



T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME ile GÖRÜNTÜ SINIFLANDIRMA

Büşra Rümeysa METE

DANIŞMAN
Dr. Öğr. Üyesi Tolga ENSARİ

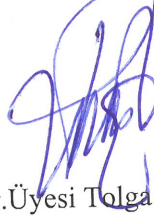
Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

İSTANBUL-2019

Bu çalışma 20.06.2019 Tarihinde ařağıdaki jüri tarafından Bilgisayar Mühendisliğı Anabilim Dalı, Bilgisayar Mühendisliğı Tezli Yüksek Lisans Programı Yüksek Lisans Tezi olarak kabul edilmiştir.

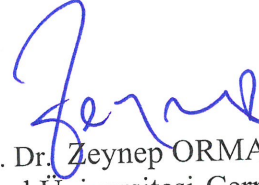
TEZ JÜRİSİ



Dr. Öğr. Üyesi Tolga ENSARİ
İstanbul Üniversitesi-Cerrahpařa
Fakülte



Prof. Dr. Murat TAYLI
İstanbul Kültür Üniversitesi
Mühendislik Fakültesi



Doç. Dr. Zeynep ORMAN
İstanbul Üniversitesi-Cerrahpařa
Mühendislik Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa’nın aboneli olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Kendisiyle çalışmaktan büyük mutluluk duyduğum, yaklaşım tarzı ve rehberliği sayesinde kendimi geliştirmemde payı büyük olan ve güler yüzünü hiçbir zaman eksik etmeyen değerli danışman hocam Dr. Öğr. Üyesi Tolga ENSARİ'ye tüm samimiyetimle teşekkür ediyorum.

Tanıştığım günden bu yana yardımsever karakterinden ödün vermeyip pozitif halleri ve bilgisiyle de ilham olan motivasyon kaynağım Arş. Gör. Doğukan AKSU'ya sonsuz teşekkürler.

Danışmanım sayesinde tanışma fırsatı bulup tez çalışmalarım esnasında bolca tartışarak kafa açtığım değerli arkadaşım Batuhan AŞIROĞLU'ya ve iş arkadaşım Güven AŞCI'ya teşekkür ederim.

Eskişehir'deki öğrencilik yıllarımın bitiminde yolumun kesiştiği ve entelektüel birikimiyle ufkumu açıp manevi desteği ile de güç veren değerli hocam Zekiye DOĞAN, iyi ki tanıdım sizi!

Bugünlere gelmemde payları büyük olan ailem, iyi ki varsınız.

Haziran 2019

Büşra Rümeyza METE

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ.....	ix
SİMGE VE KISALTMA LİSTESİ	x
ÖZET	xi
SUMMARY	xii
1. GİRİŞ	1
2. GENEL KISIMLAR.....	3
2.1 YAPILAN ÇALIŞMALAR	3
2.2 DERİN ÖĞRENME.....	5
2.2.1 YAPAY SİNİR AĞLARI.....	6
2.2.1.1 Aktivasyon Fonksiyonları	7
2.2.1.2 Kayıp Fonksiyonları (Loss Functions)	8
2.2.1.3 İleri Yayılım ve Geri Yayılım Algoritması	9
2.2.2 EVRİŞİMLİ SİNİR AĞLARI	10
2.2.2.1 Evrişim Katmanı	12
2.2.2.2 Ortaklama Katmanı (Pooling Layer).....	19
2.2.2.3 Tam Bağlantı Katmanı (Fully Connected Layer)	20
2.2.3 INCEPTION-V3 MİMARİSİ.....	21
2.2.4 ÖĞRENİM TRANSFERİ.....	25
3. MALZEME VE YÖNTEM.....	29
3.1 VERİ KÜMESİ	30
3.2 BOYUT AZALTMA İLE ÖZELLİKLERİN GÖRSELLEŞTİRİLMESİ	32
3.3 UYGULANAN ÇİÇEK SINIFLANDIRMA YAKLAŞIMI	34
3.4 VERİ ARTIRIMI	35
3.5 ÖZELLİK ÇIKARIMI	36
3.6 SINIFLANDIRMA ALGORİTMALARI.....	37
3.6.1 Tam bağlı katmanlar	38
3.6.2 Doğrusal Destek Vektör Makineleri.....	38

3.6.3 Rassal Orman.....	39
3.6.4 K-En Yakın Komşu (KNN).....	39
3.6.5 Çok Katmanlı Algılayıcı.....	40
4. BULGULAR.....	41
5. TARTIŞMA VE SONUÇ	49
KAYNAKLAR.....	51
EKLER	54
ÖZGEÇMİŞ	57



ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1 Biyolojik sinir hücresi	6
Şekil 2.2 YSA'nın matematiksel modeli	7
Şekil 2.3 İleri beslemeli bir sinir ağı.....	9
Şekil 2.4 Geri beslemeli bir sinir ağı.	10
Şekil 2.5 ESA'da katmanlar.	11
Şekil 2.6 Renkli bir görüntünün üç kanalda gösterimi.	11
Şekil 2.7 Üç kanallı bir görüntünün matris olarak ifadesi.	12
Şekil 2.8 Orijinal görüntüye 3x3 filtre ile evrişim işlemi uygulanması sonucu.	14
Şekil 2.9 5x5 boyutundaki girdiye 3x3 boyutlu filtre uygulandığında elde edilen özellik haritası.	15
Şekil 2.10 5x5 boyutundaki girdiye bir padding değeri ile 3x3 boyutlu filtre uygulandığında elde edilen özellik haritası.	16
Şekil 2.11 Adım uzunluğu iki olarak uygulandığında elde edilen özellik haritası.	17
Şekil 2.12 Üç kanallı görüntüye 3 kanallı bir adet filtre ile evrişim uygulanması.	18
Şekil 2.13 Üç kanallı görüntüye 3 kanallı üç adet filtre ile evrişim uygulanması.....	18
Şekil 2.14 Maksimum ortaklama katmanı.	20
Şekil 2.15 Ortalama ortaklama katmanı.	20
Şekil 2.16 Düzleştirme işlemi (flattening).....	21
Şekil 2.17 1x1 evrişim işlemi.	22
Şekil 2.18 5x5 evrişim işlemi.	23
Şekil 2.19 1x1 evrişim uygulandıktan sonra 5x5 evrişim uygulanması.	23
Şekil 2.20 Inception-v3 [36].....	24
Şekil 2.21 ESA'ya dayanan bir modelin mimarisi.....	26

Şekil 2.22 İnce ayar stratejileri [34].	28
Şekil 3.1 PyCharm konsol ekranından deney sırasında alınan sistem özellikleri.	30
Şekil 3.2 Oxford 102 Flowers veri kümesinden 20 farklı sınıfa ait çiçek görüntüleri.	31
Şekil 3.3 Oxford 17-Flowers veri kümesindeki farklı dört sınıftan örnek görüntüler.....	31
Şekil 3.4 Boyut azaltma ile özelliklerin görselleştirilmesi adımları.....	33
Şekil 3.5 Veri artırımı uygulanmadan veri kümelerinin iki boyutlu özellik düzlemine dönüştürülmesi, (solda) 17 Flowers (sağda)102 Flowers.....	33
Şekil 3.6 Bu çalışmada kullanılan çiçek sınıflandırma sistemi.	34
Şekil 3.7 Veri artırımı öncesi ve sonrasına ait görüntüler.	36
Şekil 3.8 Inception-v3 modelinin özellik çıkarıcı ve sınıflandırıcı kısımları [26].	37
Şekil 4.1 Oxford-17 Flowers veri kümesinde elde edilen sonuçlar.....	41
Şekil 4.2 K-En Yakın Komşu karmaşıklık matrisi.	42
Şekil 4.3 Destek Vektör Makineleri karmaşıklık matrisi.	43
Şekil 4.4 Rassal Orman karmaşıklık matrisi	44
Şekil 4.5 Çok Katmanlı Algılayıcı karmaşıklık matrisi.	44
Şekil 4.6 Oxford-102 Flowers veri kümesinde elde edilen sonuçlar.....	45
Şekil 4.7 İki veri kümesinde sınıflandırıcılara göre elde edilen başarımlar oranları.....	46
Şekil 4.8 İki veri kümesinde sınıflandırıcılara göre tüketilen zaman miktarı (sn).	47

TABLO LİSTESİ

	Sayfa No
Tablo 2.1 [18]'de önerilen Inception-v3 mimarisi	24
Tablo 4.1 Literatürdeki çalışmalarla genel bir karşılaştırma	47



SİMGE VE KISALTIMA LİSTESİ

Simgeler **Açıklama**

S : Sigmoid fonksiyonu

Kısaltmalar **Açıklama**

DWT : Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)

DVM : Destek Vektör Makineleri (Support Vector Machines)

ESA : Evrişimli Sinir Ağları (Convolutional Neural Networks)

DWT : Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)

MR8 : Maximum Response 8

ReLU : Rectified Linear Units (Doğrultulmuş Lineer Üniteler)

SIFT : Scale-Invariant Feature Transform (Değişken Olmayan Özellik Dönüşümü)

TBA : Temel Bileşen Analizi (Principle Component Analysis)

t-SNE : t-Distributed Stochastic Neighbor Embedding (t-Dağıtılmış Stokastik Komşu Gömülmesi)

YSA : Yapay Sinir Ağları (Artificial Neural Networks)

PyCharm : PyCharm, bilgisayar programlamasında, özellikle Python dili için kullanılan tümleşik bir geliştirme ortamıdır. Çek şirketi JetBrains tarafından geliştirilmiştir.

ÖZET

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME İLE GÖRÜNTÜ SINIFLANDIRMA

Büşra Rümeysa METE

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Dr. Öğr. Üyesi Tolga ENSARİ

Çiçek sınıflandırma, benzer şekil, görünüm veya yaprak, ot gibi çevresel nesnelere sahip olan geniş çeşitlilikteki çiçek türleri nedeniyle zorlu bir iştir. Bu tez çalışmasında, çeşitli türdeki çiçekleri ayırt etmek için son teknoloji bir yöntem olan öğrenim transferi stratejilerinden biri kullanılmıştır. Bunun için geleneksel yöntemlerden farklı olarak literatürde sıklıkla kullanılmış iki çiçek veri kümesinin özellikleri, veri artırımı uygulanmak suretiyle ön eğitilmiş bir derin evrişimli sinir ağı yardımıyla çıkarılmıştır. Daha sonra bu özellikler, çeşitli yapay öğrenme algoritmaları ile sınıflandırmış ve bulguları karşılaştırmalı olarak sunulmuştur. Kullanılan sınıflandırıcıların bazıları ile literatürdeki en yüksek başarımlar elde edilmiştir.

Haziran 2019, 69 sayfa.

Anahtar kelimeler: makine öğrenmesi, derin öğrenme, öğrenim transferi, çiçek sınıflandırma

SUMMARY

M.Sc. THESIS

IMAGE CLASSIFICATION WITH DEEP LEARNING

Büşra Rümeysa METE

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Computer Engineering

Supervisor : Assist. Prof. Dr. Tolga ENSARİ

Flower classification is a challenging task due to a wide variety of flower species with similar shape, appearance or environmental objects such as leaves and grass. In this thesis, one of the transfer learning methods, which is the latest technology, has been used to differentiate various types of flowers. For this purpose, unlike traditional methods, the features of the images from the two commonly used flower datasets were extracted with the help of a pre-trained deep convolutional neural network through data augmentation. Then, these features are classified by various machine learning algorithms and their results are presented comparatively. Some of the classifiers used achieved the highest performances in the literature.

June 2019, 69 pages.

Keywords: machine learning; deep learning; transfer learning; flower classification

1. GİRİŞ

Yaşam alanları bakımından çok çeşitli iklimlerde yetişebilen ve dünyanın en önemli üreticileri olan çiçeklerin sayısı birkaç yüz bin türe ulaşmaktadır. Ayrıca, dünyadaki neredeyse tüm böcek türlerini besleyerek besin zincirinde çok önemli bir rol oynamaya devam etmektedirler. Besin zincirinde oynadıkları bu önemli role ek olarak, iyileştirici özelliklerini kullanarak birçok ilaç da üretilmektedir. Bu nedenle, iyi bir çiçek bilgisine sahip olmak ve türlerini tanımak, yeni veya nadir bir bitki türünü tanımak açısından çok önemlidir. Aksi halde, birçok bitki tarım arazisinde zararlı olarak görülerek yok edilebilmekte veya insanlar tarafından toplanarak değerinden çok daha ucuz fiyatlarla satılabilmektedir. Tüm bunlar bitki türlerinin yetersiz şekilde tanınmasından kaynaklanmaktadır. Bununla beraber, doğada yetişen nice bitkinin ekilmesi gerektiği de gerçek bir olgudur. Ayrıca, Kazdağı gök narı, andız otu ve sığırkuyruğu gibi yaşamı belirli bir alanla sınırlı olan ve sadece özel iklim koşullarında yetişen sayısız endemik bitki türünün tanınma kapasitesinin artırılması da ilaç endüstrisinin gelişimini destekleyecektir. Bu alanda yapılan çalışmalar sayesinde az bilinen bitkiler hak ettikleri gerçek değeri görebileceklerdir.

Çiçek sınıflandırma sistemi araştırması botanik alanında önemli bir konu olmakla birlikte botanik çalışmaların da temeli olma özelliğini taşımaktadır. 18. yüzyıldan bu yana Carl Linneaus tarafından bir hiyerarşik bitki sınıflandırma sistemi önerilmiştir [1] ve şu ana kadar tüm dünyada yaygın olarak kullanılmaktadır. Başlangıçta, sınıflandırma metodu sadece 8.000 tür bitkiyi tanımlamaktadır, fakat şimdi 369.000 türü tanıyabilir. Nispeten sistematik sınıflandırma sistemine rağmen, bitki taksonomisi uzmanları, bitkileri sınıflandırmak için hala çok zaman ve enerji harcamalıdır. Geleneksel çiçek sınıflandırma yönteminde, botanikçiler veya taksonomistler ilk önce çiçeğin yaşam alışkanlıklarını gözlemler ve daha sonra genel özellikleri ve morfolojik yapı özelliklerini incelerler. Sonunda kaydedilen bitki örnekleriyle karşılaştırarak çiçek türlerini doğrular. Böylece çiçek sınıflandırma işlemi, zengin mesleki bilgi ve deneyime sahip bilimsel araştırmacıların rehberliğinde başarıyla gerçekleştirilebilir. Buradan hareketle çiçek sınıflandırma görevi genellikle çok az kişinin sahip olduğu uzman ve alana özgü bilgi gerektirdiği rahatlıkla söylenebilir. Bu nedenle, bu tür görevler için otomatik sınıflandırma sistemlerinin geliştirilmesi uzman olmayanlar için çok faydalıdır. Son yıllarda,

birçok kategori ihtiva eden veri kümelerinde çiçek görüntülerinin sınıflandırılması hızla iyileşmiştir.

Yukarıda bahsedilen sebepler doğrultusunda bu tez çalışmasında çiçek sınıflandırma sistemleri üzerine yoğunlaşmıştır. Literatürde geniş ölçekli çiçek veri tabanları azdır. Bununla birlikte her bir çiçek sınıfı için 1000'den çok daha az görüntü içeren ve bu sebeple zengin sayılmayacak çiçek veri kümeleri mevcuttur. Bu tez çalışmasında Oxford 17 ve Oxford 102 çiçek veri kümeleri kullanılarak son teknoloji yöntemlerle çeşitli deneyler yapılmış ve elde edilen bulgular sunulmuştur. Çalışmanın en önemli kısıtı yukarıda değinildiği üzere veri kümesi iken bir diğeri ise çalışmada kullanılan donanımın teknik kısıtlarıdır. Derin öğrenme perspektifinden bakıldığında, görüntü sınıflandırma problemi transfer öğrenme yoluyla çözülebilir. Bu tez çalışmasında son teknoloji yöntemlerden biri olan öğrenim transferinin ilerideki bölümlerde ayrıntıları verilmiş olan metotlarından biri olan ön eğitilmiş bir model yardımıyla özellik çıkarıcı olarak kullanılmasına dayalı stratejisi kullanılmıştır. Böylece veri kümelerinin elde edilen özellikleri ile çeşitli makine öğrenmesi algoritmaları kullanılarak çeşitli sınıflandırma deneyleri yapılmıştır. Bu tez çalışmasında elde edilen sonuçların, literatürdeki başarımlardan daha iyi olduğu görülmüştür.

2. GENEL KISIMLAR

Bu kısımda çiçek sınıflandırma üzerine yapılan literatür taramasına, bu alanda geçmişte ve günümüzde popüler olan tekniklere gerek özet olarak gerekse ayrıntılı olarak yer verilmiştir. Tez çalışmasında günümüzün en iyi teknolojileri kullanıldığından genel bölümlere derin öğrenme başlığı ile devam edilmiş olup derin öğrenmenin dolayısıyla makine öğrenmesinin temeli olan yapay sinir ağlarına, yapay sinir ağlarının özel bir kullanımı olan evrişimli sinir ağlarına, deneylerde kullanılan ön eğitilmiş evrişimli sinir ağı modeli olan Inception-v3 mimarisine ve öğrenim transferi yaklaşımlarına yer verilmiştir.

2.1 YAPILAN ÇALIŞMALAR

Çiçek sınıflandırmada öncü araştırmacılar, görüntüleri parçalara ayırmaya ve onları yapay olarak adlandırılabilir şekilde seçmeye yoğunlaşmışlardır. Bu geleneksel yöntem, günümüzde bilgisayar biliminin ve onunla birlikte gelen teknolojinin gelişmesi ile durmadan artmakta olduğu için daha ilkeldir; çünkü tam otomatik bir süreçte çalışmamaktadır ve insani müdahale gerektirmektedir. Ayrıca, bu yöntemle yeterince yüksek doğruluk elde edilememektedir.

