

T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**DERİN ÖĞRENME İLE İÇECEK ŞİŞE KAPAĞI
ÜZERİNDEKİ GÖRSELLERDEN ÜRÜN TİPİ TANIMA**

Yüksek Lisans Tezi

Tezi Hazırlayan
Volkan COŞKUN

İstanbul, 2020

T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**DERİN ÖĞRENME İLE İÇECEK ŞİŞE KAPAĞI
ÜZERİNDEKİ GÖRSELLERDEN ÜRÜN TİPİ TANIMA**

Yüksek Lisans Tezi

Tezi Hazırlayan
Volkan COŞKUN
17080208010
0000-0002-7976-5465

Danışman
Dr. Öğr. Üyesi Turhan KARAGÜLER

İstanbul, 2020

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Derin Öğrenme ile İçecek Şişe Kapağı Üzerindeki Görsellerden Ürün Tipi Tanıma” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullandıkları her yerde bunlara atıf yapıldığını, patent ve telif haklarını ihlal edici bir davranışımın olmadığını belirtir ve bunu onurumla doğrularım.28/12/2020

Volkan COŞKUN



T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

28.12.2020

Enstitümüz **Bilgisayar Mühendisliği** Anabilim Dalı **Bilgisayar Mühendisliği** Programı Yüksek Lisans öğrencilerinden **17080208010** numaralı **Volkan COŞKUN'un** “Beykent Üniversitesi Eğitim-Öğretim Yönetmeliği”nin ilgili maddesine göre hazırlayarak Enstitümüze teslim ettiği **“Derin Öğrenme İle İçecek Şişe Kapağı Üzerindeki Görsellerden Ürün Tipi Tanıma”** konulu tezini, Yönetim Kurulumuzun 22/12/2020 tarih ve 2020/37 sayılı toplantısında seçilen ve Online toplanan biz jüri üyeleri huzurunda, ilgili yönetmeliğin (c) bendi gereğince (45) dakika süre ile Microsoft Teams programı aracılığıyla on-line olarak aday tarafından savunulmuş ve sonuçta adayın tezi hakkında **“OYBİRLİĞİ”** ile **“KABUL”** kararı verilmiştir.

İşbu tutanak, 4 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü’ne sunulmak üzere tarafımızdan düzenlenmiştir.

DANIŞMAN
Dr. Öğr. Üyesi Turhan KARAGÜLER
(Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi Atınç YILMAZ
(Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi Serkan AYVAZ
(Bahçeşehir Üniversitesi)

Adı ve Soyadı : Volkan COŞKUN
Danışman : Dr. Öğr. Üyesi Turhan KARAGÜLER
Türü ve Tarihi : Yüksek Lisans Tezi, 2020
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Derin Öğrenme, Evrişimsel Sinir Ağları, İçecek Şişe Kapağı, Ürün Tipi, Sınıflandırma

ÖZ

DERİN ÖĞRENME İLE İÇECEK ŞİŞE KAPAĞI ÜZERİNDEKİ GÖRSELLERDEN ÜRÜN TİPİ TANIMA

Endüstriyel içecek üretim hatlarında üretilen içecekler tazeliğinin ve bütünlüğünün korunması amacı ile şişelere doldurulur ve kapakları kapatılarak tüketicilere ulaştırılır. İçecek üretici firmalar çeşitli pazarlama ve reklam faaliyetleri ya da ürün tiplerinin birbirinden ayrıştırılması için ürün yelpazesinde bulunan her farklı ürün için üzerinde farklı görsel tasarımın bulunduğu şişe kapaklarını tercih edebilir, bu nedenle doğru ürüne doğru kapağın uygulanması önem taşımaktadır. Üretim sürecinde kapak uygulama makinasında yaşanacak bir sorun neticesinde, üretim hattında o an üretimi yapılan ürün şişesine, ürüne ait olmayan bir kapağın uygulanması mümkündür, bu durumda üretici açısından çözülmesi gereken bir kalite sorunu ortaya çıkmaktadır. Bu çalışmada bahsedilen sorunun çözümüne yönelik olarak, derin öğrenme ile içecek şişe kapağı üzerindeki görsellerin tanınması ve dolayısı ile kapak ile ürün arasında ilişki kurulduğu için ürün tipinin tanınması amaçlanmıştır.

Evrişimsel sinir ağları, derin öğrenme tekniklerinin bir sınıfıdır. Ürün tipinin tanınmasında bir evrişimsel sinir ağı modeli olan VGG-16 modeli kullanılmıştır. Modelin ImageNet veri kümesi ile eğitilmiş hali üzerinden, tam bağlı katmanın son katmanı hariç öğrenme aktarması yapılmıştır. Tam bağlı katmanın son katmanı olan sınıflandırma katmanı bu çalışma için oluşturulmuş 9 farklı ürüne ait toplam 7200 adet fotoğraftan meydana gelen veri kümesi ile yeniden eğitilmiştir. Eğitim ve doğrulama sürecinde %99.99 doğruluk oranına ulaşılarak tatminkar bir sonuç elde edilmiştir. Ayrıca farklı ürüne ait bir kapağın ürün tipinin derin öğrenme ile tanınması ve o an üretimi yapılan ürüne uygulanmasını engellemeye yönelik bir prototip yazılım hazırlanarak önerilen çözümün uygulaması gerçekleştirilmiştir.

Name and Surname : Volkan COŞKUN
Supervisor : Dr. Lecturer Turhan KARAGÜLER
Degree and Date : Master Thesis, 2020
Major : Computer Engineering
Key Words : Deep Learning, Convolutional Neural Networks, Bottle Cap, Product Type, Classification, Recognition

ABSTRACT

PRODUCT TYPE RECOGNITION FROM VISUALS ON BEVERAGE BOTTLE CAP WITH DEEP LEARNING

Beverages produced in industrial beverage production lines are filled in bottles, bottle caps are closed and delivered to consumers in order to protect their freshness and integrity. Beverage companies may prefer different visual designs or figures on bottle caps for each product type or Stock Keeping Unit (SKU) because of marketing and advertising strategies or differentiating one product regarding to another in their product range. Therefore, it is important to use the right cap design on the right product. Due to a problem to be experienced in the production process, it is possible to apply a cap that does not belong to the product that is currently being produced in the production line, in this case a quality problem that needs to be solved for the manufacturer arises.

In this study, in order to solve mentioned problem, it is aimed to recognize the visuals on the beverage bottle cap by Deep Learning, accordingly a relationship is established between the cap and the product type. Convolutional neural network is a class of deep learning techniques. VGG-16 model, which is a convolutional neural network model, has been used to recognize the product type. Except for the last layer of the fully connected layer, transfer learning has been made over the model's training with the ImageNet dataset. Classification layer, which is the last layer of the fully connected layer, has been retrained with a data set consisting of a total of 7200 images of 9 different products created for this study. In the training and validation process, a satisfactory result has been achieved by achieving 99.99% training accuracy. In addition, a prototype software has been prepared to recognize product type of the cap belonging to different product type and prevent to apply on the bottle.

İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

TABLolar LİSTESİ..... iii

ŞEKİLLER LİSTESİ..... iv

KISALTMALAR v

GİRİŞ 1

1. ENDÜSTRİYEL İÇECEK ŞİŞE KAPAĞI KAPATMA

MAKİNALARI..... 5

2. DERİN ÖĞRENME..... 7

2.1. Temel Kavramlara Giriş 7

2.1.1. Yapay Zeka 7

2.1.2. Makine Öğrenmesi 8

2.1.2.1. Makine Öğrenmesinde Öğrenme Yaklaşımları 8

2.1.2.2. Makine Öğrenmesi Görevleri 9

2.2. Yapay Sinir Ağları 9

2.2.1. Biyolojik Sinir Hücresi 10

2.2.2. Yapay Sinir Hücresi 11

2.2.3. Katman Kavramı 14

2.3. Yapay Sinir Ağlarında Öğrenme 16

2.3.1. İleri Yayılım 16

2.3.1.1. Yitim (Loss) Fonksiyonu 17

2.3.2. Geri Yayılım Algoritması 17

2.4. Evrimsel Sinir Ağları (ESA) 19

2.4.1. Evrişim İşlemi 21

2.4.2. Dolgulama (Padding) 24

2.4.2.1. Dolgulama Seçenekleri 25

2.4.3. Havuzlama (Pooling) Katmanı 26

2.4.4. Düzleştirme (Flattening) 27

2.4.5. Tam Bağlı Katman 27

2.5. Tensorflow.....	28
2.6. Keras.....	28
2.7. VGGNet.....	28
3. DERİN ÖĞRENME İLE İÇECEK ŞİŞE KAPAĞI ÜZERİNDEKİ	
GÖRSELLERDEN ÜRÜN TİPİ TANIMA	30
3.1. Veri Kümesinin Hazırlanması	30
3.2. Modelin Oluşturulması.....	32
3.3. Üst Değişkenlerin Ayarlanması (Hyperparameter Tuning)	33
3.4. Eğitim ve Test Sonuçlarının Değerlendirilmesi	34
3.5. Çalışmanın Prototiplenmesi	36
SONUÇ.....	39
KAYNAKÇA	41
EKLER.....	44

TABLÖLAR LİSTESİ

	Sayfa No.
Tablo 1. Oluşturulan veri kümesi detayları.....	31
Tablo 2. Oluşturulan modelinin mimari yapısı	32
Tablo 3. Hata matrisinden elde edilen ölçüm metrikleri.....	36



ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 1. Kapak kapatma makinası.....	5
Şekil 2. Yapay Zeka ve Sınıfları.....	7
Şekil 3. Biyolojik sinir hücresi	10
Şekil 4. Yapay sinir hücresi	12
Şekil 5. Aktivasyon fonksiyonları	14
Şekil 6. Tipik ileri beslemeli yapay sinir ağı yapısı	15
Şekil 7. İleri Yayılım Yönü	16
Şekil 8. Geri Yayılım Yönü.....	18
Şekil 9. ESA yapısı.....	20
Şekil 10. Giriş ve Filtre matrislerinin konumlandırılması.....	22
Şekil 11. 2 Boyutlu evrişim işleminin görsel bir temsili	23
Şekil 12. Evrişimsel sinir ağlarında evrişim işlemi	24
Şekil 13. Maksimum havuzlama	26
Şekil 14. Ortalama havuzlama.....	27
Şekil 15. Düzleştirme	27
Şekil 16. VGG-16 mimari yapısı	29
Şekil 17. Doğruluk oranları	34
Şekil 18. Yitim miktarları.....	35
Şekil 19. Hata matrisi	35
Şekil 20. Kamera düzeneği.....	37
Şekil 21. Eşleşen ürün	38
Şekil 22. Eşleşmeyen ürün.....	38

KISALTMALAR

CCD	: Charge-Coupled Device
CMOS	: Complementary Metal Oxide Semiconductor
CPU	: Central Processing Unit
ESA	: Evriřimsel Sinir Ađı
FullHD	: Full High Definition
GPU	: Graphics Processing Unit
ILSVRC	: ImageNet Large Scale Visual Recognition Challenge
°C	: Degree Celsius
RAM	: Random Access Memory
ReLU	: Rectified Linear Unit
SKU	: Stock Keeping Unit
TPU	: Tensor Processing Unit
VGG	: Visual Geometry Group

GİRİŞ

Yarı iletken devre elemanlarının tasarım ve üretim tekniklerinde yaşanan gelişmeler, mikroişlemcilerin hesaplama güçlerinin artmasının yanı sıra maliyetlerinin azalarak erişilebilirliklerini kolaylaştırmıştır. Donanım alanında yaşanan gelişmelerle bağlantılı olarak yazılım alanında da gelişmeler yaşanmış, sofistike algoritmaların kodlanması ve geçmişte kavramsal temelleri ortaya atılmış ancak donanım yetersizleri nedeni ile durma noktasına gelen çalışmalara tekrar işlerlik kazandırılması mümkün olmuştur. Bahsedilen tüm bu gelişmeler neticesinde; veri üretebilen, üretilen verilerden öğrenerek tahmin yapabilen makine öğrenmesi tekniklerini kullanan cihaz ve uygulamalar yaygınlaşarak hayatın hemen hemen her yerinde kullanım alanı bulmuştur.

Bu çalışmada, makine öğrenmesinin (Machine Learnig) bir sınıfı olan derin öğrenme (Deep Learning) tekniklerinden evrimsel sinir ağları (Convolutional Neural Networks) kullanılarak içecek şişe kapağı üzerindeki görsel tasarımlar yardımı ile ürün tipi tespit edilerek, endüstriyel içecek üretim hatlarında yaşanan kalite probleminin çözümüne yönelik bir yöntem önerilmiş ve uygulaması gerçekleştirilmiştir.

Endüstriyel otomasyon tabanlı olarak içecek üretimi yapan fabrikalarda üretilen içecekler, otomatik dolum makinaları tarafından şişelere doldurulur ve kapak uygulayıcı makine tarafından kapakları kapatılarak tüketiciye sevk edilmeye hazır hale getirilir. Kapak uygulayıcısı, şişelere uygulanacak kapakların yerleştirildiği bir hazneye sahiptir, kapaklar üretim öncesinde bu hazneye yerleştirilir, uygulayıcı kapağı hazneden alarak mekanik düzenekler yardımı ile uygulamaya hazır konuma getirir ve üretim bandı üzerinde akan şişelere uygular.

İçecek üreticileri çeşitli pazarlama stratejileri nedeni ile ürün gamında bulunan her farklı ürün için üzerinde değişik figürlerin bulunduğu kapak tasarımları tercih edebilmektedir. Şayet kapak uygulayıcısının haznesinde o an üretimi yapılan ürüne ait olmayan bir kapak bulunuyorsa kapak uygulayıcı bu kapağı da ürüne uygular, çünkü uygulayıcının bu durumu kontrol eden dahili bir mekanizması yoktur. Saha uygulamalarında haznede yanlış kapak bulunması, bir önceki üretimden kalan farklı kapakların haznede unutulması ya da kapak tedarikçisinin talep edilen kapak tasarımı

arasına başka tasarımları karıştırması gibi durumlarda ortaya çıkmaktadır. Üretimde yaşanan bu sorunun tespit edilememesi durumunda ürün tüketiciye yanlış bir ürüne ait kapakla gitmesine neden olmakta ve dolayısı ile üretici açısından çözülmesi gereken bir kalite sorunu oluşturmaktadır. Kapak üzerindeki görsel ile ürün tipi arasında bağlantı kurulduğu için üreticinin ürün gamındaki her farklı ürün tipinde farklı kapak tasarımı kullandığı varsayılmıştır.

