

**UR5 İŞBİRLİKÇİ ROBOTLA FARKLI GEOMETRİK
ŞEKİLLERE SAHİP MALZEMELERİN
SINIFLANDIRILMASI**

**CLASSIFYING COMPONENTS WITH DIFFERENT
GEOMETRIC SHAPES BY UR5 COBOT**

AHMET ARAS AL

DR. ÖĞR. ÜYESİ ŞÖLEN KUMBAY YILDIZ
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Elektrik ve Elektronik Mühendisliği Anabilim Dalı için öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2021

ÖZET

UR5 İŞBİRLİKÇİ ROBOTLA FARKLI GEOMETRİK ŞEKİLLERE SAHİP NESNELERİN SINIFLANDIRILMASI

Ahmet Aras AL

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Şölen KUMBAY YILDIZ

Haziran 2021, 78 sayfa

Günümüzde makine öğrenmesi, literatürdeki yapay zekâyla ilgili çalışmaların büyük bir alt kümesini oluşturmaktadır. Derin öğrenme teknikleri ise, evrişimli sinir ağlarının kullanıldığı, özelleştirilmiş bir makine öğrenmesi tekniğidir. Derin öğrenme tekniklerinde, nesne tanıma ve sınıflandırma problemlerinde, eğitim için kullanılan veri kümesinden istenilen özelliklerin kullanıcı tarafından elle çıkartılmasına gerek kalmamaktadır. Derin öğrenme ile görüntü işleme tekniklerinde, sinir ağları üzerinden, eğitim verisi işlenerek, sınıflandırma için gerekli özelliklerin otomatik olarak çıkarılması önemli bir avantaj olarak değerlendirilmektedir.

Bu tez kapsamında, bir taşıyıcı banttan teker teker ve rastgele bir düzenle gelmekte olan vida, pul ve somun gibi farklı geometrik özelliklerdeki malzemelerin sınıflandırılması için, derin öğrenme tekniklerini kullanarak malzemenin ne olduğunu tespit eden MATLAB tabanlı bir nesne tanıma yazılımı hazırlanmıştır. Hazırlanan yazılımda kullanılacak nesne tanıma algoritmasına karar vermek için, literatürde derin öğrenmeyle nesne tanıma problemlerinde yaygın olarak kullanılan, Alexnet, Googlenet ve Squeezenet algoritmalarının, tez çalışmasında kullanılacak vida, pul ve somunları tanıma ve sınıflandırma başarımları incelenmiştir. Her bir algoritmanın, kendi kütüphanesinde hazır bulunan nesnelere yerine, tez çalışmasının konusu olan vida, pul ve

somunları tanıyıp sınıflandırabilmesi için, bu nesnelerin fotoğraflarından oluşan ortak bir veri kümesi hazırlanmıştır ve her bir algoritma, bu veri kümesi kullanılarak aktarım öğrenme yöntemiyle eğitilmiştir. Eğitilen algoritmaların başarımları; öncelikle veri kümesinin test için ayrılan fotoğraflarıyla yazılım ortamında test edilmiş, devamında USB tabanlı bir web kamerası kullanılarak, donanım ortamında kıyaslama testi yapılmıştır. Benzetim ve uygulamayla gerçekleştirilen testler sonrasında, tezin konusu olan nesne sınıflandırma problemi için en uygun algoritmanın Squeezenet algoritması olduğu görülmüştür.

Squeezenet algoritması kullanılarak hazırlanan nesne tanıma yazılımı, her bir tahmin durumu için, UR5 işbirlikçi robota TCP/IP üzerinden farklı bir mesaj göndererek haberleşmektedir. Robotun yapacağı hareketleri kontrol etmek amacıyla, UR5 robotun kendi multimedya arayüzü üzerinden, Polyscope grafik kullanıcı arayüzü ile robot kontrol yazılımı hazırlanmıştır. Robot kontrol yazılımı, TCP/IP üzerinden belirli bir portu sürekli olarak dinleyerek, MATLAB tabanlı nesne tanıma yazılımının çıktısı olan tahmine göre iletilen mesajları almakta ve gelen mesaja göre robot kol harekete geçerek malzemeyi alıp her malzemeyi ait olduğu kutulara atarak sınıflandırma işlemini gerçekleştirmektedir. Tezin çalışma konusu olan vida, pul ve somunların mıknatıslanma özelliği bulunduğu için robot kolun uç işlevcisi olarak gerilim kontrollü bir elektromıknatıs kullanılmıştır. Polyscope tabanlı robot kontrol yazılımında her bir malzeme için robot kolun yapması gereken işlemlere dair farklı senaryolar oluşturulmuştur. Son olarak oluşturulan sistemin başarımları test edilmiş ve sistemin tüm yazılım ve donanımlarıyla birlikte yüksek başarımla çalıştığı görülmüştür.

Anahtar Kelimeler: Yapay Zekâ, Makine Öğrenmesi, Derin Öğrenme, Evrişimli Sinir Ağları, Nesne Tanıma, Sınıflandırma, Squeezenet, Polyscope, UR5

ABSTRACT

CLASSIFYING COMPONENTS WITH DIFFERENT GEOMETRIC SHAPES BY UR5 COBOT

Ahmet Aras AL

Master of Science, Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Şölen KUMBAY YILDIZ

June 2021, 78 pages

Machine learning is a major subset of artificial intelligence studies in the literature. Deep learning is also a specialized machine learning technique which involves the use of convolutional neural networks. In deep learning techniques, it is not necessary to manually extract features from the training dataset for object recognition and classification problems. Image processing with deep learning techniques has a significant advantage of extracting the necessary features automatically for classification, by processing the training dataset through neural networks.