Nilsback ve Zisserman [2], 2004 yılında çiçekleri renkli görüntülere ayırmak için iki basamaklı bir segmentasyon modeli önermiştir. İlk basamak, ön ve arka planı ayrılması işleminden ibarettir. İkinci basamak ise çiçek taç yapısının çıkarılması işleminin gerçekleştirilmesidir. Yazarlar, bu segmentasyon algoritmasının çeşitli deformasyonlara karşı toleranslı ve genel olarak herhangi bir çiçek sınıfı için uygulanabilir olduğunu belirtmişlerdir. Çiçeğin rengi, görüntü sınıflandırmada çok önemli bir rol oynamaktadır, çünkü bölümlenme ve tanıma konusunda ek bilgi vermektedir [3]. Das ve Riseman [4], çiçeğin renginin, çiçek bölgesinde bulunan renk isimleri ve bunların nispi oranları ile tanımlandığını belirtmiştir. Bununla beraber bazı çiçek türlerinin farklı renkleri olabilir. Örnek olarak, aynı tür ebegümeci türleri kırmızı, sarı gibi farklı renklere sahipken orkide gibi bitkiler ise farklı dokulara sahip olabilmektedir. Doku da aynı zamanda çiçek görüntülerini sınıflandırmada önemli bir rol oynamaktadır; çünkü bir görüntü bölgesinde tekrar tekrar meydana gelen bağlı piksel setinin gri seviyelerinin dağılımı hakkında bilgi taşır. 2006 yılında Oxford Üniversite'sinden Nilsback ve Zisserman [5]

çiçek görüntülerinin renk, şekil ve doku özelliklerini temsil eden görsel kelimeleri çıkartarak bir çiçek sınıflandırma sistemi tasarlamışlardır. Renk sözlüğünü çıkarmak için, her bir çiçek görüntüsün HSV renk uzayına eşlenip ve her bir eğitim görüntüsünün HSV değerleri kümelenecek rengin kelime olarak ele alınması sağlanmıştır. Çiçek görüntülerinin şekil özelliklerinin kelime olarak ele alınması için SIFT tanımlayıcılarını kullanmışlardır. Doku özelliklerinin kelime olarak ele alınması için ise MR8 filtre bankasının farklı yönlerdeki tepkileri kullanılmıştır. Ayrıca yazarlar, farklı özelliklerin etkisini incelemek için bu üç görsel kelimenin hepsinin farklı ağırlıklardaki kombinasyonlarını kullanmışlardır. Nilsback ve Zisserman [5], her biri 80 görüntü içeren 17 tür veri kümesini göz önünde bulundurmuş ve üç özelliğin birleşimi için %71.76 doğruluk oranı elde etmişlerdir. Diğer yandan Tseng ve Su [6], 2005 yılında yayımladıkları çalışmalarında, farklı olarak renk, doku ve şekil özellikleri kullanmak yerine renk kümelemesi ve uzamsal bilgi özellikleri ile çok seviyeli ilişkilendirme kuralları kullanarak yeni bir görüntü sınıflandırma yöntemi önermişlerdir. Bu bağlamda doğruluk ve kapsam olarak tanımladıkları iki ölçüm birimini temel alarak doğruluk ve kapsam değerleri üzerinden bulgularını paylaşmışlardır. 2006 yılında Cho ve Lim [7], çiçek görüntülerini karakterize etmek için renk ve şekil çiçek özelliklerini içerik tabanlı görüntü alma sistemi geliştirmek için kullanmıştır. Arama verimliliğini artırmak için görüntü veri tabanını kümelemek için yeni bir Virüs Enfeksiyon Kümelemesi önermişlerdir. Sonuç olarak, hem renk hem de şekil özelliklerini kullanarak kümelemenin, yalnızca renk veya şekle göre ayrı ayrı kümelemekten daha iyi sonuç elde edeceğini açıklamışlardır. 2008 yılında Nilsback ve Zisserman [8] çalışmalarında benzer sınıfların geniş bir veri kümesinde sınıflandırma performansını iyileştirmek, sınıflandırma doğruluğunun geniş bir veri kümesinde etkisini incelemek ve özellik kombinasyonlarını araştırmak için her biri 40 ila 250 örnek içeren 103 sınıflı bir veri kümesini değerlendirmişlerdir. Özellik olarak şekil, doku, renk ve taç dağılımı özelliklerini kullanmışlardır. Birden fazla çekirdek kullanan SVM sınıflandırıcı ile %72.8 doğruluğa ulaşarak performansı önemli ölçüde iyileştirdiklerini göstermişlerdir. 2010 yılında Guru ve diğ. [9], sınıflandırıcı olarak olasılıksal sinir ağı kullanarak çiçek görüntülerinin sınıflandırılmasında doku özelliklerinin etkisini araştırmışlardır. Aynı yıl yine Guru ve diğ. [10], KNN sınıflandırıcısını kullanarak çiçekler için otomatik bir sınıflandırma modeli ortaya koymuştur. 2011 yılında Mukane ve diğ. [11], bulanık mantık sınıflandırıcı kullanarak ölçek değişmezliği doku görüntüsü alımı için DWT ve GLCM tabanlı özellik seçimini sunmuştur. 2016 yılında Liu ve diğ. [12], çalışmalarında el yapımı görsel özellikleri kullanan diğer

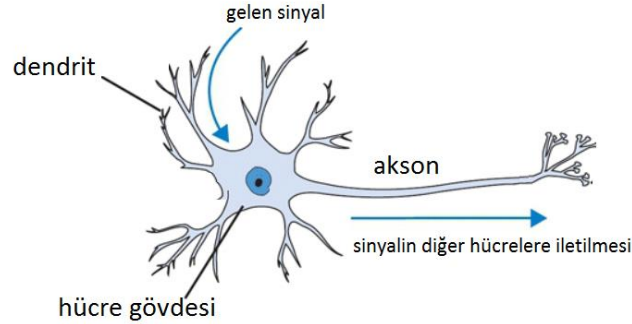
yöntemlerden farklı olarak, çiçeklerin sınıflandırılmasında doğruluğu artırmak için evrişim sinir ağını (CNN) temel alan yeni bir çerçeve önermiştir. Bu yöntemde çiçek sınıflandırması için iyi özellikleri otomatik olarak öğrenmek için evrişim sinir ağını kullanmışlardır. Kendi veri kümelerinde %76.54 başarımlarına, Oxford-102 Flowers veri kümesinde ise %84.02 başarımlarına ulaşmışlardır. 2017'de, Xia ve Xu [13], çalışmalarında çiçeklerin sınıflandırılması amacıyla Google'ın önceden eğitilmiş modeli GoogLeNet'i inception-v3 modülüyle kullanmıştır. Yazarlar Oxford 17 Flowers ve Oxford 102 Flowers isimli iki veri kümesini kullanmıştır ve sırasıyla %95 ve %94 olan oldukça yüksek doğruluk oranlarını elde etmişlerdir. 2018 yılında Wu ve diğ. [14], Oxford-17 ve Oxford-102 çiçek veri kümesi için VGG-16, VGG-19, Inception-v3 ve ResNet50 modellerine dayanarak bir ESA başlatmak suretiyle transfer öğrenme kullanmışlardır. En yüksek başarımlar ResNET50 modeli ile 17 sınıflı veri kümesi için %95.29, 102 sınıflı veri kümesi için ise %96.57 olarak elde edilmiştir. 2018 yılında yapılan başka bir çalışmada Hiary ve diğ. [15], çok çeşitli türlerin çiçeklerini ayırt etmek için iki adımlı yeni bir derinlemesine öğrenme sınıflandırıcısı önermiştir. Çiçek bölgesini çevresindeki minimum sınırlama kutusunun yerleştirilmesine izin vermek için otomatik olarak bölmek için önerdikleri çiçek segmentasyon yaklaşımı, tamamen evrişimli bir ağ çerçevesinde ikili bir sınıflandırıcı olarak modellenmiştir. Sonrasında ise farklı çiçek türlerini ayırt etmek için bir evrişimli sinir ağı sınıflandırıcısı oluşturmuşlardır. Metotlarını iyi bilinen üç çiçek veri kümesinde değerlendirmişler ve elde ettikleri sınıflandırma sonuçlarının tüm veri kümelerinde %97'yi aştığını belirtmişlerdir. Oxford 17 çiçek veri kümesinde %98.5 başarımlar elde ederken Oxford 102 çiçek veri kümesinde ise %97.1 başarımlar yakalamışlardır.

2.2 DERİN ÖĞRENME

Derin Öğrenme, yapay sinir ağları olarak bilinen beynin yapısı ve işlevinden ilham alan algoritmalarla ilgilenen bir makine öğrenmesi alt disiplini. Bu bölümde öncelikle yapay sinir ağlarına giriş yapılmış, ardından evrişimli sinir ağları üzerinde durulmuştur. Devamında derin evrişimli sinir ağlarından bahsedilip son olarak da öğrenim transferi metotlarıyla bölüm sonlandırılmıştır.

2.2.1 YAPAY SİNİR AĞLARI

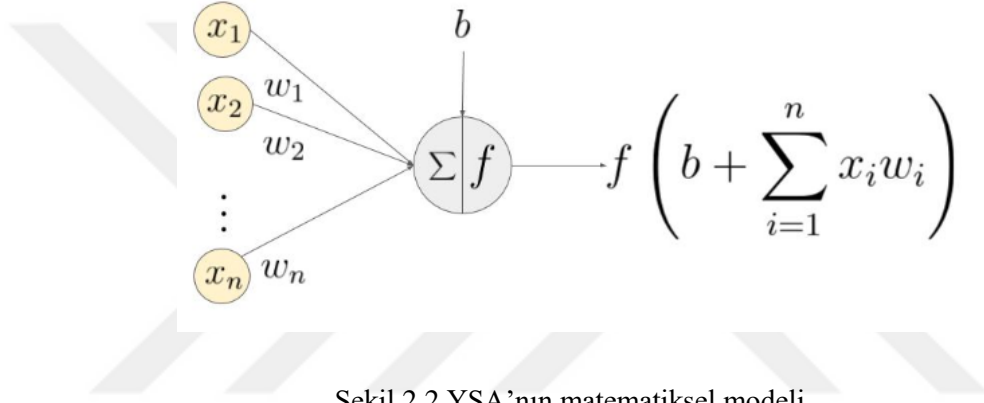
Yapay Sinir Ağları (YSA) üç temel bölümden oluşan bir matematiksel algoritmadır: giriş katmanı, gizli katman(lar) ve çıkış katmanı. Giriş katmanı, sistem girdilerinden oluşur. Bu katman basitçe girdilerin vektörü olarak düşünülebilir. Gizli katman YSA'nın özünü temsil eden katmandır. Bu katmanda nöron adı verilen birçok birim bulunur. Nöronlarda girdileri işlemek ve uygun çıkışlar üretmek üzere birtakım matematiksel işlemler gerçekleşir. Beyinde bulunan biyolojik nörondakine benzer şekilde gizli katmandaki nöron da bir önceki katmandan aldığı bir dizi değeri bir sonraki katmana gönderir. Nörondan gelen ve gönderilen değerler nörondan gelen değeri taşıyan kanalın ağırlık değerine bağlı olarak değişir. Kanalın ağırlık değeri ise ağırlık değerinin önceki nörondan gelen değer ile çarpımını ifade eder. Şekil 2.1'de beyinde bulunan biyolojik nöronun yapısı gösterilmiştir.



Şekil 2.1 Biyolojik sinir hücresi

Biyolojik sinir hücresine diğer sinir hücrelerinden gelen girdiler Şekil 2.1'de görülen *dendrit*'ler aracılığı ile sinir hücresinin gövde kısmına iletilir. Kararlılık halinin bozulmasıyla birlikte sinir hücresinin içinde kimyasal bir sürecin başlatılması tetiklenmiş olur. Bu süreçle beraber girdiler bir sonraki sinir hücresine aksonlar aracılığı ile iletilir. Sinir hücresinin akson ucu ile bir sonraki hücrenin *dendrit* arasındaki kısım sinaptik boşluk olarak bilinir. Bu boşluğa bilginin iletimi sağlayacak iletken bir madde dolar. Böylece bilginin geçişi sağlanır. Sinir hücrelerinin arasındaki mevcut sinaptik ilişkinin değişerek yeni bir ilişkinin kurulması öğrenme sürecidir.

YSA da benzer şekilde çalışmaktadır: Şekil 2.2’de gösterilmiş olan w_1x_1 , w_2x_2 diğer sinir hücrelerinden gelen verileri temsil etmektedir. Yapay sinir hücresi aldığı bu veriyi aktivasyon fonksiyonundan geçirip elde ettiği hesaplama sonucunu; yani çıktıyı diğer sinir hücrelerine iletir. Yapay sinir hücreleri biyolojik sinir hücrelerinin birebir kopyası değildir. Biyolojik sinir hücreleri oldukça karmaşık yapıdadır ve tek çeşit değillerdir, farklı davranışları vardır. Ancak şu bir gerçektir ki yapay sinir hücreleri biyolojik sinir hücrelerinden ilham alınarak tasarlanmıştır.



Şekil 2.2’de tasvir edilen işlemde W_iX_i bir matris çarpımıdır. Bu çarpıma Bayes değeri eklenir ve çıkan sonuç aktivasyon fonksiyonundan geçirilir. Buradan çıkan değerlere göre hangi sinir hücresi daha aktifse model ona göre tahmin yapacaktır. Buraya kadarki işlemler ileri besleme işlemidir. Modelin eğitilebilmesi için hatalarını inceleyip onları düzenlemesi de gerekmektedir. Bu düzenleme ise ağırlık ve Bayes değerlerinin değiştirilmesiyle yapılmaktadır.

2.2.1.1 Aktivasyon Fonksiyonları

Yapay sinir ağları eğitilirken aktivasyon fonksiyonları kullanılır. Aktivasyon fonksiyonlarının amacı w (ağırlık) ve b (Bayes) değerlerini ayarlamaktır. Şekil 2.2’de gösterildiği gibi elde edilen değer bir aktivasyon fonksiyonundan geçirilir. Bu fonksiyonlar tensörlere etki eden doğrusal olmayan operasyonlardır ve bunlar nöronların tetiklenmesi olarak düşünülebilir. Yüksek bir değer geldiği zaman nöron daha yüksek bir sinyal verir.

Sigmoid Aktivasyon Fonksiyonu

Bir sigmoid fonksiyon, karakteristik bir "S" şekilli eğri veya sigmoid eğriye sahip matematiksel bir fonksiyondur. Sigmoid aktivasyon fonksiyonu, sürekli ve türevi alınabilen bir fonksiyondur. Doğrusal olmaması sebebiyle YSA'da çok sıklıkla kullanılan bir fonksiyon olarak karşımıza çıkmaktadır. Bu fonksiyon (2.1), girdi değerleri için sıfır ile bir arasında bir değer üretilmesini sağlar.

$$S(x) = \frac{e^x}{1 + e^x} - 1 \quad (2.1)$$

ReLU Aktivasyon Fonksiyonu

ReLU günümüzün en popüler aktivasyon fonksiyonlarından biridir. Bu fonksiyon gelen değerlere öncelikle pozitif mi negatif mi diye bakar. Eğer gelen değer sıfırdan küçükse işlem sonucu olarak sıfır verir. Gelen değer sıfırdan büyükse bu değere herhangi bir sıkıştırma veya değiştirme işlemi uygulamaz. Dolayısıyla değer, fonksiyondan olduğu gibi geçer. Diğer yandan bu fonksiyonun sinir sistemimizde bulunan biyolojik nöronlara daha yakın bir yapısı vardır. Biyolojik nöronlar belli bir seviyenin altında gelen sinyalleri görmezden gelir. Bu seviyeye biyoloji de eşik şiddeti denmektedir. Sinyal bu şiddeti geçmezse yok sayılır. Bu fonksiyon ilk olarak 2012'de kullanılmaya başlanmıştır. Her yıl düzenlenen IMAGENET veri kümesi üzerinde yapılan bir yarışmada ilk defa iyi sonuçlar veren AlexNet'te de bu fonksiyon kullanılmıştır. Bu tarihten itibaren fonksiyonun popülerliği de giderek artmıştır.

$$F(x) = \begin{cases} 0; & x < 0 \\ x; & x \geq 0 \end{cases} \quad (2.2)$$

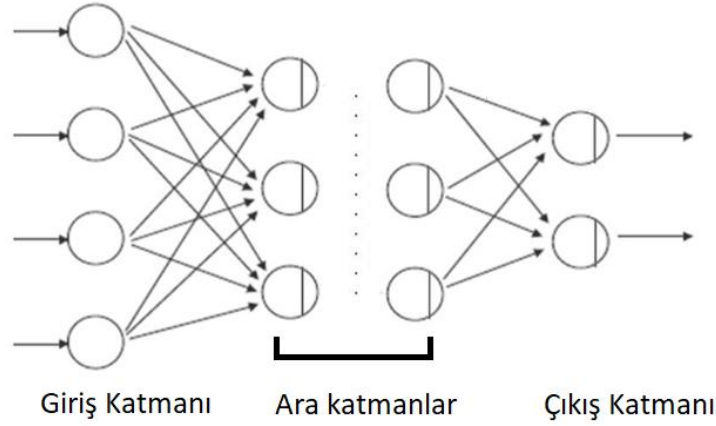
2.2.1.2 Kayıp Fonksiyonları (Loss Functions)

Kayıp fonksiyonları modelin tahmin ettiği değerlerin gerçekteki değerden ne kadar uzak olduğunu hesaplayarak hatanın büyüklüğünü gösterir. Model eğitilirken kayıp değerini sıfıra yaklaştırmak hedeflenir. Model eğitilirken karşılaştığı girdi için sonuç olarak olasılığa benzer değerler verir. Bu olasılıklardan en yükseğine bakılarak da tahmin gerçekleştirilir. Kayıp

fonksiyonu, bu tahminin ne kadar doğru olduğunu daha açık bir ifadeyle bu tahminin gerçeğe ne kadar uzak olduğunu matematiksel olarak hesaplar. Özetle kayıp fonksiyonu sayesinde model hatasını görür ve optimizasyon (iyileştirme) ile daha iyi olmaya çalışır. Bir nevi hatalarından ders alır. Bütün bu kayıp fonksiyonlarının amacı modele ne kadar hata yaptığını göstermektir.

2.2.1.3 İleri Yayılım ve Geri Yayılım Algoritması

İleri yayılım algoritması, bilginin sinir ağında yalnızca, giriş düğümleri sonra gizli katmanlar (tek veya çok sayıda katmanlar) ve son olarak da çıkış düğümleri boyunca ilerlemesi üzerine kuruludur. İki temel adımda çalışır: Girdilerinin ağırlıklı toplamını hesaplar ve ardından toplamı normalleştirmek için bir etkinleştirme işlevi uygular. İleri beslemeli sinir ağları tek yönlü bilgi akışına izin vermektedir. Çok katmanlı ileri beslemeli bir sinir ağı örneği Şekil 2.3’de görülmektedir.

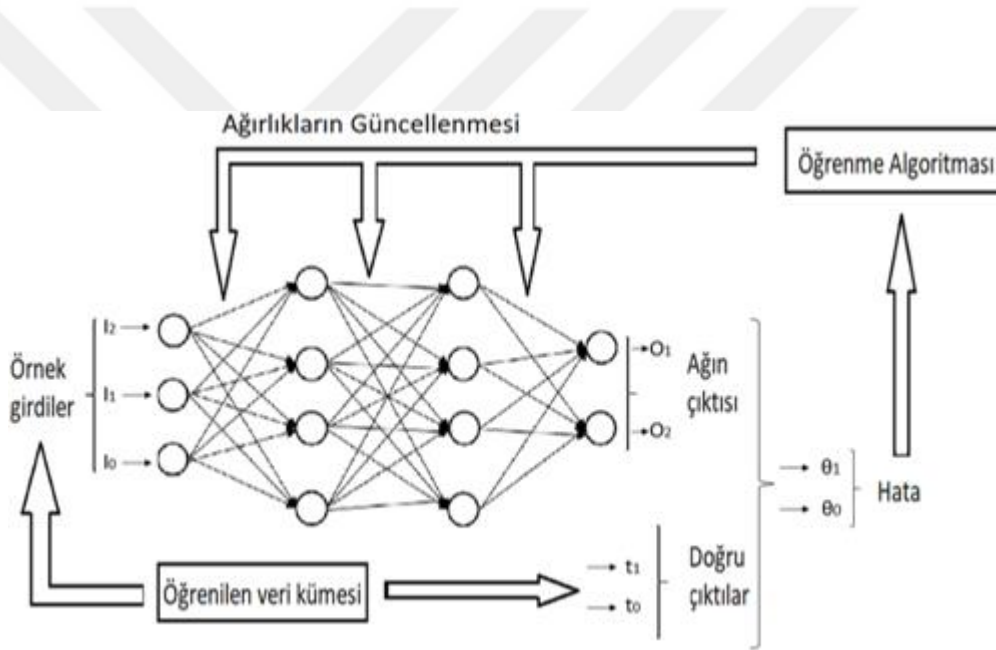


Şekil 2.3 İleri beslemeli bir sinir ağı.