Nesne tanıma alanında düzenlenmekte olan “ImageNet Large Scale Visual Recognition Competition” (ILSVRC) adlı yarışmayı 2012 yılında Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından geliştirilen AlexNet adlı modelin kazanması ile birlikte (Krizhevsky, Sutskever ve Hinton 2012), derin öğrenme teknikleri içerisinde evrimsel sinir ağları(ESA) bilgisayar görüşü alanında nesne tanıma noktasında baskın bir konuma gelmiştir (Yamashita, ve diğerleri 2018, 611-629).

Ürün tipinin tespit edilmesi için yöntem olarak bir ESA modeli olan VGG-16 modeli kullanılmıştır, her birinin üzerinde farklı görsellerin bulunduğu 9 farklı ürün tipine ait içecek kapağı, bu model ile %99.99 eğitim doğruluk (Training Accuracy) oranı ile sınıflandırılarak tatminkar bir sonuç elde edilmiştir. Görsellerin sınıflandırılmasında kullanılan VGG-16 modelinin eğitiminde tam bağlı katmanın (Fully Connected Layer) son katmanı hariç ImageNet veri kümesi ile ön eğitim(Pre trained) yapılmış hali ile öğrenme aktarması (Transfer Learning) yapılmıştır, son tam bağlı katman ise bu çalışma dahilinde oluşturulmuş 9 farklı ürüne ait 7200 adet fotoğraftan oluşan eğitim veri kümesi ve 1800 fotoğraftan oluşan doğrulama veri kümesi ile yeniden eğitilmiştir. Modelin eğitim doğruluk ve doğrulama doğruluk (Validation Accuracy) oranları grafiklerle görselleştirilmiş ve modele ayrıca uygulanmış test veri kümesine ait tahmin sonuçları bir hata matrisi ile görselleştirilmiştir.

Eğitim ve test işlemlerinin ardından önerilen tespit sistemine ait bir uygulama yazılımı hazırlanmıştır; uygulama yazılımında kullanıcı üretime başlamadan önce üretimi yapılacak ürünü menüden seçmekte ve ardından sınıflandırılacak kapak kameraya gösterilmektedir, kameraya gösterilen kapağın ürün tipi, daha önce eğitimi yapılan ESA vasıtası ile tespit edilmektedir. Seçilen kapak ile, ürün tipi tespit edilen

kapak eşleşiyorsa yazılım “Eşleşen Ürün” sonucunu döndürmekte, fakat seçilen ürün ile ürün tipi tespit edilen kapak arasında uyumsuzluk var ise yazılım “Eşleşmeyen Ürün” hata mesajını döndürmektedir. Gerçek üretim hattında ise kamera, kapak uygulayıcının haznesinde kapakların üzerindeki görselleri görecektir şekilde yerleştirilmesi ve hareket halindeki kapak tam olarak kameranın görüş alanının merkezine geldiği anda kapağın fotoğrafının alınması gerekmektedir. İstenildiği takdirde eşleşmeyen ürün tipinin tespit edilmesi ile birlikte bir alarm işareti oluşturularak görsel veya işitsel olarak bildirim yapılabilir.

Endüstriyel üretim hatlarında çeşitli amaçlarla kalite kontrolüne yönelik görüntü işleme sensörleri yaygın olarak kullanılmaktadır. Bu sensörler içlerinde CMOS kamera ve üzerinde kural tabanlı görüntü işleme algoritmalarının çalıştığı işlemci (CPU) biriminden oluşmaktadır, hızlı tepki süreleri ile avantaj sağlamakta birlikte karmaşık figürler içeren görüntüler üzerinde hata tespitinde veya görüntü sınıflandırılmasında yetersiz kaldığı durumlar ortaya çıkmaktadır. Ayrıca bu yapılarda üzerinde çalışılan görüntüden istenilen sonucun elde edilmesi için ince ayar (fine-tune) adı verilen ve uzmanlık gerektiren bir tanımlama yapılması ve tanımlamanın her farklı kalite kontrol mekanizması veya ürün için sürecin tekrar edilmesi gerekmektedir. Önerilen çalışmada, üretici sisteme yeni tanımlayacağı ürün olması durumunda ağı eğitilebilir katmanlarını yeniden eğitmek dışında görsel üzerinde bir ince ayar veya özellik/öznetelik çıkarımı(feature extraction) işlemine de ihtiyaç duyulmayacaktır. Evrimsel sinir ağlarında el yordamı ile özellik çıkarımına ihtiyaç duyulmamasının nedeni, özellik çıkarım işlemini ağı kendisi tarafından yapılmasından kaynaklanmaktadır (Yamashita, ve diğerleri 2018, 611-629). Bu bakımdan önerilen çalışma kural tabanlı görüntü işleme sensörlerine göre önemli bir üstünlük taşımaktadır.

Bu tez çalışması aşağıda belirtilen sıra ile organize edilmiş olup her bölümün içeriği hakkında kısa bir özet sunulmuştur;

Bölüm 2’de, mevcut sorunun daha net olarak ortaya konulması için endüstriyel içecek dolum ve kapak uygulayıcı mekanizmaları tanıtılmıştır.

Bölüm 3’te, derin öğrenme konusuna girmeden önce bilinmesi gereken temel kavramlar olan yapay zeka ve makine öğrenmesi kavramları açıklanmış; ardından bir

derin sinir ağı(Deep Neural Network) oluşturmak için gerekli yapılar olan: Yapay Sinir Hücresi (Artificial Neuron), Yapay Sinir Ağları (Artificial Neural Networks) ve Evrimsel Sinir Ağları açıklanarak çalışmanın üzerine oturacağı zemin için gerekli temel bilgiler verilmiştir.

Bölüm 4'te, derin öğrenme ile kapak üzerindeki görselden ürün tipinin tespit edilmesine yönelik önerilen yöntem detaylandırılmış, bu yöntem esas alınarak oluşturulan uygulama yazılımı tanıtılmıştır.

Bölüm 5'te, bu çalışmaya yönelik olarak elde edilen sonuçlar ve öneriler verilmiştir.

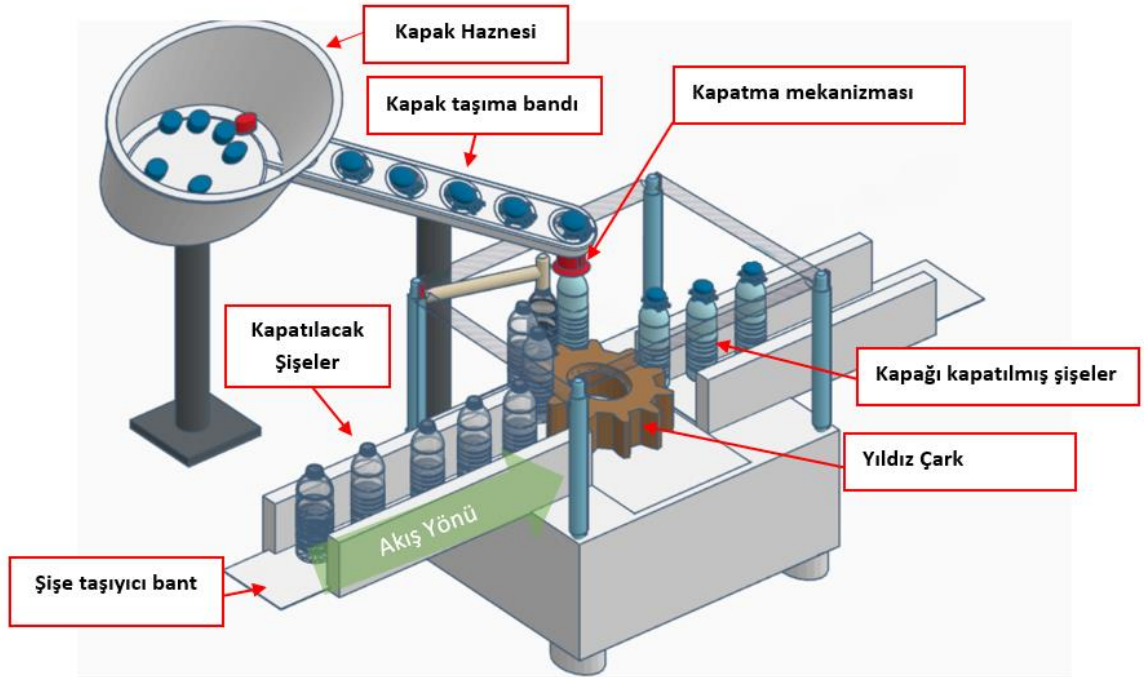


1. ENDÜSTRİYEL İÇECEK ŞİŞE KAPAĞI KAPATMA MAKİNALARI

Bu tez çalışmasında ortaya konulan problem ve önerilen çözümün detaylandırılabilmesi için endüstriyel içecek şişesi dolum ve kapatma mekanizmaları hakkında genel bilgi verilmiştir.

İçecek üretim sürecinde üretilen içeceklerin, tazeliğinin korunarak tüketicilere ulaştırılması için endüstriyel dolum makinaları yardımı ile şişelere doldurulur ve kapakları kapatılarak tüketiciye sevk edilmeye hazır hale getirilir. İçecek üretici firmalar ürün gamları içerisinde değişik içerik ve formülasyonda çeşitli ürün tiplerine sahiptirler, bu ürünlerin birbirlerinden ayrıştırılması ve çeşitli pazarlama nedenleri ile her farklı ürün tipine üzerinde farklı görsel tasarımların bulunduğu kapak tasarımları uygulayabilmektedirler.

Şekil 1’de tipik bir kapak uygulama makinası verilmiştir.



Şekil 1. Kapak kapatma makinası

Dolumu yapılan şişeler taşıyıcı bant vasıtası ile kapak kapatıcı makina girişine ulaşır, makina tarafından kapatılacak kapaklar makinanın kapak haznesinden kapak taşıyıcı bant yardımı ile alınır ve kapak besleme rayına ulaştırılır, burada kapak mekanik bir düzeneyle kapağın uygulanacağı yöne çevrilir ve şişenin üzerine

konumlandırılır. Mevcut yapı içerisinde kapak haznesine hangi kapak yerleştirilirse makina tarafından ürün tipine bakılmaksızın hazneden alınarak ürüne uygulanır, ancak kapak haznesinde herhangi bir nedenden dolayı o an üretimi yapılan ürün tipine ait olmayan bir kapak bulunuyorsa bu kapak da üretim hattında ilerleyen şişeye uygulanacaktır, bu durumda engellenmesi gereken bir kalite problemi çıkmaktadır. Eğer gözle veya başka bir yöntemle kontrol edilerek engellenmemesi durumunda tüketiciye ulaşan ürün üzerinde başka bir ürün tipine ait bir kapak olmaktadır ki bu hem üretici hem de tüketici açısından istenmeyen bir durumdur.

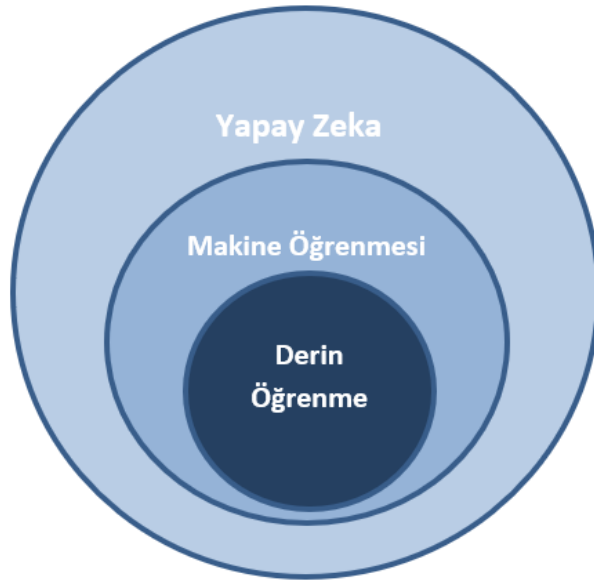
Ortaya konulan sorunun çözümüne yönelik olarak endüstride kullanılan genel maksatlı araçlar ve bu araçların kullandığı yöntemler de mevcuttur; bunların başında CCD ve CMOS görüntü sensörleri ve sensörlerden alınan verilerin işlendiği ticari yazılımlar gelmektedir. Bahsedilen ticari yazılımlarda ürün tipi, üretim hatası veya anomali tespit etmeye yönelik hazır fonksiyonlar sunulmaktadır, bu fonksiyonlar ise klasik görüntü işleme teknikleri ile oluşturulmuştur, bunların başında özellik(öznitelik) çıkarımı için gereken filtre tasarımı gibi uzmanlık gerektiren teknikler gelmektedir. Öznitelik çıkarımı, üzerinde çalışılan konuya hakim kişiler tarafından öznitelik mühendisliği teknikleri kullanılarak yapılabilir (Alpaydın 2010, 110). Böyle bir işlem ise üretici açısından maliyet artırıcı bir unsurdur.

2. DERİN ÖĞRENME

Derin öğrenme, birçok doğrusal olmayan bilgi işleme katmanından yararlanan, denetimli ya da denetimsiz öğrenmede öznelilik çıkarımı yapmak, patern analizi ve sınıflandırma yapmak amacı ile kullanılan bir makine öğrenmesi tekniğidir (Deng ve Yu 2014, 199-200). Derin öğrenme konusunda bilgi sahibi olmak ve literatürdeki yerini anlamlandırabilmek için sırası ile yapay zeka ve makine öğrenmesi gibi temel kavramlara bir göz atmak gerekmektedir.

2.1. Temel Kavramlara Giriş

Derin öğrenme, makine öğrenmesi ve yapay zeka birbiri ile ilintili kavramlardır ve bir bütün içinde her birinin nerede konumlandığı görsel olarak Şekil 2’de verilmiştir.



Şekil 2. Yapay Zeka ve Sınıfları

2.1.1. Yapay Zeka

Yapay zeka alanındaki çalışmalar çok daha eskilere gitse de, terim olarak “Yapay Zeka” ilk kez 1956 yılında John McCarthy tarafından ifade edilmiştir. Ancak yapay zeka tanımında henüz geniş kapsamlı bir uzlaşma söz konusu değildir (Nilsson 2009, 13). Bir tanım vermek gerekirse, makineleri akıllı hale getirmeye yönelik faaliyetler şeklinde ifade edilebilir.

