114	0.206	0.646	0.304	0.426	0.320	0.30
7	0			0.479	0.533	0.264	0.051	0.133	0.064	0.292	0.321	0.293	0.377	0.424	0.10
8	0			0.079	0.401	0.269	0.066	0.551	0.161	0.401	0.618	0.125	0.515	0.090	0.45
9	0			0.102	0.115	0.499	0.076	0.070	0.306	0.007	0.214	0.106	0.379	0.007	0.15
10	0			0.045	0.107	0.502	0.320	0.782	0.106	0.227	0.347	0.032	0.789	0.195	0.18
11	0			0.104	0.272	0.567	0.084	0.299	0.033	0.277	0.483	0.042	0.301	0.178	0.81
12	0			0.022	0.531	0.234	0.171	1.362	0.084	0.126	0.901	0.435	0.417	0.100	0.45
13	0			0.010	0.120	0.403	0.010	0.235	0.063	0.215	0.100	0.303	0.165	0.177	0.37
14	0			0.031	0.357	0.061	0.225	0.889	0.104	0.122	0.810	0.390	0.665	0.287	0.02
15	0			0.569	0.080	0.164	0.107	0.257	0.410	0.517	0.529	0.122	0.878	0.785	1.03
16	0			0.105	0.043	0.230	0.013	1.572	0.048	0.051	0.075	0.163	0.613	0.081	0.23
17	0			0.245	0.237	0.717	0.150	1.371	0.095	0.008	0.137	0.197	0.302	0.060	0.17
18	0			0.432	0.285	0.840	0.380	1.994	0.206	0.163	0.186	0.369	0.245	0.051	0.25
19	0			0.306	0.638	0.293	0.424	1.258	0.548	0.227	0.172	0.000	0.107	0.037	0.20
20	0			0.060	0.103	0.209	0.029	0.011	0.046	0.189	0.199	0.030	0.067	0.199	0.35
21	0			0.263	0.228	0.837	0.177	0.536	0.065	0.468	0.015	0.238	0.331	0.052	0.05
22	0			0.082	0.524	0.227	0.665	1.494	0.083	0.341	0.027	0.055	0.298	0.027	0.36
23	0			0.516	0.190	0.198	0.026	0.255	0.344	0.632	0.284	0.046	0.420	0.009	0.11
24	0			0.229	0.047	0.603	0.007	0.449	0.119	0.369	0.811	0.162	0.428	0.475	0.11
25	0			0.169	0.351	0.687	0.101	0.576	0.210	0.623	0.469	0.150	0.398	0.089	0.94
26	0			0.529	0.766	0.172	0.626	1.876	0.341	0.080	0.454	0.019	0.089	0.117	0.32
27	0			0.139	0.253	1.123	0.019	0.176	0.157	0.281	0.440	0.045	0.465	0.145	0.78
28	0			0.071	0.326	0.565	0.005	0.335	0.250	0.221	0.269	0.088	0.686	0.050	0.62
29	0			0.260	1.122	0.470	0.008	0.395	0.121	0.023	0.005	0.215	0.185	0.403	0.11
30	0			0.182	0.571	0.740	0.017	0.430	0.619	0.430	1.003	0.329	0.434	0.208	1.71
31	0			0.055	0.467	0.116	0.156	0.089	0.252	0.435	0.134	0.322	0.061	0.097	0.28
32	0			0.729	0.818	0.282	0.358	0.236	0.446	0.293	0.707	0.019	0.849	0.142	0.75
33	0			0.584	0.114	0.850	0.095	1.057	0.695	0.047	0.022	0.084	0.152	0.027	0.11
34	0			0.371	0.237	0.703	0.264	1.039	0.215	0.187	0.469	0.051	0.216	0.160	0.51
35	0			0.129	0.390	0.399	0.214	0.818	0.590	0.607	0.835	0.065	0.542	0.061	1.35
36	0			0.169	0.033	0.494	0.104	0.735	0.440	0.334	0.165	0.115	0.184	0.063	0.60
37	0			0.408	0.770	0.270	0.460	1.025	0.300	0.294	0.747	0.281	0.633	0.179	0.30
38	0			0.939	0.313	0.168	0.469	3.266	0.385	0.055	0.496	0.929	0.189	0.109	0.17

Şekil 4.8. Resimlerin Vektörlere Dönüştürülmesi

Şekil 4.8’de görüldüğü üzere, 131 örnek resim kullanılmıştır. Bu resimlerden, 2048 özellik çıkarılmıştır. Resimler vektörlere dönüştürülüp CSV formatında kaydedilerek kullanıma hazır hale getirilmiştir.