İleri beslemeli yapay sinir ağında, bir katmandaki çıkışlar bir sonraki katmana ağırlıklar üzerinden girdi olarak verilir. Bilgi, ara ve çıkış katmanında işlenir. Ağa, hem örnek girdiler, hem de bu örnek girdilerden elde edilmesi gereken çıktılar verilmektedir. Ağ, bu örneklerden

öğrenerek veya bir diğer ifade ile genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretir.

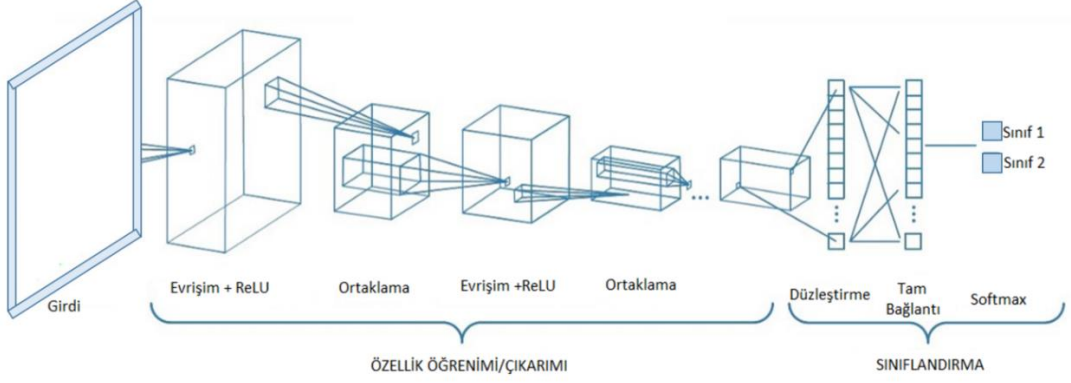
Geri yayılım ise YSA'da her nöronun hataya ne kadar katkısının olduğunu hesaplamak için kullanılır. Geri yayılımla ağırlık ve Bayes değerleri yeniden ayarlanır. Model bir giriş bilgisi alır ve bunu bir sinir ağından geçirerek bir olasılık çıktısı verir. Bu olasılığın ne kadar doğru olduğu hesaplanıp modele hatasının büyüklüğünü gösterilir. Model, hatasına bakar ve ona göre geri gidip ağırlık ve Bayes değerlerini düzenler. Eğitim esnasında bu işlem veri kümesine bağlı olarak binlerce hatta milyonlarca kez gerçekleştirilir. Aşağıda bulunan Şekil 2.4'de iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir.



Şekil 2.4 Geri beslemeli bir sinir ağı.

2.2.2 EVRİŞİMLİ SİNİR AĞLARI

Evrışimli Sinir Ağı (ESA), derin öğrenmede görsel görüntüleri analiz etmek için en sık uygulanan derin sinir ağı sınıfıdır. Bir ESA temel olarak evrişim, ortaklama ve tam bağlantı katmanlarından oluşur. Şekil 2.5'te ESA'daki katmanların bir görselleştirilmesi verilmiştir.

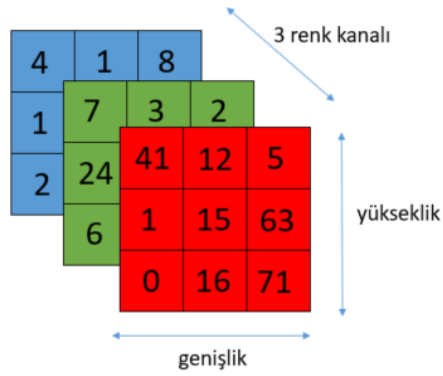


Şekil 2.5 ESA'da katmanlar.

Görüntüler matematiksel olarak matris ile ifade edilir. Bu matrisin her bir elemanı görüntüdeki bir pikselin sayısal değerine karşılık gelmektedir. Renkli bir görüntü için her bir pikselin Kırmızı, Yeşil ve Mavi (RGB) olmak üzere üç ana renk bandı bulunmaktadır. Bu renk bantları literatürde kanal ismiyle bilinmektedir. Şekil 2.6'da renkli bir görüntünün bu üç kanaldaki gösterimi verilmiştir. Üç kanallı bir görüntünün matematiksel olarak gösterimi ise Şekil 2.7'te görülebilir.



Şekil 2.6 Renkli bir görüntünün üç kanalda gösterimi.



Şekil 2.7 Üç kanallı bir görüntünün matris olarak ifadesi.

ESA'da katmanlara ve bu katmanlarda yapılan işlemlere değinmeden önce parametre ve hiper-parametre kavramlarından söz etmek gerekir. Parametreler, modellerde bulunması ve hesaplanması gereken ağırlık ve Bayes değerleridir. Hiper-parametreler ise optimum değerlerinin duruma göre farklılık gösterebilen ve aynı zamanda parametrelerin hesaplanmasına katkı sağlayarak modelin performansını da doğrudan etkileyen değerlerdir. Bunlardan en çok kullanılanlara örnek olarak öğrenme oranı (learning rate), iterasyon sayısı, gizli katman sayısı ve seçilecek aktivasyon fonksiyon verilebilir.

2.2.2.1 Evrişim Katmanı

Evrişimli Sinir Ağları'nın ana yapı taşları, bir görüntüdeki özelliklerin varlığını tespit etmek için kullanılan evrişim katmanlarıdır. Özellikler basit kenarlardan ve eğrilerden kulak, burun ve gözler gibi daha karmaşık yapılara kadar herhangi bir şey olabilir. Evrişim katmanı özellik çıkarıcı katman olarak da adlandırılmaktadır. Çünkü görüntünün özellikleri bu katman ile çıkarılır.

Bir evrişim katmanı, ağırlıkların sabit kare penceresi olan bir çekirdeğin tanımlanması ve bunun tüm giriş görüntüsü boyunca taranmasıyla oluşturulur. Görüntüdeki her konumda, çekirdek penceresindeki görüntü piksellerinde ve çekirdek ağırlıklarında bir evrişim işlemi gerçekleştirir. Bu, basitçe, çekirdek penceresi içindeki piksel değerlerinin toplamı, karşılık gelen çekirdek ağırlığı ile çarpılır.

Evrişim işlemi iki matrisin çok kanal düzeyindeki iç çarpımı (dot product)'ndan farklı bir işlem değildir. Giriş görüntüsü bir matristir. Diğeri ise evrişim filtresi veya çekirdek (kernel) olarak adlandırılan bir matristir. Bu işlemin temel noktası özellik haritası üretme yeteneğidir. Aşağıdaki formülde (2.3) iki matrisin iç çarpımı gösterilmiştir.

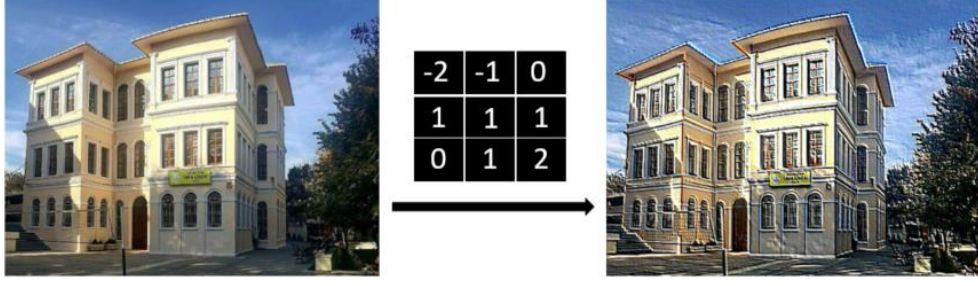
$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 \quad (2.3)$$

Evrişim işleminde, giriş görüntüsü ile onun filtre veya diğeri adıyla çekirdek ile aynı boyuta sahip yerel bir bölgesinin iç çarpımı (dot product) işleminin gerçekleştirilerek çıkış hacminin tek tamsayı değeri elde edilir. Ardından filtre, aynı giriş görüntüsü üzerinde adım uzunluğu (stride) değeri kadar kaydırılarak bu işlem tekrarlanır. Bu tekrarlama, tüm görüntü matrisi boyunca gerçekleştirilir. Sonuç olarak elde edilen çıktı ise bir sonraki katmanın girdisi olur. Ayrıca söz konusu çıktıdaki negatif değerlerin sıfıra çekilmesi için ReLU aktivasyon fonksiyonu kullanılır. Bu sebeple evrişim katmanı ReLU aktivasyon fonksiyonunu da içermektedir.

Çekirdek evrişimi yalnızca ESA'larda kullanılmaz, aynı zamanda diğeri birçok Bilgisayarlı Görü algoritmasının da kilit unsurlarından biridir. Sonuç olarak elde edilen özellik haritası değerleri, giriş görüntüsünün f ve çekirdeğin/filtrenin h ile gösterdiği aşağıdaki formüle (2.4) göre hesaplanır. Sonuç matrisinin satır ve sütun indeksleri sırasıyla m ve n olarak ifade edilmiştir.

$$[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k]f[m - j, n - k] \quad (2.4)$$

Evrişim işleminde temelde çekirdek penceresindeki bir özelliğin varlığı tespit edilmeye çalışılır. Aranılan özellik türü, çekirdek ağırlıklarının değerleri tarafından belirlenir. Elde edilen çıktıya, giriş görüntüsü boyunca verilen bir özelliğin varlığını açıklayan bir özellik haritası adı verilir. Şekil 2.8'de, bir okul binasına ait görüntüye bir filtre uygulanarak elde edilen özellik haritası görülmektedir.



Şekil 2.8 Orijinal görüntüye 3x3 filtre ile evrişim işlemi uygulanması sonucu.

Görüntü tanıma işlemini gerçekleştirmek için birçok farklı özellik türünü birbiriyle birlikte ele almak gerekir ve bu nedenle tek bir evrişimli katman tipik olarak birçok farklı özellik haritası içerir. Çekirdeklerimizin ne tür özellikler arayacağını önceden belirtilmesine gerek yoktur. Ağ eğitirken, ağın doğruluğunu artırmak için çekirdeklerin ağırlıkları ayarlanır ve böylece ağ hangi özelliklerin tespit edilmesinde önemli olduğuna karar verebilir.

Evrişim katmanında basit bir parametre hesabı yapmak için her biri 64 kanallı dört adet 3x3 filtreye sahip olunduğu düşünülürse, toplamda $(3 \times 3) \times 64 \times 4$ yani **2304 adet** hesaplanacak parametre olduğu görülür.

Üst Düzey Özellik Haritası

Evrişimli Sinir Ağları'nın ilk katmanlarında, özellik haritaları yatay veya dikey kenarlar gibi basit yapıları tespit edilirken, daha derin katmanlarda ise gözler, kulaklar veya ağızlar gibi daha karmaşık yapılara ait özellik haritaları elde edilebilir. Bu bölümde bahsedilen üst düzey özellik haritasından kasıt budur. Sözü edilen bu üst düzey özelliklerin elde edilebilmesi, çekirdek filtresinin alıcı alanı kavramıyla açıklanabilir. Birinci evrişim katmanında, 2x2 boyutlu bir filtre giriş görüntüsünden yalnızca 4 piksele erişebilir - bu nedenle bu filtrenin kulakları algılaması mümkün olmaz; ancak kenarları doğru bir şekilde algılayabilir. 2x2'lik bir ortaklama katmanından sonra, ortaklanmış özellik haritasındaki her öge orijinal giriş görüntüsünden 4 piksel etkilenir. Bir sonraki evrişim katmanındaki 2x2 boyutlu bir filtre ise giriş görüntüsünden 16 piksel etkilenir. Evrişim katmanları derinleştikçe, her bir çekirdek giriş görüntüsünün çok büyük bir bölümünü görebilir ve böylece daha karmaşık özellikleri tespit edebilir. Bu nedenle

Geçerli evrişim işleminde görüntü, her evrişim işlemiyle birlikte küçüleceği için, evrişim işlemleri yalnızca sınırlı sayıda yapılabilir. Dahası, çekirdeğin görüntü içinde nasıl hareket ettiğini incelenecek olursa, görüntünün kenarlarında bulunan piksellerin etkisinin görüntünün merkezindekilerden çok daha küçük olduğunu görülecektir. Diğer bir ifadeyle, görüntünün köşesinde bulunan pikseller, evrişim sırasında merkezi piksellere kıyasla yalnızca birkaç kez kullanılır. Bu sebeple, görüntüdeki bilgilerin bir kısmı kaybedilmektedir. Bu gibi sorunların üstesinden gelmek için, görüntüyü ek bir ya da birkaç kenarlıkla çevrelenebilir, yani tüm kenarlara bir ya da birkaç piksel eklenebilir. Diğer bir ifade ile “Aynı Evrişim” işlemi gerçekleştirilebilir. Örneğin şekil 2.10'daki 5x5 boyutundaki girdiye 1 piksel dolgu uygulanırsa görüntünün boyutu 7x7'ye yükseltilmiş olur. Böylece 3x3 filtrelili evrişim çıktısı 5x5 olur. Böylece girdi boyutu değişime uğramaz. Buna aynı evrişim denir.

0	0	0	0	0	0	0														
0	1	2	1	3	2	0					0	7	-1	10	6					
0	3	1	1	2	1	0	*	0	-1	0	=	11	-2	0	4	-2				
0	2	1	1	1	3	0		-1	5	-1		6	0	0	-2	12				
0	0	1	2	1	1	0		0	-1	0		-7	1	4	-4	1				
0	4	1	3	5	0	0						19	-3	7	21	-6				
0	0	0	0	0	0	0														
								3x3 filtre				5x5 özellik haritası								
5x5 girdi 1 dolgu ile 7x7																				

Şekil 2.10 5x5 boyutundaki girdiye bir padding değeri ile 3x3 boyutlu filtre uygulandığında elde edilen özellik haritası.

Önceki örneklerde, filtrenin bir piksel kaydırıldığı varsayılmıştı. Ancak, adım uzunluğu da evrişim katmanı hiper parametrelerinden biridir ve dikkate alınmalıdır. Şekil 2.11'de, evrişim işleminde adım uzunluğunun iki olarak belirlenmesi durumunda nasıl bir özellik haritası elde edildiği görülmektedir.



Şekil 2.11 Adım uzunluğu iki olarak uygulandığında elde edilen özellik haritası.

ESA mimarisi tasarlanırken, alıcı alanların daha az üst üste binmesini isteniyorsa veya daha küçük boyutlarda özellik haritası elde edilmek isteniyorsa, adım uzunluğunu artırmak gereklidir. Bu durumda çıktı matrisinin boyutlarını hesaplamada (2.6)'daki formül kullanılabilir.

$$W_2 = \frac{W_1 + 2P - F}{S} + 1 \quad (2.6)$$

W₁: Girdi tensörünün genişlik ve yüksekliği.

W₂: çıktının genişlik ve yüksekliği.

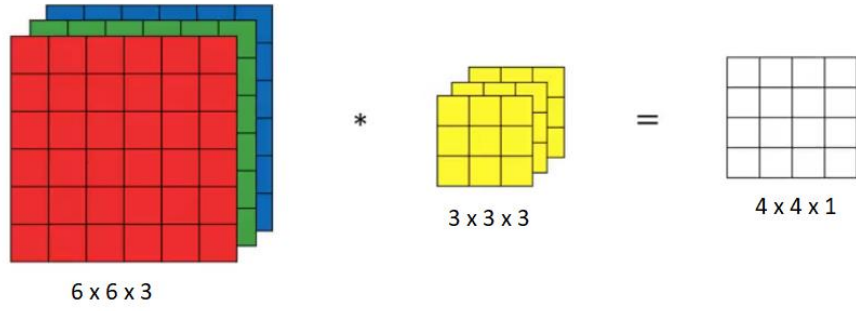
S: Adım uzunluğu (stride).

P: Dolgu miktarı (padding);

F: Filtrenin/çekirdeğin genişlik ve yüksekliği.

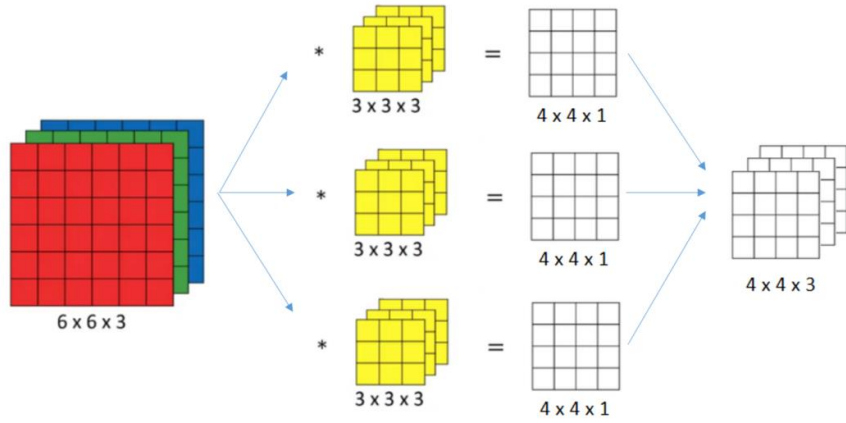
Görüntüde Üç Kanal

Hacimsel dönüşüm, renkli görüntülerle çalışılmasına izin vermekle birlikte, tek bir katmana birden çok filtre uygulanabilmesini de olanaklı kılar. Burada ilk önemli kural, görüntünün ve uygulanmak istenilen filtrenin aynı sayıda kanala sahip olması gerektiğidir.



Şekil 2.12 Üç kanallı görüntüye 3 kanallı bir adet filtre ile evrişim uygulanması.

Temel olarak, Şekil 2.12'te gösterildiği gibi çok ilerlenir; ancak bu kez üç değer uzayından gelen değer çiftleri çarpılır. Aynı görüntü üzerinde birden fazla filtre kullanılması durumunda ise her biri için evrişim işlemi ayrı ayrı yapılır ve elde edilen sonuçlar üst üste yığılarak bir bütün halinde birleştirilir. Böylece bir tensör elde edilmiş olur.