2.1.2. Makine Öğrenmesi

Makine öğrenmesi, yapay zekanın bir alt dalıdır. Makine öğrenmesi, bilgisayar tarafından yapılması gereken işlemlerin; her durumunun belirgin bir şekilde programlanması yerine, yapacağı işlemleri çeşitli kaynaklardan sunulan verilerden öğrenebilmesi ve geçmiş deneyimleri de dikkate alan kararlar verebilmesi şeklinde tanımlanır.

Bu ifade ile birlikte, makine öğrenmesinin yapay zeka kavramının akıllı makinalar ortaya çıkarma amacına hizmet ettiği açıktır. Nasıl ki makine öğrenmesi amacı itibarı ile yapay zeka kavramına hizmet ediyorsa, derin öğrenme de makine öğrenmesi kavramına farklı tekniklerle hizmet etmektedir; bu nedenle derin öğrenme de makine öğrenmesinin bir alt dalıdır, ancak sıralamanın tersi mümkün değildir, yani her yapay zeka uygulaması makine öğrenmesi olmayacağı gibi her makine öğrenmesi uygulaması derin öğrenme değildir.

2.1.2.1. Makine Öğrenmesinde Öğrenme Yaklaşımları

Gözetimli (Supervised) Öğrenme: Gözetimli öğrenmede sisteme giren verinin aslında hangi sınıfa dahil olduğu bilgisi de verilir. Sistem bir sınıf tahmini yapar ve yaptığı bu tahminin verilen sınıf ile olan ilişkisini kontrol eder, eğer üretilen tahmin değeri girişten verilen etiket değeri ile örtüşüyorsa bu ilişki kuvvetlendirilir tam tersi durumunda yani örtüşmemesi durumunda ilişki zayıflatır böylece sistem giren veri ile istenen çıktı arasındaki ilişkiyi belirleyen bir fonksiyon ortaya çıkartır. Gelecek tahminlerde öğrenilen fonksiyon yardımı ile tahmin gerçekleştirilir.

Gözetimsiz (Unsupervised) Öğrenme: Burada gözetimli öğrenmeden farklı olarak sisteme giren girdi verisinin etiket/sınıf bilgisi bilinmemektedir. Sistem girdi verileri arasındaki gizli örüntüyü tespit edip gelecek tahminlerini buna göre yapmayı amaçlar.

Yarı Gözetimli (Semi Supervised) Öğrenme: Bu teknikte etiketlenmemiş verilerin de eğitim için kullanılması mümkündür. Ağırlıklı olarak veri kümesinde çok miktarda etiketsiz veri, az miktarda etiketli veri bulunur. Bazı uygulamalarda etiketli veri bulmak konusunda zorluklar yaşanabilir, uygulamaya bağlı olarak bazı uygulamalarda etiketleme ileri uzmanlık gerektiren bir konu olabilir; örneğin radyolojik görüntülerin incelenerek etiketlenmesi uzmanlık gerektiren bir konudur.

Bahsedilen zorluklar nedeni ile algoritmayı eğitmek için yeterli etiketli veri olmadığı için etiketsiz verilerden de yararlanılır.

Pekiştirmeli (Reinforcement) Öğrenme: Bu öğrenme yönteminde, bir etmen(Agent) tanımlı bir ortam(Environment) içerisinde belli konumlarda (State) belli aksiyonlarda (Actions) bulunarak hedefe ulaşmaya çalışır, her aksiyon sonrası geçilen yeni konum için bir sayısal ödül (Reward) değeri hesaplanır, burada amaç etmenin hedefe en yüksek ödül değinde ulaşmasıdır. Etmen hedefe giden durumları deneyimleyerek amaca en yüksek puanla ulaşmayı sağlayan bir davranış kalıbı ortaya çıkarmaya çalışır.

2.1.2.2. Makine Öğrenmesi Görevleri

Sınıflandırma (Classification): Sisteme uygulanan verinin hangi sınıfa ait olduğunu belirleme amacı ile kullanılan gözetimli öğrenme tekniklerini kullanan makine öğrenmesi görevidir. Sistem sınıflandırma işlemini girişine uygulanan verilerden öğrenmekte ve çıkışında kategorik ya da kesikli bir tahmin yapmaktadır. Sınıflandırılacak veri iki sınıflı (binomial) olabileceği gibi çok sınıflı (multi-class) da olabilir. Sınıflandırma işlemi daha çok kategorik veriler için kullanılır, örneğin kadın/erkek gibi cinsiyet sınıflandırması ya da bir öğrencinin alacağı not tahmininin FF ile AA aralığında harf notu şeklinde sınıflandırılması gibi.

Regresyon (Regression): Sınıflandırma işlemi ile benzer özelliktedir, burada farklı olarak sınıflandırılacak veri kategorik değil sayısal ya da sürekli verilerdir; örneğin hava durumu tahmininin °C olarak sayısal değer olarak verilmesi şeklinde.

Kümeleme (Clustering): Benzer özelliklere sahip verilerin aynı grup içerisine yerleştirilmesi işlemidir. Temel olarak gözetimsiz öğrenme tekniklerini kullanılır, yani verinin hangi gruba dahil olacağının bilgisi başlangıçta bilinmez, veri algoritmaya uygulanır verinin özelliklerine göre hangi gruptaki verilerle daha fazla benzerlik gösteriyorsa algoritma tarafından o gruba dahil edilir.

2.2. Yapay Sinir Ağları

Derin öğrenmede doğrusal olmayan işlem birimi olarak yapay sinir ağları sıklıkla kullanılmaktadır. Yapay sinir ağları ise bir problemi çözmek üzere çok sayıda

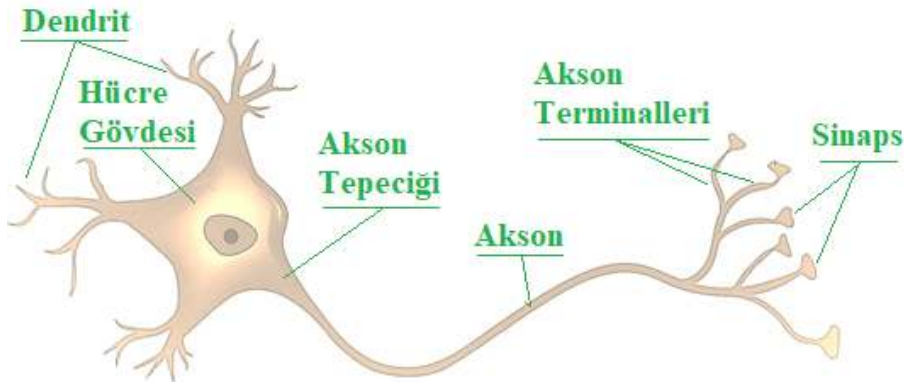
birbirleri ile bağı yapay sinir hücresinden (nöron) oluşur (Sivanandam, Sumathi ve Deepa 2008, 11).

Yapay sinir ağlarının tarihsel gelişimi hakkında ana hatlarıyla bilgi vermek gerekirse; ilk yapay sinir hücresi Warren McCulloch ve Walter Pitts tarafından 1943 yılında formüle edilmiştir. Yapay sinir hücreleri biyolojik sinir hücrelerinin taklit edilerek bir hesaplama modeli oluşturulmasını hedeflemiştir (Aggarwal 2015, 209). Frank Rosenblatt ilk tek katmandan oluşan eğitilebilir, algılayıcı (perceptron) modelini 1958 yılında önermiştir. Algılayıcı modeli AND, OR, NOT gibi basit lojik işlemleri yapabilmekte yeterli fakat XOR işleminin gerçekleşmesinde ise yetersiz olduğu 1969 yılında Marvin Minsky ve Seymour Papert tarafından gösterilmiştir. Bu çalışmanın ardından yapay sinir ağları üzerinde yapılan çalışmalar duraklama dönemi yaşamıştır (Keskenler ve Keskenler 2017, 8-18). 80'li yıllarda Geri Yayılım yönteminin kullanılmasıyla yapay sinir ağları yeniden gündeme gelmiştir (Akbaş 2018).

Yapay sinir hücresine geçiş yapmadan önce biyolojik sinir hücresi hakkında genel bilgiye sahip olunması, yapay sinir hücresi modelinin biyolojik sinir hücresinden nasıl esinlendiğini ve birbirleri ile örtüştüğü noktaları daha iyi anlaşılabilir kılmaktadır.

2.2.1. Biyolojik Sinir Hücresi

Tipik biyolojik sinir hücresini çok basit olarak tarif etmek gerekirse Dendritler (Dendrite), Hücre Gövdesi(Cell Body), Akson(Axon), ve Sinapslardan (Synapses) oluşur. Şekil 3'de biyolojik sinir hücresinin yapısı verilmiştir.



Şekil 3. Biyolojik sinir hücresi

Dendrit: Komşu sinir hücrelerinden gelen sinyallerin alınmasını sağlar.

Hücre Gövdesi: Dendritlerden gelen sinyalleri bir araya getirir, herhangi bir bilgi işleme gerçekleşmediğinde hücre zarının içi ve dışı kararlı bir elektrik yükü ile yüklüdür; hücre zarının içi dışına göre daha negatif yüklüdür, bu kararlı duruma istirahat membran potansiyeli adı verilir. Dendritlerden alınan sinyaller doğrultusunda hücre içindeki elektrik potansiyeli belli sınırlar içinde artış veya azalış gösterir, oluşan bu potansiyele kademeli potansiyel adı verilir. Kademeli potansiyelin, membran potansiyeli üzerinde artış ya da azalış yönünde bir etkisi vardır bu etki dendritlerden gelen sinyalin süresi ve şiddetine bağlıdır. Membran potansiyelinin hücre gövdesinde toplandığı son nokta (Axon hillock) akson tepeciğidir; akson tepeciği, hücre gövdesinin aksona bağlandığı özel bir bölgesindedir (Hemmings ve Egan 2012, 105). Akson tepeciğinde gerilim kapılı sodyum kapıları adı verilen bir yapı vardır. Membran potansiyeli, eşik değeri aştığı takdirde akson başlangıç segmentinde aksiyon potansiyeli adı verilen bir sinyal üretilir ve aksona doğru iletilir (Garrido, ve diğerleri 2003, 437-445).

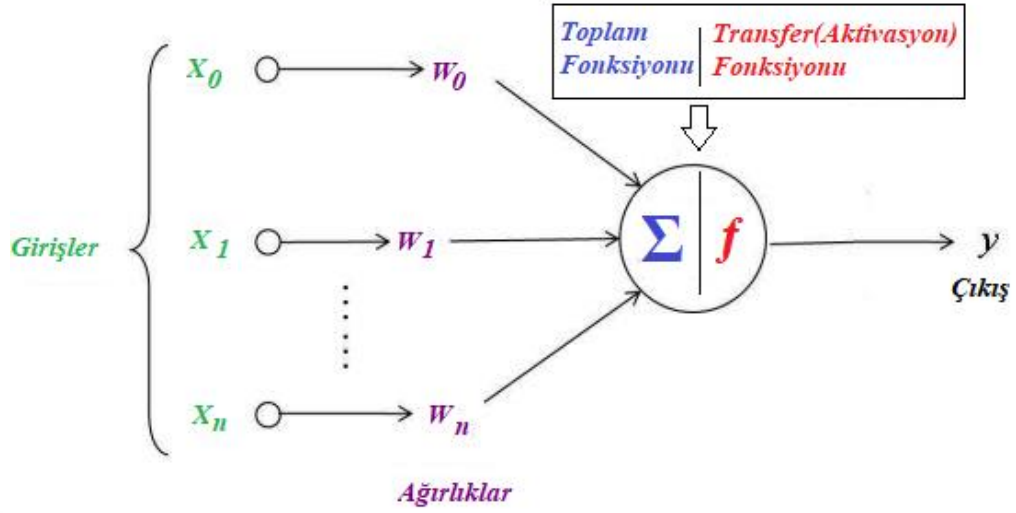
Akson: Gövdeden (akson tepeciğinden) tetiklenen sinyalin uç noktadaki hedefe taşınmasını sağlar.

Sinapslar: Nöronun aksonunun uç noktasında bulunan akson terminalleri ile hedef hücre ile bir araya geldiği bağlantı noktalarıdır, burada gerçekleşen nörotransmitter adı verilen moleküllerin salınımı ile bilgi akışı sağlanır.

Biyolojik nöron hakkında verilen bu temel bilgilerden sonra, artık biyolojik nörondan yapay nöron modeline geçiş yapılabilir.

2.2.2. Yapay Sinir Hücresi

Yapay sinir ağları, yapay sinir hücrelerinin belirlenen bir düzen içinde bağlantılar kurulması ile oluşturulur. Tipik bir yapay nöronun diyagramı Şekil 4’de verilmiştir.



Şekil 4. Yapay sinir hücresi

Bu diyagramdaki yapay nöronun, biyolojik nörondaki karşılıklarını vermek gerekirse;

“Dendritler Giriş vektörüne ($x_0 \dots x_n$) ,

Sinaps Ağırlıklara ($w_0 \dots w_n$) ,

Hücre Gövdesi Toplam fonksiyonu (Σ) ve Transfer fonksiyonuna (f),

Akson terminalleri Çıkışa (y) karşılık gelir” (Wikipedia, Artificial neuron 2020) .

Yapay nöronda x_0 ’dan x_n ’ye kadar, toplam $n + 1$ adet giriş olmak üzere, bu girişlerin her biri w_0 ile w_n arasındaki kendi ağırlık değerine karşılık gelen değer ile çarpılır ve bu çarpımların her biri toplanarak bir net çıkış elde edilir. Denklem 1’de verilen ağırlık toplamı uygulamada en yaygın kullanılan toplam fonksiyonudur, bunun yanında başka ihtiyaçlar için kullanılan toplama fonksiyonları da mevcuttur. Bu çalışmada toplam fonksiyonu olarak ağırlık toplamı ifade edilir.

$$s(k) = x_0.w_0 + x_1.w_1 + x_2.w_2 + \dots + x_n.w_n \quad (1)$$

Yapay sinir hücresinin çıkışı üzerinde etki etmek için bir yanlılık(bias) parametresi de kullanılabilir. Bu parametre (x_0) girişine $(+1)$ değeri atanarak (b) yanlılık girişi yapılır, böylece nörona (x_1) 'den (x_n) 'ye toplam n adet giriş kalmış olur, ifade buna göre yeniden düzenlenirse, Denklem 2 elde edilir.