In this thesis, a MATLAB based object recognition program is created using deep learning techniques to recognise and classify components with different geometric shapes, more specifically screws, washers and nuts, which are fed to the experiment setup randomly and one by one from a conveyor belt. The performances of three widely used algorithms in the literature for image recognition problems, namely Alexnet, Googlenet and Squeezenet are investigated in order to decide which object recognition algorithm is more suitable to be used in the software part of the thesis. A common data set consisting of the photos of the objects is prepared for recognising and classifying

screws, washers and nuts. Instead of utilising the objects that are already present in the libraries of the algorithms, each algorithm is trained by this data set with the transfer learning method. The performances of trained algorithms are initially tested on a software medium with the photos from data set reserved as test data. Subsequently, a benchmark test is carried out on the physical setup with a USB webcam for those algorithms. The tests performed on both software and hardware revealed that the Squeezenet algorithm achieved the highest performance.

The object recognition program with the Squeezenet algorithm performs communication with the UR5 collaborative robot (cobot) through TCP/IP by sending a unique message for each prediction case. A robot control program is designed with Polyscope, the graphical user interface on the robot's teach pendant to control the robot's movements. The robot control program receives the prediction output of a MATLAB based object recognition program by continuously listening to a specific port over TCP/IP and completes the classification process by moving the robot arm according to the received message to pick up the component and put it into corresponding box for that particular object. A voltage-controlled electromagnet is chosen as an end effector for the robot arm since the screws, washers and nuts used in this thesis study have ferromagnetic characteristics. Different scenarios are prepared on the Polyscope based robot control program to perform the necessary tasks of the robot arm for each object. Finally, the performance of the overall system is put to test and both the hardware and the software parts are observed to be operating successfully.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, Convolutional Neural Networks, Object Recognition, Classification, Squeezenet, Polyscope, UR5

TEŞEKKÜR

Henüz 2007 yazında üniversite tercih zamanı kararımı netleştirmek için ziyarete geldiğim Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği bölümünde bir araştırma görevlisiyken tanıştığım, iyi niyetli yaklaşımı ve samimi anlatımıyla bu bölümü tercih etmemde büyük katkısı olan, yapmak istediğim tez çalışmasını başarabileceğime inanıp bana güvenen, çalışma süresince yaşadığım zorluklar karşısında göstermiş olduğu anlayış ve rehberlikle bu çalışmayı tamamlamamı sağlayan, değerli hocam Sayın Dr. Öğr. Üyesi Şölen Kumbay Yıldız'a,

Tez çalışmasında yer alan UR5 robotu kullanabilmem için her türlü kolaylığı sağlayan, bu çalışma için ihtiyacım olan kaynakları sağlamak için bana destek veren kıymetli iş arkadaşlarım Sn. Serkan Dinçer, Sn. Sami Orçun Onur, Sn. Doğan Fethi Kaya ve Sn. Çetin Taşdelen'e,

Üniversite hayatımın ilk yılından itibaren herşeyimi paylaştığım, arkadaşlıklarıyla hayatı daha katlanılır ve eğlenceli kılan, bu tez çalışması süreci de dâhil olmak üzere her zaman manevi desteklerini yanımda hissettiğim can dostlarım Murat Senem ve Mevlüt Özgür Demiray'a,

Tüm eğitim öğretim hayatım boyunca beni destekleyen, üzerimde sonsuz emekleri olan ve haklarını asla ödeyemeyeceğim canım annem Ayşe Al'a ve biricik babam Alper Al'a, üniversite ve mühendislik tercihlerinde beni örnek alsa da benden daha büyük başarılarla imza atan dünyalar tatlısı kardeşim Alsu Ceren Al'a

Son olarak, henüz ilişkimizin başındayken dahi bana tezimi yazıp mezun olabileceğim konusunda sonsuz güvenen, ben kendime inancımı yitirmek üzereyken dahi bana olan inancını bir an olsun kaybetmeyen, sevgisi ve şefkatiyle her zaman yanımda olan, bu süreçteki ve hayattaki en büyük güç kaynağım sevgili eşim, hayat arkadaşım Şevval Didar Yıldız Al'a,

Sonsuz Teşekkürler...

Ahmet Aras AL
Haziran 2021, Ankara

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	iii
TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
SÖZLÜK DİZİNİ.....	viii
ŞEKİLLER.....	x
ÇİZELGELER.....	xi
1. GİRİŞ.....	1
1.1. Tezin Akışı.....	2
2. KURAMSAL BİLGİLER.....	4
2.1. Derin Sinir Ağları.....	4
2.2. Alexnet.....	7
2.3. Googlenet.....	16
2.4. Squeezenet.....	24
3. DERİN SİNİR AĞLARINDA AKTARIM ÖĞRENİMİ YÖNTEMİYLE SINIFLANDIRMA.....	31
3.1. Aktarım Öğrenme Yazılımının Hazırlanması.....	31
3.2. Aktarım Öğrenme Veri Kümesinin Hazırlanması.....	36
3.2.1. Veri Kümesinin Hazırlanmasında Mesafe Etkisi.....	38
3.2.2. Veri Kümesinin Hazırlanmasında Ortam Işığının Etkisi.....	41
3.3. Derin Öğrenme Algoritmalarının Yazılım Ortamında İncelenmesi.....	44
3.4. Derin Öğrenme Algoritmalarının Uygulama Ortamında İncelenmesi.....	51
3.5. Algoritma Başarımlarının Karşılaştırılması ve Uygun Algoritmanın Seçilmesi.....	54
4. UR5 İŞBİRLİKÇİ ROBOTLA SINIFLANDIRMA.....	58
4.1. UR5 İşbirlikçi Robot.....	58
4.2. Donanım Hazırlıkları.....	59
4.3. Robot Kontrol Yazılımının Hazırlanması.....	62
4.4. Sistem Başarım Testleri.....	67