Şekil 2.13 Üç kanallı görüntüye 3 kanallı üç adet filtre ile evrişim uygulanması.

Şekil 2.13'teki gibi, elde edilen tensör –ki buna üç boyut matrisi de denebilir- aşağıdaki denklemi (2.7) sağlar.

$$[n, n, n_c] * [f, f, n_c] = \left[\left[\frac{n + 2p - f}{s} + 1 \right], \left[\frac{n + 2p - f}{s} + 1 \right], n_f \right] \quad (2.7)$$

n: Görüntü boyutu

f: Filtre boyutu

n_c: Görüntünün kanal sayısı.

p: Dolgu sayısı.

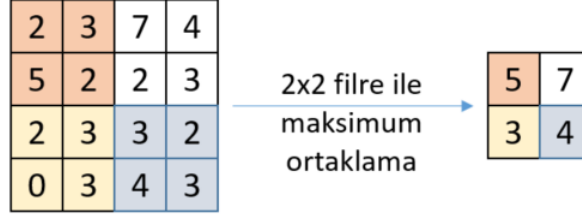
s: Adım uzunluğu.

n_f: Filtre sayısı.

Evrişim işlemlerinde duruma göre sadece bir filtre kullanmak yerine, birden çok filtre de kullanılabilir. İlk filtrenin dikey kenarları ve ikinci filtrenin ise görüntüden yatay kenarları algılayacağı varsayılırsa, yukarıdaki örnekte olduğu gibi bir 4x4x1 çıktısı almak yerine, kullanılan filtrelerin sayısı boyutunda olan bir 4x4x3 çıktısı elde edildiği görülür.

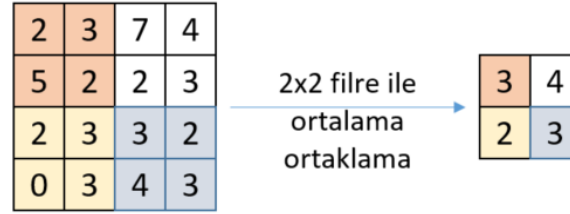
2.2.2.2 Ortaklama Katmanı (Pooling Layer)

Ortaklama katmanı, evrişim katmanından sonra uygulanan bir alt-örnekleme (downsampling) işlemidir. Özellikle tensörün boyutunu azaltmak ve hesaplamaları hızlandırmak için kullanılırlar. Havuzlama katmanına giriş tensördür. Bu katmanda yapılan işlem oldukça basittir: Görüntüyü farklı bölgelere bölerek bu parçaların her biri için maksimum veya ortalama değeri olarak sonuç matrisine koyar. Bunlardan ilki maksimum ortaklama, ikincisi ise ortalama ortaklama olarak adlandırılmaktadır (max pooling ve average pooling). Evrişim katmanında olduğu gibi filtre büyüklüğü ve adım uzunluğu olmak üzere iki tane hiper-parametresi vardır.



Şekil 2.14 Maksimum ortaklama katmanı.

Maksimum ortaklama durumunda, $n * n$ büyüklüğünde bir çekirdek, Şekil 2.14'deki örnekte 2x2 matris boyunca hareket ettirilir ve her pozisyon için maksimum değer alınır ve karşılık gelen değere konulur.



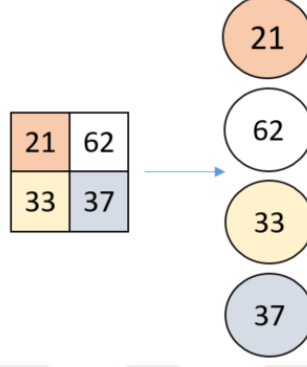
Şekil 2.15 Ortalama ortaklama katmanı.

Şekil 2.15'de gösterilmiş olan ortalama ortaklama durumunda, $n \times n$ büyüklüğünde bir çekirdek matris boyunca hareket ettirilir ve her pozisyon için ortalama tüm değerlerden alınır ve çıktı matrisinin karşılık gelen pozisyonuna konur. Bu işlemler giriş tensöründeki her kanal için tekrarlanarak çıktı tensörü elde edilir. Bu nedenle ortaklama katmanında girişin genişlik ve yüksekliği küçülürken kanal sayısı olarak ifade edilen derinlik aynı kalır.

2.2.2.3 Tam Bağlantı Katmanı (Fully Connected Layer)

Evrişimli sinir ağlarının çoğu, bir veya daha fazla tamamen bağlı katmanla sona erer. Bu, ağın, üst düzey özellik haritalarını son katmanın görüntü sınıflandırmalarının her birine eşlenir. Bu sayede elde edilen üst düzey özellik haritalarına göre en doğru tahmin yapılır. Evrişim ve

ortaklama katmanlarından elde edilen çıktı, tüm değerleri Şekil 2.16’da gösterildiği gibi düzleştirilip bir vektöre dönüştürmek için kullanılan üç boyutlu bir matristir.



Şekil 2.16 Düzleştirme işlemi (flattening).

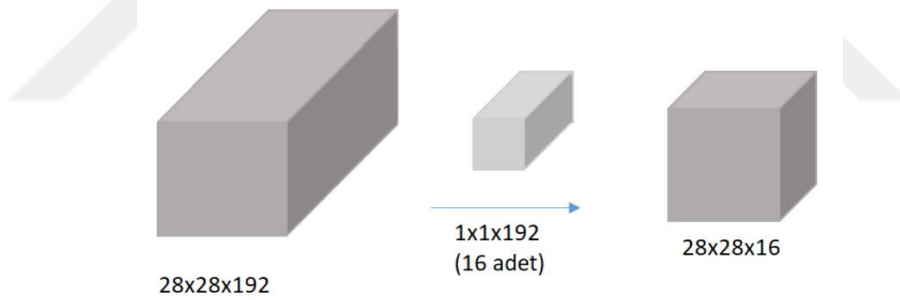
Tam Bağlantı Katmanı, en basit tanımıyla ileri beslemeli bir sinir ağından ibarettir. Tam Bağlantı Katmanının girişi, en sonuncu evrişim veya ortaklama katmanının çıktısının düzleştirilmiş olan vektör halidir. Tam Bağlantı Katmanında bu vektör, YSA’da olduğu gibi Tam Bağlantı Katmanına bağlanır. Tamamen bağlı katmanların içinden geçtikten sonra ise Şekil 2.1’de görüldüğü gibi son katman, girişin belirli bir sınıfta olma ihtimalini elde etmek için kullanılan Softmax aktivasyon fonksiyonundan geçirilir. Bu aşama ağın sınıflandırma bölümüdür. Sonuç olarak, girdi görüntüsünün farklı sınıflara ait olasılıkları elde edilmiş olur. Böylece giriş görüntüleri etiket olarak sınıflandırılabilir. Evrişimli sinir ağlarının temel işleyişi bu şekildedir.

2.2.3 INCEPTION-V3 MİMARİSİ

2013 yılında Min Lin ve diğ. [16], önerdikleri 1x1 evrişim işlemi ile önceki modellerdeki hesaplama karmaşıklıklarına yenilikçi bir çözüm getirmişlerdir. Söz konusu makalede sunulan fikir Google tarafından kullanılarak başarılı sonuçlara ulaşılmıştır. Daha önce, görüntünün genişlik ve yüksekliğini küçültmek için genellikle ortaklama katmanı kullanıldığından bahsedilmişti. Klasik modeller incelendiğinde matrislerin genişlik ve yükseklikleri ortaklama

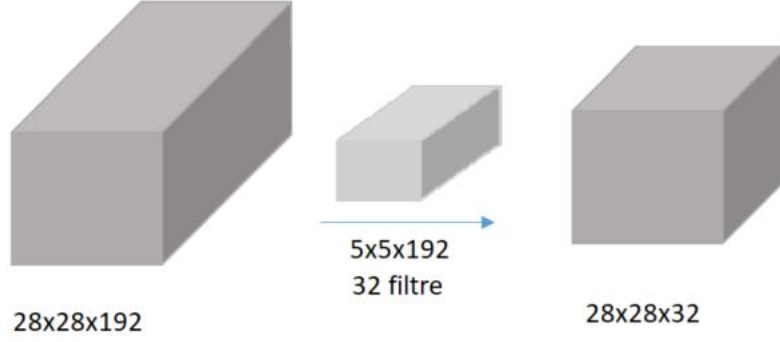
katmanları sayesinde azaltılabildiği; diğer yandan modelin başından sonuna doğru gidildikçe derinliklerin ise arttığı görülür. Ancak makalede önerilen 1x1 evrişim işlemi ile derinliğin boyutu ve dolayısıyla hesaplama karmaşıklığı büyük ölçüde azaltılmış oldu. Bu gelişmenin ardından Google, 1x1 evrişim katmanını kullanıp, derinliği istediği şekilde sınırlandırarak çok iyi ve çarpıcı sonuçlar elde etti. Bu sebeple Inception ağları anlamada 1x1 evrişim işlemini ve Network in Network isimli makaleyi irdelemek oldukça önemlidir ve bilgisayarlı görünümün bir kırılma noktasının bu makale ile gerçekleştiği söylenebilir.

1x1 evrişim işleminin temel fikri görüntüdeki kanal sayısını azaltmaktır. Böylece parametre sayısı da azaltılmış olmaktadır. Normalde 1x1 evrişim işlemi matrisin boyutunu etkilemez; ancak giriş matrisinin çok kanallı olması durumunda 1x1 evrişim işleminin çıkış matrisinin kanal sayısı uygulanan 1x1 evrişim filtresinin kanal sayısına eşit olur. Şekil 2.17’de 1x1 evrişim işlemi görülmektedir.



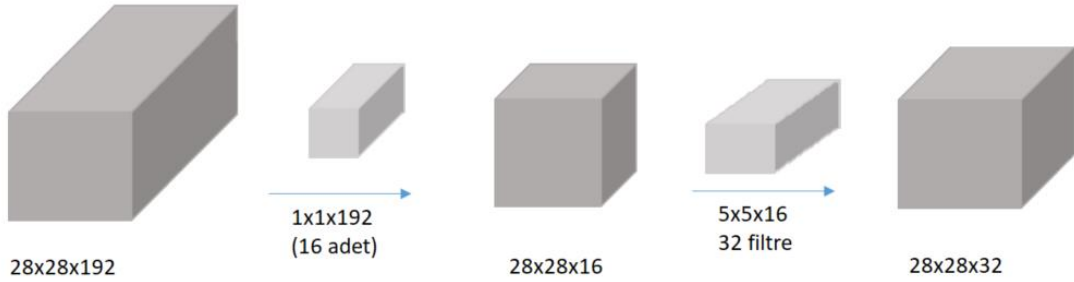
Şekil 2.17 1x1 evrişim işlemi.

1x1 evrişim işlemi uygulamaksızın Şekil 2.18’deki işlem gerçekleştirildiğinde hesaplanacak toplam parametre sayısı $(28 \times 28 \times 32) \times (5 \times 5 \times 192)$ yani **120 milyon** olmaktadır.



Şekil 2.18 5x5 evrişim işlemi.

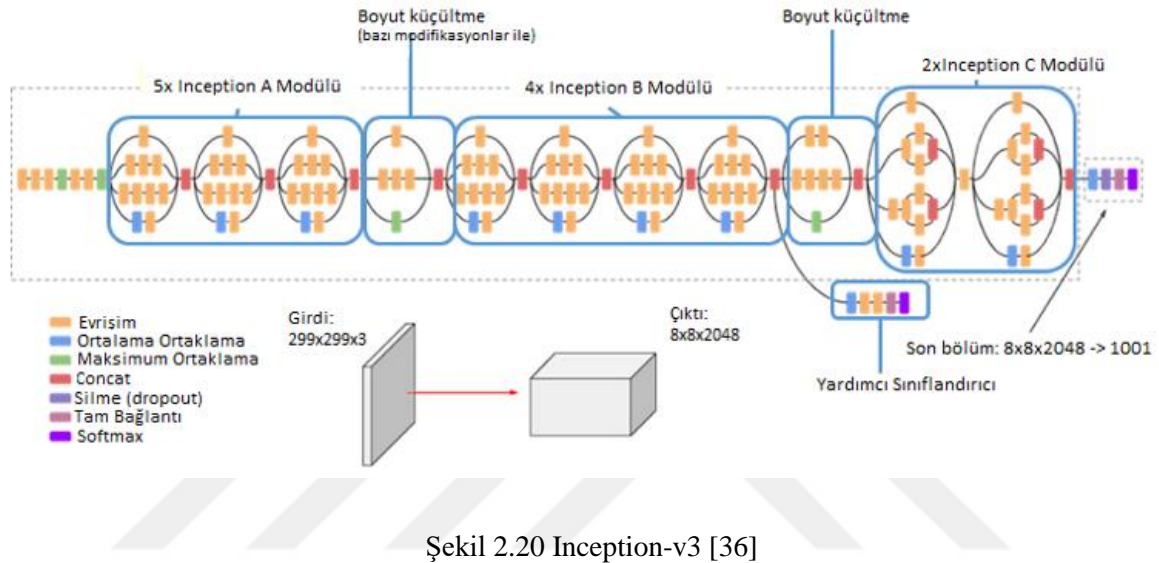
Aynı çıktıyı veren işlem Şekil 2.19'daki gibi öncesinde 1x1 evrişim katmanı uygulanarak gerçekleştirildiğinde ise toplam hesaplanacak parametre sayısı, 1x1 evrişim katmanında $(28 \times 28 \times 16) \times (1 \times 1 \times 192)$ işleminin sonucu olan 2,4 milyon parametre ile 5x5 evrişim katmanında $(28 \times 28 \times 32) \times (5 \times 5 \times 16)$ işleminin sonucu olan 10 milyon parametrenin toplamı olan **12,4 milyon** olmaktadır. Böylece ilk duruma göre 10 kat daha az parametre elde edilmiştir. Bu sonuç 1x1 evrişim işleminin hesaplamada ne kadar rahatlatma kazandırdığının bir göstergesidir.



Şekil 2.19 1x1 evrişim uygulandıktan sonra 5x5 evrişim uygulanması.

Inception olarak isimlendirilen ağ modeli çeşitli modüllerden oluşmaktadır. Bu modüllerin her birinde farklı boyutta evrişim ve maksimum ortaklama işlemleri yapılır. Literatürde GoogLeNet, inception-v1 [17] olarak da bilinmektedir. Sonradan inception-v2 [18], inception-v3 [18] ve inception-v4 [19] olarak adlandırılan versiyonları da geliştirilmiştir. Inception-v3, özellik çıkarma ve sınıflandırma bölümü olmak üzere iki bölümden oluşmaktadır. Özellik

çıkarma kısmı, evrişimli sinir ağını içerir. Inception-v3 oldukça tanınmış bir mimaridir ve ağıın girişi 299x299 piksel boyutlarında bir görüntü olmalıdır. Öte yandan, sınıflandırma kısmı tamamen bağlı ve Softmax katmanları içermektedir. Inception-v3'te belirtilen tüm katmanlar, Şekil 2.20'de gösterilmiştir.



Tablo 2.1 [18]'de önerilen Inception-v3 mimarisi

Tip	Dolgu miktarı / adım uzunluğu veya açıklama	Giriş boyutu
Evrişim	3x3 / 2	299x299x3
Evrişim	3x3 / 1	149x149x32
Aynı evrişim	3x3 / 1	147x147x32
Ortaklama	3x3 / 2	147x147x64
Evrişim	3x3 / 1	73x73x64
Evrişim	3x3 / 2	71x71x80

Evrişim	3x3 / 1	35x35x192
3 x Inception	Figür 5'deki gibi ¹	35x35x288
5 x Inception	Figür 6'deki gibi ²	17x17x768
2 x Inception	Figür 7'deki gibi ³	8x8x1280
Ortaklama	8x8	8x8x2048
Düzleştirme	Biçim	1x1x2048
Softmax	Sınıflandırıcı	1x1x1000

2.2.4 ÖĞRENİM TRANSFERİ

Öğrenim transferi, daha önce öğrenilmiş olan ilgili bir görevden elde edilen bilginin transfer yoluyla yeni bir görevde kullanılması ile öğrenimin geliştirilmesi prensibine dayanmaktadır. Makine öğrenim algoritmalarının çoğu tek bir görevi ele almak için tasarlanmış olsa da, transfer öğrenimi kolaylaştıran algoritmaların geliştirilmesi, makine öğrenimi alanında sürekli olarak ilgi duyulan bir konudur. Transfer öğrenimi üzerine kapsamlı bir inceleme Pan ve Yang [20] tarafından sağlanmıştır.

Öğrenim transferi ile öğrenme süreci sıfırdan başlatılmaz. Bu süreç, farklı bir problemi çözerken öğrenilen kalıplardan başlatılır. Bu şekilde hem önceki öğrenmelerden yararlanılmış olur hem de öğrenme sürecine sıfırdan başlamaktan kaçınılır. Bu sayede zaman da kazandırarak doğru bir model oluşturulmasına olanak tanır [21].

Öğrenim transferi, bir makine öğrenim modeli veya tekniği değil, makine öğrenimi içinde bir tasarım metodolojisidir. Bu tasarım metodolojisi ise genellikle önceden eğitilmiş modellerin kullanılmasına dayanır. Önceden eğitilmiş bir model, eldeki mevcut soruna benzer bir sorunu çözmek için büyük bir veri kümesi üzerinde eğitilmiş bir modeldir ki bu tür modellerin eğitim

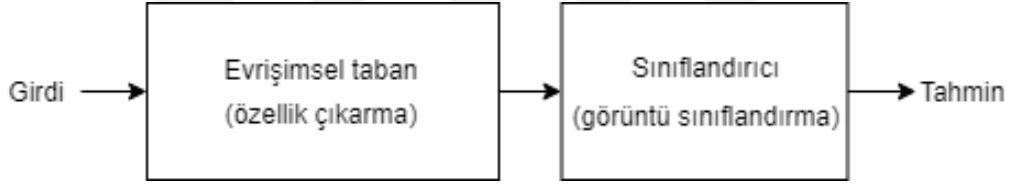
¹ EK 1 [18]

² EK 2 [18]

³ EK 3 [18]

maliyeti zaman ve gereksinim duyulan donanım bakımından hayli yüksektir. Bu sebeple Inception, VGG, ResNET gibi hazır eğitilmiş modelleri almak ve kullanmak yaygın bir uygulamadır. Bahsi geçen bu önceden eğitilmiş modeller Derin Evrişimli Sinir Ağları'na (D-ESA) dayanmaktadır.

Çok genel bir bakışla, ESA iki temel kısımda ele alınabilir: Bunlar sırasıyla evrişim tabanı ve sınıflandırıcı kısımlarıdır. İlk kısım evrişim ve havuzlama katmanları yığınınından oluşurken; ikinci kısım ise çoğunlukla tamamen bağlı katmanlardan meydana gelir. İkinci kısım olarak ele alınan sınıflandırıcının temel amacı, algılanan özelliklere göre görüntüyü sınıflandırmaktır. Tamamen bağlı bir katman, nöronları önceki katmandaki tüm aktivasyonlarla tam bağlantılara sahip olan bir katmandır. ESA'ya dayanan bir modelin yapısı için sunulan bu genel bakış Şekil 2.21'de gösterilmiştir. Gerçekte ise bu tür modeller çok daha karmaşık bir mimariye sahiptir.



