



**T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



YÜKSEK LİSANS TEZİ

NESNE TESPİTİ İLE YEMEK TARİFİ TAVSİYE SİSTEMİ

Yasin KÖKER

Enformatik Anabilim Dalı

Enformatik Programı

DANIŞMAN

Prof. Dr. Sevinç Gülseçen

II. DANIŞMAN

Dr. Murat GEZER

Temmuz, 2020

İSTANBUL

Bu alıřma 16.07.2020 tarihinde ařağıdaki jüri tarafından Enformatik Anabilim Dalı Enformatik Programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Sevin Gülseen (Danışman)
İstanbul Üniversitesi
Enformatik Anabilim Dalı

Do. Dr. iğdem Erol
İstanbul Üniversitesi
Enformatik Anabilim Dalı

Dr. Öğr. Üyesi Ayře ınar
Marmara Üniversitesi
İřletme Fakóltesi



- İntihal Programı Beyanı

20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin aboneliği olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Sayın danışman hocalarım Prof. Dr. Sevinç GÜLSEÇEN'e ve Dr. Murat GEZER'e, bu tez çalışması için gereken yönlendirici desteği sağladığı ve tez metnini inceleyerek biçim ve içerik bakımından son şeklini almasında katkıda bulundukları için teşekkür ederim.

Son olarak bu süreçte yıllarca süren çalışmalarım boyunca beni sürekli cesaretlendiren, ilgi ve sevgilerini eksik etmeyen aileme saygı, sevgi ve teşekkürlerimi sunarım.

Temmuz, 2020

Yasin KÖKER



İÇİNDEKİLER

	Sayfa No
ÖNSÖZ	iv
İÇİNDEKİLER	vii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	x
KISALTMA LİSTESİ	xi
ÖZET	xiii
SUMMARY	xv
1. GİRİŞ	1
2. GENEL KISIMLAR	4
2.1. LİTERATÜR İNCELEMESİ	4
2.2. VERİ	8
2.2.1. Web’de Veri Toplama	8
2.2.2. Veritabanı Oluşturma	9
2.2.3. Veri Madenciliğinde Ön İşleme Süreci	11
2.2.3.1. Veri Entegrasyonu	11
2.2.3.2. Veri Temizleme	12
2.2.3.3. Veri İndirgeme	12
2.3. GÖRÜNTÜ ÖN İŞLEME TEKNİKLERİ	12
2.3.1. Renk Uzayları	12
2.3.1.1. RGB Renk Uzayı	13
2.3.1.2. BGR Renk Uzayı	13
2.3.1.3. Gri Renk Uzayı	13
2.3.2. Resmi Yeniden Boyutlandırma	14
2.3.3. Fotogerçekçi Olmayan İşleyici - Üsluplaştırma Filtresi	14
2.3.4. Gaussian Bulanıklığı	14
2.3.5. Adaptif Eşikleme	14

2.3.6. Morfolojik İşlemler	15
2.3.6.1. Geniřletme	15
2.3.7. Kenar Bulma	15
2.3.7.1. Canny Kenar Dedektörü	16
2.4. DERİN ÖĞRENME	17
2.4.1. Evriřimsel Sinir Ağları Mimarisi ve Katmanları	17
2.4.1.1. Giriř Katmanı	18
2.4.1.2. Evriřim Katmanı	18
2.4.1.3. Düzleřtirilmiř Doğrusal Birim Katmanı	19
2.4.1.4. Havuzlama Katmanı	19
2.4.1.5. Tam Baęlantılı Katman	20
2.4.1.6. Seyreltme Katmanı	20
2.4.1.7. Sınıflandırma Katmanı	21
3. MALZEME VE YÖNTEM	22
3.1. VERİ ÜZERİNE İřLEMLER	22
3.1.1. Verinin Toplaması	22
3.1.1.1. Yemek Tariflerinin Toplanması	22
3.1.1.2. Malzemelere ait olan Görsellerin Toplanması	23
3.1.2. Verinin İşlenmesi	23
3.1.2.1. İşlenen Veriden Elde Edilen Bazı İstatistikler	24
3.1.3. Veritabanının Oluřturulması	25
3.2. NESNE TANIMA AřAMASI	27
3.2.1. Görüntü Ön İşleme	27
3.2.1.1. Giriř Görüntüsünü BGR Olarak Okuma	27
3.2.1.2. Resmi Yeniden Boyutlandırma	29
3.2.1.3. Gölge Kaldırma	29
3.2.1.4. Üsluplařtırma Filtresi	30
3.2.1.5. BGR'dan Gri Renk Uzayına Dönüřüm	31
3.2.1.6. Gaussian Bulanıklıęı	32
3.2.1.7. Adaptif Eřikleme	32
3.2.1.8. Gürültü Temizleme	33
3.2.1.9. Morfolojik İşlemler - Geniřletme	34

3.2.1.10. Kenar Bulma	35
3.2.1.11. Yeni Görüntü Olarak Kaydedilen Nesneler	35
3.2.2. Evrişimsel Sinir Ağları	38
3.2.2.1. Donanım ve Yazılım Özellikleri	38
3.2.2.2. Veri Kümesi	38
3.2.2.3. Modelin Oluşturulması	39
3.2.2.4. Sonuçlar	39
4. BULGULAR	41
4.1. YEMEK TARİFİ ÖNERİ AŞAMASI	41
5. TARTIŞMA VE SONUÇ	44
5.1. SONUÇLAR	44
5.2. GELECEK ÇALIŞMALAR İÇİN ÖNERİLER	45
KAYNAKLAR	46
EKLER	51
EK 1. Oluşturulan Mimari Model	51
ÖZGEÇMİŞ	52

ŞEKİL LİSTESİ

	Sayfa No
Şekil 1.1 Bu süreçte yapılan işlemler	2
Şekil 1.2 Kullanıcı ile etkileşime giren sistem	3
Şekil 2.1 DIKW piramidi [23]	8
Şekil 2.2 Web madenciliği sınıflandırması [27] [28]	9
Şekil 2.3 Tablo örneği	10
Şekil 2.4 Sorgu örneği	10
Şekil 2.5 Sorgu örneğinden dönen sonuç	11
Şekil 2.6 İlişki örneği	11
Şekil 2.7 RGB renk uzayı [39]	13
Şekil 2.8 Gaussian dağılım grafiği [44]	15
Şekil 2.9 3x3 yapısal eleman matrisi kullanılarak yapılan genişletme işlemi [47]	16
Şekil 2.10 ESA katmanları [52]	17
Şekil 2.11 Evrişim işleminin uygulanışı [53]	18
Şekil 2.12 Negatif ve pozitif girdiler için düzleştirilmiş doğrusal birim katmanının etkisi [63] ...	19
Şekil 2.13 Maksimum ve ortalama havuzlamanın girdi matrisine olan etkisi [71]	20
Şekil 2.14 Tam bağlantılı katman [67]	20
Şekil 2.15 ESA ağına seyreltme katmanı uygulanması [67]	21
Şekil 2.16 Sınıflandırma işlemi [66]	21
Şekil 3.1 Yemek tariflerinin toplanmasını gösteren diyagram	22
Şekil 3.2 Malzemelere ait olan görsellerin toplanmasını gösteren diyagram	24
Şekil 3.3 Her bir kategoriye ait toplam tarif sayısı	25
Şekil 3.4 Veritabanı ER diyagramı	26
Şekil 3.5 Görüntü ön işlemede uygulanan algoritmanın akış diyagramı	27
Şekil 3.6 Sisteme girdi olarak verilen görüntü	28
Şekil 3.7 Giriş resmini BGR'ye dönüştürülmüş hali	28

Şekil 3.8	Görüntüden gölgelerin kaldırılması	30
Şekil 3.9	Görüntüye stilizasyon filtresinin uygulanması sonucu oluşan sonuç	31
Şekil 3.10	BGR renk uzayından griye dönüşüm	31
Şekil 3.11	Gaussian Bulanıklığının uygulanması sonucu oluşan yeni resim	32
Şekil 3.12	Adaptif eşikleme uygulaması	33
Şekil 3.13	Gürültü temizlemenin uygulanması	34
Şekil 3.14	Genişletme (dilation) uygulanması	34
Şekil 3.15	Kenar bulma (Canny) uygulanması	35
Şekil 3.16	Yeni görüntü olarak kaydedilecek nesnelerin saptanan kapalı geometrileri	36
Şekil 3.17	Bulunan kenarların çıktıları	37
Şekil 3.18	Eğitim kümesindeki sınıflara ait olan görsellerin dağılımı	38
Şekil 3.19	Test kümesindeki sınıflara ait olan görsellerin dağılımı	39
Şekil 3.20	Eğitim ve geçerleme doğruluğu	40
Şekil 3.21	Eğitim ve geçerleme kaybı	40
Şekil 4.1	Girdi görüntüsündeki konturların saptanması	41
Şekil 4.2	Veritabanında yapılan sorgu işlemi	43
Şekil 4.3	Veritabanında yapılan sorgu işleminden dönen sonuçlar	44

TABLO LİSTESİ

	Sayfa No
Tablo 2.1 Öznitelik çıkarıcıların özellikleri	7
Tablo 3.1 Her bir kategoride bulunan tarifi içeriğini ilgilenilen veri alanlarına göre çeken algoritma	23
Tablo 3.2 Malzemelere ait olan görsellerin toplanmasını sağlayan algoritma.....	23
Tablo 3.3 Bir kategoriye ait olan 2 veri setini birleştiren algoritma	24
Tablo 3.4 Sebze kategorisine ait malzemelerin kullanım frekansını gösteren tablo	25
Tablo 3.5 Giriş resmini BGR uzayına çeviren algoritma	28
Tablo 3.6 Giriş görüntüsünü optimal değerde ölçeklendirme	29
Tablo 3.7 Gölge kaldırma algoritması.....	29
Tablo 3.8 Stilizasyon filtresinin algoritması.....	30
Tablo 3.9 Gri renk uzayına dönüşüm algoritması.....	31
Tablo 3.10 Gri uzaydaki görüntüye Gaussian Bulanıklığı uygulayan algoritma	32
Tablo 3.11 Adaptif eşiklemeyi uygulayan algoritma	32
Tablo 3.12 Görüntüdeki istenmeyen konturları kaldıran algoritma	33
Tablo 3.13 Görüntüdeki konturları genişleten algoritma	34
Tablo 3.14 Görüntüdeki kenarları bulan algoritma.....	35
Tablo 3.15 Görüntüdeki ROI'leri ayrı birer görüntü olarak kaydeden algoritma.....	36
Tablo 4.1 Görüntü ön işleme sonucu oluşan çıktılar ve bu çıktıların ağa verilmesi sonucu elde edilen tahminler	43
Tablo 5.1 Literatürdeki CNN kullanan çalışmalara ait başarımlar oranları	45

KISALTMA LİSTESİ

Kısaltmalar	Açıklama
-------------	----------

<i>Data</i>	: Veri
-------------	--------

<i>Data Frame</i>	: Veri Çerçevesi
-------------------	------------------

<i>WWW</i>	: Dünya Çapında Ağ (World Wide Web)
------------	-------------------------------------

<i>SQL</i>	: Yapılandırılmış Sorgu Dili (Structured Query Language)
------------	--

<i>RDBMS</i>	: İlişkisel Veritabanı Yönetim Sistemi (Relational Database Management System)
--------------	--

<i>ER</i>	: Varlık-Bağıntı Modeli (Entity–Relationship Model)
-----------	---

<i>HTML</i>	: Hiper Metin İşaretleme Dili (Hypertext Markup Language)
-------------	---

<i>px</i>	: Piksel
-----------	----------

<i>RGB</i>	: Kırmızı Yeşil Mavi (Red Green Blue)
------------	---------------------------------------

<i>BGR</i>	: Mavi Yeşil Kırmızı (Blue Green Red)
------------	---------------------------------------

<i>CNN</i>	: Evrimsel Sinir Ağı (Convolutional Neural Network)
------------	---

<i>ROI</i>	: İlgi Alanı (Region Of Interest)
------------	-----------------------------------

<i>GGÖ</i>	: Google Görseller Örumceği
------------	-----------------------------

<i>YTSÖ</i>	: Yemek Tarifi Sitesi Örumceği
-------------	--------------------------------

ÖZET

YÜKSEK LİSANS TEZİ

NESNE TESPİTİ İLE YEMEK TARİFİ TAVSİYE SİSTEMİ

Yasin KÖKER

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman: Prof. Dr. Sevinç Gülseçen

II. Danışman: Dr. Murat GEZER

Yemek yapmak, hayatımızdaki en önemli günlük faaliyetlerden bir tanesidir. Ancak, çalışanlar ve öğrenciler gibi zamanı kısıtlı ve mutfakta az zaman geçiren kişiler için ne yemek yapacaklarına ve bunu yaparken eldekilerle neler yapabileceklerine karar vermeleri uzun zaman almaktadır. Karar verilse dahi, eldeki malzemelerle bir şeyler yapılmaya kalkışıldığında malzemeler arası uyumu yakalayabilmek kolay bir iş değildir. Bu sebepten ötürü internet ortamında varolan yemek sitelerinin sağladığı yemek tariflerine gözetilmakta fakat o yemek sitelerinde tüm malzemelere yer verildiği için kişileri ek malzeme almaya yönlendirmektedir. Bu çalışma ise bu sorunu, eldeki malzemelere göre yemek tarifi araması yaptırarak ortadan kaldırmaktadır.

Bu tez çalışmasında, kullanıcıdan alınan girdi görüntüsü; öncelikle görüntü ön işleme adımlarındaki işlemlerden geçirilmektedir. Bu işlemler sonucunda çıktı olarak birden fazla kontur saptanmakta ve bu konturlar ayrı birer görüntü olarak kaydedilmektedir. Bu kaydedilen görüntüler; derin evrişimsel sinir ağı modeline girdi olarak verilmekte ve tanımlama işlemi yapılmaktadır. Tanımlama aşamasında; transfer öğrenmesi (transfer learning) ile ImageNet veri kümesinde eğitilmiş derin evrişimsel sinir ağı mimarilerinden Xception seçilmiştir. Xception mimarisi üzerine ince-ayar (fine tuning) yapılarak özelleştirilmiştir.

Bu oluşturulan yeni ağ modelinde; doğruluk (accuracy); 0.9452, kayıp (loss); 0.4930, geçerleme kaybı (validation loss); 0.4883, geçerleme doğruluğu (validation accuracy); 0.9362'lik bir başarıyı yakalanmıştır.

Bu ağdan çıkan her bir görüntüye ait olan tanımlama etiketi bir listeye kaydedilmektedir. Tanımlama işlemi tamamlandığında bu listedeki değerler veritabanına sorgu olarak gönderilmektedir. Sorgu sonucu olarak da o listedeki malzemelerle hangi yemek tariflerinin yapılabileceğine dair bir yemek tarifi öneri sonucu dönmektedir.

Sonuç olarak, kişilerin sadece mutfaklarında bulunan malzemelere göre arama yapabilecek olmaları yemek yapacak olan kişinin dışarı çıkmasını, o an için alışverişe gitmesini engelleyecektir. Böylelikle zamandan ve maliyetten tasarrufta bulunabileceklerdir.

Temmuz 2020, 67 sayfa.

Anahtar kelimeler: veri madenciliği, yapay sinir ağları, derin öğrenme, görüntü tanıma, tavsiye sistemi.

SUMMARY

M.Sc. THESIS

RECIPE RECOMMENDATION SYSTEM VIA OBJECT DETECTION

Yasin KÖKER

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor: Prof. Sevinç Gülseçen

Co-Supervisor: Dr. Murat GEZER

Cooking is one of the most important daily activities in our lives. However, it takes a long time for people, such as employees and students, to decide what to cook and what they can do with what they have in their hand for those with limited time and little experience in the kitchen. Even if a decision is made, it is not an easy task to achieve harmony between ingredients when trying to do something with the ingredients at hand. For this reason, the recipes provided by the food sites available on the internet are browsed by the people, but since all the ingredients are included in those food sites, they direct people to buy additional ingredients. This study eliminates this problem by searching recipes according to the ingredients at hand.

In this thesis, the input image taken from the user; firstly the image is processed through the image pre-processing steps. As a result of these operations, more than one contour is detected as output and these contours are saved as separate images. These saved images are given as an input to the deep convolutional neural network model and recognition process is started. At the recognition stage; Xception was selected from deep convolutional neural network architectures trained in ImageNet dataset with transfer learning. It is customized by fine tuning on the Xception architecture. In this new network model created: accuracy; 0.9452, loss; 0.4930, validation loss; 0.4883, validation accuracy; 0.9362 has been achieved.

The recognition tag of each image that comes out of this network is saved in a list. When the recognition process is completed, the values in this list are sent to the database as queries. As a result of the query, a recipe suggests which recipes can be made with the ingredients in that list.

As a result, the fact that people can search only according to the ingredients in their kitchens will prevent the person who will cook to go out and go shopping for the moment. Thus, they will be able to save time and cost.

July 2020, 67 pages.

Keywords: data mining, artificial neural network, deep learning, image recognition, recommendation system.