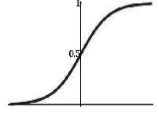
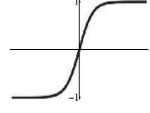
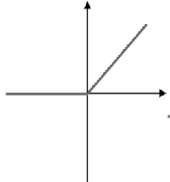
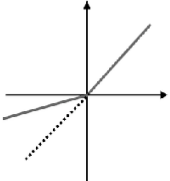
$$s(k) = b + x_1.w_1 + x_2.w_2 + \dots + x_n.w_n \quad (2)$$

Bu işlem sonrasında girişlerden gelen işaretler ağırlıklandırılarak bir net toplam elde edilmiştir, fakat nöronun $y(k)$ çıkışında nasıl bir çıkış işareti oluşacağı yani çıkışın aktive olup olmayacağı (f) aktivasyon fonksiyonuna bağlıdır. Aktivasyon fonksiyonları, girdi olarak gelen işarete göre çıkışın aktif olup olmamasını veya hangi ölçüde aktif olacağını belirleyen matematiksel fonksiyonlardır .

Aktivasyon fonksiyonu da eşitliğe dahil edildiğinde nöronun çıkışı, Denklem 3'de verilen ifade ile hesaplanır .

$$y(k) = f. (b + \sum_{k=1}^n x_k.w_k) \quad (3)$$

Aktivasyon fonksiyonu ihtiyaca göre herhangi bir fonksiyon olabilir, ancak kullanım yerine göre kendini kanıtlamış, literatürde kabul görmüş ve yaygın olarak kullanılan belli başlı fonksiyonlar vardır, bunlar; Sigmoid, tanh, ReLU, leaky ReLU, Softmax, aktivasyon fonksiyonlarıdır. Şekil 5'de bu aktivasyon fonksiyonlarının detayları verilmiştir.

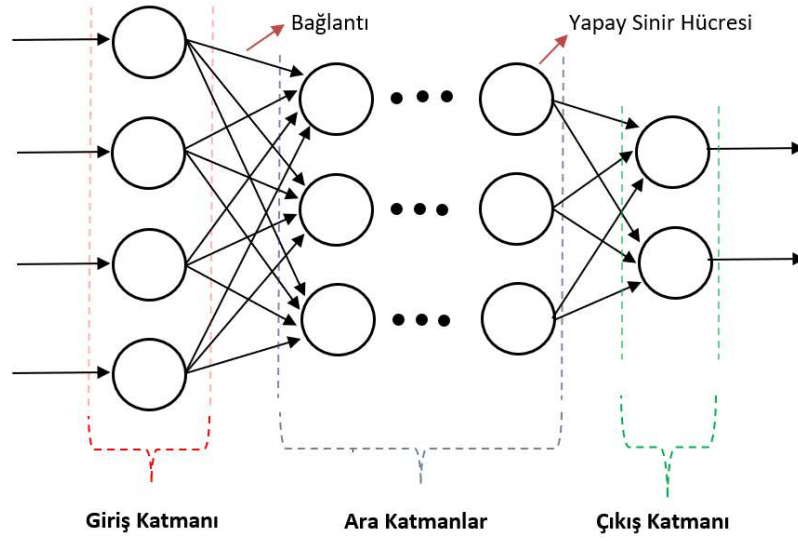
Name	Functions	Derivatives	Figure
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))^2$	
tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$	
Leaky ReLU	$f(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$	
Softmax	$f(x) = \frac{e^x}{\sum_1^j e^x}$	$f'(x) = \frac{e^x}{\sum_1^j e^x} - \frac{(e^x)^2}{(\sum_1^j e^x)^2}$	

Şekil 5. Aktivasyon fonksiyonları

Kaynak: Murugan, Pushparaja . 2017. “*Feed Forward and Backward Run in Deep Convolution Neural Network*” Computer Vision and Pattern Recognition (cs.CV). 1-20.

2.2.3. Katman Kavramı

Yapay sinir ağlarında nöron veya nöronlar katman adı verilen düzenler içerisinde bir araya getirilir. Şekil 6’da ileri beslemeli çok katmalı yapay sinir ağının diyagramı verilmiştir. Her yapay sinir hücresi bir düğüm olarak da adlandırılabilir, her düğüm aralarındaki bağlantılar vasıtası ile başka bir düğüme bağlanır, her bağlantının ağırlığı vardır, bu da bir düğümün diğeri üzerinde etki etmesini sağlar (Winston 1992, 445-446). Şekil 6’da verilen ağ için dairesel yapılar düğümü, oklar ise yapay sinir hücresinin çıkışını, kendinden sonra gelen katmandaki nöron veya nöronların girişine bağlayan bağlantıları ifade etmektedir.



Şekil 6. Tipik ileri beslemeli yapay sinir ağı yapısı

Tipik bir ileri beslemeli yapay sinir ağı üç ana katmandan oluşur.

Giriş Katmanı: Sinir ağına girişlerin yapıldığı katmandır. Genellikle giriş adedi kadar nöron vardır ve işaretler doğrudan sonraki katmanlara iletilir.

Ara Katman veya Katmanlar: Bu katman(lar) gizli katman olarak da ifade edilebilmektedir, giriş katmanının ardından bilgi bu katmana gelir. Her yapay sinir ağında ara katman bulunması gerekliliği yoktur. Ara katman adedi ve ara katmanda bulunan nöron adedi ihtiyaca göre değişiklik gösterebilir, sayının artması daha karmaşık işlemlerin çözümüne olumlu katkı sağlayabilir fakat hesaplama karmaşıklığını ve zaman karmaşıklığını arttırabilir, ayrıca her zaman daha fazla katman ve nöron eklenmesi daha iyi sonuç elde edileceği anlamına gelmez burada soruna göre bir optimizasyon gerekmektedir. Derin öğrenme kavramındaki “Derin” terimi ara katman sayısının birden fazla olmasından gelmektedir .

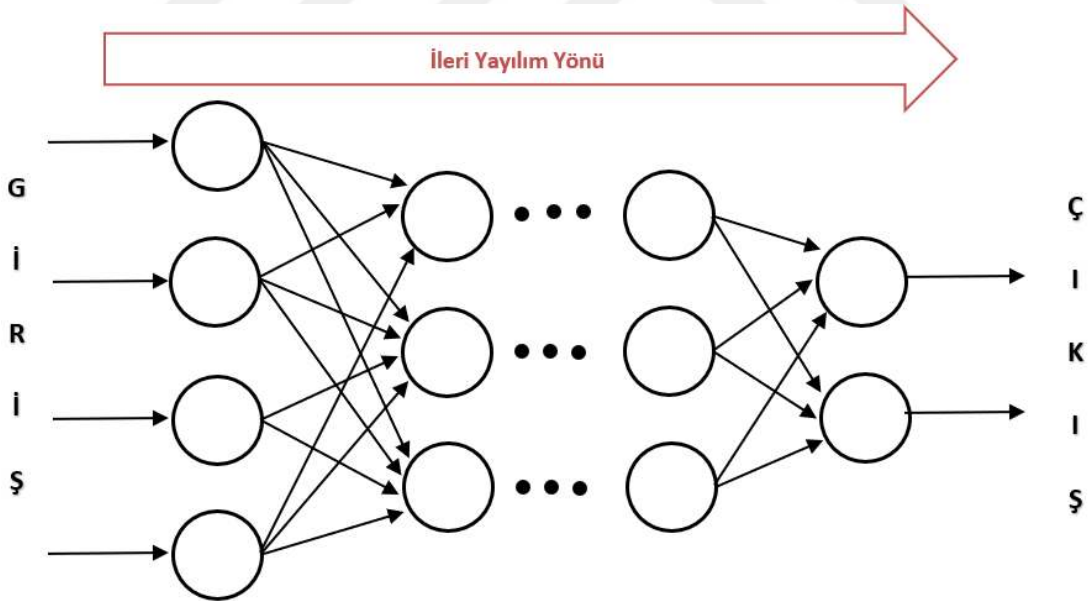
Çıkış Katmanı: Kendisinden önceki katmanda işlenen veriler bu katmandaki nöronların girişine aktarılır; genel anlamda çıkış katmanı olasılıksal bir dağılım yaparak sonucun dış ortamın beklediği çıktılara dönüşmesini sağlar.

2.3. Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağlarında öğrenme süreci anahtar bir role sahiptir çünkü yapay sinir ağları kendisine girdi olarak verilen verilerden öğrenir. Öğrenme işlemi 1949 yılında Donald O. Hebb tarafından ortaya konulan Hebbian öğrenme kuralını esas almaktadır; bu yaklaşıma göre ağırlıklar tespit edilerek hangi bağlantının hangi düğüme zayıf ya da güçlü olarak bağlanacağı belirlenmiş olur. Günümüzde yapay sinir ağları konusunda yapılan çalışmaların, ihtiyaçların da etkisi ile hızla gelişen dinamik bir yapısı vardır, bunun neticesinde farklı türde yapay sinir ağları ve değişik öğrenme teknikleri geliştirilmiştir; bu çalışmada öğrenme ileri yayılım ve geri yayılım algoritmaları üzerinden anlatılmıştır, ayrıca çalışmanın uygulama kısmında da ağıın eğitiminde bu yöntemler kullanılmıştır.

2.3.1. İleri Yayılım

Giriş katmanından çıkış katmanı yönüne doğru olmak üzere ağırlıkların ve yanlılık değerlerinin tespit edilerek bağlı olduğu diğer düğüme aktarılması ve bu değerlerin saklanması işlemidir.



Şekil 7. İleri Yayılım Yönü

2.3.1.1. Yitim (Loss) Fonksiyonu

İleri yayılım sonucunda çıkış katmanında oluşan değerler sinir ağına verilen girdilere karşı oluşturduğu tahmin değerleridir; tahmin değerinin olması gereken gerçek değere ne ölçüde yaklaştığı, sinir ağına başarımla ilgili ve tespit edilmesi gerekmektedir. Bu işlem Yitim (Loss) fonksiyonu yardımı ile gerçekleştirilir. İdealde yitim fonksiyonun değeri asgari düzeyde tutulması istenir, böylece tahmin değeri ile gerçekte olması gereken değer arasındaki fark azalmış olur.

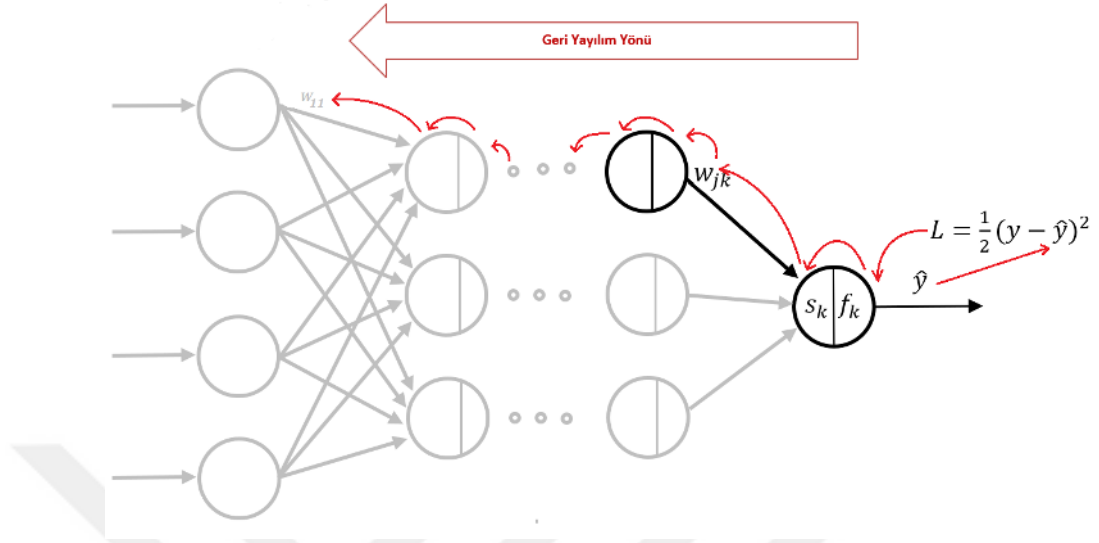
Aktivasyon fonksiyonlarında olduğu gibi değişik ihtiyaçlar için değişik yitim fonksiyonları mevcuttur. Bunlara Ortalama Karesel Hata, Log Loss, Categorical Cross Entropy gibi yitim fonksiyonları örnek verilebilir. Uygun uygulama için uygun yitim fonksiyonun seçimi önemlidir. Eğer yanlış yitim fonksiyonu tercih edilirse ağı eğitiminde beklenen başarı görülemeyebilir (Reed ve Marks 1999, 155). Bu çalışmada Categorical Cross Entropy yitim fonksiyonu kullanılmaktadır, nedeni ise; çalışmada 9 farklı ürün tipinin yani multi-class bir sınıflandırma yapılması ve sınıflandırmanın da Ürün_1, Ürün_2, ... kategorik veriler üzerinde gerçekleşmesidir. Denklem 4'de Categorical Cross Entropy yitim fonksiyonun matematiksel ifadesi verilmiştir; burada (y) veri kümesinde bulunan verilerin gerçek etiket değerlerinden oluşan bir vektör ve $(\hat{y})$ ise sınıflandırıcı çıkışındaki tahmin değerlerini göstermektedir, alt indis i ise i . elemanı, C ise örnek adedini ifade etmektedir.

$$Loss_{CrossEntropy}(y, \hat{y}) = - \sum_i^C y_i \log(\hat{y}_i) \quad (4)$$

2.3.2. Geri Yayılım Algoritması

Yitim miktarı hesaplandıktan sonra elde edilen değerlerin kaynağı ileri yayılım süreci ile belirlenen ağırlık değerleridir, öyle ise ağırlıkların yitim değerini düşürecek şekilde güncellenmesi gerekir, ancak çok katmalı bir yapı içerisinde yitimden hangi ağırlıklar ne ölçüde sorumlu olduğu sorusu ortaya çıkmaktadır. Güncellenecek ağırlık değerlerinin hesaplanması, yitim değerine göre çıkıştan girişe doğru türev zincir kuralı kullanılarak yapılır, böylece yitim değeri üzerinden geriye doğru gidilerek her ağırlığın yitim üzerindeki etkisi belirlenmiş olur. Bahsedilen bu süreç geri yayılım olarak tanımlanır. Geçmiş 1960'lı yıllarda Henry J. Kelly tarafından yapılan çalışmalara gitse de, geri yayılım terimi ve sinir ağlarında genel kullanımı D. E.