Şekil 2.21 ESA'ya dayanan bir modelin mimarisini.

Bu derin öğrenme modellerinin en büyük avantajlarından biri, özellik çıkarımının otomatik olmasıdır. Ağ, bir görev için kullanışlı özellikleri etiketli verilere ve geri yayımlara dayanarak belirleyebilir. Örneğin; bir görüntüyü sınıflandırmada öncelikle girişin hangi bölümünün önemli olduğunu belirler. Bu aynı zamanda manuel özellik tanımlama işinin otomatize edildiği anlamına gelmektedir. Ayrıca çıkartılan bu özellikler, diğer problemler için de genellikle yararlı olduğu için diğer problemlerde de tekrar kullanılabilir. Kullanışlı özellikleri belirlemede temel olarak ağın ilk katmanları kullanılır; ancak ağın çıktısı, göreve özel olması sebebiyle kullanılmaz. Sonuç olarak, ESA'nın evrişimsel tabanı - özellikle alt tabakaları (girdilere daha yakın olanlar) - genel özelliklere, sınıflandırıcı kısım ve evrişim tabanının daha yüksek tabakalarının bazıları özel özelliklere atıfta bulunur.

Probleme özgü gereksinimler için önceden eğitilmiş bir model kullanılacaksa, önce bu modelin kendi orijinal sınıflandırıcısı kaldırılır. Bu aşamadan sonra probleme özgü, uygun yeni bir sınıflandırıcı eklenir. Bu iki temel aşamadan sonra modele, üç stratejiden birine göre ince ayar yapılır. Bu stratejiler şu şekilde özetlenebilir:

i. Tüm modeli eğitme:

Bu strateji, önceden eğitilmiş modelin mimarisi kullanılarak mevcut probleme ait veri kümesine göre eğitim yapılması prensibine dayanır. Bu stratejide model sıfırdan eğitileceği için, büyük bir veri kümesiyle birlikte çok fazla hesaplama gücüne ihtiyaç duyulacaktır. Bu sebeple yeterli büyüklükte bir veri kümesine sahip olunmadığı durumlarda bu yöntemi kullanmaktan kaçınmak daha yerinde bir tercih olacaktır.

ii. Bazı katmanları donmuş halde bırakıp diğerlerini eğitme:

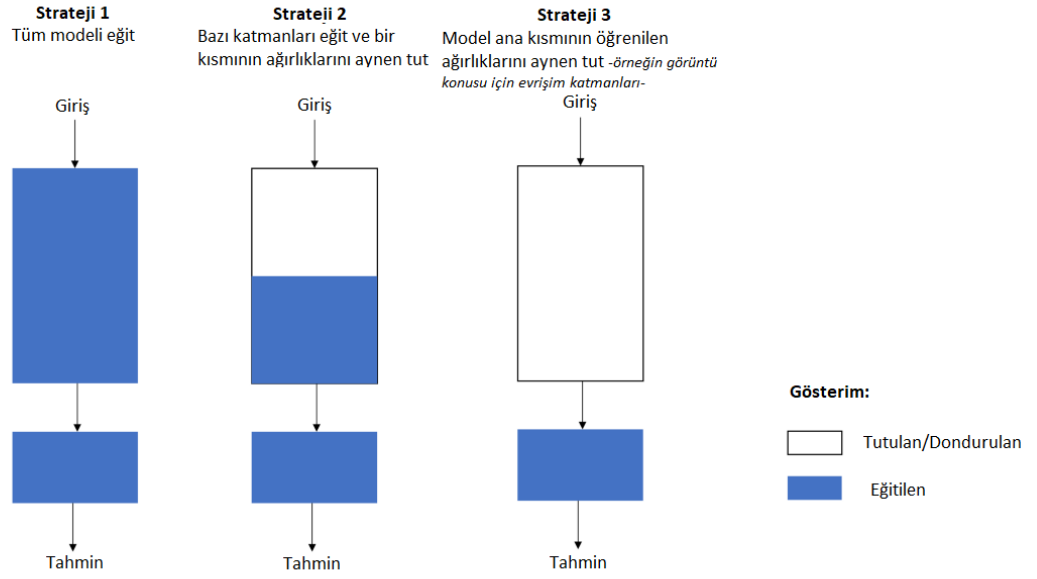
Bu stratejide eğitim sırasında, birkaç katman -genellikle bu ilk evrimsel katmanlar olmaktadır- sabit tutularak daha üst düzeydeki (daha derin olan) katmanların parametreleri optimize edilir. Böylece, daha düşük seviyeli katmanlar tekrar kullanırken optimize edilmesi gereken parametrelerin sayısının azaltılması amaçlanır. Sabit tutulacak katmanların genellikle ilk evrimsel katmanlar olmasının sebebi ise şudur: Daha düşük seviyeli katmanların problemden bağımsız olarak genellikle benzerlik gösteriyor olmasıdır. Diğer bir deyişle, düşük katmanların problemden bağımsız olarak genel özelliklere, yüksek katmanların ise probleme bağlı olarak belirli özelliklere işaret etmesi sebebiyle genellikle mimarinin ilk evrimsel katmanları dondurulur. Bu işlem ağırlıkların ayarlanarak yapılır. Donmuş bir katman eğitim sırasında değişmez. Daha genel bir ifade ile söylemek gerekirse donma ve ince ayar arasında bir denge bulmak için öğrenme oranlarının (learning rate) her katman için farklı olması sağlanabilir. Genellikle, küçük bir veri kümesine (<1.000) ve çok sayıda parametreye sahip olduğunda, aşırı öğrenmenin (ezberlemenin) önüne geçmek için daha fazla katman donmuş olarak bırakılır. Tersine, eğer veri kümesi büyükse (>1.000) ve parametre sayısı azsa, aşırı öğrenme sorunu olmayacağından, yeni görevde daha fazla katman çalıştırılması uygun olabilir.

iii. Evrişim tabanını dondurma:

Bu stratejide önceden eğitilmiş model, sabit bir özellik çıkarma mekanizması olarak kullanılır. Temel prensip ise evrişimli tabanı orijinal biçiminde tutmak ve ardından sınıflandırıcıyı beslemek için çıktılarını kullanmaktır. Bu, hesaplama gücünün az olması

veya mevcut probleme ait veri kümesinin küçük ve önceden eğitilmiş modelin veri kümesine benzer olması durumunda faydalı olmaktadır.

Şekil 2.22’te yukarıda bahsedilen üç ince ayar stratejisi özetlenmiştir. Buna göre üç stratejiden söz edilebilir: Bunlar, tüm modeli eğitme, bazı katmanları eğitip bir kısmının ağırlıklarını sabit tutma ve evrişim tabanını tamamen dondurup bu kısmı özellik çıkarıcı olarak kullanmaktır.



Şekil 2.22 İnce ayar stratejileri [34].

Bahsedilmiş olan bu stratejilerden özellikle ilk iki strateji uygulanırken dikkat edilmesi gerekmektedir; çünkü bu stratejilerde ağırlıkların kontrol etmek için öğrenme oranı (learning rate) olarak bilinen bir hiper-parametre kullanılır. Bu parametreyi özellikle önceden eğitilmiş bir ESA’ya dayalı bir model kullanırken düşük tutmak doğru olacaktır. Bunun sebebi, yüksek öğrenme oranlarının önceki bilgileri kaybettirme riskini artırmasıdır. Önceden eğitilmiş modellerin çok iyi eğitildiği varsayımı yapılırsa -ki bu varsayım oldukça adildir- öğrenme oranının düşük tutulması ESA ağırlıklarının kısa sürede çok fazla bozulmasına engel olacaktır. Aksi durumda, çok kısa sürede aşırı bir bozulma gerçekleşeceğinden önceki bilgilerin kaybedilmesi riski de artmış olacaktır.

3. MALZEME VE YÖNTEM

Bu tez çalışmasında çok sınıflı iki veri kümesine ait çiçek görüntülerinin bir veri artırımı metodu kullanılarak çoğaltılması, elde edilen tüm görüntülerin eğitim ve test kümelerine bölünerek önceden eğitilmiş bir model yardımıyla özelliklerinin çıkarılması, bu özelliklerin görselleştirilmesi ve elde edilen bu özellikler kullanılarak çeşitli sınıflandırma algoritmalarının eğitilip test işlemlerinden edinilen başarı oranlarının ve bu işlemlerin verimliliğinin ölçülmesi işlemleri *Python* dili kullanılarak gerçekleştirilmiştir. Ön eğitilmiş model kullanılarak özelliklerin çıkarılmasında *Tensorflow*, veri artırımı için kullanılan görüntü işleme tekniği için *opencv (cv2)*, zamansal verimliliğin ölçülmesinde *time*, matematiksel işlemler için gerekli olan yapılar için *numpy*, çıkarılan özelliklerin ilişkilerini görme maksadıyla gerçekleştirilen boyut azaltma ve görselleştirme işleminin yanı sıra sınıflandırma algoritmalarının test edilmesinde ise *sklearn* kütüphaneleri kullanılmıştır.

Google'ın geliştirdiği TensorFlow [35], dünyanın her yerindeki makine öğrenimi alanında büyük ilgi ve tasdik görmüştür. Bu kütüphanenin kullanımı, makine öğreniminde ve derin öğrenme programlarında ilk sırada yer almaktadır. TensorFlow, yüksek kullanılabilirlik ve yüksek esneklik avantajlarına sahiptir ve araştırmacılarının desteğiyle, verimliliği de artırılmıştır. Bugün Google, araştırmacıların farklı alanlarda kullanımını kolaylaştırmak için TensorFlow'un resmi web sitesinde pek çok önceden eğitilmiş ve tanımlanmış model yayınlamıştır. Inception-v3 [18] de bu modellerden biridir.

Deneyin gerçekleştirilmesinde önemli bir yer tutan Tensorflow kütüphanesi kullanıcıların onu CPU'da veya GPU'da kullanabilmesine olanak tanımaktadır. Bunun için mutlaka NVIDIA ekran kartına sahip olunması ve bu ekran kartının grafik işlem birimini kullanarak hesaplama performansında büyük ölçüde artışlara olanak veren paralel hesaplama mimarisi CUDA desteğine sahip olması gerekmektedir [22]. NVIDIA ekran kartının CUDA desteği olup olmadığını merak eden okuyucular "Supported GPUs" başlıklı listeyi [23] okuyabilir.

Bu çalışmada 17 ve 102 farklı sınıfa ait çiçek görüntüleri içeren iki veri kümesi [5], [8] kullanılmıştır. Tüm deneyler Şekil 3.1'de görüldüğü gibi, NVIDIA GEFORCE GTX 960M ekran kartına sahip bir bilgisayarda yapılmıştır.

```

2019-05-02 06:30:42.240732: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc
2019-05-02 06:30:42.968117: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device
name: GeForce GTX 960M major: 5 minor: 0 memoryClockRate (GHz): 1.176
pciBusID: 0000:01:00.0
totalMemory: 4.00GiB freeMemory: 3.34GiB
2019-05-02 06:30:42.969053: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device
2019-05-02 06:30:47.548492: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device
2019-05-02 06:30:47.549057: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device
2019-05-02 06:30:47.549429: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device

```

Şekil 3.1 PyCharm konsol ekranından deney sırasında alınan sistem özellikleri.

3.1 VERİ KÜMESİ

Çiçek görüntülerini sınıflandırmayı hedefleyen bu tez çalışmasında, bu alanda literatürde sıklıkla tercih edilmiş olan Oxford Visual Geometry Group'a ait 17-Flowers ve 102-Flowers veri kümeleri kullanılmıştır. Bu veri kümelerinin her ikisi de İngiltere'de yaygın olarak bulunan çeşitli çiçeklerden oluşturulmuştur.

17 tür çiçek içeren Oxford 17-Flowers veri kümesi [5], 2006 yılında Maria ve Andrew tarafından yaratılmıştır. Bu veri kümesinde, görüntünün ölçeği büyük, duruş ve aydınlatma da büyük ölçüde değişkenlik göstermektedir. Aynı zamanda farklı çiçek türleri ile aynı çiçek türleri arasındaki benzerlikler vardır. 102 tür çiçek içeren Oxford 102-Flowers veri kümesi [8] de Maria ve Andrew tarafından 2008 yılında yaratılmıştır ve türlerin her biri 40 ila 258 çiçek görüntüsüne sahiptir. Oxford 17-Flowers veri kümesi ile karşılaştırıldığında, daha fazla çiçek türü içerir ve farklı çiçek türleri arasında daha fazla benzerlik vardır. Bu nedenle çiçek sınıflandırması daha karmaşık olacaktır. Oxford 102-Flowers veri kümesinden 20 çiçek türünün örnek görüntüleri Şekil 3.2 'de, Oxford 17-Flowers veri kümesindeki dört sınıftan örnek görüntüler ise Şekil 3.3'te görülebilir.



Şekil 3.2 Oxford 102 Flowers veri kümesinden 20 farklı sınıfa ait çiçek görüntüleri.



Şekil 3.3 Oxford 17-Flowers veri kümesindeki farklı dört sınıftan örnek görüntüler.

Şekil 3.3'te verilen farklı dört sınıftan çiçek görüntüleri, ilgili veri kümesinde farklı sınıflardaki çiçeklerin renk, doku ve/veya şekil benzerliklerinin olduğunu göstermek için özellikle seçilmiştir ve yine aynı şekilde görüleceği üzere farklı sınıflardaki çiçekler birbirlerine benzer

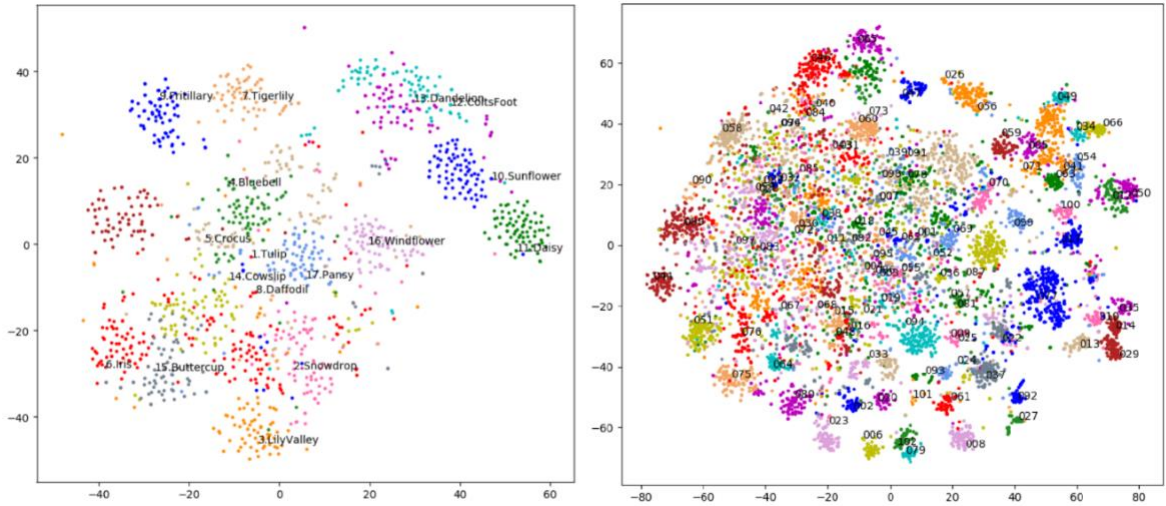
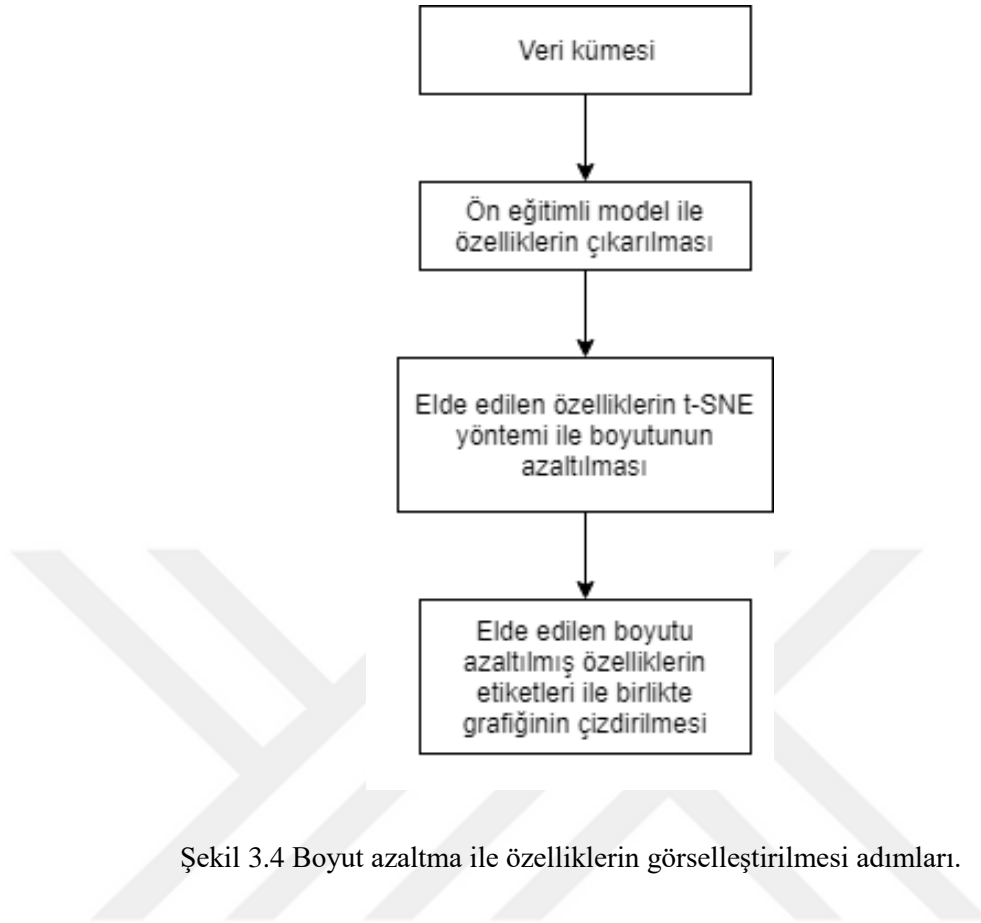
özellikler taşıyabilmektedir. Bununla beraber veri kümesinde birbirinden çok belirgin farkları olan çiçek sınıfları da mevcuttur.

3.2 BOYUT AZALTMA İLE ÖZELLİKLERİN GÖRSELLEŞTİRİLMESİ

Çok boyutlu bir veri setinin özellikleri arasındaki ilişkileri anlamak zordur; ancak bu ilişki boyut küçültülerek görselleştirilebilir. Bu sebeple, çalışma kapsamında yalnızca bilgilendirici bir adım olarak, boyut küçültme işlemi uygulanmasına karar verilmiştir.