5. SONUÇLAR VE DEĞERLENDİRMELER	71
KAYNAKLAR	73
ÖZGEÇMİŞ	78

SÖZLÜK DİZİNİ

Doğrultulmuş Doğrusal Birim	Rectified Linear Unit
Makine Öğrenmesi	Machine Learning
Aktarım Öğrenmesi	Transfer learning
Derin Öğrenme	Deep Learning
Evrişimli Sinir Ağı	Convolutional Neural Network
Tam Bağlantılı Katman	Fully Connected Layer
Son Öğrenilebilir Katman	Last Learnable Layer
Aşırı Uyum	Overfitting
Döngü	Epoch
Başarım Kıyaslama	Benchmark
Yakınlaşma	Zoom
Adım motoru	Step Motor
Geri Yayılım	Back Propagation
Zincir Kuralı	Chain Rule
Havuzlama	Pooling
Maksimum Havuzlama	Max. Pooling
Destek Vektör Makinesi	Support Vector Machine
Uzun kısa-süreli hafıza	Long Short-Term Memory (LSTM)
Grafik İşleme Birimi	Graphical Processing Unit (GPU)
Kaybolan Gradyan Problemi	Vanishing Gradient Problem
Artık Sinir Ağları	Residual Neural Networks(RNN)
Aralık	Stride
Dolgu	Padding
Yardımcı Sınıflandırıcı	Auxiliary Classifier
Alt Örnekleme	Downsampling
Uç İşlevci	End Effector
Sunucu	Server
İstemci	Client
Çalışma Alanı	Workspace

İmge Noktası

Yanlı Sinir Hücreleri

Eşiksiz En Büyük Etkilenim İşlevi

Pixel

Bias Norons

Softmax Activation Function

ŞEKİLLER

Şekil 2.1 Alexnet Derin Sinir Ağı Katmanları	8
Şekil 2.2 Evrişim İşlemi	9
Şekil 2.3 Havuzlama ile Alt Örnekleme.....	10
Şekil 2.4 Tam Bağlantılı Katman	11
Şekil 2.5 Doğrultulmuş Doğrusal Birim İşlevinin Eğitim Hızına Etkisi.....	12
Şekil 2.6 Düşürme Tekniğinin Gösterimi	13
Şekil 2.9 Başlangıç Katmanı	20
Şekil 2.10 1x1 Evrişim Süzgeçleriyle Boyut Düşürme.....	20
Şekil 2.11 Yardımcı Sınıflandırıcı Mimarisi.....	21
Şekil 2.13 Ateş Modülü.....	27
Şekil 2.14 SqueezeNet Algoritması Genel Akışı	28
Şekil 3.2 Veri Kümesinde Kullanılan Nesnelere	37
Şekil 3.3 20cm Mesafeden Fotoğraflanan Nesne Örneği.....	38
Şekil 3.4 10cm Mesafeden Fotoğraflanan Nesne Örneği.....	39
Şekil 3.5 10cm Mesafeden 2.5x Yakınlaşma ile Fotoğraflanan Nesne Örneği.....	40
Şekil 3.6 Doğrulama Verisine Yapılan Tahmin Örneği	46
Şekil 3.7 Alexnet Algoritması Benzetim Başarım Testi Sonucu	48
Şekil 3.8 GoogLeNet Algoritması Benzetim Başarım Testi Sonucu	49
Şekil 3.9 SqueezeNet Algoritması Benzetim Başarım Testi Sonucu.....	50
Şekil 4.1 UR5 İşbirlikçi Robot.....	59
Şekil 4.2 Uç İşlevci	61
Şekil 4.3 Donanım Altyapısı Genel Görünümü	62
Şekil 4.4 Robot Kontrol Yazılımı Akışı.....	63
Şekil 4.5 Nesne Tanıma Yazılımı Akışı.....	63
Şekil 4.6 Parça Alınmadı Uyarısı	67

ÇİZELGELER

Çizelge 2.1 Googlenet Katmanları.....	18
Çizelge 2.2 Googlenet Başlangıç Modülü	19
Çizelge 2.3 Squeezenet Katmanları	26
Çizelge 2.4 Squeezenet ve Alexnet Model Boyutu ve İisabetlilik Karşılaştırması	29
Çizelge 3.1 MATLAB’da Tanımlı Ön Eğitilmiş Evrişimli Sinir Ağları.....	32
Çizelge 3.2 Veri Kümesinin Hazırlanmasında Mesafe Etkisi	41
Çizelge 3.3 Algoritmaların Benzetim Başarımları Kıyaslama Çizelgesi	55
Çizelge 3.4 Algoritmaların Uygulama Başarımları Kıyaslama Çizelgesi	55
Çizelge 4.1 Sistem Başarım Testi Sonuçları.....	69

1. GİRİŞ

Evrişimli sinir ağıları, derin öğrenme uygulamalarında son yıllarda gittikçe daha popüler olmaktadır. Yüksek başarımlı bilgisayar grafik işleme birimlerinin üretilmesiyle veri işleme kapasitesinin artışı, evrişimli sinir ağlarının başarımını artırmıştır. Büyük verilerin daha hızlı ve daha verimli bir şekilde işlenebilmesi sayesinde evrişimli sinir ağlarının nesne tanıma, nesne sınıflandırma gibi bilgisayar görüşü işlemlerinde yüksek başarımlar elde etmesi mümkün olmuştur.