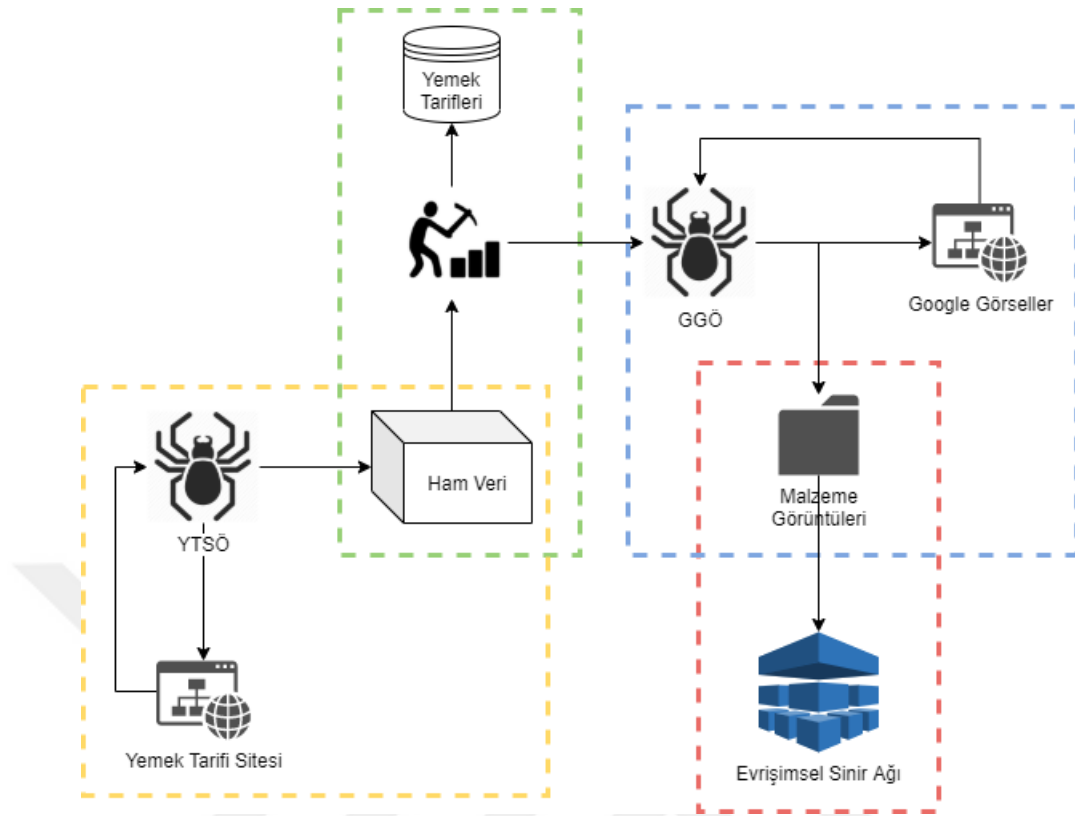
1. GİRİŞ

Yemek yapmak, hayatımızdaki en önemli günlük faaliyetlerden biridir. Ancak, çalışanlar ve öğrenciler gibi zamanı kısıtlı ve mutfakta az zaman geçiren kişiler için ne yemek yapacaklarına ve bunu yaparken eldekilerle neler yapabileceklerine karar vermeleri uzun zaman almaktadır. Karar verilse dahi, eldeki malzemelerle bir şeyler yapılmaya kalkışıldığında malzemeler arası uyumu yakalayabilmek kolay bir iş değildir. Bu sebepten ötürü internet ortamında varolan yemek sitelerinin sağladığı yemek tariflerine gözetilmakta fakat o yemek sitelerinde tüm malzemelere yer verildiği için kişileri ek malzeme almaya yönlendirmektedir. Bu çalışma ise bu sorunu, malzemelere göre yemek tarifi araması yaptırarak ortadan kaldırmaktadır. Kişilerin sadece mutfaklarında bulunan malzemelere göre arama yapabilecek olmaları yemek yapacak olan kişinin dışarı çıkmasını, o an için alışverişe gitmesini engelleyecektir. Böylelikle zamandan ve maliyetten tasarrufta bulunabileceklerdir.

Bilgi ve iletişim teknolojilerinin hızla gelişmesiyle beraber birçok alanda veri; sensörler, web teknolojileri, sosyal medya vb. araçlar vasıtasıyla üretilmektedir. Üretilen büyük miktardaki bu veri yığını ile beraber nesnelerin interneti, akıllı tanıma sistemleri vb. farklı alanlarda birçok ilgi çekici problem ortaya çıkmaktadır. Bu problemlere dair uygulamalardan biri de yemek konusudur. Yemek görüntülerinden yemek çeşidinin tanınması ve bu tanınan yemek çeşidine ait malzemelerin belirlenmesi, belirlenen bu malzemelerin mikro ve makro besin değerlerinin çıkarılması, internet ortamındaki yemek tariflerinden web içerik madenciliği yapılması vb. birçok problem mevcuttur. Bu çalışmada ise kullanıcıdan alınan girdi görüntüsündeki malzemeler tanımlanarak o malzemelerle yapılabilecek en yakın tarifleri öneren sistemin tasarımı yapılmıştır.

Bu çalışmada, yapılan işlemleri betimleyen figür **Şekil 1.1**'de gösterilmektedir ve bu işlemler sırasıyla şu adımlardan oluşmaktadır:

1. Sarı çizgilerle gösterilen alan içerisinde; yemek tarifi sitesinden içeriği toplayacak olan bir "yemek tarifi sitesi örümceği" (YTSÖ) yazılmıştır. YTSÖ vasıtasıyla 26 farklı yemek kategorisinde toplam 9457 yemek tarifi toplanmıştır. Bu yemek tariflerine ait de 11 farklı nitelik bulunmaktadır. Bu ham veri yığını 26 farklı csv dosyası altında toplanmıştır.



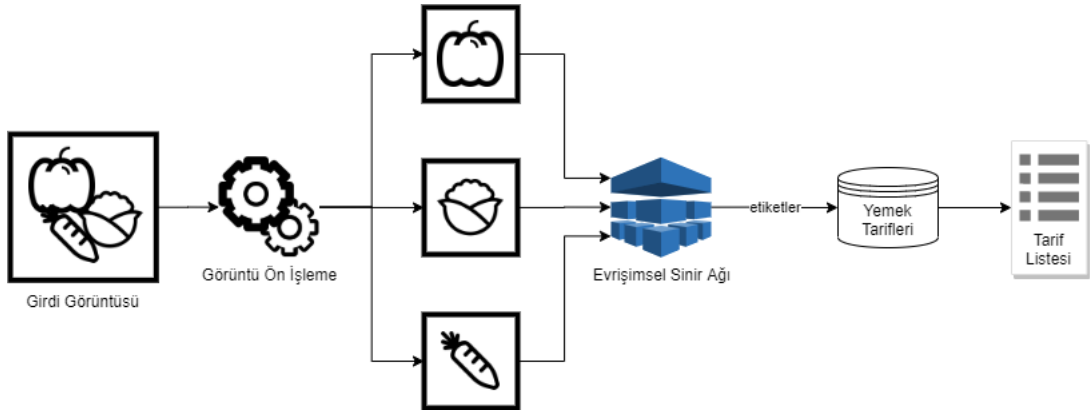
Şekil 1.1 Bu süreçte yapılan işlemler

2. Yeşil çizgilerle gösterilen alan içerisinde; ham veri yığınınında bulunan veriler; temizleme, indirgeme, bütünleştirme vb. veri madenciliği işlemlerine tabi tutularak bir yemek tarifleri veritabanı oluşturulmuştur. Ayrıca her bir emsalsiz malzeme adı da burada ayrıştırılarak 3. adımdaki örümceği beslemek üzere hazır hale getirilmiştir.

3. Mavi çizgilerle gösterilen alan içerisinde; Google Görseller'den veri toplayacak yeni bir "Google Görseller Örümceği" (GGÖ) yazılmıştır. GGÖ vasıtasıyla 2. adımda hazırlanan her bir emsalsiz malzeme adı için Google Görseller'deki görüntüler toplanmıştır. Bu toplanan görsellerin her biri kendi emsalsiz ismiyle yeni birer klasör altında toplanmışlardır.

4. Kırmızı çizgilerle gösterilen alan içerisinde; transfer öğrenimi (transfer learning) metodu kullanılarak Xception tabanlı bir evrişimsel sinir ağı mimarisi oluşturulmuştur. Bu ağı beslemek ve eğitmek için 3. adımda toplanan malzeme görselleri %80'i eğitim, %20'si de doğrulama olmak üzere ikiye ayrılarak kullanılmıştır.

Bu 4 adımda anlatılan işlemler yapıldıktan sonra oluşturulan sistemin, kullanıcı ile olan etkileşimi ise **Şekil 1.2**'de gösterilmektedir.



Şekil 1.2 Kullanıcı ile etkileşime giren sistem

Kullanıcıdan girdi olarak alınan görüntü öncelikle bir görüntü ön işleme sürecinden geçirilmektedir. Bu ön işleme sürecinde girdi görüntüsünde bulunan kenarlar saptanmakta ve her bir ayrı kapalı yüzeye sahip geometri, ayrı birer nesne olarak kabul edilmektedir. Bu nesneler, içerisinde sadece kendi geometrilerinin bulunduğu yeni birer görüntü olarak kaydedilmekte ve bir girdi listesi içerisinde kuyruğa alınmaktadırlar. Kuyruğa alınan bu görüntüler sırasıyla FIFO (ilk giren ilk çıkar) prensibine bağlı kalmak suretiyle evrişimsel sinir ağı modeline girdi olarak verilmektedir. Evrişimsel sinir ağı modelinde ise sınıflandırılma yapılmakta ve en yüksek tahmin oranına sahip olan etiket belirlenmektedir. Bu belirlenen etiketlerden bir etiketler listesi oluşturulmakta ve tanımlama işlemi bittiği zaman bu etiketler listesi yerel veritabanındaki sorguya değişkenler olarak gönderilmektedir. Bu gönderilen değişkenleri içerisinde bulunduran tarifler ise kullanıcıya gösterilmektedir.

Tez çalışmasının genel kısımlar bölümünde; veri, görüntü ön işleme teknikleri ve derin öğrenmeye ait kavramlar ve tanımlara değinilmiştir.

Malzeme ve yöntem bölümünde; verinin toplanması, verinin işlenmesi, nesne tanıma aşamasını oluşturan: görüntü ön işleme ve evrişimsel sinir ağı üzerinde durulmuştur.

Bulgular bölümünde ise yemek tarifi öneri aşamasından bahsedilmiştir.

Tartışma ve sonuç bölümünde; ortaya çıkan sonuçlar değerlendirilmiş ve gelecek çalışmalar için önerilerde bulunulmuştur.

2. GENEL KISIMLAR

2.1. LİTERATÜR İNCELEMESİ

Nesne tanıma aşaması için incelenen makalelerin çoğunda ya varolan bir yemeğin içerisindeki malzemeleri tanıma ya yemeğin kendisini tanıma üzerine yoğunlaşıldığı saptanmıştır. İncelenen çalışmalar aşağıda verilmiştir.

Chen ve diğerleri [1], Çin Mutfağına ait yemeklerin içerisindeki malzemeleri tanımaya çalışmışlardır. Bu çalışma için 3 farklı; Food-101 [2], PFID [3], VireoFood-172 [4] veri kümesinden faydalanılmıştır. Bu veri kümelerinden biri olan "VireoFood-172" [4], kendileri tarafından diğer iki veri kümesini de içine alarak oluşturulmuştur. Bu veri kümesindeki görüntülerin etiketlenmesinde 10 kişi 2 hafta boyunca çalışmıştır. Sonuç olarak 172 kategori, 353 farklı malzeme ve 110241 etiketlenmiş görüntüden oluşan bir Çin Mutfağı veri kümesi meydana getirilmiştir. Oluşturdukları modelin eğitimi için bu veri kümesindeki [4] görüntülerden rastgele olacak şekilde; %60'ı training, %10'u validation ve geriye kalan %30'u ise test için seçilmiştir. Sadece yemek içerisindeki malzemeleri tanıyarak yemeğin kategorisini ya da yemeğin kendisini tanıyarak içerisindeki malzemeleri tahmin etmektense her ikisinin de birlikte birbirini besleyebileceği bir sistem önerisi yapmışlardır. Ayrıca bir yemeğin içerisindeki malzemelerin şekli ve yapısı; doğrama, pişirme, sunum vb. şekillerde sürekli değişiklik gösterdiği için önerdikleri sistemin başarımını arttırmak adına Sıfır-Atış Öğrenmesi (zero-shot learning) kullanmayı da tercih etmişlerdir. Böylelikle sisteme daha önce tanımlanmayan bir yemek görüntüsü girdi olarak verildiğinde sistem, saptayabildiği malzemelerin birbiriyle olan semantik bağlantısından yola çıkmaya çalışacak ve kullanıcının önüne en yakın olan yemek tarifini getirecektir.

Güngör ve diğerleri [5], Türk Mutfağına ait yemekleri tanımaya çalışmışlardır. Bunun için öncelikle: Food-101 [2], Food-5K [6], Food-11 [7] veri kümelerinden Türk Mutfağına yakın olan yemekler seçilmiştir. Ardından geri kalan kısım Google Görseller'den toplanan görüntüler ile oluşturulmuştur. Son olarak da tüm bu veri kümeleri tek bir kümede "Türk Sofrası" adı altında birleştirilmiştir. Türk Sofrası veri kümesinde, 113 sınıf ve veri büyütme (image augmentation) işlemi uygulanarak oluşturulan toplam 113000

etiketli yemek görüntüsü bulunmaktadır. Eğitim aşamasında TensorFlow Kütüphanesi'nden faydalanılmıştır. Kendi mimarilerini oluşturmak yerine zamandan ve donanımdan tasarruf etmek adına öğrenme transferi yapılmıştır. Bu öğrenme transferinde mimari olarak, Google'ın Inception V3 mimarisini [8], model ağırlığı olarak da ImageNet [9] veri kümesi seçilmiştir. Ardından bu mimarideki [8] son katman atılmış ve yerine 113 sınıflandırıcıdan oluşan katman eklenmiştir. Türk Sofrası veri kümesi de %80 training, %10 validation, %10'da test için ayrılmıştır. Girdi resimleri de 299x299 boyutunda ağa verilmiştir. Test doğruluk oranında %62,7'lik bir oran yakalanmıştır.

Ergün ve diğerleri [10], Türk Mutfağına ait yemeklerden oluşan 50'si kendilerinin 51'i Food-101'den [2] olmak üzere toplamda 101 sınıflık bir veritabanı oluşturmuşlardır. Bu veritabanının içerisinde yemeklerin özellikleri, tarifleri ve resimleri de bulunmaktadır. Ayrıca semantik bir veri analizi yapılarak yemeklerle ilgili bilgiler hiyerarşik bir ontoloji modeline oturtulmuştur. Bu ontoloji sınıflarını oluşturmak için Protege programı kullanılmıştır. Sıfırdan bir derin öğrenme mimarisi oluşturmak yerine Google'ın Inception V3 mimarisi [8], model ağırlığı olarak da ImageNet [9] veri kümesi seçilmiştir. Ardından bu mimarideki [8] son katman atılmış ve yerine 101 sınıflandırıcıdan oluşan katman eklenmiştir. Test doğruluk oranında ortalama %78'lik bir oran yakalanmıştır. Sonuç olarak bu eğitim sonucunda girdi olarak verilen bir yemek görüntüsünü sistem tanımlayıp, ontoloji aracılığı ile saptanan yemeğe ait semantik verilere erişilebilmektedir.

Maruyama ve diğerleri [11], Android işletim sistemine sahip mobil cihazlar için gerçek zamanlı nesne tanımlaması yaparak yemek tarifi öneren bir uygulama geliştirmişlerdir. Bunun için içerisinde 30 farklı yemek malzemesi barındıran deneysel bir veritabanı oluşturmuşlardır. Uygulamaya girdi görüntüsü cihazın kamerası vasıtasıyla verilmektedir. Görüntüye ait özellikleri çıkartmak için Hızlandırılmış Gürbüz Öznitelikler (SURF) [12], görüntüye ait özellikleri barındıran vektörleri tek bir vektöre indirgemek için de Öznitelik Torbası (BoF) [13] kullanılmıştır. Sınıflandırma işlemi için Destek Vektör Makinesi (SVM) [14] kullanılmaktadır. En yüksek orana sahip 6 aday için %83.93'lük bir başarımlık oranı yakalanmıştır. Malzeme tanımlandıktan sonra malzeme adı bir web api üzerinden ilgili yemek sitesine gönderilmekte ve o websitesinden de sonuç olarak yemek tarifi dönmektedir.

Beijbom ve diğerleri [15], girdi olarak restoranda sipariş edilen yemeğin görüntüsüyle beraber kullanıcının lokasyon bilgisini de ek bağlam olarak alıp tanımlama performansını

arttıran ve çıktı olarak da toplam kalori hesabını veren bir sistem geliştirmişlerdir. Bu sisteme özel veri kümesini oluşturmak için 3 farklı restorana gidilip restoranların menülerindeki her bir yemeğin fotoğrafı farklı açılarda çekilmiştir. Bunun sonucunda 41 yiyecek kategorisinde 646 farklı görüntü ve bu görüntülerdeki malzemelerin etiketlenmesinden oluşan 1386 malzemelik bir veri kümesi oluşturulmuştur. Malzemelerin kalori miktarları ise o restoranlar ile çalışan diyetisyenler tarafından oluşturulmuştur. Tanıma çerçevesi (recognition framework) için görüntülerden beş tür temel özellik çıkartılmıştır. Bunlar sırasıyla: renk, Gradyan Yönlendirme Histogramı (HOG) [16], Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) [17], Yerel İkili Örüntü (LBP) [18], Maksimum Tepki-8 Filtreleme Bankası'dır (MR8 Filter Bank) [19]. Veri kümesi %90'ı eğitim, %10'u da test olarak ayrılmıştır. Görüntüleri sınıflandırma için 205 sınıflık (41 yemek kategorisi * 5 temel özellik) bir SVM [14] kullanılmıştır. En yüksek orana sahip 5 aday için %92'lik bir başarı oranı yakalanmıştır.

Hashimoto ve diğerleri [20], kullanıcı odaklı bir yemek hazırlama destek sistemi önerisi yapmışlardır. Buna göre sistem; diğer yardımcı sistemlerdeki [21], [22] gibi kullanıcıyı belli bir sıraya mahkum etmek yerine, kullanıcının bulunduğu alanı ve de kullanmakta olduğu malzemelerin, hangi alanlar arasında geçiş yaptığını kameralar vasıtasıyla takip etmekte ve yapılmakta olan yemek tarifinin hangi adımında olduğunu anlamaktadır. Böylelikle bir sonraki adımı tahmin ederek kullanıcıyı yönlendirmektedir. Sistem 3 aşamadan meydana gelmektedir. Bunlar yiyeceği takip etme (kullanıcının elindeki malzeme ocak alanında mı yoksa diğer alanlarda mı vb), malzemeyi tanıma ve yapılan eylemi (doğrama, soyma vb) tanımadır. Bunun için mutfak; eviye, tezgah ve ocak alanı olmak üzere 3 ayrı kısma ayrılmıştır. Bu alanlardaki kullanıcı aktivitesini ve malzemeleri tanımak için her alana birer adet düşecek şekilde toplamda 3 ayrı optik kamera kullanılmıştır. Ocak alanındaki optik kameraya ek olarak bir adette termal kamera eklenmiştir. Böylelikle ocak alanındaki sıcaklık farkından hangi pişirme yönteminin uygulandığı tanınmaktadır. Yemek tarifi adımları silsilesinde ve yapılan eylemi tanımadaki karar ağaçları kullanılmaktadır. Sisteme 5 eylem türü tanımlanmış ve %84,6 oranında hangi eylemin yapılmakta olduğu belirlenmiştir. Ayrıca 4 farklı malzeme için 10 hazırlama adımı tanımlanmış ve bu hazırlama adımları takip edilerek %93,8 oranında hangi malzemelerin kullanıldığı tanınmıştır.