Rumelhart ve arkadaşları tarafından 1986 yılında yayınlanan “Learning representations by back-propagating errors” makalesi ile olmuştur.



Şekil 8. Geri Yayılım Yönü

Şekil 8’de verilen ileri beslemeli ağ üzerinde $\hat{y}$ tahmin değeri, L yitim miktarı, f_k aktivasyon fonksiyonu, s_k k . nöronun net girdisi ve w_{jk} ise k nöronu ile j nöronu arasındaki ağırlığı ifade etmek üzere; L yitim fonksiyonunun w_{jk} ağırlığına göre kısmi türevi Denklem 5’de verilen eşitlik ile hesaplanabilir. Bu yöntem izlenerek istenilen ağırlığa göre yitim fonksiyonunun gradyanı hesaplamak mümkündür.

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial f_k} \frac{\partial f_k}{\partial s_k} \frac{\partial s_k}{\partial w_{jk}} \quad (5)$$

Yitim fonksiyonunda minimuma ulaşmak için ağırlıkların ne kadar ve ne yönde değiştirileceğini belirleme işlemi ise optimizasyon olarak adlandırılır. Yapay sinir ağlarında optimizasyon amacı ile kullanılan çeşitli yöntemler mevcuttur. Uygulamada en çok kullanılan optimizasyon yöntemi ise Gradyan İniş (Gradient Descent) algoritmasıdır. Denklem 6’da Gradyan İniş algoritmasının matematiksel ifadesi verilmiştir; burada α öğrenme oranını ifade etmektedir, bir çeşit ince ayar parametresi gibi düşünülebilir. Hesaplanan gradyanın yeni ağırlık w_{jk+1} ’in değişiminde ne ölçüde etkili olmasını belirler, eğer çok küçük seçilirse eğitim işlemi uzun süre alır, büyük seçilirse salınım yaşanması yani fonksiyonun minimum değerine ulaşamaması riskini doğurur.

$$w_{jk+1} = w_{jk} - \alpha \frac{\partial L}{\partial w_{jk}} \quad (6)$$

Gradyan İniş Algoritmasının veri kümesine nasıl uygulanacağına bağlı olarak değişik türleri mevcuttur. Bunlar;

Toplu Gradyan İniş (Batch Gradient Descent): Veri kümesindeki tüm örnekler ağı girdikten sonra ağırlıklarda güncelleme işlemi yapılır.

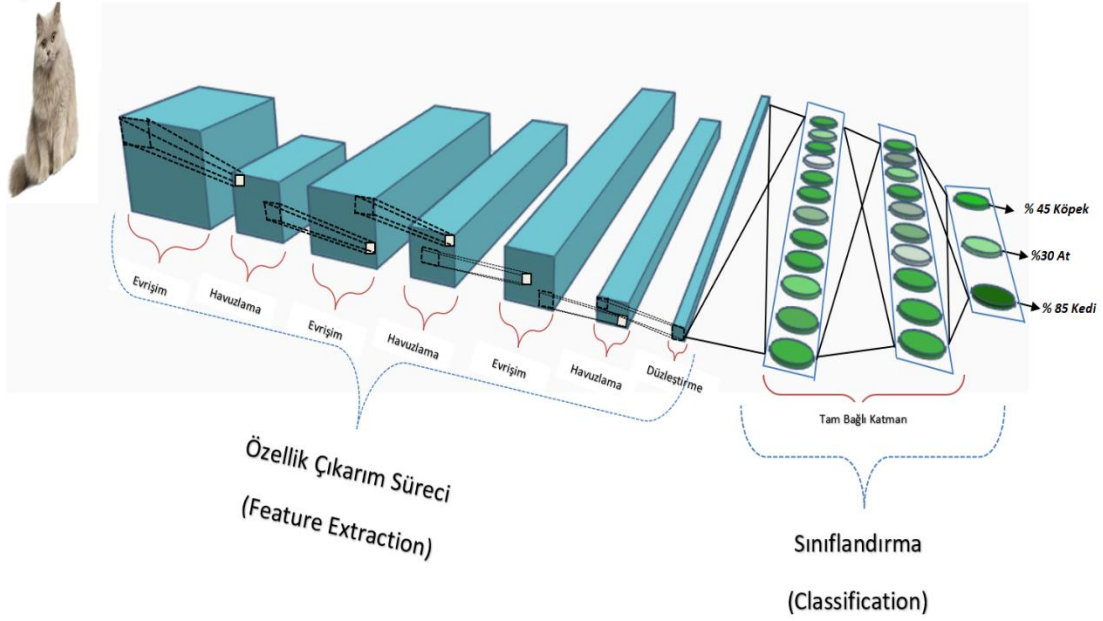
Olasılıksal Gradyan İniş (Stochastic Gradient Descent) : Ağı uygulanan her örnekten sonra güncelleme yapılır.

Mini-Toplu Gradyan İniş (Mini-Batch Gradient Descent): Veri kümesindeki örnekler paketler halinde ağı uygulanır ve her paketten sonra güncelleme yapılır.

Gradyan İniş Algoritmasında bazı koşullarda sorunlar ortaya çıkabilmektedir. Örneğin, yitim fonksiyonun değerinin lokal minimumda kalması varsa global minimuma ulaşamaması gibi; bunun yanında yitim fonksiyonunun değerinin minimuma daha salınımsız ve/veya daha hızlı ulaşılması istenebilir bu tarz ihtiyaçlara atfen Gradyan İniş'i esas alan ve yerine göre eksikliklerini gidermeye yönelik algoritmalar da mevcuttur. Bunlara Momentum, RMS-Prop, Adagrad, Adam, Nadam algoritmaları örnek verilebilir.

2.4. Evrişimsel Sinir Ağları (ESA)

Evrişimsel Sinir Ağları ağırlıklı olarak bilgisayar görüşü alanında görüntü sınıflandırma ve görüntü tanıma amacı ile kullanılan derin sinir ağlarının bir sınıfıdır, bunun yanında doğal dil işleme ve zaman serisi analizlerinde de kullanılabilmektedir. ESA'lar genel matris çarpımı yerine katmanlarının en az birinde evrişim işlemi yapılarak oluşturulan sinir ağlarıdır (Goodfellow, Bengio ve Courville 2016, 326). Bu çalışmada görüntü sınıflandırması yapıldığından dolayı evrişimsel sinir ağları ve onu oluşturan bileşenlerin izahı, görüntülerin sınıflandırılma işlemi esas alınarak yapılmıştır. Şekil 9'da temsili bir ESA yapısı verilmiştir.



Şekil 9. ESA yapısı

ESA'nın beyindeki görsel kortekste bulunan sinir hücresi organizasyonundan ilham alan bir yapısı vardır (Kim, ve diğerleri 2016). Hubel ve Wiesel hayvan beyinleri üzerinde yaptığı çalışmalarda görme işleminin görsel sistemde bulunan değişik yapılardaki hücrelerin kendi alıcı alanlarında bulunan uyarıcılara daha farklı tepki verdiklerini göstermiştir (Hubel ve Wiesel 1962, 106-154). Alıcı alanı, çeşitli uyaranlara karşı sinirsel tepkiler üretebilen duyuusal alanın bir parçasıdır (Alonso ve Chen 2009, 5393). Duyusal alan ise basitçe beynin duyu organlarından gelen verileri işleyen bölgeleridir. Görsel sistemdeki basit hücreler, kendi algılama alanları içinde bulunan belirli yönelimlerdeki düz çizgilere daha çok duyarlıdır, kompleks hücreler adı verilen hücreler ise belli yönde hareket eden yönelimlerin hareketli uyaranlarına duyarlıdır ve sabit uyaranlara yanıt vermez; hiperkompleks hücreler ise yönelim, hareket ve yöne duyarlıdır (Wikipedia, Hypercomplex cell 2017). Bu şekilde farklı seviyedeki hücrelerden elde edilen farklı bilgiler beynin karmaşık görüntü merkezilerine iletilerek görüntü oluşturulur (Gerrig ve Zimbardo 2012, 100).

Bilgisayar görüşü ya da görüntü işlemede görüntüler üzerinde işlem yaparken doğrudan görüntünün tümü ya da ham veri üzerinde işlem yapmak yerine görüntüyü temsil eden özellikler tespit edilir; bu özellik ve temsil değerleri üzerinden gerekli işlemler yapılır.

Özellik Tespiti (Feature Detection) uzmanlık gerektiren bir süreçtir, uygun özellik tespiti için uygun filtre tasarımı gerektirmektedir. Geliştirilen bir filtre o uygulama için iyi sonuç verse bile başka bir uygulamada çok genel veya çok özel kalabilmektedir. Ancak ESA ile filtreleri oluşturan değerler geri yayılım yolu ile ağ tarafından öğrenilebilmektedir. Bunun yanında evrişim işlemi ham verinin doğrudan ileri beslemeli sinir ağına uygulanması yerine, belli seviyelerde görüntüyü temsil eden özelliklerin elde edilmesi, ardından bu özelliklerin ileri beslemeli ağa girdi olarak verilmesi nedeni ile hesaplama karmaşıklığı ve kaynak kullanımı üzerinde de olumlu etkileri vardır.

Ham görüntünün doğrudan ileri beslemeli sinir ağına uygulanması durumunda zaman karmaşıklığı ve kaynak kullanımı üzerindeki etkiyi bir örnekle detaylandırmak gerekirse; (224×224) piksel boyutunda ve kırmızı, yeşil, mavi 3 kanadan oluşan bir fotoğraf için giriş katmanında $224 \times 224 \times 3 = 150528$ adet girişe ihtiyaç duyulur, bu fotoğraf çözünürlüğü günümüz görüntü sensörlerinin geldiği nokta dikkate alındığında mütevazı bir değerdir, girişlerin her biri 1000 nörona sahip 2 ara katmana ve ardından 10 nöronlu bir çıkış katmanına sahip tam bağlı yapay sinir ağı modeline uygulandığında hesaplanacak ağırlık değerleri: $150528 \times 1000 \times 1000 \times 10 = 1.5 \times 12^{12}$ seviyesine çıkmaktadır ki bu hesapta ağırlıklarla birlikte hesaplanması gereken yanlılık değeri ihmal edilmiştir. Ayrıca hesaplanan değerlerin bellekte saklanması gerektiğini de gözden kaçırmamak gerekir

2.4.1. Evrişim İşlemi

Matematikte evrişim işlemi; $f(x)$ ve $g(x)$ olmak üzere, iki fonksiyondan birinin diğeri üzerine nasıl etki ettiğini bulmak üzere gerçekleştirilen bir işlemdir. İşaret işlemede ise $h[n]$ gibi dürtü cevabı bilinen ayrık zamanlı doğrusal zamanla değişmeyen bir sistemin, verilen bir $x[n]$ giriş işaretine karşı sistemin üreteceği çıkış işaretinin tespit edilmesinde kullanılır. Denklem 7’de tek boyutlu ayrık zamanlı bir sistemin evrişim işlemi verilmiştir

$$x[n] * h[n] = \sum_{k=-\infty}^{k=\infty} x[k] h[n - k] \quad (7)$$

Sayısal görüntü işlemede fotoğraflar piksel adı verilen iki boyutlu dizinin elemanlarından oluşur, bu nedenle evrişim işlemi iki boyutta yapılması

gerekmektedir, temelde bu işlem Denklem 7’de verilen tek boyutlu evrişim işleminin boyut eklenmiş halidir. Bu çalışmada görüntü sınıflandırılacağı için iki boyutlu evrişim işlemine değinmek gerekmektedir. İki boyutlu evrişim işlemi Denklem 8’de verilmiştir, burada I giriş matrisi olmak üzere $[i, j]$ giriş matrisinin sırası ile [sütun,satır] konumu, m ve n değerleri ise F filtre matrisinin elemanlarının konumlarıdır.

$$(I * F)_{m,n} = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} I[i, j] F[m - i, n - j] \quad (8)$$

Şekil 10’da giriş ve filtre matrisi konumlandırma işlemleri gösterilmiştir. Yaygın olarak uygulamalarda filtrenin merkezi (0,0) pozisyonu kabul edilerek konumlandırma yapılır (Ahn 2018).

Giriş (I)

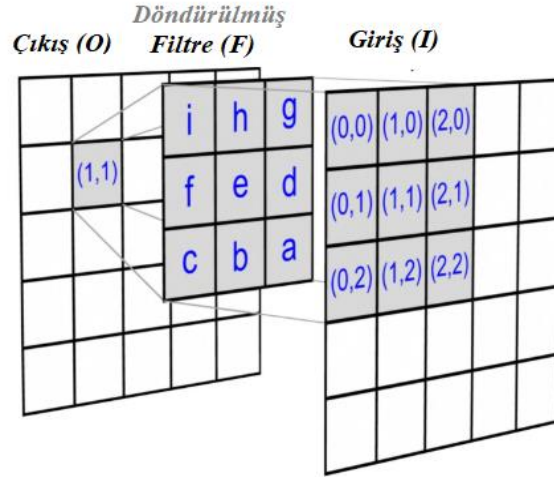
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

Filtre (F)

	m	-1	0	1
n	-1	a	b	c
	0	d	e	f
	1	g	h	i

Şekil 10. Giriş ve Filtre matrislerinin konumlandırılması

Örneğin, çıkış $O(1,1)$ noktasındaki değeri hesaplanmak istendiğinde filtrenin merkezi yani (0,0) konumundaki “e” elemanı, giriş matrisinde $I(1,1)$ elemanının üzerine gelecek şekilde konumlandırılır ve hesaplamalar buna göre yapılır. Bu gösterimde konum [sütun,satır] referansı ile yerleştirilmiştir.



Şekil 11. 2 Boyutlu evrişim işleminin görsel bir temsili

Kaynak: Ahn, Song Ho, 2018, "Convolution",

<http://www.songho.ca/dsp/convolution/convolution.html> , Erişildi: 30/11/2020

Şekil 10'da dikkat edilirse başlangıçta yani işlem öncesinde filtre matrisinin $(-1, -1)$ konumunda "a" elemanı varken evrişim işlemi ile birlikte Şekil 11'de gösterildiği gibi bu konuma "i" elemanı gelmiştir. Görüleceği üzere giriş matrisindeki değerler filtre matrisindeki değerlerle çarpılmadan önce filtre matrisinin yatayda ve dikeyde döndürülmesi işlemini gerçekleştirmektedir.