Nesne sınıflandırma amacıyla geliştirilen evrişimli sinir ağlarının, kendi ön eğitilmiş kütüphanesinde bulunan nesnelere daha farklı nesnelere sınıflandırabilmesi için aktarım öğrenme yöntemiyle ağ yeniden eğitilebilmektedir. Belirli nesnelere tanıması için aktarım öğrenmesiyle eğitilen ağın nesnelere tanıma başarımında, ağın eğitiminde kullanılan veri kümesi büyük önem taşımaktadır. Veri kümesindeki her bir nesne sınıfına ait fotoğraf miktarı, nesnenin fotoğraf kadrajında kapladığı alan, nesnenin bulunduğu ortam ışığı gibi faktörler, eğitilen algoritmanın başarımını doğrudan etkileyebilmektedir.

Alexnet, Googlenet ve Squeezenet evrişimli sinir ağları, hız ve verimlilik açısından getirdiği çözümlerle literatürde sağlık, tarım, trafik sistemleri, robotik gibi çeşitli alanlarda nesne sınıflandırma, nesne tespiti, hareket takip ve tespiti gibi amaçlarla tercih edilmişlerdir [1–19]. Tezin ilerleyen bölümlerinde bu çalışmalarla ilgili daha ayrıntılı bilgiye değinilecektir. Bu tez çalışmasında Alexnet, Googlenet ve Squeezenet evrişimli sinir ağlarının, vida pul ve somun gibi farklı geometrik özelliklere sahip nesnelere sınıflandırma başarımları ve kullanılan veri kümesinin bu başarımlara etkileri araştırılmış, benzetim ve uygulama olarak başarımların kıyaslamaları yapılarak sonuçlar ortaya konmuştur. Yapılan araştırmalar sonucunda en yüksek başarımları gösteren Squeezenet evrişimli sinir ağının, UR5 işbirlikçi robotla birlikte kullanımıyla nesne sınıflandırmasının donanım ortamında hızlı ve verimli bir şekilde gerçekleştirilebilmesi için bir yöntem önerilmiştir.

1.1. Tezin Akışı

Bu tez çalışmasının akışı şu şekilde organize edilmiştir. Bölüm-2’de tez çalışmasında incelenen Alexnet, Googlenet ve Squeezenet evrişimli derin sinir ağları hakkında kuramsal bilgiler verilmiştir. İlgili algoritmaların katmanları ve mimari yapıları detaylı olarak ele alınarak çalışma yapıları incelenmiştir. Bu derin sinir ağlarının nesne sınıflandırma işlemlerini nasıl ele aldığı, sahip oldukları kritik özellikleri gibi konulardan bahsedilmiştir. Ek olarak, ilgili ağların literatürde daha önce hangi alanlarda ne gibi uygulamalarda kullanıldığı, işbirlikçi robotlarla birlikte kullanılıp kullanılmadığı ve yapılan çalışmalarda ne gibi başarımlar elde edildiğine dair bilgiler verilmiştir.

Üçüncü bölüm, çalışmada kullanılan derin sinir ağlarının, vida, pul ve somun nesnelere sınıflandırabilmesi için kullanılan aktarım öğrenme yazılımının nasıl hazırlandığını işleyerek başlamaktadır. Devamında, veri kümesini oluştururken çekilen fotoğrafların nesnelere olan uzaklığının ve ortam ışığının, aktarım öğrenme başarımlarına etkisi ele alınmaktadır. Son olarak, hazırlanan aktarım öğrenme yazılımıyla eğitilen derin sinir ağlarının benzetim ve uygulamalı başarımlarını karşılaştırmalı olarak incelenmekte ve sonuçları ortaya konmaktadır. Bölümün sonunda, elde edilen bulgular doğrultusunda UR5 robotla yapılacak sınıflandırma işleminde hangi derin sinir ağının tercih edildiği, nedenleriyle birlikte açıklanmaktadır.

Dördüncü bölümde tez kapsamında incelenen nesnelere fiziksel olarak sınıflandırmasını gerçekleştiren UR5 işbirlikçi robotun sahip olduğu teknik yetkinlikler ve fiziksel özelliklerinden kısaca bahsedilmiştir. Robotun sınıflandırma operasyonunu gerçekleştirmesi için kurulan donanım düzeneği kullanılan tüm ekipmanların özellikleri ve görevleriyle birlikte okuyucuya aktarılmıştır. Bölümün son kısmında ise, UR5 robotun kendi yazılım arayüzünden hazırlanan robot kontrol yazılımı detaylı olarak anlatılmıştır. Robot kontrol yazılımının hangi senaryoda nasıl davrandığı, standart akışın nasıl planlandığı, test bilgisayarında MATLAB üzerinde koşturma olan nesne sınıflandırma yazılımıyla nasıl haberleştiği ve alınan güvenlik önlemleri gibi konular ele alınmıştır. Bölüm sonunda ise gerçekleştirilen sistem başarımlarından bahsedilmiş ve sonuçları incelenmiştir.