Öznitelik Çıkarıcı Adları	Özellikleri
Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT)	Ölçekleme ve dönme değişimlerinden etkilenmeyen bir tanımlayıcıdır. Öznitelik noktası algılama (anahtar noktası), öznitelik noktası yerelleştirilmesi, yön atama ve öznitelik tanımlayıcı oluşturma olmak üzere 4 aşamadan meydana gelmektedir [17].
Hızlandırılmış Gürbüz Öznitelikler (SURF)	SIFT'in hızlandırılmış bir sürümüdür. Kutu filtreleri kullanarak hesaplamaları daha hızlı yapmaktadır. Böylelikle takip ve nesne tanımlama gibi gerçek zamanlı uygulamaları mümkün kılmaktadır [12].
Gradyan Konumu ve Yönlendirme Histogramı (GLOH)	Histogramlar için uzamsal bölgeleri daha fazla dikkate alan SIFT benzeri bir tanımlayıcıdır. Bir ara vektörü toplam 272 boyut için 17 konum ve 16 yönlendirme kutusundan hesaplamaktadır [72].
Gradyan Yönlendirme Histogramı (HOG)	Bir görüntünün yerel kısımlarındaki gradyan yönelim oluşumlarını saymaktadır. SIFT'e ve şekil bağlamlarına benzemektedir. Yüksek doğruluk için örtüşen yerel kontrast normalizasyonu kullanılmaktadır. [16]
Yerel İkili Örüntü (LBP)	3x3'lük bir pencerede, ortadaki pikseli merkez alıp komşu piksellere göre eşiklemenin yapıldığı ve bu eşikleme sonucunu ikili olarak sonuç döndüren bir doku operatörüdür [73].
Hızlandırılmış Segment Testinden Öznitelikler (FAST)	Öznitelik noktalarını çıkarmak ve daha sonra birçok bilgisayar görme görevinde nesneleri izlemek ve eşlemek için kullanılabilen bir köşe algılama yöntemidir. Ayrıca makine öğrenme teknikleri uygulandığında, hesaplama süresi ve kaynakları açısından üstün performans elde edilmektedir. Bu yüzden gerçek zamanlı video işleme uygulamaları için çok uygun bir yöntem olmaktadır [75].
İkili Gürbüz Bağımsız Temel Öznitelikler (BRIEF)	SIFT'teki gibi anahtar noktalar tespit edildikten sonra bunları ikili öznitelik vektörlerine çevirmektedir. Her bir anahtar noktası 128-512 bit dizisi (string) olan bir öznitelik vektörü ile tanımlanmaktadır. Hız ve tanıma oranı açısından SURF ve SIFT gibi diğer tanımlayıcılardan daha iyi performans göstermektedir [68].
Yönlendirilmiş FAST ve Döndürülmüş BRIEF (ORB)	SIFT ve SURF patentli algoritmalar olduğu için OpenCV Labs tarafından geliştirilmiştir. ORB, FAST anahtar nokta dedektörü ve BRIEF tanımlayıcısı üzerine kurulmuştur [61].

Tablo 2.1 Öznitelik çıkarıcıların özellikleri

2.2. VERİ

Data (veri), Latince vermek anlamında kullanılan dare fiilinin geçmiş zaman çekimi olan datum kelimesinin çoğuludur. Veri; tecrübe, gözlem ve deney sonucunda elde edilen, sayı, kelime veya görsellerden oluşan, düzenlenmiş enformasyon kümesini ifade etmektedir [57].

Veriden bilgiye giden patikada aşamalar genel olarak enformasyon piramidi ile gösterilir. Bu piramidin en temel hali sadece veri (data), enformasyon (information), bilgi (knowledge) ve bilgelik (wisdom) bileşenlerinden meydana geldiği için, piramit DIKW Piramidi olarak da isimlendirilmektedir [24].

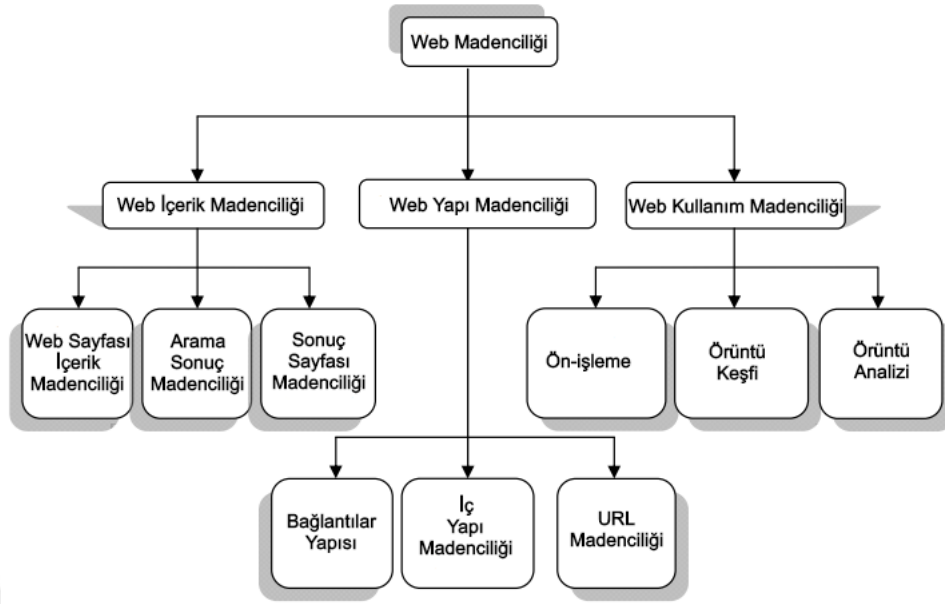


Şekil 2.1 DIKW piramidi [23]

2.2.1. Web’de Veri Toplama

İnternet ortamında üretilen verinin her geçen gün artmasıyla birlikte, üretilen bu veriden; arzulanan bilgi kaynaklarına erişmek, kullanıcı hareketlerini takip ve analiz edip bir kullanım deseni oluşturmak, bunu yaparken de bu işlemi otomatize hale getiren araçlar kullanmak elzem hale gelmektedir. Bu faktörler de, hem sunucu hem de istemci tarafında bu ihtiyaçları doyuracak akıllı sistemler yaratma gereğini doğurmaktadır [25].

Web madenciliği genel olarak, World Wide Web’deki bilgilerin keşfi ve analizi olarak tanımlanmaktadır. Web madenciliği toplanacak bu bilgiyi; web bağlantılarından, sayfa içeriklerinden ve web kullanımından elde etmektedir. Web madenciliği üç ana kolda ele alınmaktadır. Bunlar; web içerik madenciliği, web yapı madenciliği ve web kullanım madenciliğidir [26].



Şekil 2.2 Web madenciliği sınıflandırması [27] [28]

Web içerik madenciliğinde faydalı ve kullanılabilir veriler, web sayfalarının doğrudan içeriğinden veya kaynak kodlarındaki HTML etiketlerinden elde edilmektedir [29].

Web yapı madenciliği kullanılabilir verileri, web sayfalarında bulunan bağlantılardan çıkarmaktadır. Bu bağlantılar kullanılarak hangi sayfanın daha önemli olduğu vb. bilgiler elde edilebilmektedir [30].

Web kullanım madenciliğinde kullanılan birincil veri kaynağı, genellikle bir web sunucusu günlüğünde toplanan ziyaretçilerin web sitesine yaptıkları isteklerin bir kaydından meydana gelmektedir [31].

Yukarıda anlatılan web madenciliği yöntemlerinin uygulanabilmesi için ihtiyaçlara özel olarak oluşturulmuş "örümcek" adı verilen araçlar mevcuttur. Bu içerik toplama araçlarından biri olan "Scrapy Web Spider"; web sitelerini tarayarak yapısal veriyi ortaya çıkartmaktadır. Böylelikle bu veriyi; veri madenciliği, bilgi işleme veya tarihsel arşivleme gibi çok çeşitli alanlarda kullanabilmek mümkün hale gelmektedir [32].

2.2.2. Veritabanı Oluşturma

Edgar Frank Codd tarafından 1970 yılında öne sürülen ilişkisel veri modeline dayanan veritabanlarına ilişkisel veri tabanı yönetim sistemi (Relational Database Management

System) denmektedir. Tablo (table), anahtar (key), sorgu (query) ve ilişki (relation) adı verilen temel kavramlara sahiptir [33].

Tablo: Veriler, satır ve sütunlar hâlinde tablolarda saklanmaktadır. Tablonun her satırı bir kaydı, her bir sütunu ise o kayda ait bir nitelik verisini tutmaktadır. **Şekil 2.3**'te bir tablo örneği görülmektedir.

recipe_id	author_name	340x940px_de	recipe_link	recipe_name	how_to	cooking_time	prep_time	serving_size
Fil...	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Emre Turan	https://cdn...	https://...	Fırında Patates C...	Patateslerin kab...	30 dakika	15 dakika	1 kase
2	Emre Turan	https://cdn...	https://...	Mantar Kızartması	Mantarlarının yü...	15 dakika	10 dakika	4 kişilik
3	Zeynep Çetin	https://cdn...	https://...	Baharatlı Patlam...	Mısır patlatma ...	10 dakika	10 dakika	4 kişilik
4	Cemre Özgüzel	https://cdn...	https://...	Cajun (Kajun) B...	Patateslerin kab...	25 dakika	15 dakika	4 kişilik

Şekil 2.3 Tablo örneği

Anahtar: Birincil anahtar (Primary Key) ve yabancı anahtar (Foreign Key) olmak üzere 2 adettir. Birincil anahtar bir kaydı başka bir kayıttan ayırmayı sağlayan birbirinden farklı, emsalsiz bir niteliklerdir (**Şekil 2.3**'teki recipe_id sütunu). Dış anahtar ise, aynı veya başka bir tablodaki birincil anahtar ile ilişkide olan niteliklerdir. Birincil anahtardan farklı olarak dış anahtarda aynı değer başka kayıtlarda tekrar edebilmektedir (**Şekil 2.6**'daki recipe_ingredient_id sütunu).

Sorgu: Veritabanının bir veya daha çok tablosunda tutulan kayıtlar içerisinde istenen niteliklere ait değerlerin gösterilmesi için oluşturulan komutlar bütünüdür. İlişkisel veritabanı yönetim sistemlerinde sorgulama dili olarak SQL (Structured Query Language) kullanılmaktadır. Örnek bir SQL sorgusu; **Şekil 2.4**'te ve bu sorgudan dönen sonuç ise **Şekil 2.5**'te görülmektedir.

```

1 SELECT recipe_item_qty, recipe_item_name
2 FROM
3     RecipeTable,
4     RecipeIngredientTable,
5     SyncRecipeIngredientTable
6 WHERE
7     RecipeTable.recipe_id = SyncRecipeIngredientTable.recipe_id
8     AND RecipeIngredientTable.recipe_ingredient_id = SyncRecipeIngredientTable.recipe_ingredient_id
9     AND SyncRecipeIngredientTable.recipe_id = 9457;
```

Şekil 2.4 Sorgu örneği

İlişki: Tablolar arasındaki veri tutarlılığını sağlama amacıyla kurulan bağlantılardır. Bir tablodaki birincil anahtar ile başka bir tablodaki dış anahtarın birbirine bağlanması

	recipe_item_qty	recipe_item_name
1	10 adet	dolmalık biber
2	300 mililitre	pirinç
3	7 adet orta boy	kuru soğan
4	2 çay kaşığı	yenibahar
5	2 tutam	tarçın
6	2 çay kaşığı	karabiber
7	3 çay kaşığı	kuru nane
8	2 çay kaşığı	tuz
9	1 avuç	dolmalık fıstık
10	1 avuç	kuş üzümü
11	1 tatlı kaşığı	şeker
12	1 demet	maydanoz

Şekil 2.5 Sorgu örneğinden dönen sonuç

recipe_id	recipe_ingredient_id
Filter	Filter
8	67
8	68
8	69
8	70
8	71
8	72
8	73
8	74
9	75
9	76
9	77
9	78

Şekil 2.6 İlişki örneği

ile oluşmaktadır. Bir ilişki örneği Şekil 3.4'te "SyncRecipeCategoryTable" ve "SyncRecipeIngredientTable"da görülmektedir.

Bu tezde, veritabanı yönetim sistemi olarak SQLite kullanılmaktadır. SQLite'ı diğer yönetim sistemlerinden (MySQL, Oracle, PostgreSQL) ayıran en önemli nokta sunucuya ihtiyaç olmaksızın, kullanılmakta olan cihazın içerisinde gömülü olarak kullanılabilesidir [34].

2.2.3. Veri Madenciliğinde Ön İşleme Süreci

Veri madenciliği çalışmalarında iyi bir sonuç elde edilebilmesi başlangıçta hazırlanan verinin uygunluğuna ve kalitesine bağlıdır. Veride ne kadar az gürültü mevcutsa elde edilebilecek sonuçların kalitesi o denli artacaktır. Sorunlu verinin temizlenmesi, eksik verinin bir şekilde tamamlanması ve yeni özniteliklerinin oluşturulması bu aşamanın önemli öğeleridir [24].

Veri ön işleme, veri analizi sürecinin süratle tamamlanabilmesi ve veri analizi sonuçlarının kalitesinin artırılması amacı ile veri üzerinde gerçekleştirilen veri entegrasyonu, veri temizleme ve veri indirgeme işlemleridir [35].

2.2.3.1. Veri Entegrasyonu

Veri entegrasyonu ya da diğer adıyla veri bütünleştirme tanımlanan problem için verinin bir çok kaynaktan toplanması, seçilmesi ve bütünleştirilerek tek bir kaynaktan bir araya getirilmesi işlemidir. Günümüzde entegre sistemlerin kullanılması önem taşısa da, on yılların

birikimi ile gelen çeşitli iç ve dış veritabanlarının bir araya getirilmesi ve bunların özellikle veri ambarı gibi veri depolarında saklanması, veri ön işlemenin zaman ve işgücü açısından en ağır işlemidir [24].

2.2.3.2. Veri Temizleme

Kullanılacak olan veri dizilerinin farklı kaynaklardan toplanması, doğal olarak veri uyumsuzluklarına neden olacaktır. Bu uyumsuzlukların başlıcaları farklı zamanlara ait olmaları, kodlama farklılıkları ve farklı ölçü birimlerinde olmalarıdır [24]. Ayrıca yanlış ve aşırı uçta bulunan verilerin ortadan kaldırılması da verilerin temizlenmesi konusuna girmektedir [36].

2.2.3.3. Veri İndirgeme

Günümüzde özellikle bellek kapasitelerinin artması ve fiyatlarının ucuzlaması sonucunda, had safhada veri obezitesi problemi yaşanmaktadır. Çok büyük ölçülerde gerekli/gereksiz sağlanan bu veri yığınları, giderek veri analizi çalışmalarını boğmaktadır. Artan boyut sayısı ve toplamda veri miktarı, çeşitli veri indirgeme yöntemleri ortaya çıkarmaktadır [24]. Bunlar; gereksiz değerlerin silinip orjinal verinin karakterinin korunduğu özniteliklerin silinmesi, kayıtların silinmesi ve özniteliklere ait değerlerin silinmesi olmak üzere üç farklı yöntem kullanılmaktadır [37].

2.3. GÖRÜNTÜ ÖN İŞLEME TEKNİKLERİ

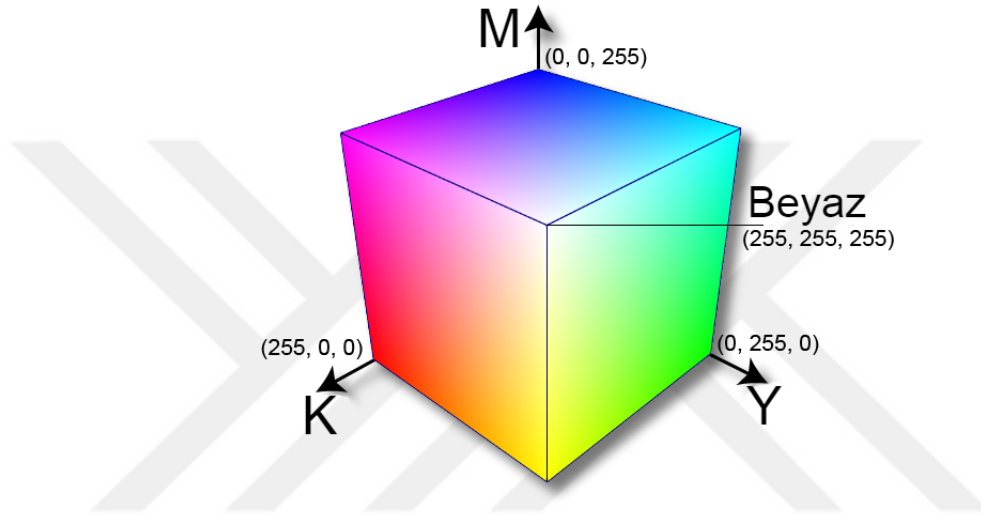
Bu bölüm; ilgi alanı (Region Of Interest, ROI) haricinde kalan nesne, gölge vb etmenlerin tespit edilip, girdi görüntüsünden kaldırılmasını ve ROI'deki her bir nesnenin ayrı ayrı farklı birer görüntü olarak kaydedilmesini içermektedir.

2.3.1. Renk Uzayları

Renk çeşitliliklerinin fazla olması sebebiyle bu renkleri belirli kümeler altında gruplama ihtiyacı doğmuştur. Bu renkleri standartlaştırma ve gruplama kavramına da renk uzayı denmektedir. Her bir renk uzayı, renk kümesini tanımlamak için kendine has bir yapıya sahiptir [38].

2.3.1.1. RGB Renk Uzayı

Bu renk uzayı; Red(Kırmızı), Green(Yeşil) ve Blue(Mavi) renklerinin baş harfi ile adlandırılmıştır. Her bir piksele ait olan renkler; 3 renk olan kırmızı, yeşil ve mavinin belirli yoğunluklarda karıştırılması ile elde edilen renk kodunu tutmakta ve katmanda saklanmaktadır. **Şekil 2.7**'de her bir renk için 256 farklı değer bulunduran renk uzayı küpü görülmektedir.



Şekil 2.7 RGB renk uzayı [39]

2.3.1.2. BGR Renk Uzayı

OpenCV girdi görüntülerini BGR; Blue(Mavi), Green(Yeşil), Red(Kırmızı) formatında okumaktadır. Çünkü OpenCV ilk geliştirilmeye başlandığında, kamera üreticileri ve görüntü işleme üzerine yazılım geliştiren firmalar BGR renk formatını tercih etmiştir. Kırmızı kanal, en az önemli renk kanallarından biri olarak kabul edilmekte ve en son sırada listelenmektedir [40].

2.3.1.3. Gri Renk Uzayı

Bu renk uzayı karanlıktan aydınlığa doğru monokromatiktir. Böylelikle, renkli görüntülerde olduğu gibi sadece gri ölçek görüntüsü üç değil, tek bir kanaldan oluşur. Siyah ve beyaz olmak üzere 2 rengin birbiriyle olan karışımlarından meydana gelmektedir. 0; siyah renge, 255; beyaz renge ve bu değerler arasındaki gri seviyeler ise gri tonlara karşılık gelmektedir [41].

2.3.2. Resmi Yeniden Boyutlandırma

Resmi yeniden boyutlandırma, resimlerin ölçeklendirilmesini ifade etmektedir. Resim ölçekleme, bir resimdeki piksel sayısını azaltmaya yardımcı olmaktadır. Böylelikle, resmin işlenmesi için gereken işlem yükü azalmaktadır.