ESA uygulama pratiklerinde filtre döndürülmez, esasında filtre döndürme işlemi yapılmadığında yapılan işlem Denklem 9'da verilen çapraz doğrulama işlemine karşılık gelir.

$$(I \otimes F)_{m,n} = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} I[i, j] F[m + i, n + j] \quad (9)$$

Denklem 9 incelendiğinde evrişim işleminden farklı olarak filtrenin yatayda ve dikeyde döndürülmeden işleme sokulduğu görülür.

Evrişimsel sinir ağlarında filtre değerleri ağ tarafından öğrenildiği için filtre döndürme işlemi olsun ya da olmasın eğer filtreler uygun bir şekilde eğitilirse filtreleri oluşturan değerlerde aynı sonuca ulaşılmış olur. Çapraz doğrulama işlemi, evrişim işlemine göre daha anlaşılır ve kolay uygulanabilirliği nedeni ile tercih edilir. Ancak, bir terminolojik mutabakat olarak çapraz doğrulama işlemi yapılmasına rağmen uygulamada evrişim terimi kullanılır (Bai 2019). Literatüre uygunluk açısından bu çalışmada da evrişim terimi kullanılacaktır.

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2		

Kaydırma 1

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	

Kaydırma 2

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	

Kaydırma 3

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-1	4	-1

Kaydırma 4

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-5		

Kaydırma 5

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-5	4	

Kaydırma 6

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-5	4	-5

Kaydırma 7

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-5	4	-5
2		

Kaydırma 8

Giriş (I)

1	0	0	0	1
2	1	0	1	2
2	0	1	0	2
2	1	0	1	2
1	0	0	0	1

Filtre (F)

0	-1	0
-1	4	-1
0	-1	0

Çıkış(O)

2	-3	2
-5	4	-5
2	-3	

Kaydırma 9

Şekil 12. Evrişimsel sinir ağlarında evrişim işlemi

Evrişim işleminin görsel olarak açıklamak gerekirse; Şekil 12’de verilen bir 5×5 boyutundaki I giriş matrisi ve 3×3 boyutundaki F filtre matrisi olsun. Filtre, her adımda belirlenen kaydırma miktarı kadar kaydırılır, bu örnekte kaydırma adımı 1’dir. Her kaydırma adımında, giriş matrisindeki her eleman, filtre matrisinde kendi üstüne gelen eleman ile çarpılır; ardından bu değerler toplanarak elde edilen değer, çıkış matrisinin ilgili konumuna yazılır.

2.4.2. Dolgulama (Padding)

Derin evrişimsel sinir ağlarında ardışık olarak pek çok evrişim katmanı olduğu düşünülürse; bu işlemler sonucunda girişteki görüntünün önemli ölçüde küçülüp daralması anlamına gelir. Ayrıca kenarlarda kalan pikseller, görüntünün merkezine yakın olan piksellere oranla daha az evrişim işlemine tabi tutulması dolayısı ile kenarlardaki bilgilerde kayıp yaşanmasına neden olur, bu nedenle giriş görüntüsü evrişim işlemine tabi tutulmadan önce dolgulama adı verilen işlem uygulanır.

Bir $i \times i$ boyutundaki giriş matrisi, $f \times f$ boyutundaki bir filtre ile $s = 1$ (birim) kaydırma değerinde evrişim işlemine tabi tutulduğunda, çıkış matrisi "o": $o = (i - f + 1 \times i - f + 1)$ boyutunda olur.

Şekil 11'de gösterildiği gibi 5×5 boyutundaki (I) giriş matrisi, 3×3 boyutunda (F) filtresi ile evrişim işlemine tabi tutulduğunda, çıkış (O) matrisinin boyutu 3×3 olduğu görülmektedir.

2.4.2.1.Dolgulama Seçenekleri

Geçerli (Valid) Dolgulama: Bu uygulamada aslında dolgulama işleminin yapılmadığını ifade eder, çıkış matrisinin daralacağı anlaşılır.

$$s = 1 \text{ için çıkış boyutu: } o = (i - f + 1 \times i - f + 1) \quad (10)$$

$$s > 1 \text{ için çıkış boyutu: } o = \left\lfloor \frac{i - f}{s} \right\rfloor + 1 \quad (11)$$

Yarı Dolgulama (Same Padding): Giriş matrisi ile çıkış matrisinin boyutunun aynı olduğu dolgulama uygulamasıdır.

p dolgulama miktarı, ($f = 2n + 1, f \in N$) ve $s = 1$ olmak üzere çıkış boyutu;

$$o = (i - f + 1 + 2p \times i - f + 1 + 2p) \quad (12)$$

Çıkış matrisi giriş matrisine eşit olsun istendiğinden Denklem 11'de "o" çıkış yerine, "i" yazılabilir; bu durumda dolgulama miktarı "p", Denklem 12 ile bulunabilir.

$$i - f + 1 + 2p = i \quad (13)$$

$$p = \frac{f-1}{2} \quad (14)$$

Kaydırma miktarı $s > 1$ için çıkış boyutu "o" Denklem 15'de verilen ifade ile bulunur.

$$o = \left\lfloor \frac{i + 2p - f}{s} \right\rfloor + 1 \quad (15)$$

Tam dolgulama: Bazı uygulamalarda çıkış matrisinin boyutunun artması istenebilir bu durumda tercih edilen bir uygulamadır (Dumoulin ve Visin 2016).

Dolgulama miktarı p , $p = f - 1$ ve kaydırma miktarı s , $s = 1$ olmak üzere çıkış boyutu Denklem 16 ile hesaplanabilir.

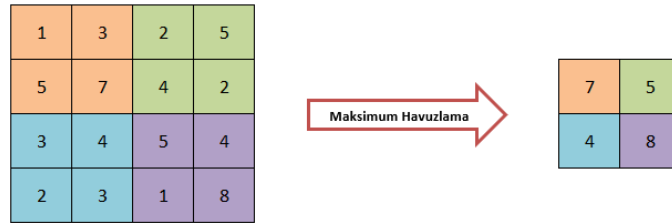
$$o = i + (f - 1) \quad (16)$$

2.4.3. Havuzlama (Pooling) Katmanı

Bu katmanın amacı evrişim işleminden sonra ortaya çıkan matrisin boyutunu azaltmak ve dolayısı ile hesaplama hızını artırmaktır. Bunun yanı sıra havuzlama katmanının aşırı öğrenmeyi engelleyen bir özelliği vardır (İnik ve Ülker 2017). Yaygın olarak kullanılan iki çeşidi mevcuttur; bunlar, maksimum ve ortama havuzlama yöntemleridir. Her iki yöntemde de havuzlama yapılacak matris üzerinde belirlenen boyutlarda bir filtre, belirli kaydırma adımlarında gezdirilir. Maksimum havuzlama Evrişimsel Sinir Ağlarında en çok tercih edilen yöntemdir (Suárez-Paniagua ve Segura-Bedmar 2018). Denklem 17’de havuzlama sonrası elde edilecek çıkış matrisinin boyutları verilmiştir.

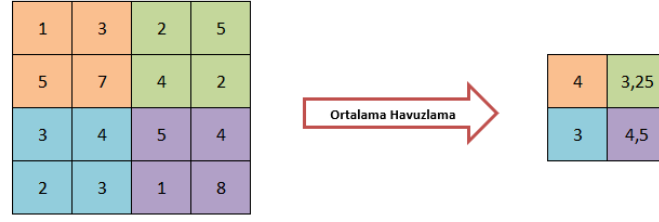
$$o = \left\lfloor \frac{i - f}{s} \right\rfloor + 1 \quad (17)$$

Maksimum havuzlamada filtrenin denk düştüğü alandaki hücreler içinden, değeri en büyük olan alınarak çıkışa aktarılır;



Şekil 13. Maksimum havuzlama

Ortalama havuzlamada ise filtrenin denk düştüğü alandaki değerlerin aritmetik ortalaması alınır ve bu değer çıkışa aktarılır.



Şekil 14. Ortalama havuzlama

2.4.4. Düzleştirme (Flattening)

Havuzlama işleminin ardından elde edilen matris artık tam bağlı katmana uygulanması gerekir, ancak tam bağlı katmanın girişi tek boyutlu bir dizi şeklindedir bu nedenle havuzlama işleminin ardından elde edilen matris tek boyutlu dizi haline getirildiği kısımdır.



Şekil 15. Düzleştirme

2.4.5. Tam Bağlı Katman

ESA’larda tam bağlı katman, öncesindeki aşamalarda gerçekleşen özellik çıkarımından sonra sınıflandırmanın gerçekleştirildiği katmandır. Temelde tam bağlı katman, bir ileri beslemeli yapay sinir ağıdır. Tam bağlı katman, bir aktivasyon ünitesinin çıkışının kendinden sonra gelen katmandaki aktivasyon ünitelerinin tamamının girişine uygulanmasından dolayı bu şekilde adlandırılır. Ağın son katmanında bir Softmax aktivasyon işlemi gerçekleşir ve ağ tarafından tahmin edilen sınıfın olasılıksal değeri ortaya çıkar bu değer üzerinden ağın tahmin sonucu elde edilir.

2.5. Tensorflow

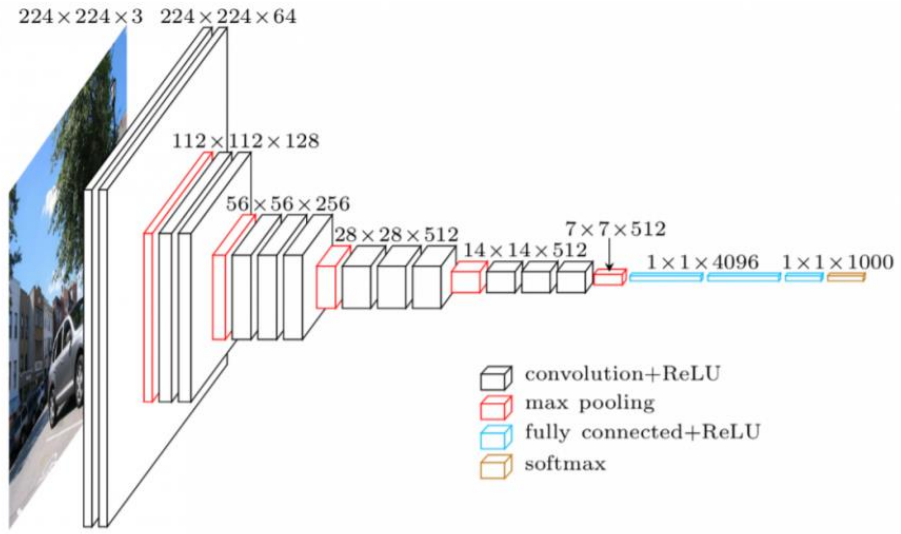
Tensorflow bir makine öğrenmesi kütüphanesidir. Uçtan uca, açık kaynak bir platformdur. Derin öğrenme ve yapay zeka üzerinde çalışmalar yapan “Google Brain” ekibi tarafından geliştirilmiştir. Çoklu CPU, GPU ve TPU kullanımını destekleyerek modelin dağıtılmasını imkan sağlar (Tensorflow 2020) . Ayrıca Web, mobil ve masaüstü uygulamalarına yönelik paketleri de mevcuttur.

2.6. Keras

Bir derin öğrenme Uygulama Programlama Arayüzüdür. Python dilini kullanmaktadır. Tensorflow, Microsoft Cognitive Toolkit, R ve Theano kütüphaneleri ile birlikte çalışabilir. Hızlı deneyimleme, kullanıcı dostu olma, modüler yapısı ve genişletilebilirliği ile ön plana çıkmaktadır (Keras 2020). Geliştiricisi ve devam ettiricisi olan François Chollet’e göre başlangıçta bağımsız bir makine öğrenmesi çerçevesinden ziyade bir arayüz olarak tasarlanmıştır, yüksek seviyeli yapısı ile daha sezgisel bir soyutlama katmanı sağlar. Bu çalışmada kullanılacak ESA modeli olan VGG-16 Keras yardımı ile oluşturulmuştur, Keras ise arka tarafta modelin oluşturulmasında Tensorflow’dan yararlanmaktadır. Ayrıca Keras yaygın olarak kullanılan ve kendini kanıtlamış VGG-16, VGG-19, ResNet50, InceptionV3, MobileNet, InceptionResNetV2 gibi modellerin yine yaygın kullanılan veri kümeleri ile ön eğitimi yapılmış modellerini içinde barındıran uygulamaları mevcuttur ve ortaya çıkan yeni modeller zaman içinde eklenmektedir.

2.7. VGGNet

Nesne sınıflandırma amacı ile Oxford Üniversitesi’ndeki Visual Geometry Group tarafından geliştirilmiştir. 2014 yılı ImageNet, ILSVRC2014 yarışmasında lokalizasyon kategorisinde birinci, sınıflandırma kategorisinde ise GoogLeNet’in ardından ikinci olmuştur (ImageNet 2014). Bu modelde, evrişim katmanlarında görece daha küçük boyutlu filtrelerin (3x3 gibi) kullanılması, derinliğin ise artırılması (16- 19 katman gibi) esasına dayanır (Simonyan ve Zisserman 2014). Şekil 16’da VGG-16 Modelinin yapısı verilmiştir.



Şekil 16. VGG-16 mimari yapısı

Kaynak: Loukadakis, Cano ve O’Boyle(2018), “*Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures*”, 11th International Workshop on Programmability and Architectures for Heterogeneous Multicores (MULTIPROG-2018).

3. DERİN ÖĞRENME İLE İÇECEK ŞİŞE KAPAĞI ÜZERİNDEKİ GÖRSELLERDEN ÜRÜN TİPİ TANIMA

Bu bölümde, içecek şişe kapaklarının derin öğrenme ile tanınmasına yönelik önerilen uygulama tanıtılmıştır. Bu uygulama ile şişe üzerine uygulanan kapağın evrişimsel sinir ağları ile tipinin tanınması, şişe içine doldurulan ürüne ait olmayan bir kapak uygulanması durumunda yani ürün ile uygulanan kapağın eşleşmemesi durumunda bir uyarı işareti üretilmesi sağlanacaktır.