Beşinci ve son bölümde ise kurulan sistemin, tüm donanım ve yazılım özellikleriyle birlikte ele alındığında ortaya koyduğu başarımlar değerlendirilmiş ve yapılan tüm çalışmalar biraraya getirilerek kısaca özetlenmiştir. Bu bölümde ayrıca geliştirilen sistemin nasıl daha iyi başarımlar gösterebileceğine dair ileride yapılabilecek çalışmalara yönelik öneriler sunulmuştur.

2. KURAMSAL BİLGİLER

2.1. Derin Sinir Ağları

Derin sinir ağları, girdi ve çıktılar arasındaki doğru matematiksel ilişkiyi oluşturmayı hedefleyen çok katmanlı yapay sinir ağlarına verilen genel isimdir. Derin öğrenmenin yapı taşları, Walter Pitts ve Warren McCulloch tarafından, 1943 yılında yapılan çalışmada atılmıştır. Pitts ve McCulloch, insan beynindeki sinir ağlarını temel alarak geliştirdikleri modellemede, beynin düşünce sürecini, eşik mantığı olarak adlandırdıkları matematiksel algoritmaların birleşimiyle ifade etmişlerdir [20]. 1960 yılında Henry J. Kelley, ideal uçuş güzergâhlarının hesaplanmasıyla ilgili yaptığı çalışmalar sırasında, sürekli geri yayılım kavramının ilk versiyonunu oluşturmuştur [21]. Bundan 2 yıl sonra 1962’de Stuart Dreyfus, sadece zincir kuralını temel alan daha basit bir geri yayılım kavramı geliştirmiştir [22]. 1960’lı yıllarda kullanılan her iki geri yayılım kavramı da, günümüzde derin öğrenmede eğitim sırasında hatalardan yararlanılarak yapılan geri yayılım süreciyle kıyaslandığında yetersiz ve hantal kalmıştır. Bilinen ilk derin öğrenme algoritması 1965 yılında geliştirilmiştir. Ivakhnenko ve Lapa, geliştirdikleri modelde karmaşık denklemlerden oluşan çoklu etkilenim işlevi kullanarak istatistiksel analizler yapmıştır [23]. Bu sayede her bir katmanda istatistiksel olarak en iyi özelliklerin bir sonraki katmana aktarıldığı yavaş ve manuel bir süreç tanımlamışlardır. 1970’li yıllar, yapay zekâ araştırmacılarının karşılaştıkları problemleri çözememesi ve buna bağlı olarak yatırımcıların ilgisini kaybetmesi ve teşviklerin azalmasıyla derin sinir ağları alanındaki gelişmeler açısından da durağan geçmiştir [24]. Özellikle 1974-1980 yılları arasını kapsayan bu dönem, birinci yapay zekâ kışı olarak adlandırılmaktadır.

Bilinen ilk evrişimli sinir ağı, Kunihiko Fukushima tarafından geliştirilmiştir. Fukushima, geliştirdiği sinir ağlarında çoklu havuzlama ve evrişimli katmanlar kullanmıştır. Fukushima, 1979 yılında geliştirdiği Neocognitron isimli yapay sinir ağında hiyerarşik çok katmanlı bir tasarım oluşturmuştur [25]. Bu tasarım, bilgisayarın görsel şablonları tanıyabilmesini mümkün kılmıştır. Ek olarak Fukushima’nın tasarladığı evrişimli sinir ağı, ağ üzerindeki belirli bağlantılarının ağırlıklarını artırarak istenilen önemli özelliklerin elle ayarlanabilmesini sağlamıştır.

Derin sinir ağlarının eğitiminde hataların kullanılmasını ifade eden geri yayılım kavramı, Seppo Linnainmaa'nın geri yayılım için yazdığı FORTRAN kodunu içeren tez çalışmasıyla birlikte büyük bir ilerleme kaydetmiştir [26]. Yine de bu kavramın derin sinir ağlarında uygulanması, 1985 yılına kadar mümkün olmamıştır. 1985 yılında Rumelhart, Williams ve Hinton'un bir sinir ağında geri yayılmanın ilginç dağıtım temsilleri sağlayabileceğini gösterdiği çalışma, derin sinir ağlarında günümüzdeki geri yayılım kavramının öncüsü olmuştur [27]. Felsefi olarak, bu çalışma sayesinde, bilişsel psikolojide, insan anlayışının sembolik mantığa mı (hesaplama) yoksa dağıtılmış temsillere mi (bağlantısallık) dayandığı sorusu gün yüzüne çıkmıştır. 1989 yılında Yann LeCun, geri yayılım kavramının ilk somut uygulamasını gerçekleştirmiştir. LeCun, bu çalışmada el yazısıyla yazılmış rakamların bilgisayar tarafından okunması için evrişimli sinir ağlarını geri yayılım ile birleştirmiştir [28]. Bu sayede el yazısıyla yazılan rakamları okuyarak bir hata olup olmadığını kontrol ettirmiştir. 1987-1993 yılları arası, sinir ağları ve derin öğrenme araştırmalarını olumsuz etkileyen ikinci yapay zekâ kışı yaşanmıştır. Bu dönemde yapay zekânın mevcut potansiyelinin yatırımcılara abartılarak anlatılması sonucu yükselen beklentilerin karşılanmaması, yatırımların azalmasına ve yapay zekâ araştırmacılarının ekonomik zorluklar yaşamasına neden olmuştur. Tüm zorluklara rağmen yapay zekâ ve derin sinir ağlarıyla ilgili çalışmalarına devam eden araştırmacılar sayesinde önemli gelişmeler devam etmiştir. 1995 yılında Cortes ve Vapnik, benzer verilerin tanınması ve haritalandırmasını sağlayan destek vektör makinesini geliştirmişlerdir [29]. 1997 yılında ise, Hochreiter ve Schmidhuber tarafından tekrarlayan sinir ağları için yetersiz ve azalan hata geri yayılımı problemini çözmek amacıyla “uzun kısa süreli bellek” modeli geliştirilmiştir [30].