2.3.3. Fotogerçekçi Olmayan İşleyici - Üsluplaştırma Filtresi

Üsluplaştırma filtresi (Stylization Filter); her nesnenin kenarını, dış çizgisini veya konturunu pürüzsüz ve aynı zamanda keskin hale getirmektedir [42]. Bu filtrenin uygulanması, resim kenarlarının yanı sıra nesne anahatlarını da çok görünür ve keskin hale getirmektedir. Böylece, her nesnenin anahattını veya konturunu algılamak ve ayırmak için kullanılan kenar saptama algoritmasına yardımcı olmaktadır [43].

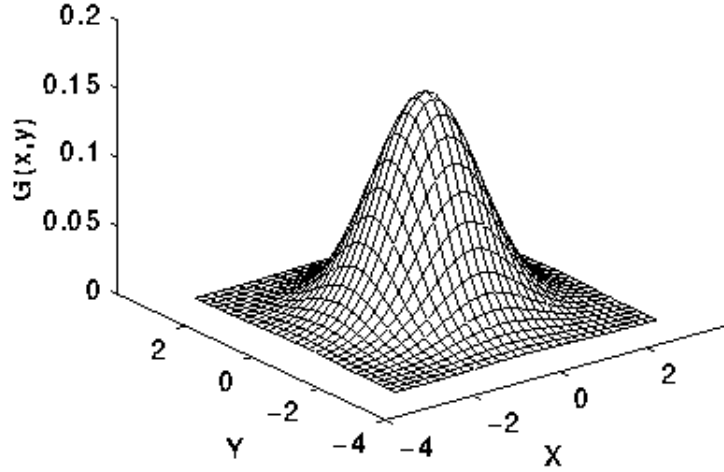
2.3.4. Gaussian Bulanıklığı

Resmi bulanıklaştırarak, gürültüyü azaltmak için kullanılmaktadır. Bir resim üzerinde uygulanan 2D Gaussian formülü 2.1'de, bu denkleme ait olan dağılımın grafiği ise **Şekil 2.8**'de gösterilmektedir [44].

$$[!h]G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

2.3.5. Adaptif Eşikleme

Eşikleme, girdi olarak verilen renkli ya da gri renk uzayına sahip bir görüntüyü ikili (binary) hale getiren bir yöntemdir. Görüntüdeki her piksel için bir eşik hesaplanmalıdır. Piksel değeri eşik altına düşerse arkaplan olarak değilse de ön plan olarak ayarlanmaktadır. Adaptif eşikleme ise, görüntü üzerindeki eşik değerini dinamik olarak değiştirmektedir. Böylelikle görüntüdeki değişen aydınlatma koşullarına; ışık patlamaları, gölgeler vb. uyum sağlayabilmektedir [45].



Şekil 2.8 Gaussıan dağılım grafiğı [44]

2.3.6. Morfolojik İşlemler

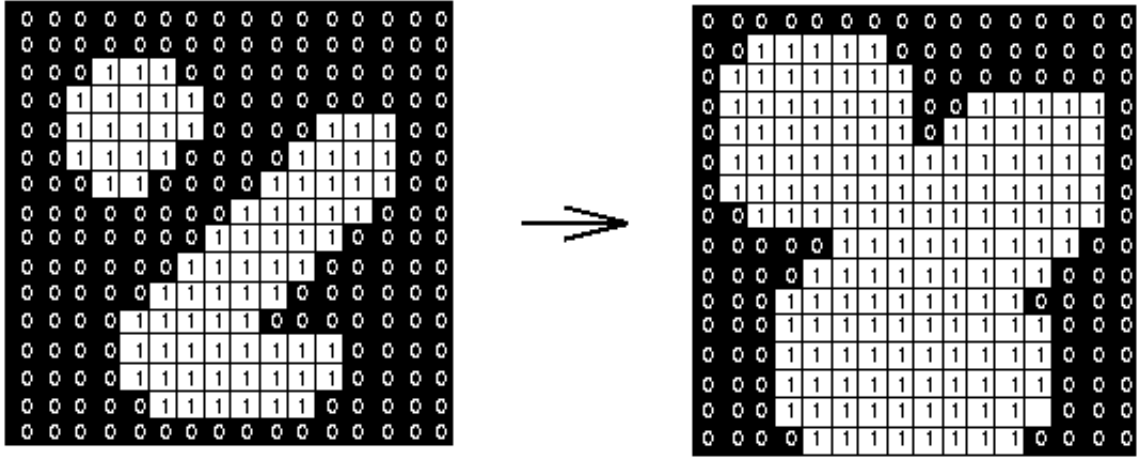
İkili (binary) bir görüntü ve bir yapılandırma ögesini girdi olarak almakta ve bunları küme operatörü (kavşak, birleşim, dahil etme, tamamlayıcı) kullanarak birleştirmektedir. Giriş görüntüsündeki nesneleri, yapılandırma elemanında (3x3, 5x5) kodlanan şeklin özelliklerine göre işlemektedir [46].

2.3.6.1. Genişletme

Girdi olarak verilen ikili bir görüntüdeki ön planda olan nesnenin piksellerini (genelde beyaz renkte) kademeli olarak genişletme işlemidir. Böylece ön plan piksellerinin alanları büyürken, bu bölgelerdeki delikler küçülmektedir. Şekil 2.9'da genişletme işleminin uygulanışı görülmektedir [47].

2.3.7. Kenar Bulma

Kenarlar, görüntüde iki farklı alan arasındaki sınırda gerçekleşmiş olan yoğunluk kontrastına sahip yerlerdir [48]. Kenarlar genellikle nesne sınırlarını temsil eden görüntü konumlarında meydana geldiğinden, görüntüyü farklı nesnelere karşılık gelen alanlara bölmek istediğimizde görüntü algılamada, kenar algılama yaygın olarak kullanılmaktadır. Bir görüntünün kenarlarından temsil edilmesi, görüntü bilgilerinin çoğunu korurken veri miktarının önemli ölçüde azaltılması avantajına da sahip olmaktadır [49].



Şekil 2.9 3x3 yapısal eleman matrisi kullanılarak yapılan genişletme işlemi [47]

2.3.7.1. Canny Kenar Dedektörü

Canny kenar bulma algoritması 4 adımdan meydana gelmektedir.

1. Görüntü (I), gürültüyü azaltmak adına Gaussian Bulanıklığı ile yumuşatılır [54].

$$G * I \quad (2.2)$$

2. Her piksel için yerel kenar normal yönleri bulunur.

$$\bar{\mathbf{n}} = \frac{\nabla(G * I)}{|\nabla(G * I)|} \quad (2.3)$$

3. Kenar büyüklükleri hesaplanır.

$$|\nabla(G * I)| \quad (2.4)$$

4. Kenarın normal yönleri boyunca sıfır-geçişler (zero-crossings) bularak kenarlar bulunur [55]. (en büyük olmayan bastırma)

$$\frac{\partial^2(G * I)}{\partial \bar{\mathbf{n}}^2} = 0 \quad (2.5)$$

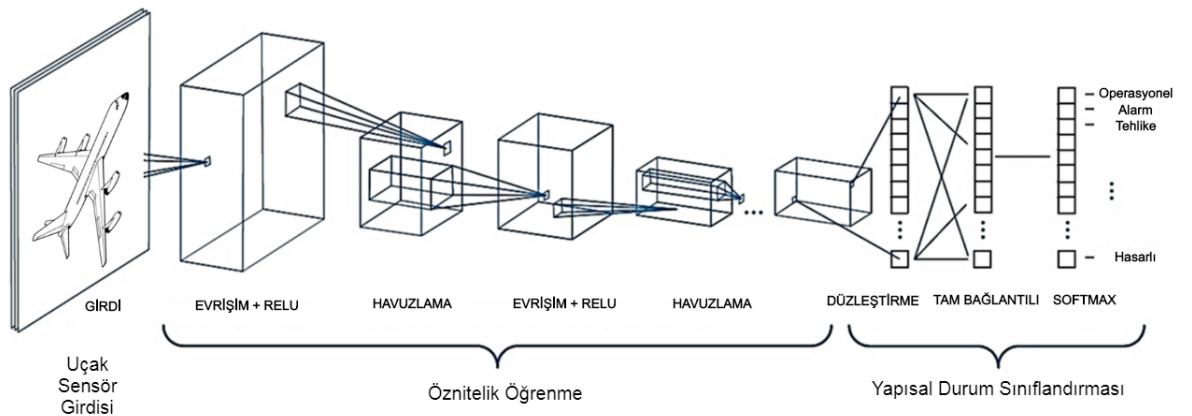
2.4. DERİN ÖĞRENME

Yapay Sinir Ağları (YSA), insan beyninin sinir yapısını taklit ederek beynin öğrenme ve hatırlama yoluyla yeni bilgiler üretebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilmiş yazılımlardır [50]. Von Neumann bilgisayarlarında olduğu gibi program komutlarının sırasal olarak icra edilmesi yerine, YSA modelleri birçok hesaplama elemanının değişken ağırlıklarla birbirlerine bağlanmasını ve yoğun paralel işlemlerin yapılmasını öngören çeşitli varsayımlara dayandırılmaktadır [24].

Derin öğrenme, çok büyük miktarlarda işaretlenmiş verilerden özellik saptayabilen sistemler oluşturmak için derin sinir ağlarının kullanılmasıdır [58]. Buradaki derin kavramı, ağdaki katman sayısını ifade etmektedir. Ağdaki katman sayısı arttıkça ağın yapısı da derinleşmektedir. YSA'lar en fazla iki veya üç katmandan meydana gelirken derin ağlar yüzlerce katmandan meydana gelmektedir.

2.4.1. Evrişimsel Sinir Ağları Mimarisi ve Katmanları

Evrişimsel Sinir Ağları (ESA) yapısı ilk olarak 1989 yılında Lecun tarafından kullanılmıştır. ESA yapısı; giriş katmanı (input layer), evrişim katmanı (convolution layer), düzleştirilmiş doğrusal birim katmanı (relu), havuzlama katmanı (pooling layer), tam bağlantılı katman (dense layer), seyreltme katmanı (dropout layer) ve sınıflandırma katmanından (classification layer) meydana gelmektedir [51].



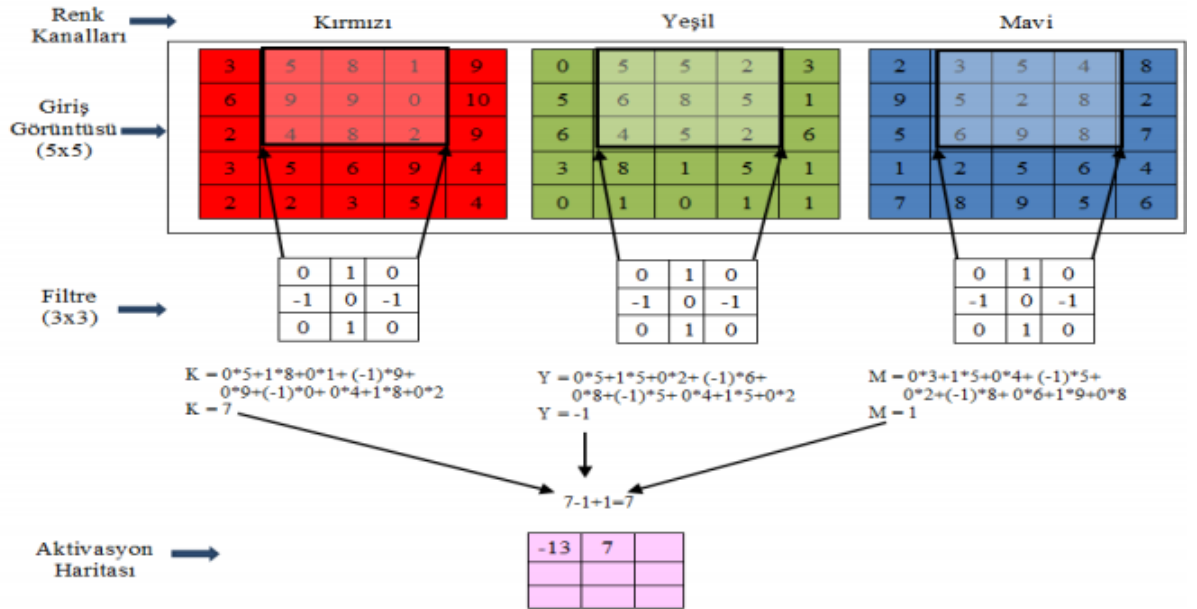
Şekil 2.10 ESA katmanları [52]

2.4.1.1. Giriş Katmanı

Giriş katmanı, ESA'nın ilk katmanını oluşturmaktadır. Oluşturulmak istenen modelin başarılı bir şekilde sonuç verebilmesi için giriş katmanındaki verinin boyutu önem kazanır hale gelmektedir. Bu girdideki görüntünün boyutu arttıkça, ihtiyaç duyulan bellek miktarı ve hesaplama zamanı da artmakla beraber başarımlar oranı da doğrusal olarak artış göstermektedir [53].

2.4.1.2. Evrişim Katmanı

Evrişim katmanı, dönüşüm katmanı olarak bilinmektedir. Burdaki dönüşüm işlemi, 2x2'lik ya da 3x3'lük bir filtrenin girdi görüntüsü üzerinde yatay ve dikey olarak kaydırılması işlemine dayanmaktadır. Her bir kaydırılarak (stride) dolaşmada resmin değerleri (genişlik, yükseklik, kanal sayısı) ile filtrenin değerleri çarpılıp sonuçları toplanarak çıkış değerleri tablosu oluşturulmaktadır. Bu çıkış değerleri sonucu oluşan tabloya da aktivasyon haritası (özellik haritası) denmektedir. Bu aktivasyon haritasında ise her bir filtreye özgü olan özellikler bulunmaktadır [53] [69].



Şekil 2.11 Evrişim işleminin uygulaması [53]

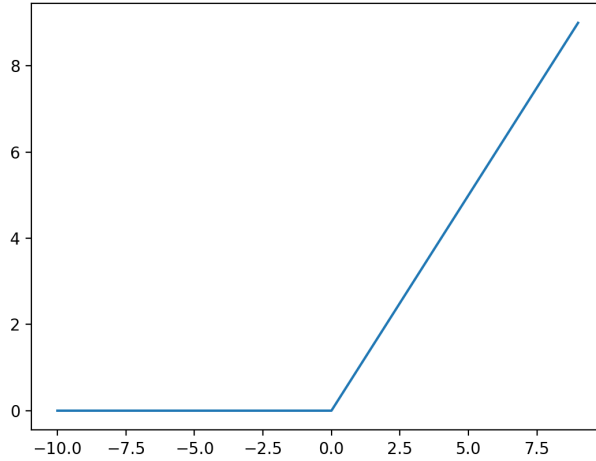
Şekil 2.11'de 5x5 boyuta sahip RGB bir girdi (5x5x3) görüntüsünün her bir renk kanalına, 3x3'lük bir filtre vasıtasıyla evrişim işlemi için kaydırma yapılmaktadır. Bu dolaşım sırasında filtre, sınıra ulaştığında bir alt basamağa geçerek dolaşıma devam etmektedir. Bu işlem

her renk kanalında da gerçekleştirildikten sonra, üç kanalın toplamı ile aktivasyon haritası oluşturulmaktadır.

2.4.1.3. Düzleştirilmiş Doğrusal Birim Katmanı

Düzleştirilmiş Doğrusal Birim Katmanı (ReLU), evrişim katmanlarından sonra gelmekte ve ağı doğrusal olmayan bir hale sokmak için uygulanmaktadır. Girdi verisine yapmakta olduğu etki matematiksel olarak 2.6'da tanımlanmakta, grafiği ise **Şekil 2.12**'de görülmektedir [64].

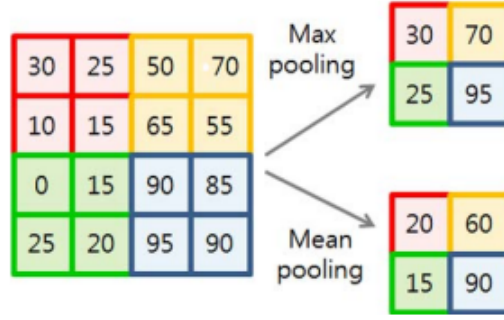
$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \quad (2.6)$$



Şekil 2.12 Negatif ve pozitif girdiler için düzleştirilmiş doğrusal birim katmanının etkisi [63]

2.4.1.4. Havuzlama Katmanı

Havuzlama katmanı, ReLU katmanından sonra gelmekte ve bir sonraki evrişim katmanı için girdi boyutunu (genişlik x yükseklik) azaltmaktadır. Böylelikle bilginin ezberlenmesinin önüne geçmekte ve bir sonraki evrişim katmanı için daha az hesaplama yükü oluşturmaktadır. Havuzlama katmanına ait maksimum havuzlama ve ortalama havuzlama olmak üzere iki adet filtre bulunmaktadır ve bunların matrise olan etkisi **Şekil 2.13**'te gösterilmektedir [53] [69].

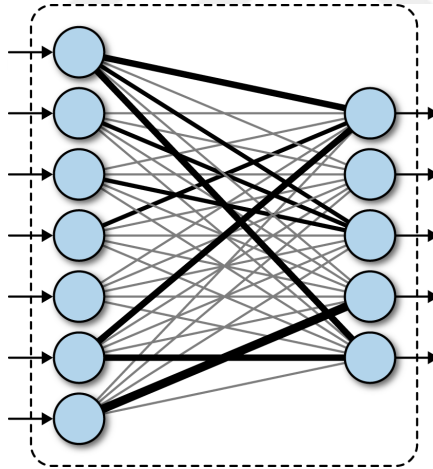


Şekil 2.13 Maksimum ve ortalama havuzlamanın girdi matrisine olan etkisi [71]

2.4.1.5. Tam Bağlantılı Katman

Tam bağlantılı katman, ESA mimarisindeki bir modelde arka arkaya gelmekte olan evrişim, ReLu ve havuzlama katmanlarından sonra gelmekte olup kendinden önceki katmanın tüm alanlarına bağlı olmaktadır. Tam bağlantılı katman öncesinde özellik çıkarılımı yapılmaktayken bu katman ile birlikte sınıflandırma işlemleri başlamaktadır [56].