Boyut küçültmede Temel Bileşen Analizi (TBA) doğrusal bir özellik çıkarım tekniği iken t-SNE, boyutsallığın azaltılması için uygulanabilecek doğrusal olmayan diğer bir tekniktir ve yüksek boyutlu veri kümelerinin görselleştirilmesi için daha uygundur [24].

Bu çalışmada, ESA özellikleri Inception-v3 modeli kullanılarak elde edildikten ve kaydedildikten sonra t-dağılımlı stokastik komşu gömme (t-SNE) tekniği uygulanmıştır. Bu işlem veri kümelerinin her ikisi için de veri artırımı işlemi uygulanmaksızın ön eğitilmiş model yardımıyla özelliklerinin çıkarılması sonrasında bu özelliklerin TSNE kütüphanesindeki `fit_transform` isimli fonksiyona parametre olarak gönderilip elde edilen boyutu azaltılmış özelliklerin etiketlerine göre grafiğinin çizdirilmesi ile gerçekleştirilmiştir. Ancak bu adım, yalnızca bilgilendirici bir adım olduğundan sonuç olarak elde edilecek başarı ve verimliliğe bir etkisi olmadığı göz önünde bulundurulmalıdır. Boyut azaltma ile özellikleri görselleştirme işlemine ait adımlar Şekil 3.4'te görülmektedir.



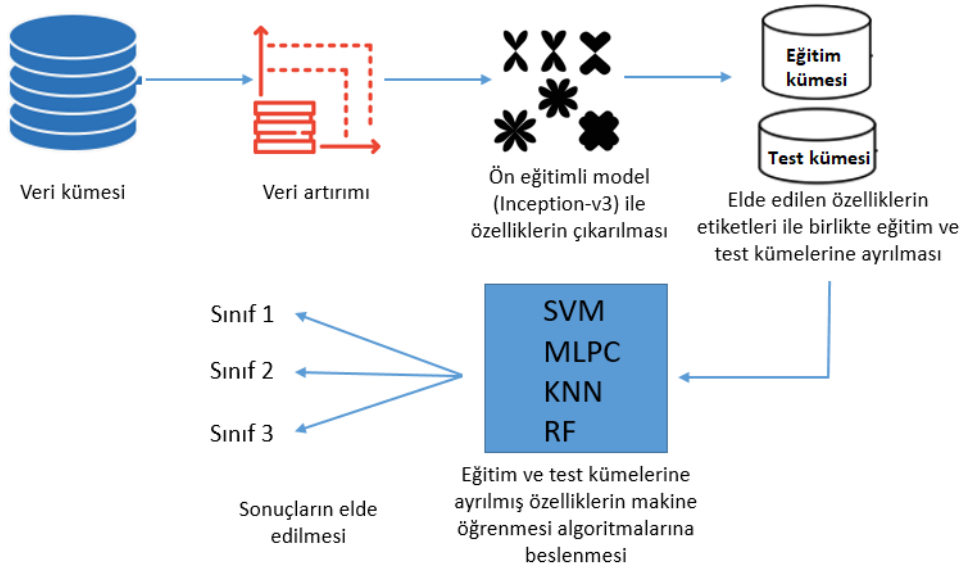
Şekil 3.5 Veri artırımı uygulanmadan veri kümelerinin iki boyutlu özellik düzlemine dönüştürülmesi, (solda) 17 Flowers (sağda) 102 Flowers

Boyut azaltma adımlarının izlenmesi neticesinde iki veri kümesi için de birer görüntü elde edilmiştir. Şekil 3.5'te sırasıyla Oxford 17-Flowers [5] ve Oxford 102-Flowers [8] veri

kümeleri için yüksek boyutlu özelliklerin iki boyutlu özellik düzlemine dönüştürülmesi işlemine ait çıktılar verilmiştir. Burada aynı sınıfa ait özelliklerin birbirine yakın olarak kümelenmiş olduğu görülmektedir.

3.3 UYGULANAN ÇİÇEK SINIFLANDIRMA YAKLAŞIMI

Bu tez çalışması, literatür araştırmasında paylaşılan yöntemlerden farklı olarak Şekil 3.6’te görülen akışın izlenmesi sonucu elde edilen sonuçları sunmaktadır. Bu akıştan söz etmek gerekirse: Öncelikle belirtilmiş olan iki veri kümesine ait çiçek görüntüleri veri artırımı tekniği ile iki sayıca katına çıkarılmıştır. Bu özellikler sırasıyla %80 ve %20 oranlarında eğitim ve test kümelerine ayrılmıştır. Bu noktada herbirinin etiketi ve özellikleri bilinen iki farklı alt kümenin olduğu göz önünde bulundurulmalıdır. Bu aşamadan sonra yapılan işlem eğitim kümesi ile ileride belirtilecek olan sınıflandırma algoritmaları eğitilip test kümesi ile de test edilerek elde edilen başarımlar ve verimliliğe ilişkin sonuçlar gözlemlenmiştir.



Şekil 3.6 Bu çalışmada kullanılan çiçek sınıflandırma sistemi.

3.4 VERİ ARTIRIMI

Veri artırımı, veri kümesindeki görüntülerin değiştirilmiş sürümlerinin oluşturularak veri kümesinin boyutunun yapay olarak genişletilmesi için kullanılan bir tekniktir. Bu tez çalışmasında da etiketli veri eksikliğinin dezavantajlarını gidermek için doğal görüntü sınıflandırmalarında daha iyi performans elde edilmesi amacıyla kullanılmıştır. Bu yöntem, görüntü sınıflandırmada etiketlenmiş veri azlığı sorununu ortadan kaldırmak için kullanılmıştır. Literatürde birçok veri artırım yöntemi bulunmaktadır. Bu işlem, görüntünün eksenler üzerinde farklı açılarda döndürülmesi, görüntüye çeşitli tiplerde gürültü eklenmesi veya stil transferi yapılması [25] suretiyle gerçekleştirilebilmektedir.

Bu tez çalışmasında ise görüntünün dikey ekseninde 90 derecelik açıyla döndürülmesi suretiyle bir veri artırımı işlemi gerçekleştirilmiştir. Şekil 3.7’de Oxford 102-Flowers [8] veri kümesinden iki farklı türde çiçeğin veri artırımı işlemi öncesi ve sonrasına ait görüntüleri verilmiştir.

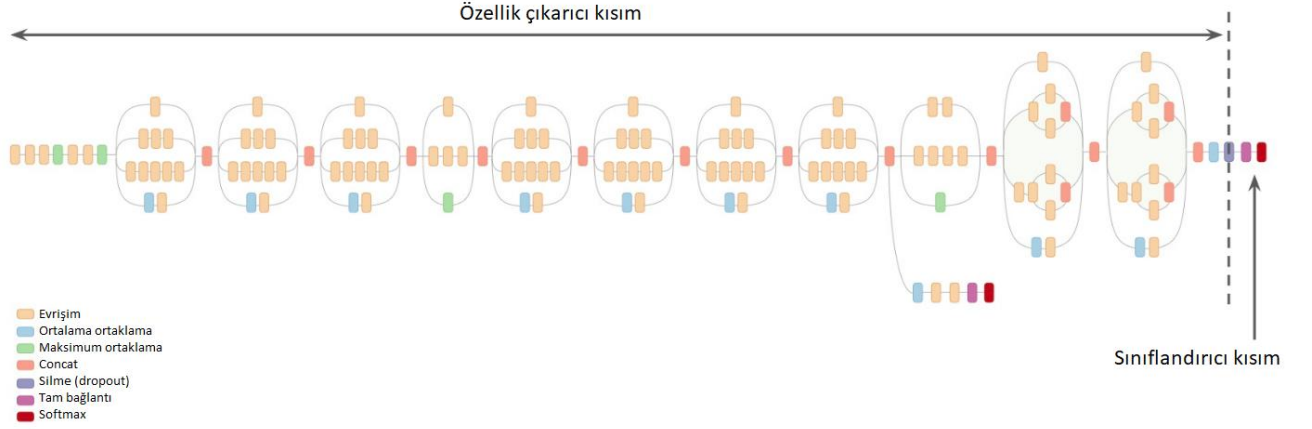


Şekil 3.7 Veri artırımı öncesi ve sonrasına ait görüntüler.

3.5 ÖZELLİK ÇIKARIMI

Şekil 2.5'te gösterildiği gibi ESA katmanları temel olarak özellik çıkarıcı ve sınıflandırıcı işlevlerini yerine getirir. Bu çalışmada, veri kümelerinin özelliklerinin çıkarılmasında ön eğitilmiş bir ESA modeli olan inception-v3 kullanılmıştır. Bunun için daha önce de belirtildiği gibi, Tensorflow Kütüphanesi kullanılarak sınıflandırma bölümü kaldırılmıştır. Kalan diğer katmanlar ise özellik çıkarıcı olarak kullanılmıştır. Bu yöntem Şekil 2.22'te gösterilen üçüncü stratejinin uygulanması işlemidir. Şekil 3.8'de ön eğitilmiş bir ESA modeli olan inception-v3'ün özellik çıkarma bölümü ve deneyde kullanılmayan kısmı olan sınıflandırma bölümü görülmektedir.

Özetlemek gerekirse, bu çalışmada, belirtilmiş olan veri kümelerindeki görüntülerden etiketli özellik vektörü üretmek amacıyla Inception-v3 modelinin özellik çıkarıcı bölümü kullanılmış ve buradan elde edilen özellikler çeşitli sınıflandırma algoritmalarına beslenmiştir.



Şekil 3.8 Inception-v3 modelinin özellik çıkarıcı ve sınıflandırıcı kısımları [26].

3.6 SINIFLANDIRMA ALGORİTMALARI

Özelliklerin çıkarılması aşamasından sonra kullanılacak sınıflandırıcı modeli için kullanılabilir yöntemler araştırmacılar tarafından çoğaltılabilir. Rawat ve Wang'a [20] göre, farklı sınıflandırıcıların Derin ESA üzerindeki performansını karşılaştırmak hala daha fazla araştırma gerektirmektedir ve böylece ilginç bir araştırma yönü meydana gelmektedir⁴. Bu sebeple bu tez çalışmasında, alt başlıklarda incelenmiş olan çeşitli sınıflandırıcıların çiçek sınıflandırma probleminde gösterdiği performanslar ortaya koyulmuştur. Derin ESA üzerinde performansı araştırılan ve bu doğrultuda kayda değer bir başarı hedeflenen sınıflandırıcılar ise şunlardır: DVM, Rassal Orman, K-En Yakın Komşu ve Çok Katmanlı Algılayıcı.

Önceden eğitilmiş ESA'ya dayanan bir öğrenim transferi yaklaşımından doğan görüntü sınıflandırma modellerinin, kabaca özellik çıkarımı yapan evrişimli taban ve buradan çıkarılan özelliklere göre giriş görüntüsünü sınıflandıran sınıflandırıcı olmak üzere iki bölümden oluşur.

Sınıflandırıcı oluşturmak için farklı yaklaşımlar izlenebilmektedir:

⁴ “Notwithstanding these attempts, comparing the performance of different classifiers on top of DCNNs still requires further investigation and thus makes for an interesting research direction.” [20].

3.6.1 Tam bağı katmanlar

ESA’larda temel olarak kullanılan sınıflandırıcı tam bağı katmanlar ile inşa edilmektedir. Birçok görüntü sınıflandırma problemi için standart yaklaşım, Softmax fonksiyonu ile aktive edilmiş bir katman ve ardından tamamen bağı katmanlardan oluşan bir yığın kullanmaktır [27], [28], [29]. Softmax aktivasyon fonksiyonu, her olası sınıf etiketi üzerindeki olasılık dağılımını verir ve ardından görüntü en olası sınıfa göre sınıflandırılır. Sınıflandırıcı olarak tam bağlantı katmanının kullanıldığı bir yapı Şekil 2.5’te verilmiştir.

Bu yaklaşıma, genel ve klasik olması sebebiyle değinilmiştir. Çalışmada kullanılan sınıflandırıcılar, takip eden alt başlıklarda belirtilmiştir.

3.6.2 Doğrusal Destek Vektör Makineleri

Doğrusal Destek Vektör Makineleri (DVM), kullanılabilir başka bir olası sınıflandırıcı olarak değerlendirilebilir. Tang’a [3] göre, evrişimli taban tarafından çıkarılan özellikler üzerine doğrusal bir DVM sınıflandırıcısını eğiterek sınıflandırma doğruluğu artırılabilir.

Destek Vektör Makineleri (DVM), sınıflandırmada kullanılan en etkili ve en basit denetimli öğrenme algoritmalarından biridir ve sınıflandırma problemlerinde sıklıkla kullanılır. Destek Vektör Makineleri’nde, girdi verilerinin dağıtımını hakkında önceden bir bilgi veya varsayım yoktur. Giriş verileri doğrusal veya doğrusal olmayan olarak ayrılabilir. Ek olarak, DVM’de aşırı öğrenme sorunu da yoktur. Yapay sinir ağlarında, çapraz doğrulama uygulanmadıkça aşırı öğrenme meydana gelebilir. DVM yaklaşımının avantajları ve dezavantajları hakkında daha ayrıntılı bilgi için çalışma [30] okunabilir.

Bu çalışmada kullanılan sınıflandırıcılardan biri DVM’dir. Deneyde sklearn kütüphanesindeki LinearSVC işlevi varsayılan şekilde kullanılmıştır. Bu işlevin varsayılan parametreleri [31]’den incelenebilir.

3.6.3 Rassal Orman

Rastgele Orman, makine öğrenmenin popüler modellerinden biridir, çünkü hem regresyon hem de sınıflandırma problemlerine uygulanabilir. Ayrıca, hiper parametre tahmini olmadan iyi sonuçlar verir. Rastgele Orman'da, sınıflandırma sürecinde birden fazla karar ağacı kullanılarak sınıflandırma başarısının artırılması amaçlanmaktadır. Karar ağaçlarının en büyük sorunlarından biri ise, verileri ezberlemek veya daha teknik olarak aşırı uydurma olarak da adlandırılabilir aşırı öğrenmedir. Bu sorunu çözmek için, rastgele orman modeli, hem veri kümesinden hem de özellik kümesinden onlarca hatta yüzlerce farklı rastgele kümesi seçer ve eğitir. Bu yöntemle, yüzlerce karar ağacı oluşturulur ve her karar ağacı ayrı ayrı tahmin edilir.

3.6.4 K-En Yakın Komşu (KNN)

K-En Yakın Komşu algoritması, hem sınıflandırma hem de regresyon tahmini problemleri için kullanılabilir. Ancak, endüstrideki sınıflandırma problemlerinde daha yaygın olarak kullanılmaktadır.

Algoritma, aşağıdaki adımlardan meydana gelmektedir:

1. Veriyi yükle.
2. K değerini başlat.
3. Sınıf tahminini elde etmek için 1'den toplam eğitim toplam eğitim kümesi sayısına kadar ilerde.
 - a. Test verileri ile her bir eğitim verisi arasındaki mesafeyi hesapla.
 - b. Hesaplanan mesafeleri, değerlerine göre artan sırada sırala.
 - c. Sıralanan diziden ilk k satırı al.
 - d. Bu satırlar için frekansı en yüksek sınıfı bul.
 - e. Bu sınıfı tahmin sonucu olarak döndür.

Bu algorithmada, Öklid Mesafesi (3.1), Manhattan Mesafesi (3.2) ve Minkowski Mesafesi (3.3) gibi farklı mesafe denklemleri kullanılabilir.

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (3.1)$$

$$\sum_{i=1}^k |x_i - y_i| \quad (3.2)$$

$$\left(\sum_{i=1}^k (|x_i - y_i|^q) \right)^{1/q} \quad (3.3)$$

Yapılan deneyde sklearn kütüphanesindeki **KNeighborsClassifier** işlevi şu parametreler dışında varsayılanlar ile kullanılmıştır: `n_neighbors=1`, `n_jobs=4`, `metric='manhattan'`. Bu işlevin diğer varsayılan parametreleri [32]'den incelenebilir.

3.6.5 Çok Katmanlı Algılayıcı

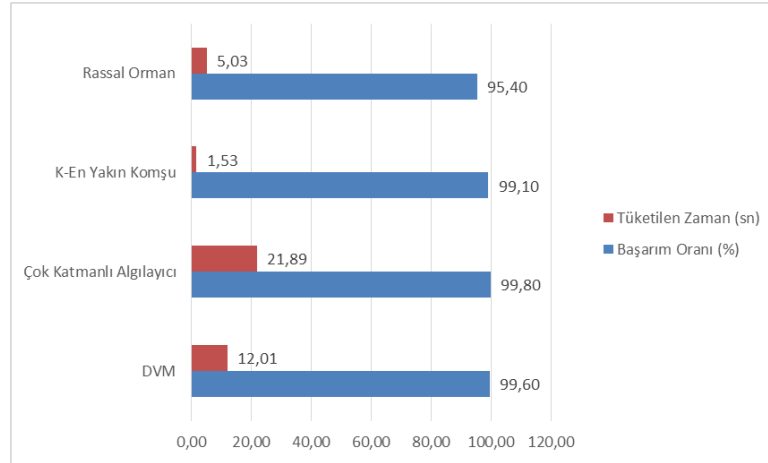
Modelin eğitiminde temel olarak üç adım vardır: İleri besleme, hata veya kaybı hesaplama, geri besleme. Modelin eğitim adımı olan ilk adım aşamasında girdiler modellere gönderilerek her katmanda ağırlıklar ile çarpılıp bayes değeri ile toplanır. Böylece modelin hesapladığı çıktı bulunur. Bu çıktıya tahmin edilen çıktı denmektedir. Diğer yandan olması gereken çıktının etiketi de zaten bilinmektedir. Elde edilen çıktı ile elde edilmesi beklenen çıktılar bilindiğine göre bu değerler baz alınarak loss (kayıp) değeri hesaplanır ve geri yayılım algoritması (backpropagation) ile hesaplanan kayıp geri yayılır ve gradyan (gradient) ile ağırlıkların güncellenmesi sağlanır. Bu, modelin eğitimindeki ana adımdır. Bu adımda, ağırlıklar gradyan akışına göre ayarlanır.

Deneyde sklearn kütüphanesindeki **MLPClassifier** işlevi varsayılan parametreler ile kullanılmıştır. Bu sebeple ReLU aktivasyon fonksiyonu kullanılmıştır. Bu işlevin diğer varsayılan parametreleri [33]'den incelenebilir.

4. BULGULAR

Bu bölümde Oxford 17-Flowers [5] ve Oxford 102-Flowers [8] veri kümeleri için yapılan deneylerden elde edilen bulgular sunulmuştur.