4.3. Algoritmaların Uygulanması

Test & Score (1)		Evaluation Results					
Sampling		Method	AUC	CA	F1	Precision	Recall
<input checked="" type="radio"/> Cross validation	Number of folds: 10	Logistic Regression	0.955	0.878	0.837	0.877	0.878
<input checked="" type="checkbox"/> Stratified		kNN	0.833	0.763	0.710	0.773	0.763
<input type="radio"/> Random sampling	Repeat train/test: 10	Random Forest Learner	0.864	0.779	0.642	0.790	0.779
<input type="checkbox"/> Stratified	Training set size: 66 %	Naive Bayes	0.835	0.809	0.786	0.842	0.809
<input type="radio"/> Leave one out		SVM Learner	0.832	0.840	0.792	0.840	0.840
<input type="radio"/> Test on train data							
<input type="radio"/> Test on test data							
Target Class							
(Average over classes)							
Report							

Şekil 4.9. Test Sonuçları

4.4. Sonuçların Karşılaştırılması

Şekil 4.9’da uygulanan orange3 aracını kullanarak 5 farklı algoritmaya ait test sonuçları şekil 4.5’ te görülmektedir. Sınıflandırma başarısı (classification accuracy), CA başlığı altında gösterilmektedir.

F1 skoru istatistiksel terimler olan precision ve recall’ın ağırlıklı ortalamasıdır. hatalı pozitif (false positive) ve hatalı negatif (false negative) sayıları hesaba katarak bulunur. Sınıf verileri dengesiz olduğunda kullanışlıdır.

Precision doğru pozitiflerin bütün pozitiflere oranıdır.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall doğru pozitiflerin, doğru pozitif ve hatalı negatiflerin toplamına oranıdır.

$$\text{Recall} = \frac{tp}{tp + fn}$$

	0	1	Σ
Actual 0	74	7	81
Actual 1	9	41	50
Σ	83	48	131

Şekil 4.10. Logistic Regression Confusion Matrix

Şekil 4.10’de görüldüğü üzere. Logistic regression algoritması uygulandığında, 74 veriye hastalıklı tespiti yapılmış ve doğru çıkmıştır. 41 tane de doğru bir şekilde sağlıklı bitki tespiti yapılmıştır. 131 veride 16 tane hata yapılmıştır.

Actual		0	1	Σ
	0	62	19	81
	1	12	38	50
	Σ	74	57	131

Şekil 4.11. KNN Confusion Matrix

Şekil 4.11’de görüldüğü üzere. KNN algoritması uygulandığında, 62 veriye hastalıklı tespiti yapılmış ve doğru çıkmıştır. 38 tane de doğru bir şekilde sağlıklı bitki tespiti yapılmıştır. 131 veride 31 tane hata yapılmıştır.

Actual		0	1	Σ
	0	76	5	81
	1	24	26	50
	Σ	100	31	131

Şekil 4.12. Random Forest Confusion Matrix

Şekil 4.12’de görüldüğü üzere. Random Forest algoritması uygulandığında, 76 veriye hastalıklı tespiti yapılmış ve doğru çıkmıştır. 26 tane de doğru bir şekilde sağlıklı bitki tespiti yapılmıştır. 131 veride 29 tane hata yapılmıştır.

Actual		0	1	Σ
	0	60	21	81
	1	4	46	50
	Σ	64	67	131

Şekil 4.13. Naive Bayes Confusion Matrix

Şekil 4.13’de görüldüğü üzere. Naive Bayes algoritması uygulandığında, 60 veriye hastalıklı tespiti yapılmış ve doğru çıkmıştır. 46 tane de doğru bir şekilde sağlıklı bitki tespiti yapılmıştır. 131 veride 25 tane hata yapılmıştır.

	0	1	Σ
Actual 0	70	11	81
Actual 1	10	40	50
Σ	80	51	131

Şekil 4.14. SVM Confusion Matrix

Şekil 4.14’de görüldüğü üzere. Naive Bayes algoritması uygulandığında, 70 veriye hastalıklı tespiti yapılmış ve doğru çıkmıştır. 40 tane de doğru bir şekilde sağlıklı bitki tespiti yapılmıştır. 131 veride 21 tane hata yapılmıştır.

Orange programı yardımıyla yapılan işlemlerden sonra, Jupyter Notebook üzerinde Python programlama dili kullanılarak; Cross Validation, Neural Network ve Logistic Regression yöntemleri uygulanmıştır. Bu işlemler için, Orange programında kullanılan veri seti kullanılmıştır. Kodlar şekil 4.15-4.18’de gösterilmiştir.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model, datasets

import pandas as pd
X=pd.read_csv("TrainYaprakData.csv")
Y=pd.read_csv("TrainYaprakTarget.csv")
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y,
random_state=2)
```

Şekil 4.15. Veri Setinin Yüklenmesi

Orange’dan farklı olarak veri seti, data ve target olarak ikiye bölünmüştür. Data, verinin özelliklerini tutarken, target verinin sınıf bilgisini tutmaktadır. Daha sonrasında, veri seti, eğitim ve test olarak ayrılmıştır.