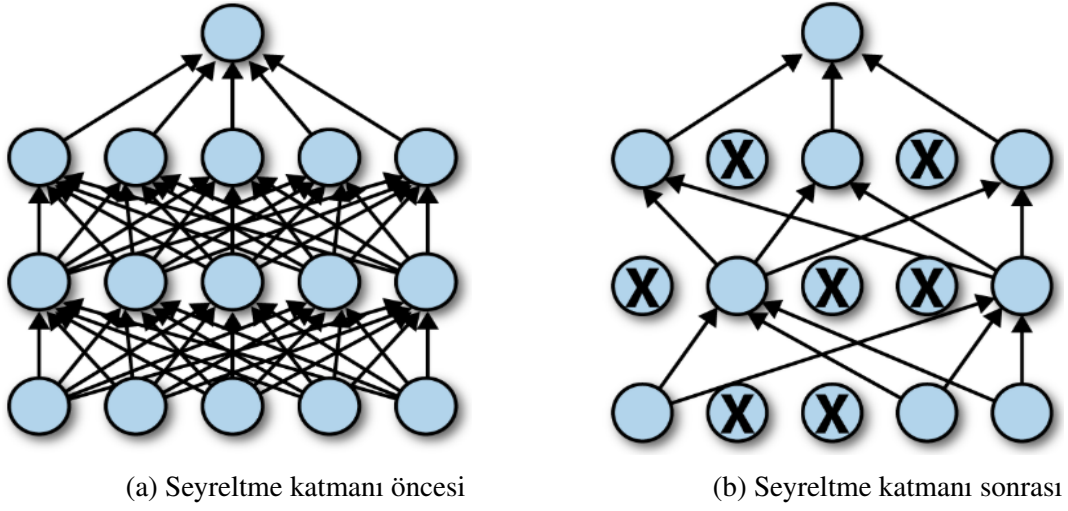
Şekil 2.14’te tam bağlantılı katman gösterilmektedir.



Şekil 2.14 Tam bağlantılı katman [67]

2.4.1.6. Seyreltme Katmanı

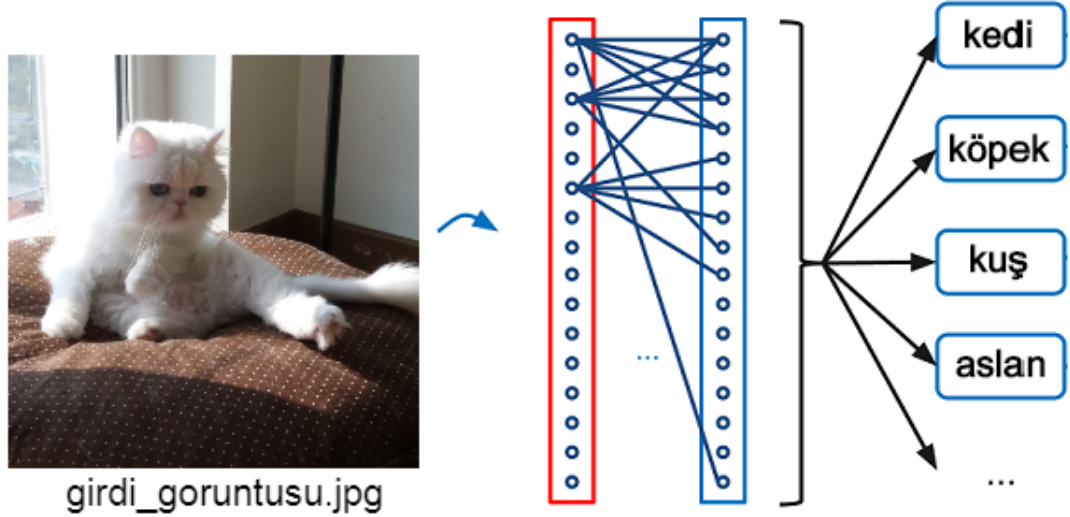
Seyreltme katmanı, ağırlık bilgileri ezberlemesini engellemek için kullanılmaktadır. Bunu yapabilmek içinde ağırdan bazı düğümleri ortadan kaldırmaktadır [74]. Şekil 2.15’te ağa seyreltme katmanı uygulanmadan öncesi ve uygulandıktan sonrası gösterilmektedir.



Şekil 2.15 ESA ağına seyreltme katmanı uygulanması [67]

2.4.1.7. Sınıflandırma Katmanı

Sınıflandırma katmanı, tam bağlantılı katmandan sonra gelmekte ve bu katmanda sınıflandırma işlemi yapılmaktadır. Bu katmandaki çıkış değeri sınıflandırılacak nesne sayısına eşit olmaktadır. Şekil 2.16'da modele verilen görüntünün sınıflandırma işlemi görülmektedir [56].



Şekil 2.16 Sınıflandırma işlemi [66]

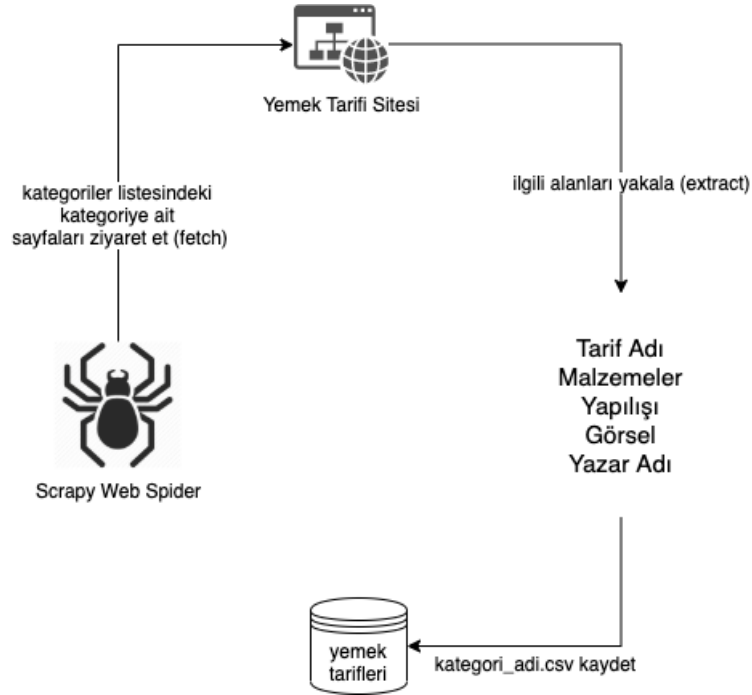
3. MALZEME VE YÖNTEM

3.1. VERİ ÜZERİNE İŞLEMLER

3.1.1. Verinin Toplaması

3.1.1.1. Yemek Tariflerinin Toplanması

Web madenciliği yapılacak olan sitenin içeriği incelendikten sonra içerisinde yemek tariflerinin çekileceği 24 adet kategori saptanmıştır. Bu kategorilere ait olan kategori sayfalarının linkleri "kategoriler" adlı bir listede tutulmaktadır. Ardından sitenin kaynak kodu incelendikten sonra içerisinde ilgili tarife ait olan verilerin çekileceği ilgili alanlar saptanmıştır. Bunlar: yazar adı, görsel resmi, tarifin linki, tarifin adı, malzeme adı, malzeme adedi, nasıl hazırlanacağı, pişirme süresi, hazırlama süresi, servis sayısı olmak üzere 10 adettir. Her bir ilgi alanına ait olan HTML etiketinin adı "veri_cekilecek_alanlar" adlı bir listede tutulmaktadır. Uygulanan algoritma **Tablo 3.1**'de, diyagramı ise **Şekil 3.1**'de görülmektedir.



Şekil 3.1 Yemek tariflerinin toplanmasını gösteren diyagram

Girdi:	kategoriler, veri_cekilecek_alanlar
Çıktı:	kategori_(kategori).csv
Başlangıç	
Adım 1	for kategori in kategoriler:
Adım 2	for alan in veri_cekilecek_alanlar:
Adım 3	kayıtlar= yakala(kategori, alan)
Adım 4	return kayıtlar
Bitir	

Tablo 3.1 Her bir kategoride bulunan tarifi içeriğini ilgilenilen veri alanlarına göre çeken algoritma

3.1.1.2. Malzemelere ait olan Görsellerin Toplanması

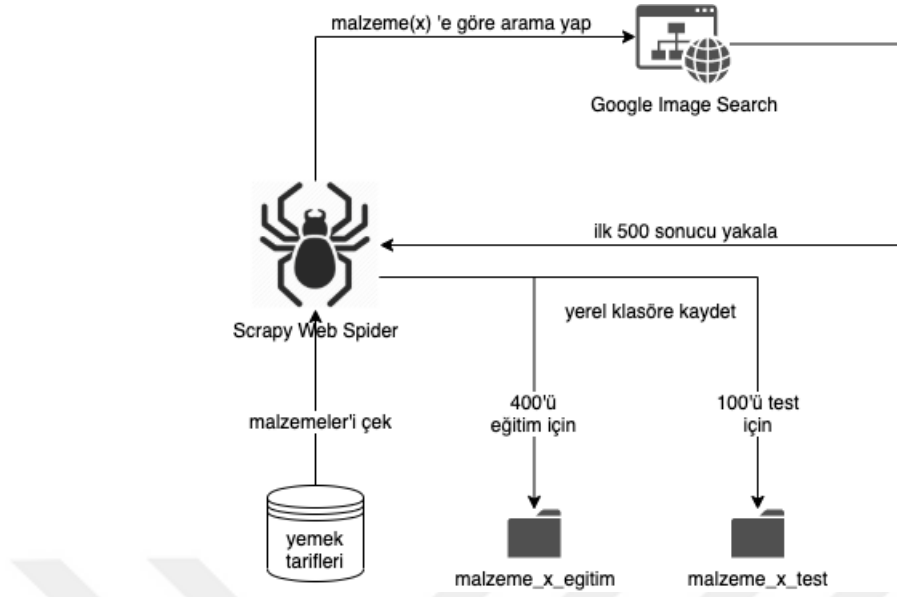
Her bir emsalsiz malzeme, "Verinin İşlenmesi" bölümünde anlatılan yöntemle ayrıştırıldıktan sonra örümceğe parametre olarak verilmektedir. Örümcek ise **Tablo 3.2**'deki gösterilen algoritmaya göre hareket ederek; Google Resim Arama Motoru'nda arama yapmaktadır. Ardından arama motorundan dönen ilk 500 sonuç; 400'ü eğitim, geri kalan 100'ü de test için olmak üzere iki ayrı klasör altında, kendi adıyla toplanmaktadır. Burada toplanan görseller CNN ağını eğitmek için kullanılmaktadır.

Girdi:	List emsalsiz_malzemeler
Çıktı:	emsalsiz_malzeme_x.png
Başlangıç	
Adım 1	emsalsiz_malzemeler= veritabanının malzemeler sütunundan emsalsiz malzemeleri al.
Adım 2	for malzeme in emsalsiz_malzemeler:
Adım 3	resim= orumcek(malzeme)
Adım 4	write(resim)
Bitir	

Tablo 3.2 Malzemelere ait olan görsellerin toplanmasını sağlayan algoritma

3.1.2. Verinin İşlenmesi

Verilere dair işlemler, Jupyter Notebook üzerinde Pandas ve Numpy kütüphaneleri kullanılarak yapılmaktadır. Her bir yemek kategorisi için örümcekte gelen 2 farklı csv dosyası bulunmaktadır. Bunlardan ilki kategori_adı.csv iken diğeri kategori_adı_detaylar.csv olarak adlandırılmaktadır. Sırasıyla, ilk csv dosyasında: yazar adı, yemek resminin linki, malzeme adları, hazırlanışı, hazırlanış süresi, tarif adı; ikinci csv dosyasında ise: hazırlanışı,



Şekil 3.2 Malzemelere ait olan görsellerin toplanmasını gösteren diyagram

kullanılan malzemelerin adetleri, malzemelerin isimleri, servis boyutu, hazırlama ve pişirme süresi bulunmaktadır. Verilerin temizlenmesi, istenmeyen kısımların atılması ve son olarak da 2 dataframe'in birleştirilmesini gösteren algoritma **Tablo 3.3**'te gösterilmektedir.

Girdi:	kategori_adi.csv, kategori_adi_detay.csv
Çıktı:	kategori_adi_birlestirilmis.csv
Başlangıç	
Adım 1	her iki csv dosyasını da birer dataframe olarak yükle. (df1=kategori_adi.csv, df2=kategori_adi_detay.csv)
Adım 2	bos_alanlari_kontrol_et(df1, df2) ? satırı sil : devam et.
Adım 3	df1= kaldır(tekrar_edan_sütun(df1))
Adım 4	df3= birlestir(df1, df2)
Adım 5	kaydet(df3, 'kategori_adi_birlestirilmis.csv')
Bitir	

Tablo 3.3 Bir kategoriye ait olan 2 veri setini birleştiren algoritma

3.1.2.1. İşlenen Veriden Elde Edilen Bazı İstatistikler

Örümcek vasıtasıyla 24 kategoride toplam 9457 yemek tarifi toplanmıştır. Her bir kategoriye ait yemek tarifi sayısı **Şekil 3.3**'te görülmektedir.

Keyfi olarak seçilen sebze kategorisine ait malzemelerin tariflere göre kullanım frekansı **Tablo 3.4**'te görülmektedir.

```

1. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_atistirmaliklar.csv => 721
2. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_balik.csv => 148
3. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_borek.csv => 199
4. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_corba.csv => 234
5. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_diyet.csv => 300
6. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_dunya_mutfaklari.csv => 264
7. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_et.csv => 307
8. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_hamur_isi.csv => 1283
9. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_kahvaltilik.csv => 539
10. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_kek.csv => 429
11. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_kofte.csv => 175
12. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_kurabiye.csv => 451
13. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_makarna.csv => 129
14. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_meze.csv => 205
15. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_pasta.csv => 166
16. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_pilav.csv => 115
17. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_pizza.csv => 67
18. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_pogaca.csv => 163
19. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_pratik_yemek.csv => 1455
20. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_salata.csv => 227
21. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_sebze.csv => 315
22. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_tatli.csv => 1228
23. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_tavuk.csv => 238
24. C:\Users\Yasin\Desktop\yemek_csv_ops\merged_of_csv_files\category_zeytinyagliilar.csv => 99

```

Total Recipe Count: 9457

Şekil 3.3 Her bir kategoriye ait toplam tarif sayısı

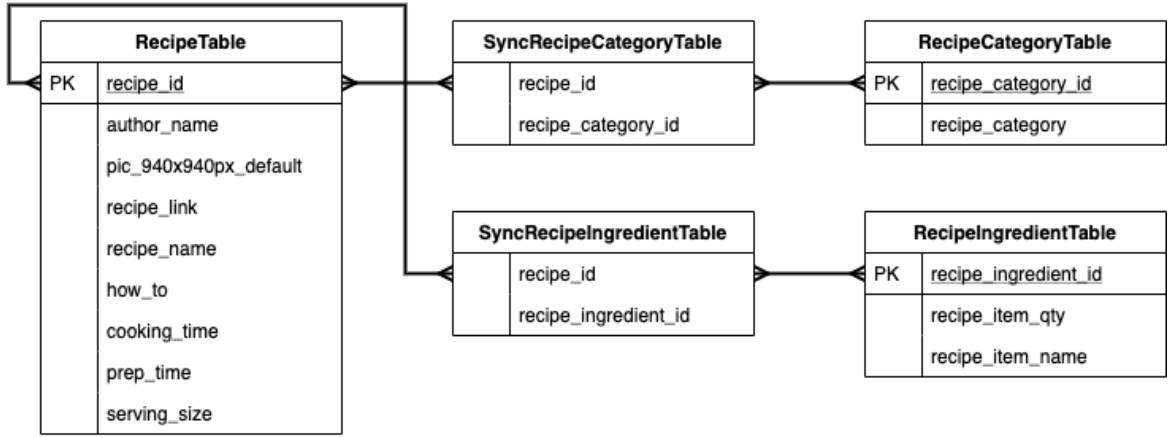
malzeme	frekans	malzeme	frekans
tuz	289	domates	46
zeytinyağı	232	maydanoz	44
karabiber	148	havuç	42
sarımsak	131	patates	37
un	67	sıvı yağ	34
soğan	65	biber salçası	33
yumurta	62	rendelenmiş kaşar peyniri	31
domates salçası	60	süt	30
sıcak su	59	orta boy kuru soğan	29
su	56	pirinç	29
tereyağı	55	pul biber	29
dereotu	50	taze soğan	28
kabak	48	patlıcan	28

Tablo 3.4 Sebze kategorisine ait malzemelerin kullanım frekansını gösteren tablo

3.1.3. Veritabanının Oluşturulması

Veri madenciliği işlemleri sonucunda oluşturulan veritabanı 3 ana ve 2 yardımcı olmak üzere 5 ayrı tablodan meydana gelmektedir. Ana tablolardan olan RecipeTable bünyesinde; tarif id'sini, yazar adı, tarif linki, tarif adı, nasıl yapıldığı, pişme süresi, hazırlık süresi ve servis miktarınının kayıtlarını tutmaktadır. RecipeCategoryTable ise;

tariflerin kategori id'sini ve kategori adını tutmaktadır. RecipeIngredientTable ise; tarif malzemesinin id'sini, malzeme adedini ve malzeme ismini tutmaktadır. Ara tablolardan olan SyncRecipeCategoryTable; RecipeTable ve RecipeCategoryTable arasında çoktan çoğa (many-to-many), SyncRecipeIngredientTable'da ise; RecipeTable ve RecipeIngredientTable arasında gene çoktan çoğa ilişki mevcuttur ve bu ilişkilere karşılık düşen kayıtları ilgili alanlara denk gelecek şekilde tutulmaktadır. Oluşturulan veritabanına ait ER Diyagramı Şekil 3.4'te görülmektedir.



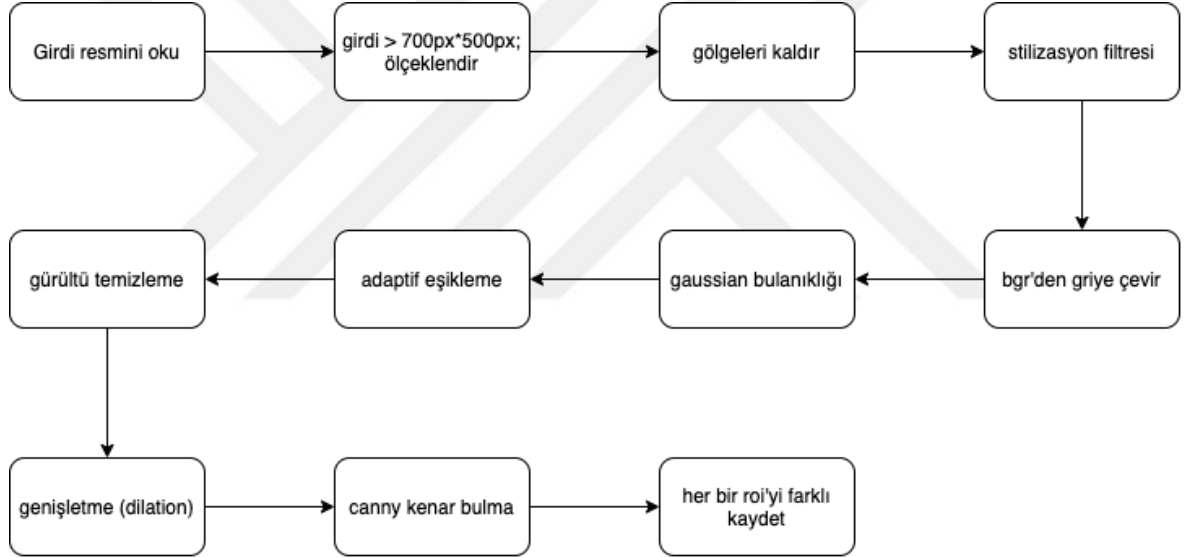
Şekil 3.4 Veritabanı ER diyagramı

3.2. NESNE TANIMA AŞAMASI

Nesne tanıma aşaması, birbirini tamamlayan iki ayrı aşamadan meydana gelmektedir. Kullanıcıdan gelen görüntü öncelikle görüntü ön işleme aşamasına gelir ve burada çeşitli filtreleme ve kenar saptama algoritmalarından geçirilir. Kenar saptama algoritmasının saptadığı kenarlar ayrı birer görüntü olarak kaydedilir. Kaydedilen bu görüntüler sırasıyla CNN ağına gönderilir.