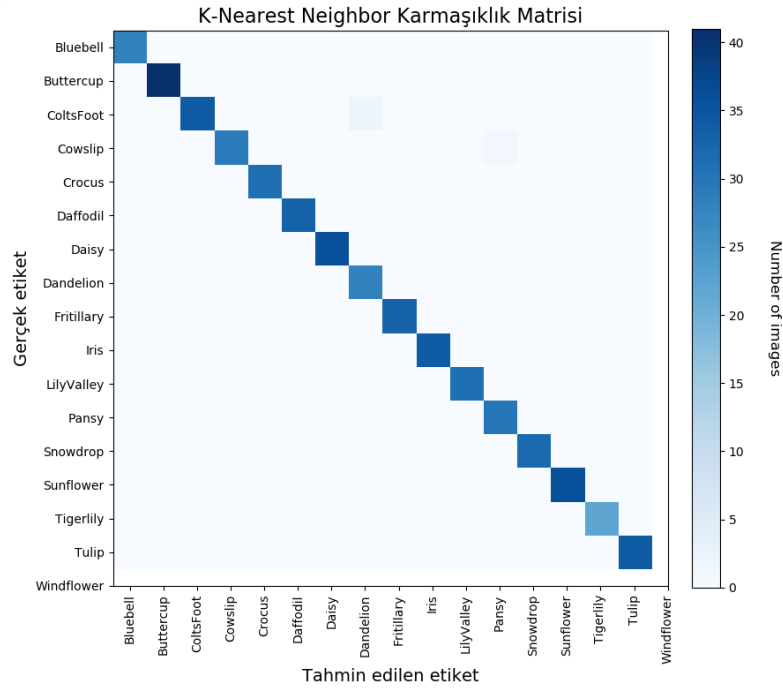
Oxford 17-Flowers veri kümesine veri artırımı uygulanarak görüntüler sayıca iki katına çıkarıldıktan sonra inception-v3 ön eğitilmiş model özellik çıkarıcı olarak kullanılmıştır. Bu model kullanılarak elde edilen özellikler ve onların etiketleri, %80 ve %20 oranlarında eğitim ve test kümelerine ayrılmıştır. Eğitim ve test kümelerine ayrılan bu özellikler, etiketleri ile birlikte çeşitli sınıflandırıcılara girdi olarak verilmiştir. Bu sınıflandırıcılardan en yüksek başarımlar %99,8 ile Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)'dan gelirken en yüksek ikinci başarımlar ise %99,6 ile DVM (Destek Vektör Makineleri - SVM) sınıflandırıcısından geldiği gözlemlenmiştir. En yüksek üçüncü başarımlar ise %99,1 ile K-En Yakın Komşu (KNN) sınıflandırıcısı ile elde edilirken en düşük başarımlar ise %95,4 ile Rassal Orman (Random Forest) sınıflandırıcısından gelmiştir.



Şekil 4.1 Oxford-17 Flowers veri kümesinde elde edilen sonuçlar.

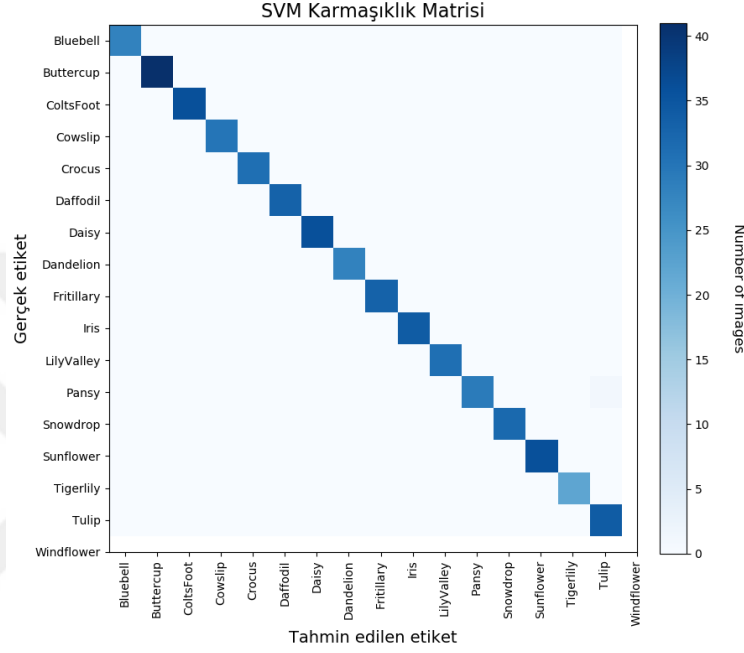
Tüketilen zaman açısından ise en verimli algoritma 1,53 saniye ile K-En Yakın Komşu algoritması iken ikinci en verimli algoritmanın 5,03 saniye ile Rassal Orman (Random Forest) olduğu görülmüştür. Zaman tüketiminde 12,01 saniye ile DVM (Destek Vektör Makineleri - SVM) üçüncü sırada yer alırken 21,89 saniye ile en çok zaman tüketen algoritmanın ise Çok

Katmanlı Algılayıcı algoritması (Multi-Layer Perceptron) olduğu tespit edilmiştir. Şekil 4.1’de başarımların ve tüketilen zaman miktarlarının karşılaştırmalı bir gösterimi verilmiştir. Bu şekle göre Çok Katmanlı Algılayıcı ve Destek Vektör Makinelerinin sınıflandırmada daha önde oldukları görülmektedir. Rassal Orman sınıflandırıcısının başarımları ise diğer sınıflandırıcılardan belirgin bir farkla geride kaldığı tespit edilmiştir. Bu sebeple derin ESA mimarilerinde sınıflandırıcı olarak Çok Katmanlı Algılayıcı veya Destek Vektör Makineleri’nin tercih edilmesinin Rassal Orman sınıflandırıcısının tercih edilmesinden başarımlar anlamında daha verimli olacağı söylenebilir. Ancak en yüksek iki oranın elde edildiği bu algoritmaların zaman tüketimi açısından ise diğer algoritmalara göre daha verimsiz çalıştığı görülmüştür. Bununla beraber en yüksek iki başarımların alındığı DVM ve Çok Katmanlı Algılayıcı zaman tüketimi açısından karşılaştırıldığında Destek Vektör Makinelerinin zaman tüketiminin neredeyse %50 oranında az olması ve başarımların Çok Katmanlı Algılayıcıdan yalnızca %0,2 az olması sebebiyle DVM’nin daha tercih edilebilir bir algoritma olduğu çıkarımı yapılabilir.



Şekil 4.2 K-En Yakın Komşu karmaşıklık matrisi.

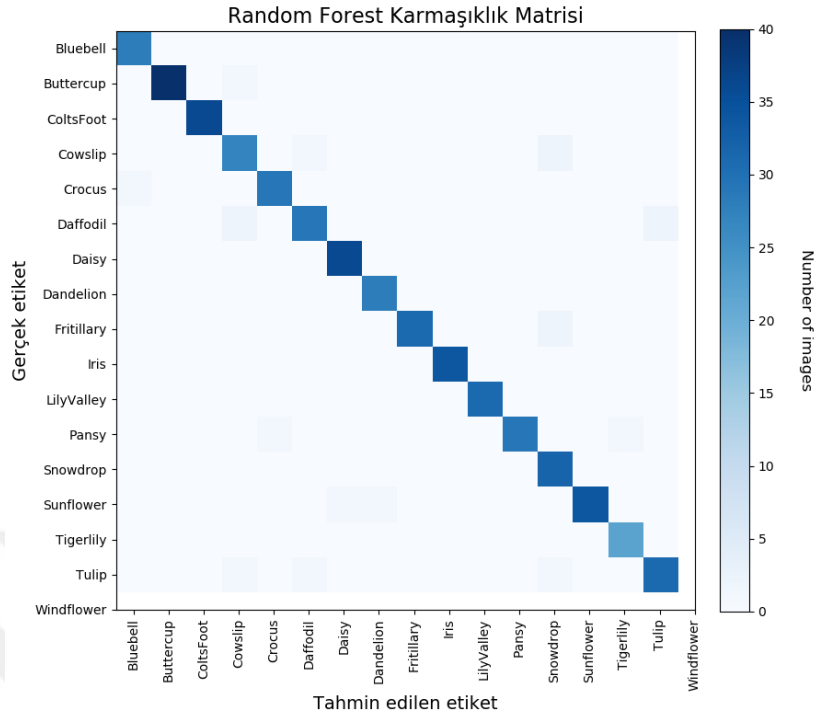
K-En Yakın Komşu algoritmasına ait karmaşıklık matrisi Şekil 4.2’de verilmiştir. Bu matrise göre modelin en isabetsiz tahminleri yaptığı çiçek sınıfı Tigerlily olmuştur. Buna karşın en isabetli tahminler Buttercup isimli çiçek sınıfında gerçekleşmiştir.



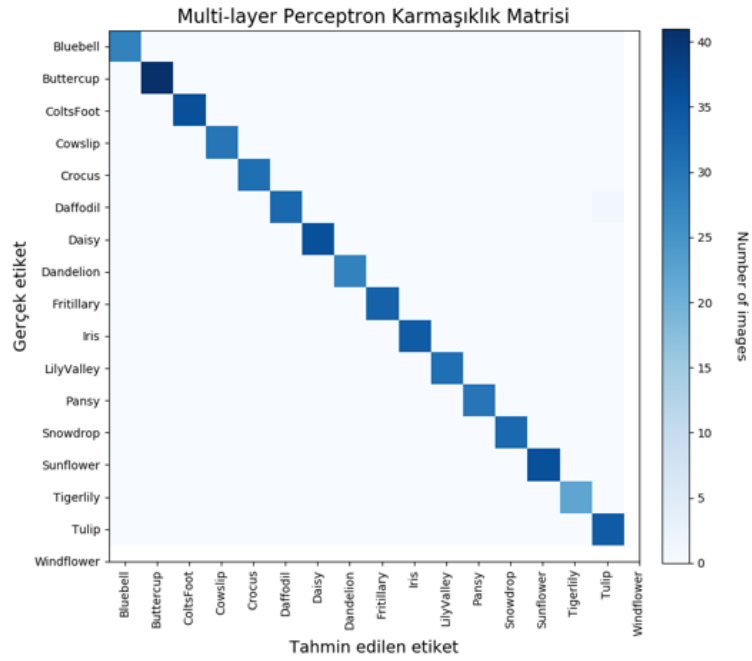
Şekil 4.3 Destek Vektör Makineleri karmaşıklık matrisi.

Destek Vektör Makineleri algoritmasına ait karmaşıklık matrisi Şekil 4.3’te verilmiştir. Bu matrise göre modelin en isabetsiz tahminleri yaptığı çiçek sınıfı Tigerlily olmuştur. Buna karşın en isabetli tahminler ise Buttercup isimli çiçek sınıfında gerçekleşmiştir.

Şekil 4.4 ve Şekil 4.5’te sırasıyla Rassal Orman ve Çok Katmanlı Algılayıcı sınıflandırıcılarından elde edilen karmaşıklık matrisleri verilmiştir. Bu karmaşıklık matrislerine göre yukarıdakilere benzer şekilde en isabetli tahminlerin yapıldığı sınıf Buttercup olurken, en isabetsiz tahminler ise yine Tigerlily sınıfında gerçekleşmiştir.



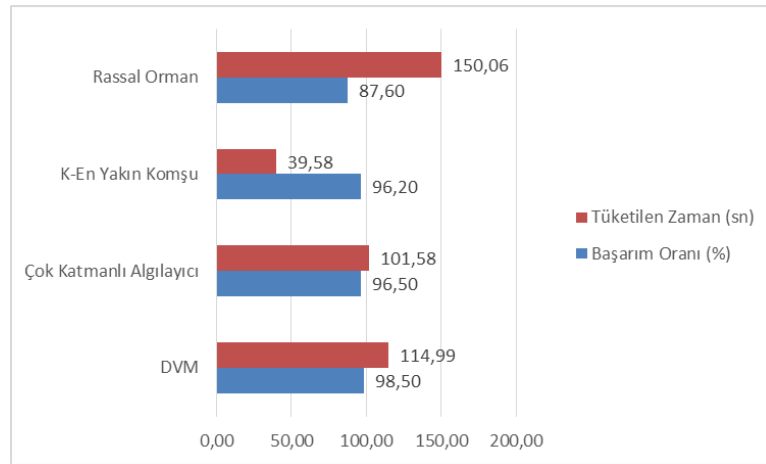
Şekil 4.4 Rassal Orman karmaşıklık matrisi



Şekil 4.5 Çok Katmanlı Algılayıcı karmaşıklık matrisi.

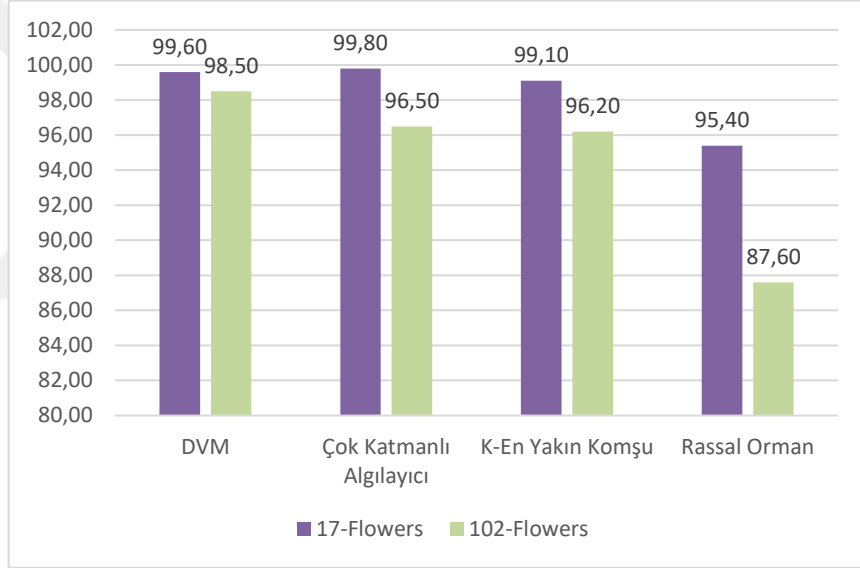
İlk veri kümesi için yapılan tüm işlemler aynı şekilde Oxford 102-Flowers veri kümesinde de gerçekleştirilmiştir. 17 sınıflı veri kümesine nazaran hem çiçek sınıf sayısı hem de sınıflara ait görüntü sayıları bakımından daha zengin olan bu veri kümesinde elde edilen başarımların ve tüketilen zaman miktarlarının kendi içinde karşılaştırmalı bir gösterimi Şekil 4.6'da verilmiştir. İlk olarak veri artırımı uygulanarak görüntüler sayıca iki katına çıkarıldıktan sonra Inception-v3 isimli ön eğitilmiş model özellik çıkarıcı olarak kullanılmıştır. Bu model kullanılarak elde edilen özellikler ve onların etiketleri, sırasıyla %80 ve %20 oranlarında eğitim ve test kümelerine ayrılmıştır. Eğitim ve test kümelerine ayrılan bu özellikler, etiketleri ile birlikte belirtilen makine öğrenmesi algoritmalarına girdi olarak verilmiştir. Bu sınıflandırıcılardan en yüksek başarımlar %98,5 ile DVM (Destek Vektör Makineleri - SVM)'den gelirken en yüksek ikinci başarımlar ise %96,5 ile Çok Katmanlı Algılayıcı (Multi-Layer Perceptron) sınıflandırıcısından geldiği gözlemlenmiştir. En yüksek üçüncü başarımlar ise %96,2 ile K-En Yakın Komşu (KNN) sınıflandırıcı ile elde edilirken en düşük başarımlar ise %87,6 ile Rassal Orman (Random Forest) sınıflandırıcısından gelmiştir.

Tüketilen zaman açısından ise en verimli algoritma 39,58 saniye ile K-En Yakın Komşu algoritması iken ikinci en verimli algoritmanın 101,58 saniye ile Çok Katmanlı Algılayıcı algoritması (Multi-Layer Perceptron) olduğu görülmüştür. Zaman tüketiminde 114,99 saniye ile DVM (Destek Vektör Makineleri - SVM) üçüncü sırada yer alırken, en çok zaman tüketen algoritma ise 150,06 saniye ile Rassal Orman (Random Forest) olduğu tespit edilmiştir.



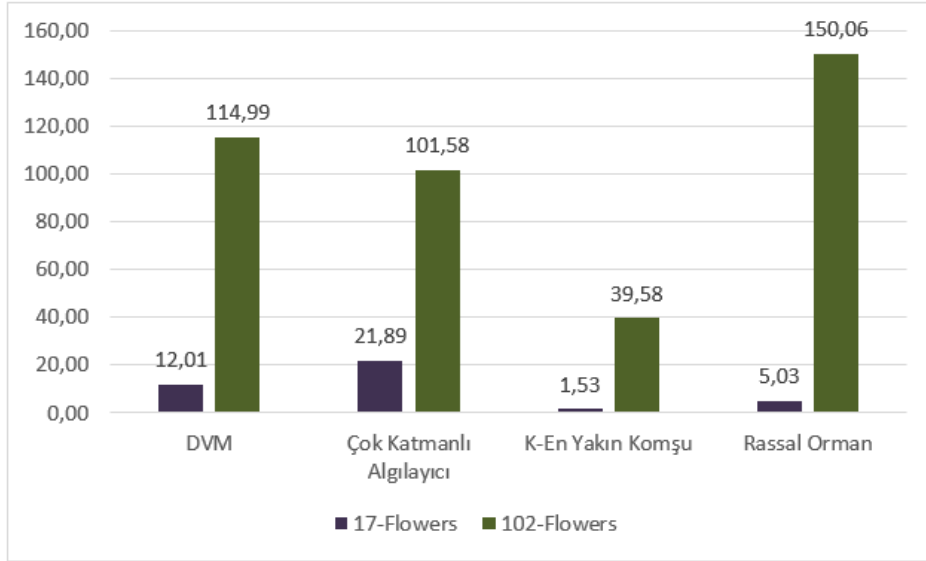
Şekil 4.6 Oxford-102 Flowers veri kümesinde elde edilen sonuçlar.

Şekil 4.6’da görüldüğü gibi Destek Vektör Makineleri sınıflandırma başarımında diğer sınıflandırıcılara göre daha öndedir. Bununla beraber DVM’de, tüketilen zaman anlamında 17 sınıflı veri kümesine göre performans düşüşü de tespit edilmiştir. Diğer veri kümesi ile elde edilen sonuçlara benzer şekilde en düşük başarım oranı yine Rassal Orman sınıflandırıcısı ile gerçekleşmiştir. Buna ek olarak, bir önceki karşılaştırmalı sonuçlara benzer şekilde, 102 sınıflı veri kümesinde Rassal Orman sınıflandırıcısı ile elde edilen bu başarım oranının diğer sınıflandırıcılarla elde edilenlere göre belirgin bir oranda düşük olduğu gözlenmiştir. Zaman tüketimi noktasında da en verimsiz çalışan algoritmanın yine Rassal Orman olduğu görülmüştür.



Şekil 4.7 İki veri kümesinde sınıflandırıcılara göre elde edilen başarımları oranları.