Derin öğrenme alanındaki bir sonraki önemli evrimsel adım, bilgisayarların veri işleme kapasitesinin artmaya başladığı ve grafik işleme birimlerinin geliştirildiği 1999 yılında gerçekleşmiştir. Grafik işleme birimlerinin resimleri işleyebilme yeteneğiyle, hesaplama hızları 10 yıllık bir süreçte 1000 kat artan bilgisayarlar sayesinde sinir ağları, vektör destek makineleriyle rekabet edebilmeye başlamıştır. Destek vektör makinelerinin hız avantajına karşı, sinir ağları aynı verileri kullanarak daha iyi sonuçlar ortaya koyma ve daha fazla eğitim verisi eklendikçe gelişmeye devam etme avantajıyla öne çıkmıştır.

2000’li yılların başında gradyan tabanlı öğrenme yöntemlerine sahip derin sinir ağlarında “Kaybolan Gradyan Problemi” ortaya çıkmıştır [31]. Ağın alt katmanlarında oluşan özelliklerin, üst katmanlara öğrenme sinyalinin ulaşmamasından dolayı üst katmanlar tarafından öğrenilemediği keşfedilmiştir. Sorunun kaynağının, belirli etkilenim işlevlerinin, son derece küçük bir aralıkta haritalanan yoğun girdi alanları oluşturduğu ve bu girdi alanında büyük bir değişiklik olsa dahi çıktısında küçük bir değişikliğe indirgendiği ve bunun da kaybolan bir gradyanla sonuçlanmasından kaynaklandığı ortaya çıkmıştır. Bu problemin üstesinden gelmek için ilerleyen yıllarda her bir katmanın ayrı ayrı ön eğitimi [32], etkilenim işlevi olarak “Doğrultulmuş Doğrusal Birim” kullanımı [33], “Artık Sinir Ağları” [34] gibi yöntemler geliştirilmiştir.

2001 yılında Laney, veri artışının getirdiği zorluklar ve fırsatları; veri hacmi, veri hızı ve veri çeşitliliği olmak üzere üç farklı ekseninde tanımladığı bir rapor yayınladı [35]. Raporda, veri çeşitliliğinin ve veri kaynaklarının artışıyla veri büyüklüklerinin ve veri hızlarının artışı bahsedildi. Bu rapor, henüz başlamakta olan “Büyük Veri” dönemine bir hazırlık çağrısı niteliğinde kabul edilmiştir.

2009 Yılında Li, 14 milyondan fazla tanımlı görüntüden oluşan ücretsiz bir veritabanı olan ImageNet’i oluşturdu [36]. Oluşturulan bu veri tabanı derin sinir ağlarının eğitilmesinde kullanıldı. 2012 yılından itibaren de Imagenet Geniş Çaplı Görsel Tanıma Yarışması (İngilizce açılımı - ILSVRC) adıyla nesne tespit ve görüntü sınıflandırma alanlarında dünya çapında yarışmalar düzenlenmiştir.

2011 yılından itibaren, grafik işleme birimlerinin hızlarının büyük bir artış göstermesi, evrimsel sinir ağlarının katman katman ön eğitime ihtiyaç duymadan eğitilebilmesini mümkün kılmıştır. Bilgisayarların hesaplama hızlarının artışıyla birlikte derin öğrenme yöntemleri, hız ve verimlilik açısından oldukça avantajlı duruma gelmiştir. Nitekim bu durum ILSVRC yarışmalarında da belirgin bir şekilde ortaya çıkmış ve 2012 yılından itibaren, evrimsel sinir ağları ILSVRC yarışmalarında büyük üstünlük elde etmişlerdir [37]. Bu tez çalışması kapsamında incelenen derin öğrenme ağlarından Alexnet 2012 yılında, Googlenet ise 2014 yılında Imagenet Geniş Çaplı Görsel Tanıma Yarışması’nın (ILSVRC) kazanan algoritmaları olmuştur [37].