3.2.1. Görüntü Ön İşleme

Bu bölümde, kullanıcı tarafından girdi olarak verilen görüntü üzerinde **Şekil 3.5**'te gösterilen adımlar uygulanmaktadır.



Şekil 3.5 Görüntü ön işlemede uygulanan algoritmanın akış diyagramı

İşbu ön işleme algoritmasını en kötü senaryoda dahi efektif bir şekilde çalışmasını sağlamak adına düzenlenen kurulum **Şekil 3.6**'da gösterilmiştir. Bu kurulumda: birbirine temas eden nesneler olmasına, ilgi bölgesi (ROI) dışında kalan obje/ler olmasına ve görece kötü ışıktandırmadan sebep; gölgeler bulundurmasına özellikle dikkat edilmiştir.

3.2.1.1. Giriş Görüntüsünü BGR Olarak Okuma

Kullanıcı tarafından girdi olarak gönderilen görüntü varsayılan olarak RGB renk uzayında olmaktadır. OpenCV kütüphanesinde bulunan içeriye görüntü aktarma algoritması bunu BGR uzayında **Tablo 3.5**'te gösterildiği şekilde içeriye aktarmaktadır [62]. Bu işlem sonucu da **Şekil 3.7**'de gösterilmektedir.



Şekil 3.6 Sisteme girdi olarak verilen görüntü

Girdi:	RGB Resim (img_rgb)
Çıktı:	BGR Resim (img_bgr)
Başlangıç	
Adım 1	img_bgr= cvtColor(img_rgb, cv2.COLOR_RGB2BGR)
Adım 2	return img_bgr
Bitir	

Tablo 3.5 Giriş resmini BGR uzayına çeviren algoritma



(a) RGB Giriş Resmi



(b) BGR Dönüşümü

Şekil 3.7 Giriş resmini BGR'ye dönüştürülmüş hali

3.2.1.2. Resmi Yeniden Boyutlandırma

Kullanılan teknolojik aletin kamera özelliklerine bağlı olarak giriş resminin boyutları çeşitli çözünürlüklerde olabilmektedir. Yüksek çözünürlüklü görüntüleri işleyebilmekte uzun zaman almaktadır. Bundan dolayı algoritmadaki işlem adımlarını daha hızlı gerçekleştirilebilmesi adına giriş görüntüsü, belirlenen optimal çözünürlükten (700x500 px) daha büyük ise yeniden ölçeklendirilecektir [70] [65]. Şayet değilse, yeniden boyutlandırma adımı pas geçilip diğer ön işleme adımlarına geçilecektir. Uygulanan algoritma **Tablo 3.6**'da gösterilmektedir.

Girdi:	BGR Resim (img_bgr)
Çıktı:	Ölçeklendirilmiş Resim (img_rsz)
Başlangıç	
Adım 1	if img_bgr.height > optimal.height & img_bgr.width > optimal.width
Adım 2	img_rsz= resize(img_bgr)
Adım 3	return img_rsz
Adım 4	else return img_bgr
Bitir	

Tablo 3.6 Giriş görüntüsünü optimal değerde ölçeklendirme

3.2.1.3. Gölge Kaldırma

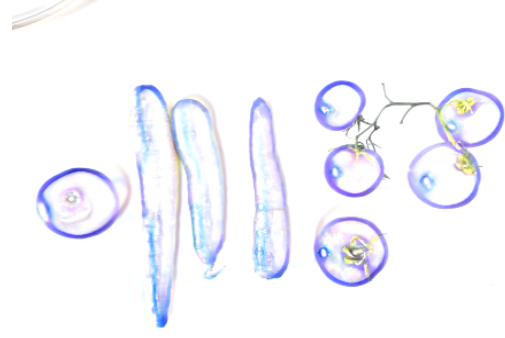
Arkaplandaki gölgeleri vurgulayıp daha rahat yakalayabilmek için görüntü önce morfolojik olarak genişletilmekte ardından medyan ile bulanıklaştırılmaktadır. Bu işlemler sonucu elde edilen görüntü ile orjinal görüntü arasındaki fark alınıp yeni bir değişkene atanmakta ardından da bu görüntüyü tutan değişken normalize edilmektedir.

Girdi:	Ölçeklendirilmiş Resim (img_rsz)
Çıktı:	Gölgesi Kaldırılan Resim (img_shadow_removed)
Başlangıç	
Adım 1	dilated_img= dilate(img_rsz)
Adım 2	bg_img= medianBlur(dilated_img)
Adım 3	diff_img= 255 - absdiff(img_rsz, bg_img)
Adım 4	img_shadow_removed= cv2.normalize(diff_img)
Bitir	

Tablo 3.7 Gölge kaldırma algoritması



(a) BGR Gölgesi



(b) BGR Gölgesiz

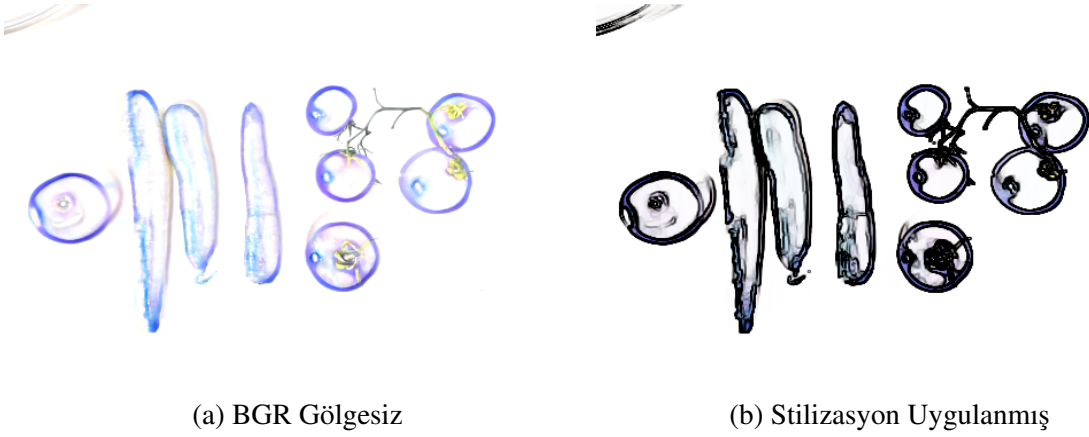
Şekil 3.8 Görüntüden gölgelerin kaldırılması

3.2.1.4. Üsluplaştırma Filtresi

Üsluplaştırma filtresi, her nesnenin kenarını, dış çizgisini veya konturunu pürüzsüz ve aynı zamanda keskin hale getirmektedir [42]. Bu filtrenin uygulanması, görüntüdeki kenarların yanı sıra nesne anahatlarını da çok görünür ve keskin hale getirir. Böylece, her nesnenin anahattını veya konturunu algılamak ve ayırmak için kullanılan kenar saptama algoritmasına yardımcı olur [43]. Uygulanan algoritma **Tablo 3.8**'de, işlem sonucuna ait çıktı ise **Şekil 3.9**'da gösterilmektedir.

Girdi:	Gölgesi Kaldırılan Resim (img_shadow_removed)
Çıktı:	Stilizasyon Filtresi Uygulanmış Resim (img_stylization)
Başlangıç	
Adım 1	img_stylization= stylization(img_shadow_removed)
Adım 2	return img_stylization
Bitir	

Tablo 3.8 Stilizasyon filtresinin algoritması



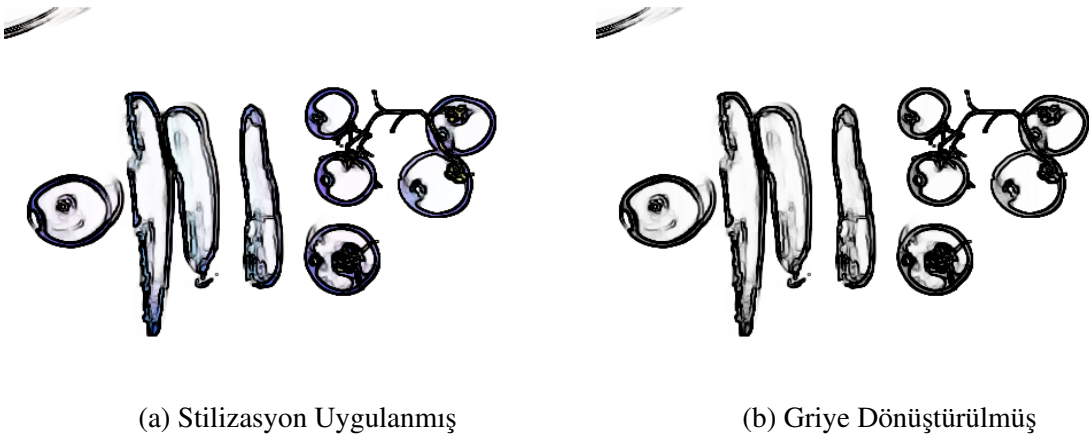
Şekil 3.9 Görüntüye stilizasyon filtresinin uygulanması sonucu oluşan sonuç

3.2.1.5. BGR'dan Gri Renk Uzayına Dönüşüm

BGR uzayından, gri renk uzayına geçerek hesaplama gereksinimleri azalmasından dolayı hesaplamalar daha hızlı bir şekilde gerçekleştirilir. Kullanılan algoritma **Tablo 3.9**'da ve işlem sonucu da **Şekil 3.10**'da görülmektedir.

Girdi:	Stilizasyon Filtresi Uygulanmış Resim (img_stylization)
Çıktı:	Gri Renk Uzayına Dönüştürülen Resim (img_gray)
Başlangıç	
Adım 1	img_gray= cvtColor(img_stylization, cv2.COLOR_BGR2GRAY)
Adım 2	return img_gray
Bitir	

Tablo 3.9 Gri renk uzayına dönüşüm algoritması



Şekil 3.10 BGR renk uzayından griye dönüşüm

3.2.1.6. Gaussian Bulanıklığı

Görüntüyü bulanıklaştırarak, gürültüyü azaltmak için kullanılmaktadır. Kullanılan algoritma **Tablo 3.10**'da ve bu algoritmanın sonucu oluşan görüntünün sonucu da **Şekil 3.11**'de görülmektedir.

Girdi:	Gri Renk Uzayına Dönü stürülen Resim (img_gray)
Çıktı:	Gaussian Bulanıklığı Uygulanmış Resim (img_gaussian_blur)
Başlangıç	Adım 1 img_gaussian_blur= GaussianBlur(img_gray, cv2.BORDER_REPLICATE)
	Adım 2 return img_gaussian_blur
Bitir	

Tablo 3.10 Gri uzaydaki görüntüye Gaussian Bulanıklığı uygulayan algoritma



(a) Griye Dönüştürülmüş

(b) Gaussian Bulanıklığı Uygulanmış

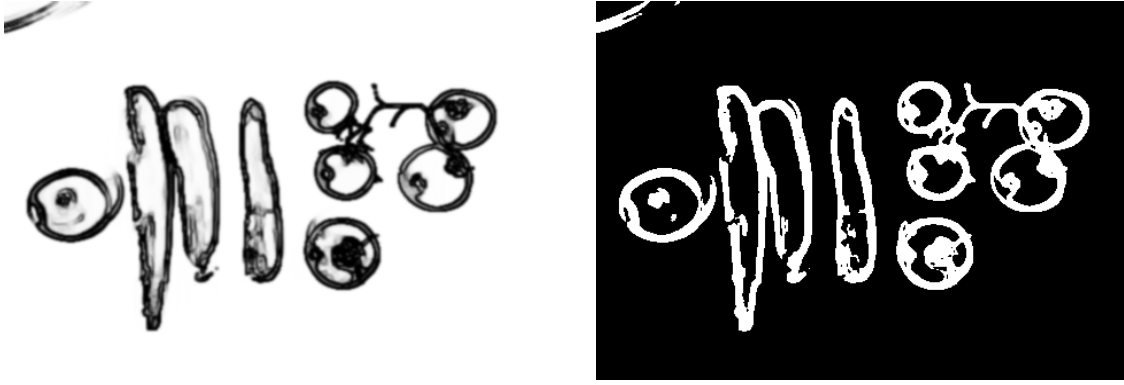
Şekil 3.11 Gaussian Bulanıklığının uygulanması sonucu oluşan yeni resim

3.2.1.7. Adaptif Eşikleme

Arkaplan ve önplandaki nesneleri birbirinden ayırmak için kullanılmaktadır. Kullanılan algoritma **Tablo 3.11**'de ve bu algoritmanın sonucu oluşan görüntü de **Şekil 3.12**'de görülmektedir.

Girdi:	Gaussian Bulanıklığı Uygulanmış Resim (img_gaussian_blur)
Çıktı:	Adaptif Eşikleme Uygulanmış Resim (img_adaptive_threshold)
Başlangıç	Adım 1 img_adaptive_threshold= adaptiveThreshold(img_gaussian_blur)
	Adım 2 return img_adaptive_threshold
Bitir	

Tablo 3.11 Adaptif eşiklemeyi uygulayan algoritma



(a) Gaussian Bulanıklığı Uygulanmış

(b) Adaptif Eşikleme Uygulanmış

Şekil 3.12 Adaptif eşiklemenin uygulanması

3.2.1.8. Gürültü Temizleme

Adaptif eşikleme ile ön plandakilerin ortaya çıktığı görüntüden gelen tüm konturlar saptanır. Ardından her biri bir dikdörtgen içerisine alınır. Bu dikdörtgenlerin alanları hesaplanır ve sonra ortalaması alınır. Eğer dikdörtgenler bu ortalamaadaki eşik değerin altındaysa piksel değerleri 0 ile doldurulur. Böylelikle istenmeyen gürültüler görüntüden kaldırılmış olur. Uygulanan algoritma **Tablo 3.12**'de ve işlem sonucu da **Şekil 3.13**'te görülmektedir.

Girdi:	Adaptif Eşikleme Uygulanmış Resim (img_adaptive_threshold)
Çıktı:	İstenmeyen Konturları Kaldırılmış Resim (img_outside_noise_removal)
Başlangıç	
Adım 1	tüm konturları bul
Adım 2	bulunan her bir konturu bir dikdörtgen içerisine al
Adım 3	bu dikdörtgenlerin alanını hesapla
Adım 4	alanları hesaplanan bu dikdörtgenlerin ortalamasını al
Adım 5	for dikdortgen_alani in dikdortgenler
Adım 6	if dikdortgen_alani < ortalama
Adım 7	dikdörtgen alanının içerisini 0 ile doldur
Adım 8	else istenenler.add(dikdortgen)
Adım 9	return istenenler
Bitir	

Tablo 3.12 Görüntüdeki istenmeyen konturları kaldıran algoritma



(a) Adaptif Eşikleme Uygulanmış



(b) Gürültü Temizleme Uygulanmış

Şekil 3.13 Gürültü temizlemenin uygulanması

3.2.1.9. Morfolojik İşlemler - Genişletme

Gürültü temizleme algoritmasının döndürdüğü görüntüyü girdi olarak almakta ve konturların kenarlarını genişletmektedir. Bu genişletme algoritması (Tablo 3.13) sayesinde piksel grupları büyümekte ve pikseller arası boşluklar da küçülmektedir. İşlem sonucu Şekil 3.14'te görülmektedir.

Girdi:	İstenmeyen Konturları Kaldırılmış Resim (img_outside_noise_removal)
Çıktı:	Konturları Genişletilmiş Resim (img_dilation)
Başlangıç	
Adım 1	img_dilation= dilate(img_outside_noise_filter)
Adım 2	return img_dilation
Bitir	

Tablo 3.13 Görüntüdeki konturları genişleten algoritma



(a) Gürültü Temizleme Uygulanmış



(b) Genişletme (Dilation) Uygulanmış

Şekil 3.14 Genişletme (dilation) uygulanması

3.2.1.10. Kenar Bulma

Konturları morfolojik olarak genişletilmiş resmi, **Tablo 3.14**'te görülmekte olan Canny kenar bulma algoritmasına girdi olarak almakta ve işlem sonucu da **Şekil 3.15**'te görülmektedir.

Girdi:	Konturları Genişletilmiş Resim (img_dilation)
Çıktı:	Kenarları Tespit Edilmiş Resim (img_edges)
Başlangıç	Adım 1 img_edges= cv2.Canny(img_dilation, lower_thresh, higher_thresh)
	Adım 2 return img_edges
Bitir	

Tablo 3.14 Görüntüdeki kenarları bulan algoritma



(a) Genişletme (Dilation) Uygulanmış



(b) Kenar Bulma (Canny) Uygulanmış

Şekil 3.15 Kenar bulma (Canny) uygulanması

3.2.1.11. Yeni Görüntü Olarak Kaydedilen Nesneler

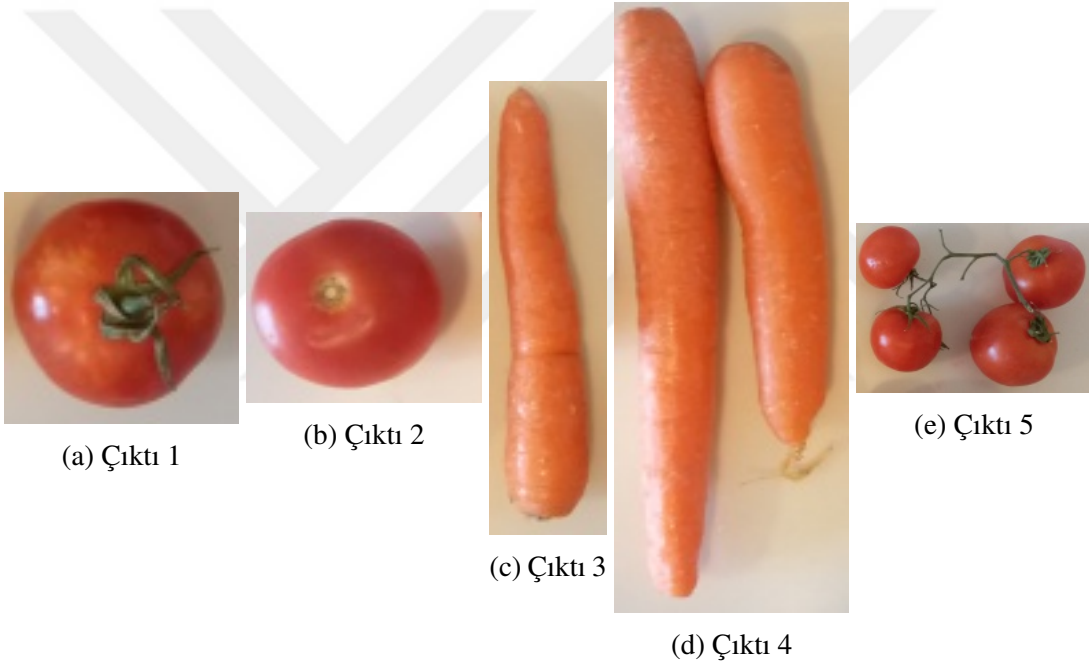
Kenar bulma algoritmasında işlendikten sonra dönen ROI'leri bulunan görüntüler **Şekil 3.16**'da görülmektedir. Bu görüntüdeki ROI'ler ise **Tablo 3.15**'teki algoritma ile ayrı birer görüntü olarak kaydedilmiştir. CNN'e girdi olarak verilmeye hazır hale gelen ayrı görüntüler de **Şekil 3.17**'de görülmektedir.