Şekil 4.7’de Oxford-17 Flowers ve Oxford-102 Flowers veri kümelerinde elde edilen başarımları oranları karşılaştırmalı olarak verilmiştir. Buna göre en yüksek başarımları oranları DVM ve Çok Katmanlı Algılayıcı’da elde edilmiştir. Ayrıca tüm algoritmalarda veri kümesi büyüdüğünde başarımları oranlarının da azaldığı görülmektedir. Bu bağlamda, DVM’deki başarımları oranı düşüşü Çok Katmanlı Algılayıcı’daki düşüşe göre daha az olduğu tespit görülmüştür.



Şekil 4.8 İki veri kümesinde sınıflandırıcılara göre tüketilen zaman miktarı (sn).

Şekil 4.8’de ise makine öğrenmesi algoritmalarının iki veri kümesi üzerindeki zaman tüketimleri saniye cinsinden gösterilmiştir. Tüm algoritmalarda veri kümesi büyüdükçe zaman tüketiminin de arttığı görülmektedir. En yüksek başarımların elde edildiği iki algoritma zaman tüketimi açısından karşılaştırıldığında Çok Katmanlı Algılayıcı’daki performans düşüşünün DVM’ye göre daha az olduğu tespit edilmiştir.

Tablo 4.1 Literatürdeki çalışmalarla genel bir karşılaştırma.

Referans Çalışma	Yöntem	Veri Kümesi	Başarım Oranı
[12]	ESA	Oxford 102-Flowers	%84.02
[13]	Inception-v3	Oxford 17-Flowers	%95
		Oxford 102-Flowers	%94
[14]	ResNET50	Oxford 102-Flowers	%95.29
		Oxford 17-Flowers	%96.27

[15]	Segmentasyon + ESA	Oxford 102-Flowers	%98.5
		Oxford 17-Flowers	%97.1
Bu çalışma	Inception-v3 + SVM	Oxford 17-Flowers	%99.6
		Oxford 102-Flowers	%98.5
	Inception-v3 + MLPC	Oxford 17-Flowers	%99.8
		Oxford 102-Flowers	%96.5
	Inception-v3 + KNN	Oxford 17-Flowers	%99.1
		Oxford 102-Flowers	%96.2
	Inception-v3 + RF	Oxford 17-Flowers	%95.4
		Oxford 102-Flowers	%87.6

Bu tez çalışmasında elde edilen sonuçlara ait literatürdeki aynı veri kümelerini kullanan diğer çalışmalarla kıyas tablosu Tablo 4.1’de verilmiştir. Tabloda kırmızı ile görülen oranlar diğer çalışmalar arasından en yüksek başarımların alındığı çalışmaya aittir. Bunun yanında koyu olarak görülen değerler ise bu tez çalışmasında elde edilen ve literatürdeki en yüksek sonuçlardan daha iyi olan sonuçlardır. Bu tablo incelendiğinde, ön işlemede segmentasyon ve sınıflandırmada ise Evrişimli Sinir Ağı kullanan çalışmanın [15] elde ettiği başarımların oranlarından daha iyi sonuçlar elde edildiği görülmektedir. Bu sonuçların elde edilmesinde ön işlemede veri artırımı uygulamanın ve ön eğitilmiş sinir ağı kullanmanın yararlı olduğu sonucu çıkarılabilir.

5. TARTIŞMA VE SONUÇ

Derin öğrenme yöntemleri bilgisayar görüşünde önemli başarılarla yol açmıştır. Soruna özgü özellikleri manuel olarak tanımlamak yerine derin öğrenme ile uygulayıcıların ham görüntüleri girdi olarak alan modelleri eğitmeleri sağlanarak bu işlem otomotize edilir. Evrişimli Sinir Ağı (ESA), özellik çıkarmak ve onları otomatik olarak birleştirmek için çok katmanlı evrişim işlemi yapar. Böylece daha yüksek bir tanıma oranına ve daha geniş bir uygulama yelpazesine sahip olur. Günümüzde özellik tanımlamanın orijinal karmaşıklığı, ağı tanımlamanın karmaşıklığına doğru evrilmiş durumdadır. Mimariler sıklıkla yeniden kullanılırken, ağ mimarilerini oluşturmada tek bir strateji yoktur. Tipik olarak, derin veri öğrenme teknikleri, ilgili araştırma ortamlarında ImageNet gibi muazzam veri kümeleri ile keşfedilmiş ve uygulanmıştır. Bu büyük veri kümelerindeki performansı artırmak için, araştırmacılar artan derinliği ve karmaşıklığı olan ağ mimarileri geliştirmişlerdir. Bu mimariler, milyonlarca parametrenin hesaplanmasıyla sonuçlanmaktadır.

Öğrenim transferi, bir ortamdan öğrenilen mevcut bilgiyi kullanabilen ve farklı olan fakat eski problemle bir ilişkisi olan diğer yeni problemi çözebilen yeni bir makine öğrenme yöntemidir. Motosiklet tanıma probleminden öğrenilen bilgileri bisiklet probleminin çalışmasına uygulanabilmesi, öğrenim transferine örnek gösterilebilir.

Bilgisayarlı görü alanında, ince taneli tanıma (fine grained recognition), özellikle son yıllarda daha popüler hale gelen bir alan olarak bilinmektedir. Bu çalışmada, oldukça küçük ve ince görsel farklılıkları olan sınıflarda bir çalışma yapabilmek için farklı çiçek türlerini içeren iki temel veri kümesi kullanılmıştır: Oxford 17-Flowers ve Oxford 102-Flowers. Bu çalışmada önerilen sınıflandırma sistemi ile deneylerde kullanılmış olan ve oldukça fazla sınıf içeren iki veri kümesi ile de literatürdeki en yüksek sonuca ulaşılmıştır. Derin evrişimli sinir ağı modeli kullanılarak özellik çıkarma, manuel özellik çıkarma yönteminden daha rahat ve pratiktir. Geleneksel yöntemlerle karşılaştırıldığında, Oxford çiçek veri kümesinde çiçek tanıma doğruluğu açıkça geliştirilmiştir. Deneyler, bu yöntemin çiçek tanıma doğruluğunu arttırdığını ve belirgin bir düzelme sağladığını göstermektedir.

Ayrıca bu çalışmada önerilen sınıflandırma sistemi çiçek sınıflandırma dışındaki alanlarda da kullanılabilir. Bu yönüyle çalışma, sadece çiçek sınıflandırma alanına değil etiketli verinin az olduğu diğer alanlara da uygulanabilir niteliktedir.

Tez kapsamında herkese açık olan yalnızca iki veri kümesi kullanılmıştır; ancak literatürde çiçek görüntüleri üzerine çalışmaların yapılmış olduğu diğer veri kümeleri de kullanılarak çalışma daha da genişletilebilir. Bunun yanı sıra literatürde hali hazırda bulunan etiketli çiçek veri kümelerinin azlığı sebebi ile özellikle görüntü sayısının daha az olduğu ve karmaşıklık matrisinde görülen, modelin en çok hatalı tahmin yaptığı çiçek sınıflarının görüntüleri artırılarak ve/veya iyileştirilerek sınıf bazında hatalı tahmin oranını düşürme yoluna gidilebilir. Böylece elde edilen başarı oranının artırılması da sağlanabilir. Çalışmada elde edilen bulgulara göre sınıflara göre iyi ve kötü tahmin oranlarının altında yatan sebeplerin bulunmasına yönelik çalışmalar yapılarak elde edilecek bulguların farklı veri kümeleriyle de yapılmış çalışmalardaki etkisi incelenebilir.



KAYNAKLAR

- [1]. Manktelow, M., 2010. History of taxonomy. Lecture from Dept. of Systematic Biology, Uppsala University, 29.
- [2]. Nilsback, M. E. and Zisserman, A. 2004. Delving into the whorl of flower segmentation. In the Proceedings of British Machine Vision Conference, Vol. 1, pp. 27-30.
- [3]. R. Al-Tayeche and A. Khalil, "CBIR: Content Based Image Retrieval," Project Report, Department of Systems and Computer Engineering, Faculty of Engineering, Carleton University, April 4, 2003.
- [4]. M. Das, R. Manmatha, E. M. Riseman, "Indexing Flower Patent Images Using Domain Knowledge," IEEE Intelligent Systems, Vol. 14(5), 1999, pp.24-33.
- [5]. Nilsback, M. E. and Zisserman, A. 2006. A Visual Vocabulary for flower Classification In the Proceedings of Computer Vision and Pattern Recognition, Vol. 2, pp. 1447-1454.
- [6]. V. S Tseng, M. Wang and J. Su, "A new method for Image Classification by using Multilevel Association Rules," Proc. of the 21st International Conference on Data Engineering (ICDE '05).
- [7]. S. Cho and P. Lim, "A novel Virus Infection Clustering for Flower Images Identification," Proc. of the 18th international Conference on Pattern Recognition. Vol.2, 2006, pp. 1038-1041.
- [8]. Nilsback, M. E. and Zisserman, A. 2008. Automated flower classification over a large number of classes. In the Proceedings of Sixth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 722 – 729.
- [9]. D.S. Guru, Y.H. Sharath Kumar, S. Manjunath, "Textural features in flower classification", Mathematical and Computer Modeling Journal, 2010
- [10]. D.S.Guru, Y.H.Sharath, and S. Manjunath. " Texture features and KNN in classification of flower images", IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition (PTTIPPR), Vol. 1, pp. 21- 29, 2010.
- [11]. Shailendrakumar M.Mukane, Sachin R. Gengaje, Dattatraya S. Bormane," On Scale Invariance Texture Image Retrieval using Fuzzy Logic and Wavelet Co- occurrence based Features", International Journal of Computer Applications (0975 – 8887) Volume 18– No.3, March 2011
- [12]. Liu, Y., Tang, F., Zhou, D., Meng, Y., & Dong, W. "Flower classification via convolutional neural network.", International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications IEEE, pp.110-116. 2017
- [13]. Xia, X., Xu, C. and Nan, B., 2017, June. Inception-v3 for flower classification. In 2017 2nd International Conference on Image, Vision and Computing (ICIVC) (pp. 783-787). IEEE.

- [14]. Wu, Y., Qin, X., Pan, Y. and Yuan, C., 2018, July. Convolution Neural Network based Transfer Learning for Classification of Flowers. In 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP) (pp. 562-566). IEEE.
- [15]. HIARY, Hazem, et al. Flower classification using deep convolutional neural networks. IET Computer Vision, 2018, 12.6: 855-862
- [16]. Lin, M., Chen, Q. and Yan, S., 2013, Network in network, arXiv preprint arXiv:1312.4400.
- [17]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 1-9).
- [18]. Christian Szegedy, Vincent Vanhoucke, et al, Rethinking the Inception Architecture for Computer Vision. arXiv: 1512.00567, 20 15.
- [19]. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-First AAAI Conference on Artificial Intelligence.
- [20]. Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345-1359.
- [21]. Rawat, W. and Wang, Z., 2017, Deep convolutional neural networks for image classification: A comprehensive review, Neural computation, 29(9), 2352-2449.
- [22]. CUDA Paralel Hesaplama , <https://www.nvidia.com.tr/object/cuda-parallel-computing-tr.html>, [Ziyaret Tarihi: 16 Nisan 2019].
- [23]. Supported GPUs, https://www.geforce.com/hardware/technology/cuda/supported-gpus?field_gpu_type_value=All, [Ziyaret Tarihi: 16 Nisan 2019].
- [24]. Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.
- [25]. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
- [26]. Google codelab ekibi , <https://codelabs.developers.google.com/codelabs/cpb102-txf-learning/index.html#1> Başlık: Image Classification Transfer Learning with Inception v3, [Ziyaret Tarihi: 16 Nisan 2019].
- [27]. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012, Imagenet classification with deep convolutional neural networks, In Advances in neural information processing systems (pp. 1097-1105).
- [28]. Simonyan, K. and Zisserman, A., 2014, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

- [29]. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.
- [30]. Tang, Y., 2013, Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.
- [31]. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>, [Ziyaret Tarihi: 16 Nisan 2019].
- [32]. <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
Başlık: sklearn.neighbors.KNeighborsClassifier, [Ziyaret Tarihi: 16 Nisan 2019].
- [33]. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html, [Ziyaret Tarihi: 16 Nisan 2019]
- [34]. Kızrak, A., 2018, Fine-Tuning ve Transfer Öğrenme Nedir, https://github.com/ayyucekizrak/Udemy_DerinOgrenmeyeGiris/blob/master/TransferOgrenme_FineTuning/ReadMe.md, [Ziyaret Tarihi: 16 Nisan 2019].
- [35]. Girija, S.S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Software available from tensorflow. org.
- [36]. Tsang, SH, 2018, Review: Inception-v3—1st Runner Up (Image Classification) in ILSVRC 2015, <https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c>, [Ziyaret Tarihi: 16 Nisan 2019]

EKLER

EK 1. Tablo 2.1'de [18]'e referans verilen Şekil 5.

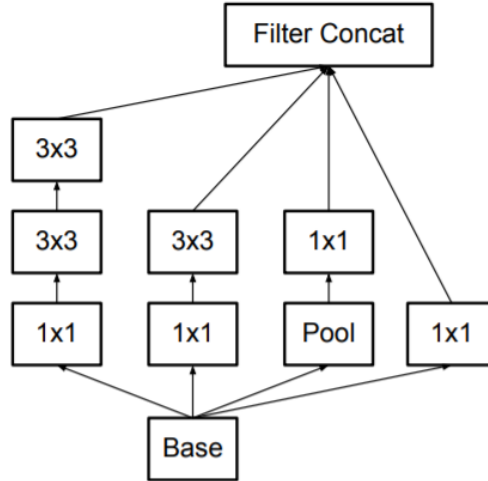


Figure 5. Inception modules where each 5×5 convolution is replaced by two 3×3 convolution, as suggested by principle 3 of Section 2.

EK 2. Tablo 2.1'de [18]'e referans verilen Şekil 6.

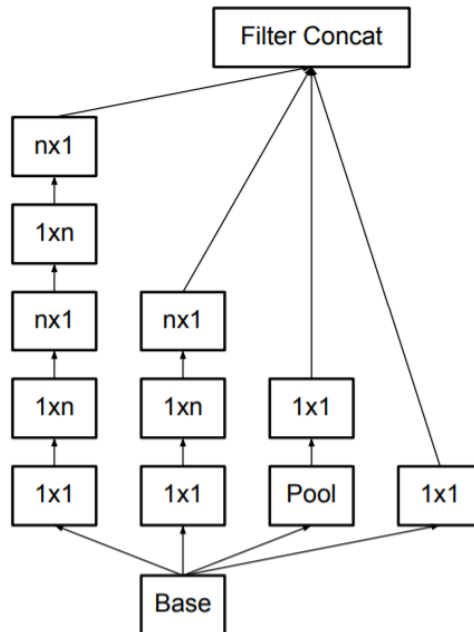


Figure 6. Inception modules after the factorization of the $n \times n$ convolutions. In our proposed architecture, we chose $n = 7$ for the 17×17 grid. (The filter sizes are picked using principle 3)

EK 3. Tablo 2.1’de [18]’e referans verilen Şekil 7.

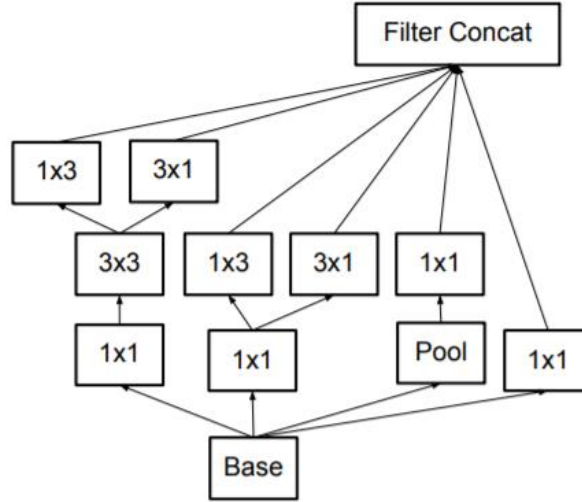


Figure 7. Inception modules with expanded the filter bank outputs. This architecture is used on the coarsest (8×8) grids to promote high dimensional representations, as suggested by principle 2 of Section 2. We are using this solution only on the coarsest grid, since that is the place where producing high dimensional sparse representation is the most critical as the ratio of local processing (by 1×1 convolutions) is increased compared to the spatial aggregation.

EK 4. Veri artırımı için yazılan Python kodu.

```

from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
from pathlib import Path

def rotate(image_path, degrees_to_rotate, saved_location):
    image_obj = Image.open(image_path)
    rotated_image = image_obj.rotate(degrees_to_rotate)
    rotated_image.save(saved_location)
    rotated_image.show()

numberOfFolder = ['001', '002', '003', ..., '100', '101', '102']

#--ilgili dizinde klasörleri oluşturma

```

```
#for x in numberOfFolder:
#  os.mkdir("C:/Users/Busra/Desktop/102FLOWER_Augmented/" + str(x))

for x in numberOfFolder:
    recordpath = "C:/Users/Busra/Desktop/102FLOWER_Augmented/" + str(x) + '/'
    print('recordpath: ', recordpath)
    y = "C:/Users/Busra/Desktop/102Flowers/" + str(x)
    print('y: ', y)
    pathlist = Path(y).glob('*/*/*.jpg')
    print(str(pathlist))
    i = 1
    for path in pathlist:
        path_in_str = str(path)
        # -- Görüntüyü oku
        img = cv2.imread(path_in_str)
        img = np.array(img)
        height, width, channels = img.shape
        print(height, width, channels)
        rotated_image = img.rotate(90)
        cv2.imwrite(recordpath + '{:d}'.format(i).zfill(4) + ".jpg", rotated_image) # rotated image
```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Büşra Rümeysa METE
Doğum Yeri	Bakırköy/İSTANBUL
Doğum Tarihi	14.03.1991
Uyruğu	<input checked="" type="checkbox"/> T.C. <input type="checkbox"/> Diğer:
Telefon	
E-Posta Adresi	b.mete@iku.edu.tr
Web Adresi	https://www.linkedin.com/in/busramete/

Eğitim Bilgileri	
Lisans	
Üniversite	Eskişehir Osmangazi Üniversitesi
Fakülte	Mimarlık-Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	2016

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi-Cerrahpaşa
Enstitü Adı	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı

Makale ve Bildiriler	
<p>Aşıroğlu, B., Mete, B. R., Yıldız, E., Nalcakan, Y., Tekindağ, M., & Ensarı, T. (2019, April). Automatic HTML Code Generation from Mock-up Images Using Machine Learning Techniques. In: 2019 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT). IEEE, 2019. p. 1-4.</p>	
<p>Günay, M., Yıldız, E., Nalcakan, Y., Aşıroğlu, B., Zencirli, A., Mete, B. R., & Ensarı, T. (2018, October). Digital Data Forgetting: A Machine Learning Approach. In 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-4). IEEE</p>	