Bu amaçla, ürünlere ait kapakların fotoğrafları çekilerek eğitim için ihtiyaç duyulan veri kümesi oluşturulmuştur. Ardından önerilen çözüm için ImageNet veri kümesi ile ön eğitimi yapılan VGG-16 modeli oluşturulmuş, son katmanı ise bu çalışma için oluşturulan şişe kapağı veri kümesi ile yeniden eğilmiş, müteakiben ağı eğitimi ve doğrulama doğruluk oranları verilmiştir.

Ayrıca bu çalışmada önerilen çözümün bir prototipi oluşturularak, çalışması hakkında bilgi verilmiştir.

3.1. Veri Kümesinin Hazırlanması

Modelin eğitiminin ve test işlemlerinin yapılması için 9 farklı ürün kapağına yani 9 sınıfa ait 7200 adet fotoğraftan oluşan eğitim ve 1800 fotoğraftan oluşan doğrulama veri kümesi oluşturulmuştur. Tüm örnekler dikkate alındığında fotoğrafların %80'i eğitim, %20'si doğrulama veri kümesinde olacak şekilde ayrılmıştır.

Fotoğraflar 3 farklı ışık kaynağı altında elde edilmiştir, bunlar sırası ile günışığı, sarı ışık (3000 Kelvin) ve beyaz ışık (6000 Kelvin) şeklindedir. Veri kümesindeki fotoğraflar değişik oryantasyonlarda ve net bir şekilde elde edilen fotoğraflardan oluşturulmuştur. Fotoğraflar VGG-16 modelinin giriş boyutu olan 224×224 piksel boyutlarında boyutlandırılmıştır.

Veri kümesinde bulunan ürünler Tablo1'de verilmiştir. Kapak üzerinde bulunan görsel tasarımlar ve bu ürünlere ait ticari kullanım hakları ürünlerin üretici firmalarına aittir, bu çalışmada ticari amaç taşımayan akademik bir faaliyet için kullanılmıştır. Ürünlerin üretici bilgileri Ekler kısmında verilmiştir.

Tablo 1. Oluřturulan Veri Kumesi Detayları

Kapak	Ürün Tanımı	Eğitim Veri Kumesi Örnek Adedi	Doğrulama Veri Kumesi Örnek Adedi
	Ürün_1	800	200
	Ürün_2	800	200
	Ürün_3	800	200
	Ürün_4	800	200
	Ürün_5	800	200
	Ürün_6	800	200
	Ürün_7	800	200
	Ürün_8	800	200
	Ürün_9	800	200
		Toplam=7200	Toplam=1800

3.2.Modelin Oluşturulması

Bu bölümde Keras kütüphanesi kullanılarak VGG16 modeli oluşturulmuştur, modelin “dense_1” katmanı öncesindeki katmanlar ImageNet veri kümesi ile eğitilmiş hali üzerinden transfer öğrenesi yapılmıştır, bu nedenle bu katmanlar yeniden eğitilmemesi için dondurulmuştur. Klasik olarak VGG-16 modelinin son katmanı yani tahmin işleminin gerçekleştiği katman önce modelden çıkarılmış ardından eğitilebilir bir katman olarak modele yeniden eklenmiştir. Oluşturulan modelin katmanları ve özellikleri Tablo 2’de gösterilmiştir.

Tablo 2. Oluşturulan Modelinin Mimari Yapısı

Katman (Tür)	Çıktı boyutu	Parametre
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160

block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
dense_1 (Dense)	(None, 9)	36873

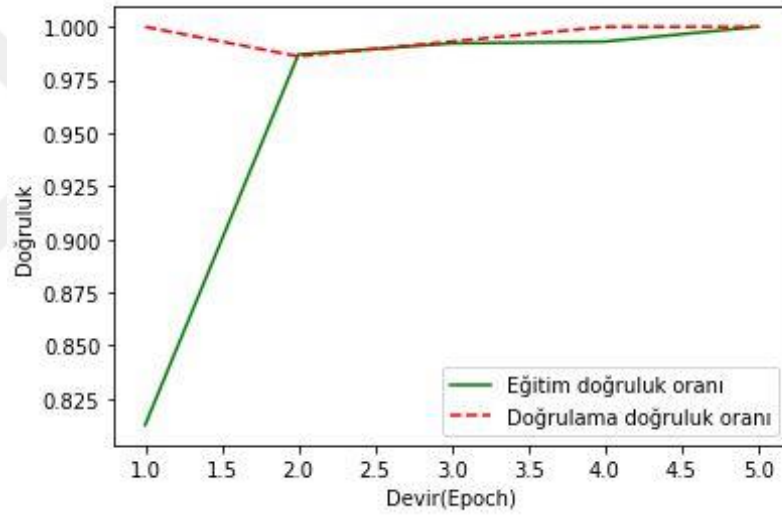
3.3.Üst Değişkenlerin Ayarlanması (Hyperparameter Tuning)

Bu çalışmada ürün tipini tespit etmeye yönelik kullanılan VGG-16 ESA modeli, Tensorflow+Keras yapısı yardımı ile oluşturulması ve bu modelin Imagenet veri kümesi ile eğitilmiş hali üzerinden transfer öğrenmesi yapılmasından dolayı mevcut üst değişkenlere müdahale edilmemiştir. Yalnızca modele eklenen ve eğitilebilir olarak ayarlanan “dens_1” katmanının eğitimi için yitim fonksiyonu olarak kategorik sınıflandırma yapıldığından Catagorical Cross Entropy yitim fonksiyonu kullanılmıştır. Optimizasyon algoritması olarak RMS-Prop tercih edilmiştir, bunun nedeni ise yapısı itibarı ile RMS-Prop fonksiyonunun, Gradyan İniş veya Olasılıksal Gradyan İniş fonksiyonlarına göre minimuma daha hızlı yakınsaması nedeniyledir. Son katmanda aktivasyon fonksiyonu olarak Softmax aktivasyon fonksiyonu kullanılmıştır, Softmax aktivasyon fonksiyonu girdi olarak verilen fotoğrafın hangi

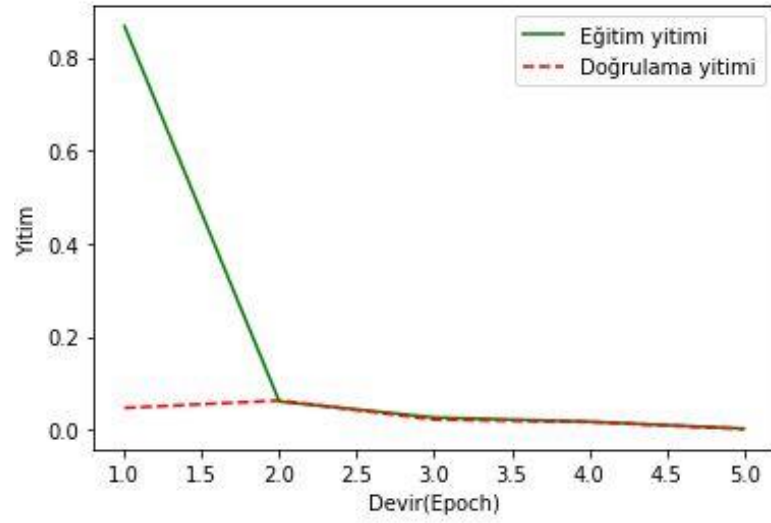
sınıfa ait olduğu tahminini 0 ile 1 aralığında bir olasılıksal değer olarak döndürür. Modelin toplam 138,357,544 adet parametresi bulunmaktadır. Bu parametrelerin 36873 adedi eğitilebilir olarak belirlenmiştir ve Intel Core i7-6600U CPU ve 16 Gigabyte RAM'a sahip bir bilgisayar üzerinde eğitilmiştir. Eğitim esnasında tüm eğitim veri kümesi 32'lik gruplar (batch) halinde modele uygulanmış ve bu 5 devir (epoch) halinde gerçekleştirilmiştir. Devir ve batch boyutu deneme yanılma yöntemi ile en iyi performansı veren değerler gözlemlenerek tespit edilmiştir.

3.4. Eğitim ve Test Sonuçlarının Değerlendirilmesi

Eğitim işlemi sonrasında elde edilen eğitim ve doğrulama doğruluk oranları grafiği Şekil 17'de, eğitim ve doğrulama yitim değerleri grafiği ise Şekil 18'de verilmiştir.

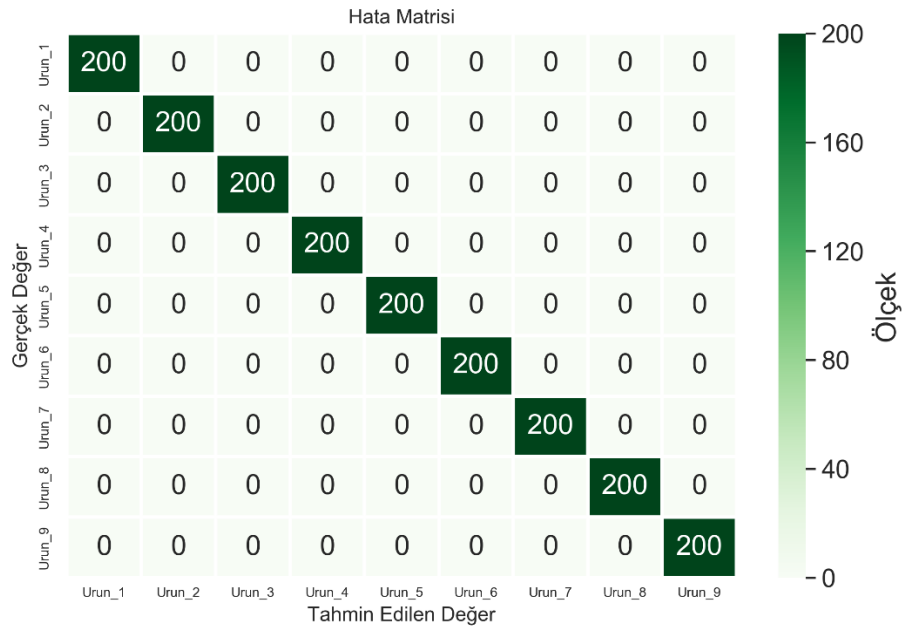


Şekil 17. Doğruluk oranları



Şekil 18. Yitim miktarları

Doğruluk grafiğinden de görüleceği üzere model daha 2. Devirden itibaren %98 oranında kapakları doğru sınıflandırmaktadır, bu oran belirlenen son devir olan 5. Devirde %99,99 oranına ulaşmaktadır ki bu sonucun gayet başarılı olduğunu yani modelin uygulanan her bir örneği doğru sınıflandırdığını göstermektedir. Ayrıca test amacı ile 9 sınıfa ait 1800 adet fotoğraftan oluşan örnekler modele uygulanmış ve modelin bu örnekleri de %100 doğru sınıflandığı tespit edilmiştir. Test sonuçları Şekil 19’da verilen hata matrisinde gösterilmiştir.



Şekil 19. Hata matrisi

Hata matrisine göre elde edilen ölçüm metrikleri ise Tablo 3’de verilmiştir.

Tablo 3. Hata matrisinden elde edilen ölçüm metrikleri

	Hassasiyet	Geri çağırma	f1-skoru	destek
Urun_1	1.00	1.00	1.00	200
Urun_2	1.00	1.00	1.00	200
Urun_3	1.00	1.00	1.00	200
Urun_4	1.00	1.00	1.00	200
Urun_5	1.00	1.00	1.00	200
Urun_6	1.00	1.00	1.00	200
Urun_7	1.00	1.00	1.00	200
Urun_8	1.00	1.00	1.00	200
Urun_9	1.00	1.00	1.00	200
doğruluk			1.00	1800
macro avg	1.00	1.00	1.00	1800
weighted avg	1.00	1.00	1.00	1800

3.5. Çalışmanın Prototiplenmesi

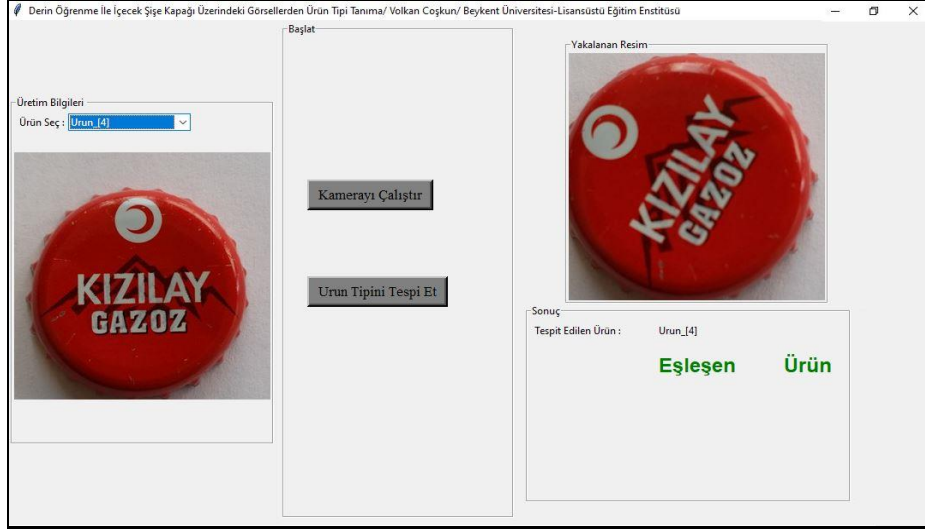
Bu çalışmada önerilen, derin öğrenme ile içecek şişe kapağı üzerindeki görsellerden ürün tipi tanıma işlemine yönelik olarak bir prototip uygulaması yapılmıştır. Oluşturulan ESA yapısı ve eğitim işlemi sonucunda elde edilen ağırlıklar, sırası ile .json ve .h5 dosyaları şeklinde kaydedilmiştir. Böylece ürün tipini tanıma işlemi esnasında her defasında ağın yeniden oluşturulması ve eğitilmesi gerekmemektedir.

Şekil 20’de prototip program ve kapakların üzerindeki görsellerin elde edildiği kamera düzeneği gösterilmektedir. Kamera olarak 2 Megapiksel FullHD Web kamerası kullanılmıştır, lens odağını el ile ayarlamak mümkündür.