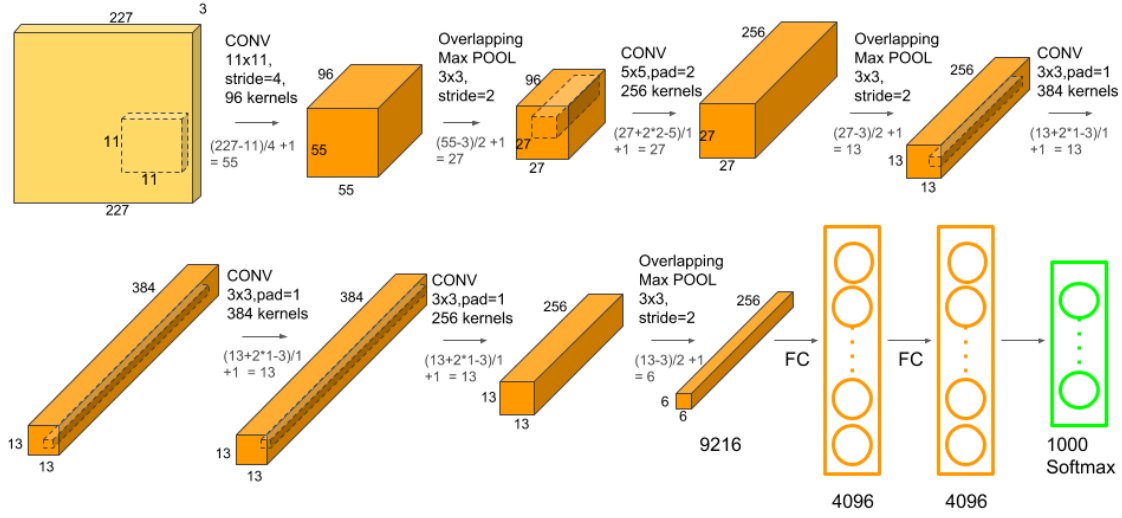
2012 yılında Google Brain ekibi tarafından gerçekleştirilen “The Cat Experiment” isimli projenin sonuçları yayınlandı [38]. Projede, tanımlı veriler

kullanılarak eğitilen evrişimli sinir ağlarını ifade eden denetimli öğrenme yerine, tamamen denetimsiz öğrenme yöntemi kullanılarak, bu yöntemin zorlukları ve sonuçları araştırılmıştır. 1000 adet bilgisayar kullanılmış ve 3 gün boyunca internetten izletilen videolarla derin sinir ağının tanımsız verilerden neler öğrenebildiği araştırılmıştır. Çalışma sonucunda sinir ağındaki bazı nöronların insan yüzü, insan bedeni, kedi yüzü gibi belirli nesnelere karşı detektör görevi gördüğü ve yüksek tanıma başarısı gösterdiği gözlemlenmiştir.

Derin sinir ağlarının geçmişten günümüze gelişimi incelendiğinde, hem büyük verinin işlenmesi hem de yapay zekânın evriminin, derin sinir ağlarıyla doğrudan bağlantılı olduğu görülmektedir. Derin sinir ağları hala gelişmekte ve yaratıcı fikirlerle yeni gelişmelere açık durumdadır. Bu bölümün devamında, derin sinir ağlarının günümüze olan yolculuğuna da önemli katkılarda bulunan, literatürde birçok çeşitli alanda uygulamalarda da kullanılmış olan Alexnet, Googlenet ve Squeezenet derin sinir ağlarının çalışma yapısı ve özellikleri incelenecektir.

2.2. Alexnet

Alexnet, 2012 yılında Krizhevsky ve arkadaşları tarafından geliştirilmiş bir derin sinir ağıdır [39]. ImageNet kütüphanesinde yer alan görselleri sınıflandırmak amacıyla tasarlanmıştır. Girdi olarak Imagenet kütüphanesindeki 1000 farklı görsel sınıfına ait resimleri alıp, bu resmin mevcut sınıflardan hangisine ait olabileceğine dair yüzdesel olasılık çıktıları üretmektedir. Sinir ağlarında o güne kadar kullanılan geleneksel yöntemlere getirdiği yenilikçi yaklaşımlarla 2012 yılında düzenlenen ILSVRC Imagenet Geniş Çaplı Görsel Tanıma Yarışması'nı büyük başarı göstererek kazanmıştır. Yarışmada, Imagenet kütüphanesindeki tanımlı görsellere, algoritmanın ürettiği en yüksek olasılıklı ilk 5 tahminde doğru tahmin edilmemesi anlamına gelen Top-5 hata oranı %15,3 olan Alexnet, yarışmadaki bir sonraki en düşük top-5 hata oranının %26,2 olmasıyla yarışmayı etkileyici bir farkla kazanmıştır. Alexnet derin sinir ağı, Şekil 2.1'de görüldüğü üzere temel olarak 5 evrişimli katman, 3 havuzlama katmanı ve 3 tam bağlantılı katman olmak üzere 11 temel katmandan oluşmaktadır. Bu başlık altında Alexnet derin sinir ağının katmanlarında gerçekleştirilen işlemler detaylı olarak ele alınacaktır. Alexnet derin sinir ağı katmanları Şekil 2.1'de gösterilmiştir [40].