Girdi:	Kenarları Tespit Edilmiş Resim (img_edges)
Çıktı:	Yeni Oluşturulan Resimler
Başlangıç	Adım 1 for (i, kenar) in enumerate(kenarlar): Adım 2 (x, y, w, h)= cv2.boundingRect(kenar) Adım 3 cropped_image= img_resized_rgb[y:y+h, x:x+w] Adım 4 image_name= path_output + "output_" + str(i+1) + ".jpg" Adım 5 cv2.imwrite(image_name, cropped_image)
Bitir	

Tablo 3.15 Görüntüdeki ROI'leri ayrı birer görüntü olarak kaydeden algoritma



Şekil 3.16 Yeni görüntü olarak kaydedilecek nesnelerin saptanan kapalı geometrileri



Şekil 3.17 Bulunan kenarların çıktıları

3.2.2. Evrişimsel Sinir Ağları

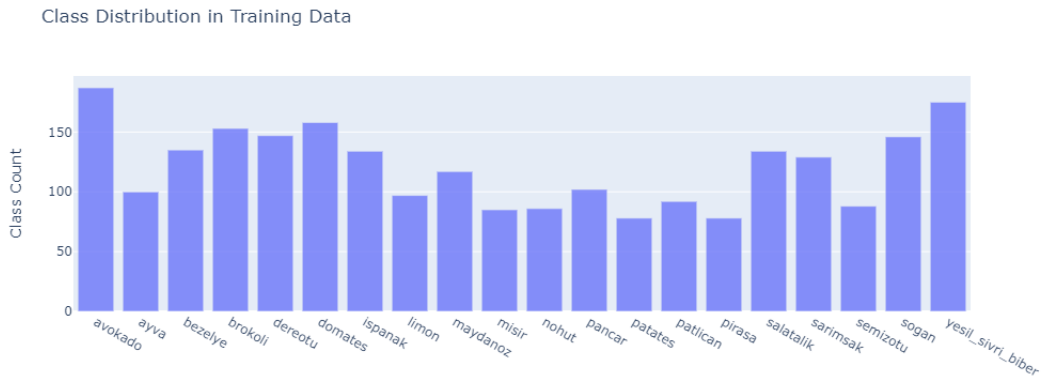
3.2.2.1. Donanım ve Yazılım Özellikleri

Bu tez çalışmasında modelin eğitimi için Kaggle’ın [59] çevrimiçi uzak sunucusundaki sanal makineden faydalanılmıştır. Sanal makinenin özellikleri: Tesla P100 16GB VRAM ekran kartı, Intel Xeon işlemci (2 Çekirdek atanmış), 5GB dahili depolama alanı ve 13GB RAM şeklindedir.

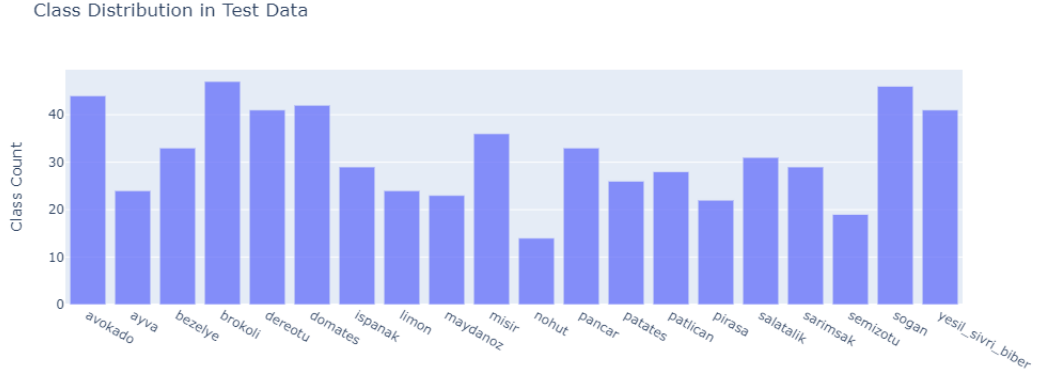
Yazılım aşamasında ise kullanılan kütüphaneler: Python 3.7.6 yazılım dili, Tensorflow 2.1.0, Keras 2.3.1, cv2, numpy, pandas, matplotlib, seaborn, graphviz ve pydot’tur.

3.2.2.2. Veri Kümesi

YTSÖ’nün topladığı 26 farklı yemek kategorisinden bu çalışma için sebze kategorisi seçilmiştir. Sebze kategorisi incelendiğinde 544 farklı emsalsiz malzeme bulunmuştur. Bu malzemelere ait ön ekler (orta boy, rendelenmiş vb.) temizlendikten sonra geriye 180 malzeme kalmıştır. Bu malzemeler arasından da sınırlar, baharatlar ve sebze cinsleri (dolmalık biber, çarliston biber vb.) kaldırılmıştır. Kalan kümeden de en sık kullanılan 20 malzeme seçilerek, görselleri toplaması için GGÖ’ye gönderilmiştir. GGÖ tarafından 20 sınıf altında 3053 görüntü toplanmıştır. Her bir sınıfın kendi klasörünün içerisindeki görüntüler, %80’i eğitim ve %20’si test olmak üzere iki ayrı klasörde toplanmıştır. Bu işlem sonucunda sırasıyla her sınıfa ait eğitim ve test veri kümesinde oluşan görüntü dağılımı **Şekil 3.18**’de ve **Şekil 3.19**’da görülmektedir.



Şekil 3.18 Eğitim kümesindeki sınıflara ait olan görsellerin dağılımı



Şekil 3.19 Test kümesindeki sınıflara ait olan görsellerin dağılımı

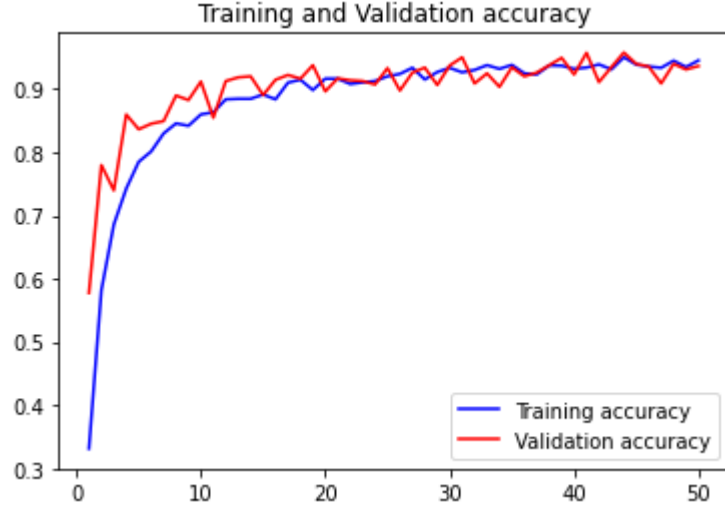
3.2.2.3. Modelin Oluşturulması

Bu çalışmada, veri kümesinin görece az sayıda olmasından ve sıfırdan eğitilecek bir modele göre daha önceden eğitilmiş bir ağın vereceği tahmin sonuçlarının daha yüksek olmasından dolayı transfer öğrenimi tercih edilmiştir. Transfer öğreniminde kullanılacak olan mimariye karar verilirken Keras Applications [60] sayfasındaki ImageNet [9] veri kümesiyle eğitilen modellerin tahmin çıktı sonucu olarak verdiği top-1 ve top-5 skorları dikkate alınmıştır. Bu skorlara göre NASNetLarge modeli ilk sırada gelirken, Xception modeli ise ikinci sırada yer almaktadır. Eğitim, ilk olarak NASNetLarge modeli üzerinde yapılmıştır fakat belli bir sayıdaki iterasyondan sonra Kaggle'daki [59] kernel'in RAM'inin yetersiz gelmesinden dolayı eğitim tamamlanamamıştır. Bu yüzden de görece daha düşük boyuta ve katman sayısına sahip olan Xception modeli seçilmiştir. Xception tabanlı ihtiyaca uygun olarak özelleştirilen modelde ilk olarak Xception'a ait olan sınıflandırıcı katman çıkarılmıştır. Ardından giriş katmanını düzeltmek için bir düzleştirici katman (flatten layer), birbirleriyle iç içe olacak şekilde 3 adet tam bağlantılı katman ve 3 adet seyreltici katman eklenmiştir. Son olarak da sınıf sayısı kadar düğüme sahip olan (20 sınıf) bir tam bağlantılı katman eklenmiştir. Oluşturulan modelin görseli, ekler sayfasında bulunmaktadır.

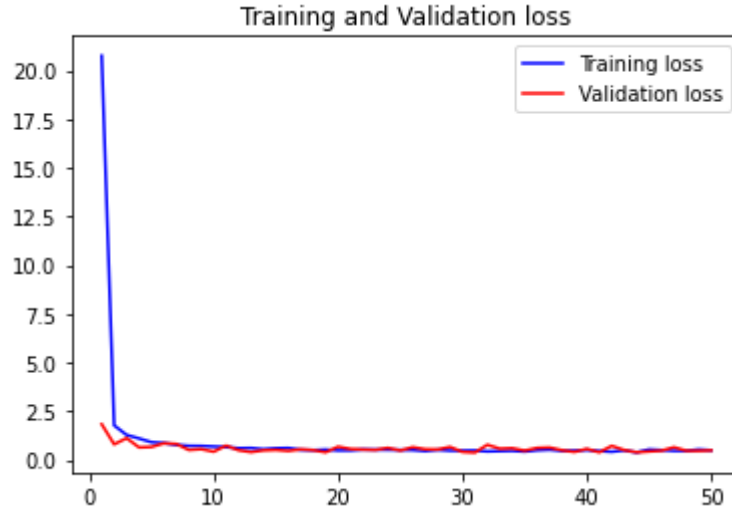
3.2.2.4. Sonuçlar

Eğitimi yapılan modelin model uyarlama (fine-tuning) değerleri sırasıyla şunlardır: küme büyüklüğü (batch size); 32, ağa girdi olarak verilen görüntü çözünürlükleri 299x299px, iterasyon sayısı (epoch); 50 ve öğrenme oranı (learning rate) ise 0.001'dir. Ağ eğitildikten sonra ortaya çıkan sonuçlar sırasıyla: doğruluk (accuracy); 0.9452, kayıp (loss); 0.4930,

geçerleme kaybı (validation loss); 0.4883, geçerleme doğruluğu (validation accuracy); 0.9362'dir. Bu sonuçlara ait grafikler sırasıyla **Şekil 3.20** ve **Şekil 3.21**'de gösterilmiştir.



Şekil 3.20 Eğitim ve geçerleme doğruluğu



Şekil 3.21 Eğitim ve geçerleme kaybı

4. BULGULAR

4.1. YEMEK TARİFİ ÖNERİ AŞAMASI

Cep telefonu kamerasıyla fotoğrafı çekilip sisteme girdi olarak verilen 3963x2883px çözünürlüğe sahip görüntü **Şekil 4.1a**'da görülmektedir. Girdi, nesne tanıma aşamasında anlatılan görüntü ön işleme adımlarından geçirilmiş ve bu işlemler sonucunda 11 adet birbirinden ayrı kontur saptanmıştır. Bu konturlar **Şekil 4.1b**'de gösterilmektedir.



(a) Kullanıcıdan gelen girdi görüntüsü



(b) Girdi görüntüsündeki konturlar

Şekil 4.1 Girdi görüntüsündeki konturların saptanması

Ardından her bir kontur ayrı birer görüntü olarak kaydedilmiştir. Bu kaydedilen 11 görüntü sırasıyla derin ağ modeline verilmiştir. Her bir girdi görüntüsü için; ağdan dönen ilk-5 tahmin sonucu azalan sırada olacak şekilde ait olduğu görüntünün altına yazılmıştır. Yeşil renk ile vurgulanan başlıklar; doğru tahmin, kırmızı renk ile vurgulanan başlıklar ise yanlış tahmin yapıldığını göstermektedir.

Yukarıda anlatılan bu aşamalardan sonra sistemin gerçek hayattaki uygulamasında ise kullanıcıya, her bir girdi görüntüsüne karşılık düşen tahmin sonucunun doğru olup olmadığı sorulacaktır. Eğer o görüntüye ait tahmin doğru ise o tahmine ait olan etiket sorguya gönderilmek üzere "sorgu listesi"ne alınacak, değilse kullanıcıdan o görüntüye ait olan etiketin doğrusunun girilmesi istenecek ve kullanıcı tarafından girilen bu etiket değeri sorgu listesine eklenecektir. Ardından bu sorgu listesindeki etiket değerleri, veritabanındaki tanımlı sorgu listesinde değer döndürmek üzere öntanımlı sorguda kullanılacaktır.

Çıktı 1

**semizotu**

ıspanak
bezelye
maydanoz
soğan

Çıktı 2

**dereotu**

maydanoz
semizotu
pırasa
yeşil sivri biber

Çıktı 3

**bezelye**

ıspanak
yeşil sivri biber
salatalık
semizotu

Çıktı 4

**patates**

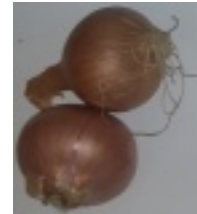
soğan
pancar
salatalık
avokado

Çıktı 5

**domates**

avokado
pancar
limon
salatalık

Çıktı 6

**soğan**

pancar
sarımsak
patates
domates

Çıktı 7



yeşil sivri biber

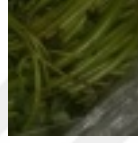
bezelye
salatalık
avokado
patlıcan

Çıktı 8



ağa verilmedi

Çıktı 9



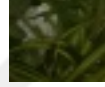
ağa verilmedi

Çıktı 10



ağa verilmedi

Çıktı 11



ağa verilmedi

Tablo 4.1 Görüntü ön işleme sonucu oluşan çıktılar ve bu çıktıların ağa verilmesi sonucu elde edilen tahminler

Kullanıcıdan gelen **Şekil 4.1a**'daki girdi görüntüsü sonucu elde edilen yeni görüntüler ve kullanıcının yapılan yanlış tahminleri düzeltmesi sonucu oluşan sorgu listesi sırasıyla; "maydanoz, dereotu, semizotu, patates, domates, soğan, yeşil sivri biber" değerlerinden meydana gelmektedir. Bu değerler **Şekil 4.2**'de gösterilen öntanımlı sorguya verilmiştir.

```

1 SELECT RecipeTable.recipe_id, recipe_name, how_to, cooking_time, prep_time, serving_size
2 FROM
3     RecipeTable,
4     RecipeIngredientTable,
5     SyncRecipeIngredientTable
6 WHERE
7     RecipeTable.recipe_id = SyncRecipeIngredientTable.recipe_id
8     AND RecipeIngredientTable.recipe_ingredient_id = SyncRecipeIngredientTable.recipe_ingredient_id
9     AND RecipeIngredientTable.recipe_item_name in ("maydanoz", "dereotu", "semizotu", "patates", "domates", "soğan", "yeşil sivri biber")
10 GROUP by RecipeTable.recipe_id HAVING COUNT(*) = 4;
```

Şekil 4.2 Veritabanında yapılan sorgu işlemi

Veritabanında yapılan bu sorgu sonucunda, sorgu listesindeki en az 4 malzemenin de bir arada bulunduğu 20 yemek tarifi sonuç olarak döndürülmüştür. Bu dönen sonuçlara ait olan ilk 10 tarif **Şekil 4.3**'te görülmektedir.

	recipe_id	recipe_name	how_to	cooking_time	prep_time	serving_size
1	213	Üç Farklı Hatay Mezesi	Bir kasede kırık yeşil zeytin, doğranmış taze soğā...	25 dakika	30 dakika	4 kişilik
2	2007	Kuşbaşı Etli Karniyarik	Pijamalı soyduğunuz patıcanları limonlu ve tuzlu ...	40 dakika	40 dakika	10 adet
3	3371	Ev Usulü Tavadā Gözleme	Patatesli harç için; haşlanmış ve rendelenmiş ola...	20 dakika	50 dakika	8 kişilik
4	4466	Fırında Sebzeli Köfte	Soğan ve sarımsağı rondodan geçirin. Suyunu iyi...	30 dakika	15 dakika	4 kişilik
5	5185	Üç Farklı Hatay Mezesi	Bir kasede kırık yeşil zeytin, doğranmış taze soğā...	25 dakika	30 dakika	4 kişilik
6	5228	Üç Çeşit Meze	Bir kapta doğradığınız domates, biber, soğan ve ...	10 dakika	1 saat	4 kişilik
7	6088	Patates Mantısı	Haşlanmış patatesleri bir kasede püre haline geti...	30 dakika	20 dakika	5 kişilik
8	6104	Fırında Sebzeli Köfte	Soğan ve sarımsağı rondodan geçirin. Suyunu iyi...	30 dakika	15 dakika	4 kişilik
9	6621	Patates Dizme	Fırını 180 dereceye ayarlayın.,Soğanı, havucu ve ...	1 saat	20 dakika	8 kişilik
10	6638	Fırında Kabak Dolması	Kabakları yıkayın ve bir çatal yardımıyla kabukları...	1 saat	25 dakika	4 kişilik

Şekil 4.3 Veritabanında yapılan sorgu işleminden dönen sonuçlar

5. TARTIŞMA VE SONUÇ

5.1. SONUÇLAR

Bu tez çalışmasında, kullanıcıdan alınan girdi görüntüsü; öncelikle görüntü ön işleme adımlarındaki işlemlerden geçirilmektedir. Bu işlemler sonucunda çıktı olarak birden fazla kontur saptanmakta ve bu konturlar ayrı birer görüntü olarak kaydedilmektedir. Bu kaydedilen görüntüler; evrişimsel derin ağ modeline girdi olarak verilmektedir ve tanımlama işlemi yapılmaktadır. Her bir görüntüye ait olan tanımlama etiketi bir listeye kaydedilmektedir. Tanımlama işlemi tamamlandığında bu listedeki değerler veritabanına sorgu olarak gönderilmektedir. Sorgu sonucu olarak da o listedeki malzemelerle hangi yemek tariflerinin yapılabileceğine dair bir yemek tarifi öneri sonucu dönmektedir.