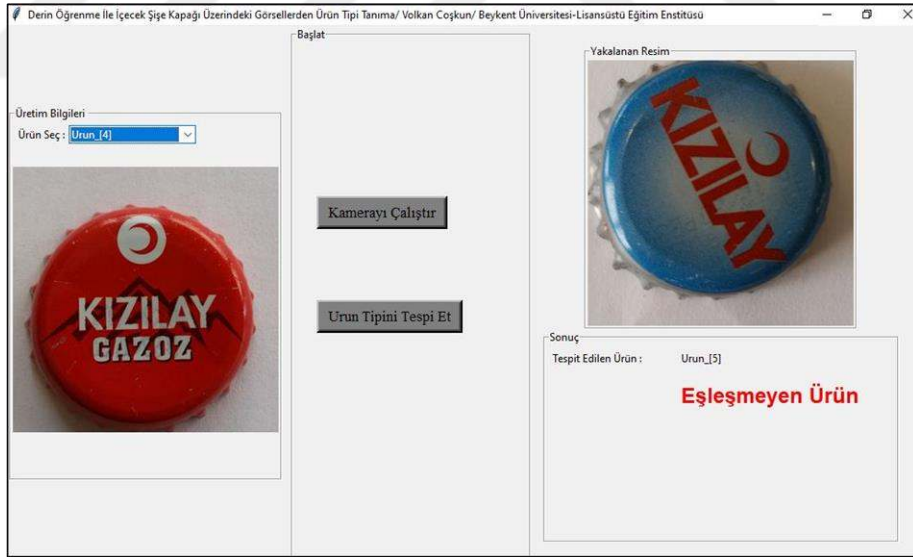
Şekil 20. Kamera düzeneği

Program, bir içecek üretim hattı dikkate alınarak tasarlanmıştır. Öncelikle program arayüzünden “Üretim Bilgileri” kısmından o an gerçekte üretimi yapılacak olan ürün seçimi yapılır, ardından kamera altına ürün tipi tanınacak kapak yerleştirilir, bu gerçek üretim hattında kapak uygulayıcı makinanın henüz uyguladığı kapak olduğu varsayılır ve bu kapağın fotoğrafı çekilerek “Yakalanan Resim” kısmında gösterilir, ardından “Ürün Tipi Tespit Et” butonun basılır bu işlemle birlikte yakalanan fotoğraf ürün tipi tespit edilmek üzere modele uygulanır. Şekil 21’de gösterildiği gibi eğer seçilen ürün ile tespit edilen/sınıflandırılan ürün eşleşirse program “Eşleşen Ürün” mesajı verir.



Şekil 21. Eşleşen ürün

Şayet Şekil 22’de olduğu gibi seçilen ürün tipi ile tespit edilen ürün eşleşmiyorsa “Eşleşmeyen Ürün” mesajı verecektir ki bu durumda yine gerçek üretim hattı düşünüldüğünde eşleşmeyen ürünün ayıklanması ve doğru kapağın uygulanması sonucunu ortaya çıkarır.



Şekil 22. Eşleşmeyen ürün

SONUÇ

Bu tez çalışmasında, endüstriyel içecek üretim hatlarında şişe üzerine uygulanan kapak üzerindeki görselin derin öğrenme ile hangi ürün tipine ait olduğu tespit edilmiş, buradan hareketle uygulanan kapağın o an üretimi yapılan ürüne ait olup olmadığı saptanmıştır. Şayet üretilen ürün ile şişe üzerine uygulanan kapağın ürün tipi eşleşmiyorsa, ürüne ait doğru kapağın uygulanması için bir uyarı mesajı verilmesi sağlanmıştır.

Kapak üzerindeki görsellerin sınıflandırılmasında kullanılan VGG-16 modelinin, ImageNet veri kümesi gibi yaklaşık 14 milyon fotoğraftan oluşan geniş bir veri kümesi ile eğitilmiş hali üzerinden eğitim aktarması yapıldığı için modele eklenen son tam bağlı katmanın eğitim sürecinde daha 2. Devirde ağırlıkların %98'lere ulaşan eğitim doğruluk oranı, belirlenen son devir olan 5. Devir ile %99.99 doğruluk oranına ulaştığı görülmüştür. Elde edilen bu doğruluk oranı üzerinde, çalışma için oluşturulan kapak veri kümesi içindeki kapak fotoğraflarının net olarak elde edilen görüntülerden oluşmasının ve her ürün tipi için tek çeşit kapak tasarımının kullanılmasının da olumlu etkileri vardır. Sonucu, bu çalışmada verilen uygulama ve saha uygulamaları özelinde değerlendirmek gerekirse; şişe üzerine uygulanacak kapaklar yeni üretilmiş, üzerinde deformasyon olmayan kapaklardan oluşmaktadır, bu nedenle veri kümesindeki örnekler yeni üretilmiş üzerinde deformasyon olmayan kapaklardan toplanmıştır. Çalışmanın, üzerindeki görsel yıpranmış ve deformasyona uğramış kapaklar ile yapılması, doğruluk oranını azaltıcı etki gösterecektir, ancak bu uygulamada şişe üzerine uygulanacak kapaklar doğal olarak yeni üretilmiş olması nedeni ile deformasyon olan veya eskimiş kapakların sınıflandırılması başka bir uygulama konusu olduğu varsayılarak bu durumdaki kapaklar kapsam dışı bırakılmıştır, bu durumdaki kapakların kapsam dışı bırakılmasının da doğruluk oranı üzerinde olumlu etki gösterdiği düşünülmektedir.

Kapak üzerindeki görselden ürün tipini tespit etme işleminin gecikme süresi modelin üzerinde çalıştırıldığı bilgisayarın hesaplama gücüne bağlı olmakla birlikte 260 milisaniye olarak ölçülmüştür. Modelin bu konfigürasyona sahip bir bilgisayar üzerinde çalıştırılması durumunda 260 milisaniyeden daha hızlı kapak uygulayan makine için hızı yetersiz kalmaktadır, bu hali ile gecikme süresi güncel saha

uygulamaları dikkate alındığında kabul edilebilir değerlerden uzaktır, ancak bu değer tespit algoritmasının daha yüksek işlem gücüne sahip bir bilgisayar veya dağıtık bir yapı üzerinde çalıştırılması durumunda gecikme süresinin azalması, dolayısı ile hızlı kapak kapatma makinelerine uygulanması mümkün olabilecektir. Ayrıca kapak üzerindeki görselin sınıflandırılması için bu çalışmada kullanılan VGG-16 modelinden daha az sayıda parametre bulunan bir modelin kullanılması, tespit gecikmesinin azalması yönünde bir etkiye bulunabilir.

Çalışmanın uygulama kısmında, ürün üzerindeki görsellerin fotoğrafının çekilmesi esnasında kameranın ortam ışığından etkilenmesi nedeni ile yetersiz veya aşırı ışık alan fotoğraflar çekilmesi mümkündür, bu durumda sınıflandırıcıya uygulanacak net bir fotoğraf olmadığı için sınıflandırıcı yanlış bir tahminde bulunabilir. Bu etkinin önüne geçmek için kamera ortam koşullarından ve ortam ışığındaki değişikliklerden etkilenmeyecek şekilde sabit ışık kaynağı içeren kapalı bir muhafaza içine alınması durumunda ışık şiddeti daha stabil hale gelecek ve daha net fotoğraflar alınmasını sağlayacaktır.

Ayrıca daha ileri bir kontrol için önerilen yönteme ek olarak, üretilen ürünün kapağının yanı sıra ürüne uygulanan etiketin de tanınması durumunda ikili bir mekanizma ile kontrol noktasının artırılarak hem uygulanan etiketin doğruluğu hem de sistemlerin birbirleri üzerindeki kontrolü sağlanabilir.

KAYNAKÇA

- Aggarwal, Charu C. . 2015. «Data Classification Algorithms And Applicaitons.» 209. CRC Press.
- Ahn, Song Ho . 2018. «Convolution.» Erişildi: 11 2020, 30.
<http://www.songho.ca/dsp/convolution/convolution.html>.
- Akbaş, Emre. 2018. *Derin Yapay Sinir Ağları ve Derin Öğrenme 'ye Kısa Bir Giriş*. BMO Semineri. Erişildi: 11 27, 2020.
https://www.emo.org.tr/ekler/b6d1307298e4886_ek.pdf?tipi=2&turu=X&sube=14
- Alonso, Jose-Manuel , ve Yao Chen. 2009. «Receptive field.» *Scholarpedia* 4 (1): 5393. http://www.scholarpedia.org/article/Receptive_field.
- Alpaydın, Ethem . 2010. «Introduction to Machine Learning.» 110. London: The MIT Press.
- Bai, Kunlun. 2019. «A Comprehensive Introduction to Different Types of Convolutions in Deep Learning.» Erişildi: 11 30, 2020.
<https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>.
- Deng, Li, ve Dong Yu. 2014. «Deep Learning: Methods and Applications.» *Foundations and Trends in Signal Processing*, 30 06: 199-200.
- Dumoulin, Vincent, ve Francesco Visin. 2016. «A guide to convolution arithmetic for deep.»
- Garrido, Juan José , Fanny Fernandes, Anissa Moussif, Marie-Pierre Fache, Pierre Giraud, ve Bénédicte Dargent. 2003. «Dynamic compartmentalization of the voltage-gated sodium channels in axons.» *Biology of the Cell* 437-445.
- Gerrig, Richard J., ve Philip G. Zimbardo. 2012. *Psikoloji ve Yaşam*. Çeviren Gamze Sart. NOBEL AKADEMİK YAYINCILIK.
- Goodfellow, Ian, Yoshua Bengio, ve Aaron Courville. 2016. «Deep Learning.» 326. MIT Press.

- Hemmings, Hugh C, ve Talmage D Egan. 2012. *Pharmacology and Physiology for Anesthesia*. Saunders.
- Hubel , D. H., ve T. N. Wiesel. 1962. «Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.» *The Journal of physiology* 160 (1): 106-154.
- ImageNet. 2014. «Results of ILSVRC2014.» Eriřildi: 12 2020, 02.
<http://www.image-net.org/challenges/LSVRC/2014/results#clsloc>.
- İnik, Özkan , ve Erkan Ülker. 2017. «Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme.» *GAZİOSMANPAŞA BİLİMSEL ARAŞTIRMA DERGİSİ* 6 (3): 85-104.
- Keras. 2020. «About Keras.» Eriřildi: 11 2020, 02. <https://keras.io/about/>.
- Keskenler, Mustafa Furkan, ve Eyüp Fahri Keskenler. 2017. «Geçmişten Günümüze Yapay Sinir Ağları ve Tarihçesi.» *Takvim-i Vekayi*, 29 12: 8-18.
- Kim, Jonghong, O. Sangjun, Yoonnyun Kim, ve Minho Lee. 2016. «Convolutional Neural Network with Biologically Inspired Retinal Structure.» *Procedia Computer Science* 88: 145-154.
- Krizhevsky, Alex , Ilya Sutskever, ve Geoffrey E. Hinton. 2012. «ImageNet Classification with Deep Convolutional.» *Advances in Neural Information Processing Systems* 25.
- Loukadakis, Manolis, Jos'e Cano, ve Michael O'Boyle. 2018. «Accelerating Deep Neural Networks on Low.» *11th International Workshop on Programmability and Architectures for Heterogeneous Multicores (MULTIPROG-2018)*.
- Murugan, Pushparaja . 2017. «Feed Forward and Backward Run in Deep Convolution Neural Network.» *Computer Vision and Pattern Recognition (cs.CV)*. 1-20.
- Nilsson, Nils J. . 2009. «A HISTORY OF IDEAS AND ACHIEVEMENTS.» 13.
- Reed, Russell, ve Robert J Marks. 1999. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Bradford Book.
- Simonyan, Karen, ve Andrew Zisserman. 2014. «VERY DEEP CONVOLUTIONAL FOR LARGE-SCALE IMAGE IMAGE RECOGNITION.» *ICLR*.

- Sivanandam, S N, S Sumathi, ve S N Deepa. 2008. «Introduction To Neural Networks Using Matlab 6.0.» 11. New Delhi: Tata McGraw-Hill Publishing.
- Suárez-Paniagua, V., ve I. Segura-Bedmar. 2018. «Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction.» *BMC Bioinformatics* 19 (209).
- Tensorflow. 2020. «Distributed training with TensorFlow.» Erişildi: 12 2020, 02. Tens https://www.tensorflow.org/guide/distributed_training.
- Wikipedia, The Free Encyclopedia. 2020. *Artificial neuron*. Erişildi: 11 27, 2020. https://en.wikipedia.org/wiki/Artificial_neuron.
- Wikipedia, The Free Encyclopedia. 2017. *Hypercomplex cell*. Erişildi: 11 20, 2020. https://en.wikipedia.org/wiki/Hypercomplex_cell.
- Winston, Patrick Henry. 1992. «Artificial Intelligence.» 445-446. Addison Wesley.
- Yamashita, Rikiya , Mizuho Nishio, Richard Kinh Gian Do, ve Tagoshi Kaori. 2018. «Convolutional neural networks: an overview and application in radiology.» *Insights into Imaging* 611–629.

EKLER

EK-1

Bu çalışma için oluşturulan veri kümesinde bulunan ürünlerin üretici bilgileri.

Kapak	Ürün Tanımı	Üretici Bilgisi
	Ürün_1	Kızılay İçecek Sanayi Ve Ticaret Anonim Şirketi
	Ürün_2	Erbak-Uludağ İçecek Sanayi Anonim Şirketi
	Ürün_3	Freşa İçecek Sanayi ve Ticaret Anonim Şirketi
	Ürün_4	Kızılay İçecek Sanayi Ve Ticaret Anonim Şirketi
	Ürün_5	Kızılay İçecek Sanayi Ve Ticaret Anonim Şirketi
	Ürün_6	Niğde Meşrubat ve Gıda Sanayi Ticaret Limited Şirketi
	Ürün_7	Erbak-Uludağ İçecek Sanayi Anonim Şirketi
	Ürün_8	Erbak-Uludağ İçecek Sanayi Anonim Şirketi
	Ürün_9	Erbak-Uludağ İçecek Sanayi Anonim Şirketi

ÖZGEÇMİŞ

1986 yılında İstanbul’da doğdu. Lise eğitimini Ümraniye Teknik ve Endüstri Meslek Lisesi, Elektronik bölümünde tamamladı. Marmara Üniversitesi Teknik Bilimler Meslek Yüksekokulu’nda Endüstriyel Elektronik Bölümünden Önlisans, Anadolu Üniversitesi İktisat Fakültesi Kamu Yönetimi Lisans ve Beykent Üniversitesi Elektronik ve Haberleşme Mühendisliği Lisans programlarından mezun oldu.

Volkan COŞKUN