```

from sklearn.model_selection import cross_val_score

from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
scores = cross_val_score(logreg, X_train, y_train)
print("cross-validation scores: ", scores)

cross-validation scores: [ 0.82352941  0.875          0.96875   ]

```

Şekil 4.16. Cross Validation

```

logisticregression=LogisticRegression(C=1,random_state=2)

logisticregression.fit(X_train, y_train)
print("training set score: %f" % logisticregression.score(X,Y))
print("test set score: %f" % logisticregression.score(X_test, y_test))

training set score: 0.992366
test set score: 0.969697

```

Şekil 4.17. Logistic Regression

```

from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_moons
mlp = MLPClassifier(solver='adam', random_state=0).fit(X_train, y_train)

print("training set score: %f" % mlp.score(X,Y))
print("test set score: %f" % mlp.score(X_test, y_test))

training set score: 0.977099
test set score: 0.909091

```

Şekil 4.18. Neural Network MLP Classifier

5. SONUÇ VE DEĞERLENDİRME

Proje sürecinin başından bu zamana, veri madenciliği konusunda birçok araştırma yapılmıştır. Bu alanın alt dallarında çeşitli literatür taramaları yapılmıştır. Öğrenilen bilgiler; odada insan tespiti, cinsiyet tespiti, müşterinin davranışlarına yönelik gözetimsiz öğrenme gibi örnekler çözülerek pekiştirilmiştir. Bu son raporda anlatılan, bitkilerde hastalık tespit örneği ile, öğrenilen bilgiler harmanlanarak kullanılmıştır.

Sonuç olarak, makine öğrenmesi ve görüntü işleme yöntemleri, teknik ve teorik bilgiler araştırılarak, pratik uygulamalar ile kavranmıştır. Geline nokta, belirlenen probleme yönelik, özel veriler toplanabilir ve modeller oluşturularak test edilebilir düzeye ulaşılmıştır.

Bitki hastalıklarının tespitine yönelik gerçekleştirilen bu çalışmada, elde edilen test sonuçları incelendiğinde, en iyi sonucu %95 başarı oranıyla logistic regression algoritması elde etmiştir.

Doğru ve yanlış tahminler göz önüne alınarak confusion matriksi incelendiğinde, başarı yüzdesinin, seçilen yaprak fotoğrafına göre ciddi şekilde değiştiği gözlenmiştir. Arka planı düzgün, uygun bir mesafeden güzel bir ışıkla çekilen fotoğraflarda, başarı oranı yüksekken, çözünürlüğü kötü olan ve yaprak dışında başka odak noktaları bulunan kötü fotoğraflarda, başarı oranı düşüktür. Bu durumda, bütün resimler için çeşitli karar filtreleri uygulanarak, kusursuz bir veri seti toplanırsa, başarı oranının yükselmeye sabitleneceği açıktır.

Bu aşamada, bitkilerde hastalık tespiti için çözülen örnekte başarılı bir test sonucu elde edilmiştir. Bundan sonraki süreçte, veri seti büyütülüp, özellik seçim aşamalarına yoğunlaşarak, daha iyi bir test sonucu elde edilmesi gündemdedir. Böylece, günlük hayatta kullanılabilir bir uygulama sunulabilecektir.

Projenin eksik kısımları giderildikten sonra, projenin ikinci döneme sarkması düşünülen sonraki aşamasında, mobil platformda kullanılabilir, bitkilerde hastalık tespit sistemi hayata geçirilecektir. Böylece modern tarımda, büyük bir maliyetten kurtulma fırsatı doğacaktır.

6. KAYNAKLAR

1. İnternet: Tarımsal İlaç Kullanımı. Web: http://www.tuik.gov.tr/PreIstatistikTablo.do?istab_id=2288
2. İnternet: Sannakki S. S. , Rajpurohit V. S. , Nargund V. B. , Kulkarni P. (2013). Diagnosis and Classification of Grape Leaf Diseases Using Neural Networks. Web: https://www.researchgate.net/publication/269332436_Diagnosis_and_classification_of_grape_leaf_diseases_using_neural_networks
3. İnternet: Arivazhagan S. , Shebiah R. N. , Ananthi S. , Varthini S. V. (2013). Detection of Unhealthy Region of Plant Leaves and Classification of Plant Leaf Diseases Using Texture Features. Web: https://jp.mathworks.com/matlabcentral/answers/uploaded_files/25743/FinalAlgorithm.pdf
4. İnternet: Revathi P. , Hemalatha M. (2012). Classification of Cotton Leaf Spot Diseases Using Image Processing Edge Detection Techniques. Web: <http://www.ijcttjournal.org/2017/Volume43/number-3/IJCTT-V43P121.pdf>
5. İnternet: Şeker Ş.E. , Yıldız M. (2016). Veri Madenciliği Araçları(Data Mining Tools). Web: <http://ybsansiklopedi.com/wp-content/uploads/2017/04/veri-m>
6. İnternet: Acar E. , Özerdem M. S. (2011). Zambak Yaprığı İmgelerinde Pas Hastalıklarının GLCM Tabanlı Sınıflandırma Yöntemleri İle Tespiti. Web: <http://www.dicle.edu.tr/muhendislikdergisi/cilt2sayi2/md-12-001.pdf>
7. İnternet: (2017). Makine Öğrenmesi Modern Tarımı Nasıl Değiştirir? Web: <https://agtechtr.wordpress.com/2017/09/19/makine-ogrenmesi-modern-tarimi-nasil-degistirir/>
8. İnternet: Test & Score. Web: <https://docs.orange.biolab.si/3/visual-programming/widgets/evaluation/testandscore.html>