Yemek tariflerini ve bu yemek tariflerine ait olan malzemelerin görüntülerini toplamak için 2 farklı örümcek yazılmıştır. Bu örümceklerden ilki olan YTSÖ, önceden belirlenen yemek sitelerindeki içerikleri (yemek adı, malzeme listesi, yapılışı vb.) toplayıp bir ham veri yığını oluşturmuştur. Oluşturulan bu ham veri yığını, çeşitli veri madenciliği yöntemleriyle temizlenmiş, indirgenmiş ve bütünleştirilmiştir. Ardından işlenmiş ve arzu edilen hale getirilmiş bu veriyle bir "yemek tarifleri veritabanı" oluşturulmuştur. Oluşturulan bu veritabanındaki yemek tariflerine ait olan malzemelerin görsellerini toplamak amacıyla da ikinci örümcek olan GGÖ yazılmıştır. GGÖ, her biri emsalsiz olan malzeme adlarını sırasıyla; Google Görseller üzerinde aramaktadır. Arama sonucu çıkan sonuçları ise araması

yapılan malzemenin kendi adıyla "Malzeme Görüntüleri" adındaki klasörün altında olacak şekilde kaydetmektedir. "Malzeme Görüntüleri" adındaki klasörde toplanan bu görüntülerin içerisinde alakasız olan görüntüler temizlenmiştir. Bu temizleme işleminden sonra bu görüntülerin %80'i eğitim, %20'si test için olmak üzere ikiye ayrılmış ve bir veri kümesi oluşturulmuştur.

Literatürdeki Çalışmalar	Kullanılan Mimari	Başarım Oranları
Chen ve diğerleri [1]	AlexNet	%64,91
Güngör ve diğerleri [5]	Inception V3	%62,7
Ergün ve diğerleri [10]	Inception V3	%78

Tablo 5.1 Literatürdeki CNN kullanan çalışmalara ait başarımlar

Veri kümesinin görece az görüntü barındırmasından ve önceden eğitilmiş bir ağı ince-ayar (fine-tuning) yapılarak özelleştirilmesinin, tahmin oranı başarımına etkisi daha yüksek olacağından transfer öğrenmesi yöntemi tercih edilmiştir. Derin evrimsel sinir ağı modeli oluşturulurken mimari tabanı olarak Xception tercih edilmiştir. Xception'ın son katmanı olan sınıflandırıcı katmanı ortadan kaldırılmıştır. Bunun yerine 1 düzleştirici, 3 adet tam bağlantılı ve seyreltici katman eklenmiştir. Son olarak da 20 düğümden oluşan bir softmax sınıflandırıcı katmanı eklenip, eğitime başlanılmıştır. Bu eğitim sonucunda: doğruluk (accuracy); 0.9452, kayıp (loss); 0.4930, geçerleme kaybı (validation loss); 0.4883, geçerleme doğruluğu (validation accuracy); 0.9362 çıkmıştır.

5.2. GELECEK ÇALIŞMALAR İÇİN ÖNERİLER

Bu tez kapsamında yapılan işlemlerin dışında gelecekte yapılacak olan çalışmalarda daha iyi sonuçlar elde edilebilmesi adına motorlu döner levhaya sahip bir düzenek kurulup bir "malzeme kayıt ortamı" yaratılabilir ve farklı açılarda bir sürü görüntü alınarak veri kümesi zenginleştirilebilir. Böylelikle eğitim sırasında ağ daha iyi eğitilebilir ve test görüntüleri üzerinde daha yüksek doğrulamaya sahip başarımlar yakalanabilir. Bununla birlikte Sıfır-Atış Öğrenmesi üzerine de yoğunlaşılabilir. Böylelikle ağ daha önce görmediği malzemeler üzerine de tahminlerde bulunabilir hale getirilebilir.

Uygulamalar alanında ise; kullanıcının tercihlerine (glutensiz, diyabetik vb.) özel yemek tarifleri önerilebilir. Kalori, günlük alınması gereken makro ve mikro besinlerin takibi yapılabilir.

KAYNAKLAR

- [1]. Chen J. and Ngo C., 2016, Deep-based ingredient recognition for cooking recipe retrieval, 32–41.
- [2]. Bossard L., Guillaumin M., and Gool L., 2014, Food-101 - mining discriminative components with random forests, 446–461.
- [3]. Chen M., Dhingra K., Wu W., and Sukthankar R., 2009, Pfid: Pittsburgh fast-food image dataset, volume 289-292, 289–292.
- [4]. NGO C. and Chen J., 2016, Deep-based ingredient recognition for cooking recipe retrieval, *ACM Multimedia*, City University of Hong Kong, Hong Kong.
- [5]. Güngör C., Baltacı F., Erdem A., and Erdem E., 2017, Turkish cuisine: A benchmark dataset with Turkish meals for food recognition, In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 1–4.
- [6]. Singla A., Yuan L., and Ebrahimi T., 2016, Food/non-food image classification and food categorization using pre-trained googlenet model, 3–11.
- [7]. Food-11 dataset, 2019, grebvm2.epfl.ch/lin/food/Food-11.zip, [Ziyaret Tarihi: 4 Nisan 2020].
- [8]. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., and Wojna Z., 2015, Rethinking the inception architecture for computer vision, *CoRR*, abs/1512.00567.
- [9]. Deng J., Dong W., Socher R., Li L., Li K., and Li F., 2009, Imagenet: a large-scale hierarchical image database, 248–255.
- [10]. Ergun O. and Ozturk B., 2018, An ontology based semantic representation for Turkish cuisine, 1–4.
- [11]. Maruyama T., Kawano Y., and Yanai K., 2012, Real-time mobile recipe recommendation system using food ingredient recognition, 27–34.
- [12]. Bay H., Ess A., Tuytelaars T., and Gool L., 2008, Surf: Speeded up robust features, *Computer Vision and Image Understanding*, 110, 346-359.
- [13]. Csurka G., Dance C., Fan L., Willamowski J., and Bray C., 2004, Visual categorization with bags of keypoints, *Work Stat Learn Comput Vision, ECCV*, Vol. 1.
- [14]. Hearst M., 1998, Support vector machines, *IEEE Intelligent Systems*, 13(4), 18-28.
- [15]. Beijbom O., Joshi N., Morris D., Saponas S., and Khullar S., 2015, Menu-match: Restaurant-specific food logging from images, *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 844-851.

- [16]. Dalal N. and Triggs B., 2005, Histograms of oriented gradients for human detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2.
- [17]. Lowe D., 2001, Object recognition from local scale-invariant features, *Proceedings of the IEEE International Conference on Computer Vision*, 2.
- [18]. Ojala T., Pietikainen M., and Maenpaa T., 2000, Gray scale and rotation invariant texture classification with local binary patterns, volume 1842, 404–420.
- [19]. Varma M. and Zisserman A., 2005, A statistical approach to texture classification from single images, *International Journal of Computer Vision*, 62, 61–81.
- [20]. Hashimoto A., Mori N., Funatomi T., Yamakata Y., Kakusho K., and Minoh M., 2008, Smart kitchen: A user centric cooking support system, Kyoto University, Kyoto.
- [21]. Siio I., Itiro, Hamada, Reiko, Mima N., and Noyuri, 2017, Kitchen of the future and applications, 946.
- [22]. Bradbury J., Shell J. and Knowles C., 2003, Hands on cooking: Towards an attentive kitchen, 996.
- [23]. Enformasyon piramidi, https://en.wikipedia.org/wiki/DIKW_pyramid#/media/File:DIKW_Pyramid.svg, [Ziyaret Tarihi: 4 Nisan 2020].
- [24]. Akpınar H., 2014, *Data Veri Madenciliği - Veri Analizi*, Papatya Bilim, İstanbul.
- [25]. Cooley R., Mobasher B., and Srivastava J., 1997, Web mining: Information and pattern discovery on the world wide web, *Proceedings of the International Conference on Tools with Artificial Intelligence*, 558–567.
- [26]. Berendt B., Hotho A., and Stumme G., 2002, Towards semantic web mining, *Lecture Notes in Computer Science*.
- [27]. Kavita E. A., 2011, Web mining taxonomy figure, https://www.researchgate.net/figure/Taxonomy-of-Web-Mining-Source-Kavita-et-al-2011_fig1_282357293, [Ziyaret Tarihi: 4 Nisan 2020].
- [28]. Gezer M., Erol Ç., Gülseçen S., 2007, Bir web sayfasının veri madenciliği ile analizi, Akademik Bilişim 2007, Dumlupınar Üniversitesi, Kütahya 31 Ocak-2 Şubat 2007.
- [29]. Chakrabarti S., 2000, Data mining for hypertext: A tutorial survey, *SIGKDD Explorations*, 1.
- [30]. Pirolli P., Pitkow J., and Rao R., 2000, Silk from a sow's ear: Extracting usable structures from the web, *Conference on Human Factors in Computing Systems - Proceedings*.
- [31]. Mobasher B., Jain N., Han E., and Srivastava J., 1996, Web mining: Pattern discovery from world wide web transactions.
- [32]. Scrapy, 2020, Scrapy web spider, <https://docs.scrapy.org/en/latest/intro/overview.html>, [Ziyaret Tarihi: 4 Nisan 2020].

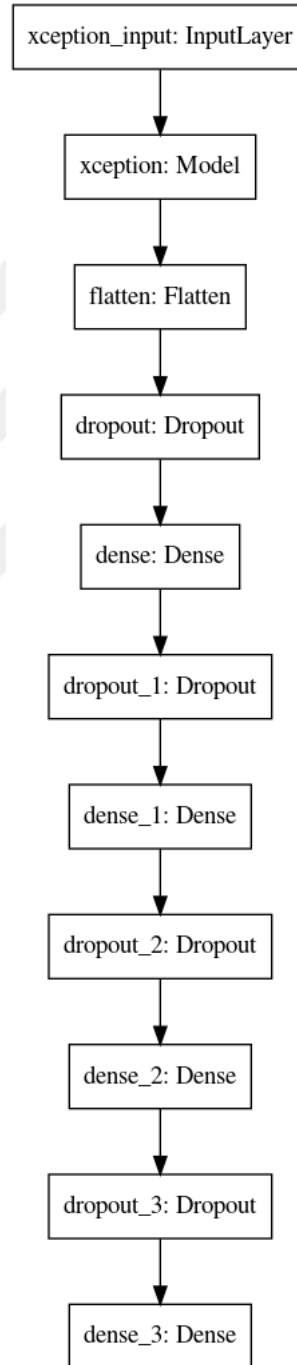
- [33]. Codd E. F., 1983, A relational model of data for large shared data banks, *Communications of the ACM*, 26(1), 64–69.
- [34]. Bhosale S., Patil M., and Patil P., 2015, International journal of computer science and mobile computing sqlite: Light database system, *International Journal of Computer Science and Mobile Computing*, 44, 882–885.
- [35]. Fayyad U., Piatetsky-Shapiro G., and Smyth P., 1996, From data mining to knowledge discovery in databases, *AI magazine*, 17(3), 37–37.
- [36]. Silahtaroglu G., 2006, *Veri Madenciliği: Kavram ve Algoritmaları*, Papatya Bilim, İstanbul.
- [37]. Kantardzic M., 2020, *Data mining: concepts, models, methods, and algorithms*, John Wiley & Sons, Inc, USA.
- [38]. Kaehler A. and Bradski G., 2017, *Learning OpenCV 3: Computer vision in C with the OpenCV library*, O'Reilly Media, Inc, USA.
- [39]. Rgb renk uzayı resmi, 2020, https://favpng.com/png_view/light-rgb-color-space-rgb-color-model-light-png/BsYUHtec, [Ziyaret Tarihi: 4 Nisan 2020].
- [40]. Mallick S., 2020, Why did they choose bgr color space in opencv? <https://www.learnopencv.com/why-does-opencv-use-bgr-color-format/>, [Ziyaret Tarihi: 4 Nisan 2020].
- [41]. Hasan S. N., 2019, Derin Öğrenme yöntemleri ile cilt lezyon bölütleme, Master's thesis, İstanbul Üniversitesi.
- [42]. OpenCV, 2020, Stylization filter, https://docs.opencv.org/master/df/dac/group__photo__render.html#gacb0f7324017df153d7b5d095aed53206, [Ziyaret Tarihi: 4 Nisan 2020].
- [43]. Khan R., Raisa T. and Debnath R., 2018, An efficient contour based fine-grained algorithm for multi category object detection, 6, 127–136.
- [44]. Fisher R., Perkins S., Walker A. and Wolfart E., 2003, Gaussian bulanıklığı. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>, [Ziyaret Tarihi: 4 Nisan 2020].
- [45]. Efford N., 2000, Adaptive Threshold. <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic3.htm#adaptive>, [Ziyaret Tarihi: 4 Nisan 2020].
- [46]. Walker A., Perkins S., Fisher R. and Wolfart E., 2003, Morfolojik İşlemler, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>, [Ziyaret Tarihi: 4 Nisan 2020].
- [47]. Agu E., 2009, Genişletme İşlemi. <https://web.cs.wpi.edu/~emmanuel/courses/cs545/S14/slides/lecture07.pdf>, [Ziyaret Tarihi: 4 Nisan 2020].
- [48]. Fisher R., Perkins S., Walker A. and Wolfart E., 2003, Kenar bulma. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/featops.htm>, [Ziyaret Tarihi: 4 Nisan 2020].

- [49]. Niebles J. C. and Krishna R., 2017, Kenar bulma dedektörleri, http://vision.stanford.edu/teaching/cs131_fall1718/files/05_edges.pdf. [Ziyaret Tarihi: 4 Nisan 2020].
- [50]. Bulut M. and Başoğlu B., 2017, Kısa dönem elektrik talep tahminleri İçin yapay sinir ağları ve uzman sistemler tabanlı hibrid tahmin sistemi geliştirilmesi, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 32.
- [51]. Lecun Y., Matan O., Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L., and Baird H., 1990, Handwritten zip code recognition with multilayer networks, volume 2.
- [52]. Tabian I., Fu H., and Khodaei Z., 2019, A convolutional neural network for impact detection and characterization of complex composite structures, *Sensors*, 19, 4933.
- [53]. İnık Ö. and Ülker E., 2017, Derin Öğrenme ve görüntü analizinde kullanılan derin Öğrenme modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6, 85–104.
- [54]. Fisher R., Perkins S., Walker A. and Wolfart E., 2003, Canny kenar bulma dedektörü. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm>. [Ziyaret Tarihi: 4 Nisan 2020].
- [55]. Narasimhan S., 2006, Canny kenar bulma dedektörü, <http://www.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/>, [Ziyaret Tarihi: 4 Nisan 2020].
- [56]. Amidi S., 2019, Convolutional neural networks cheatsheet, <https://stanford.edu/~shervine/teaching/cs230/cheatsheet-convolutional-neural-networks>, [Ziyaret Tarihi: 4 Nisan 2020].
- [57]. Data, <https://tr.wikipedia.org/wiki/Veri>, [Ziyaret Tarihi: 4 Nisan 2020].
- [58]. Kurt F., 2020, Derin Öğrenme (deep learning) nedir?, <http://www.derinogrenme.com/2015/07/21/derin-ogrenme-deeplearning-nedir/>, [Ziyaret Tarihi: 4 Nisan 2020].
- [59]. Kaggle, 2020, <https://kaggle.com>, [Ziyaret Tarihi: 4 Nisan 2020].
- [60]. Keras, 2020, Keras applications. <https://keras.io/api/applications/>. [Ziyaret Tarihi: 4 Nisan 2020].
- [61]. OpenCV, 2020, Orb algoritması, https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html#orb, [Ziyaret Tarihi: 4 Nisan 2020].
- [62]. OpenCV, 2020, Reading and writing images and video, https://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html?highlight=imwrite#cv2.imwrite, [Ziyaret Tarihi: 4 Nisan 2020].

- [63]. Brownlee J., 2020, A Gentle Introduction to the Rectified Linear Unit (ReLU), <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>, [Ziyaret Tarihi: 4 Nisan 2020].
- [64]. DeepAI, 2020, Relu tanımı, <https://deepai.org/machine-learning-glossary-and-terms/relu>, [Ziyaret Tarihi: 4 Nisan 2020].
- [65]. OpenCV, 2020, Resize image function, https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html?highlight=resize#cv2.resize, [Ziyaret Tarihi: 4 Nisan 2020].
- [66]. Sınıflandırma işlemi görseli, 2020, <https://adeshpande3.github.io/assets/Cover.png>, [Ziyaret Tarihi: 4 Nisan 2020].
- [67]. Ramsundar B., Zadeh R. B., 2020, Tensorflow for deep learning, <https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>, [Ziyaret Tarihi: 4 Nisan 2020].
- [68]. Calonder M., Lepetit V., Strecha C., and Fua P., 2010, Brief: Binary robust independent elementary features, volume 6314, 778–792.
- [69]. Goodfellow I., Bengio Y., and Courville A., 2016, *Deep Learning*. MIT Press, <http://www.deeplearningbook.org>, [Ziyaret Tarihi: 18 Mayıs 2020].
- [70]. Khan R. and Debnath R., 2019, Multi class fruit classification using efficient object detection and recognition techniques, *International Journal of Image, Graphics and Signal Processing*, 11, 1–18.
- [71]. Kim Y. and Moon T., 2015, Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks, *IEEE Geoscience and Remote Sensing Letters*, 13, 1–5.
- [72]. Mikolajczyk K. and Schmid C., 2005, A performance evaluation of local descriptors, *IEEE transactions on pattern analysis and machine intelligence*, 27, 1615–30.
- [73]. Pietikainen M., Hadid A., Zhao G., and Ahonen T., 2011, *Computer Vision Using Local Binary Patterns*, volume 40.
- [74]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014, Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15, 1929–1958.
- [75]. Viswanathan D., 2011, Features from accelerated segment test (fast). Proceedings of the 10th workshop on Image Analysis for Multimedia Interactive Services, London, UK, 6–8.

EKLER

EK 1. Oluşturulan Mimari Model



ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Yasin KÖKER
Doğum Yeri	İstanbul
Doğum Tarihi	15.06.1993
Uyruğu	☒ T.C. ☐ Diğer:
Telefon	0539 557 53 49
E-Posta Adresi	yasinkoker@hotmail.com
Web Adresi	-



Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Ticaret Üniversitesi
Fakülte	Mühendislik ve Tasarım Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	2015

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri
Anabilim Dalı	Enformatik Anabilim Dalı
Programı	Enformatik Programı
Mezuniyet Tarihi	2020