Şekil 2.1 Alexnet Derin Sinir Ağı Katmanları

Evrişim katmanlarında, evrişim operasyonu eğitilebilir çekirdek veya süzgeçler kullanılarak yapılmaktadır. Bazı sinir ağlarında ise her bir katmana eklenen ve sabit bir değere sahip yanlı sinir hücreleri de evrişim işlemine eklenir [41]. Bu evrişim işlemi, süzgeçlerin, girdi görseli üzerinde belirli bir imge noktası sayısı kadar kaydırılarak girdi üzerinde hareket ettirilmesi şeklinde gerçekleştirilmektedir. Bu kaydırma aralığı ne kadar büyük olursa, süzgeçin her bir evrişim işleminden sonra aldığı yol o kadar fazla olmaktadır. Bu da toplamda daha az sayıda evrişim işlemi yapılarak çıktı boyutunun küçültülmesini sağlamaktadır. Süzgeçin, evrişim işlemleri için girdi üzerindeki her bulunduğu konumda, girdinin ilgili imge noktası değeri ve süzgeçin değeri arasında çarpma işlemi gerçekleştirilmektedir. Çıkan sonuca yanlı sinir hücresinin değeri eklenerek evrişim işlemi yapılmaktadır. Bu sayede evrişim işleminin sonucunda bir özellik haritası ortaya çıkmaktadır. Çıkarılan özellik haritasına, bir etkilenim işlevi uygulanarak bir sonraki katman için gerekli girdi sağlanmaktadır. Şekil 2.2 bir evrişim işlemi örneğini göstermektedir [42].

Şekil 2.2’de 6x6x3’lük bir girdi üzerine 3x3x3 boyutunda bir süzgeç, aralık değeri 1, dolgu değeri 0 olarak uygulanmıştır. Süzgeç, girdi üzerinde soldan sağa ve yukarıdan aşağıya doğru hareketlerini 1’er imge noktası ilerleyerek yatayda ve dikeyde 4 seferde tamamlayacağı için, özellik haritasının boyutu 4x4x3 olmaktadır. Yanlı sinir

hücrelerinde bulunan değer, süzgeç ve girdi arasında yapılan toplam 16 evrişim işleminin her birinin sonucuna ayrı ayrı eklenmiştir.

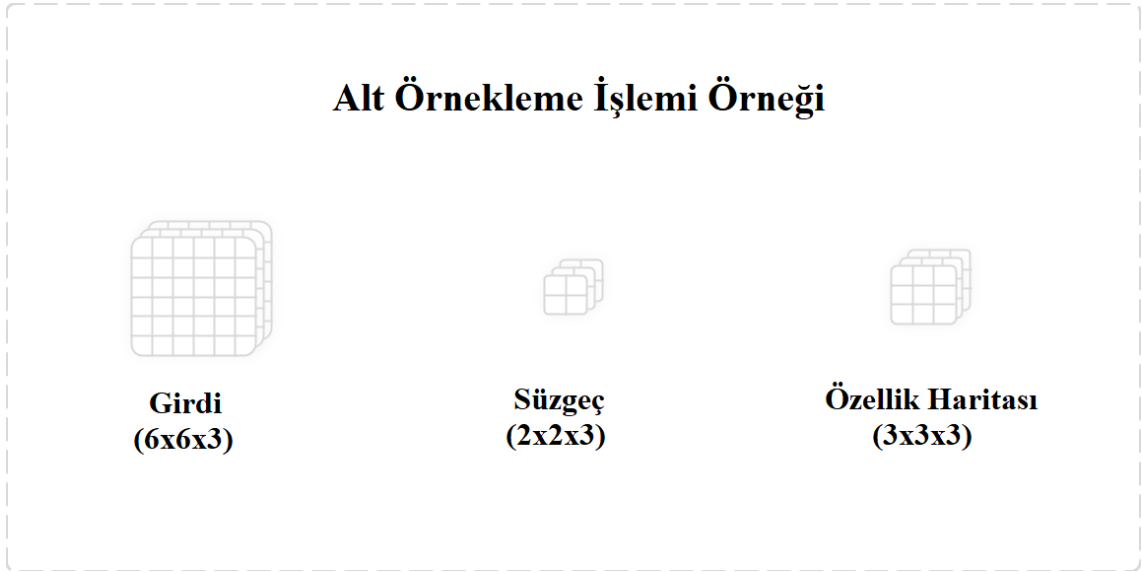


Şekil 2.2 Evrişim İşlemi

Havuzlama katmanları, eğitim amaçlı olmayan süzgeçlerin kullanıldığı alt örnekleme katmanlarıdır [42]. Kullanılmasındaki temel amaç, girdi verisinden bir sonraki katmana taşınacak olan özellik sayısını azaltmak ve ağın içerisindeki parametre sayısını ve hesaplama yükünü azaltmaktır. Ortalama havuzlama ve maksimum havuzlama olarak bilinen iki yöntem bulunmaktadır. Ortalama havuzlamada süzgeçin girdi üzerinde bulunduğu konum içerisindeki imge noktası değerlerinin ortalaması alınırken, maksimum havuzlamada süzgeçin bulunduğu konum içerisindeki imge noktalarından en yüksek değere sahip olan imge noktası seçilmektedir. Alexnet derin sinir ağında maksimum havuzlama yöntemi kullanılmaktadır [39]. Maksimum havuzlama katmanının özellik haritasının boyutları evrişimli katmandakine benzer şekilde hesaplanmaktadır. Şekil 2.3, örnek bir maksimum havuzlama katmanı işlemini göstermektedir [42].

Şekil 2.3'te 6x6x3 boyutundaki bir girdiye 2x2x3 boyutundaki süzgeç, 2 imge noktası aralıklarla uygulanmıştır. Böylece süzgeç yatayda ve dikeyde 3'er kez kayarak toplamda 9 kez yer değiştirmiş ve her bulunduğu konumun içerisinde bulunan en

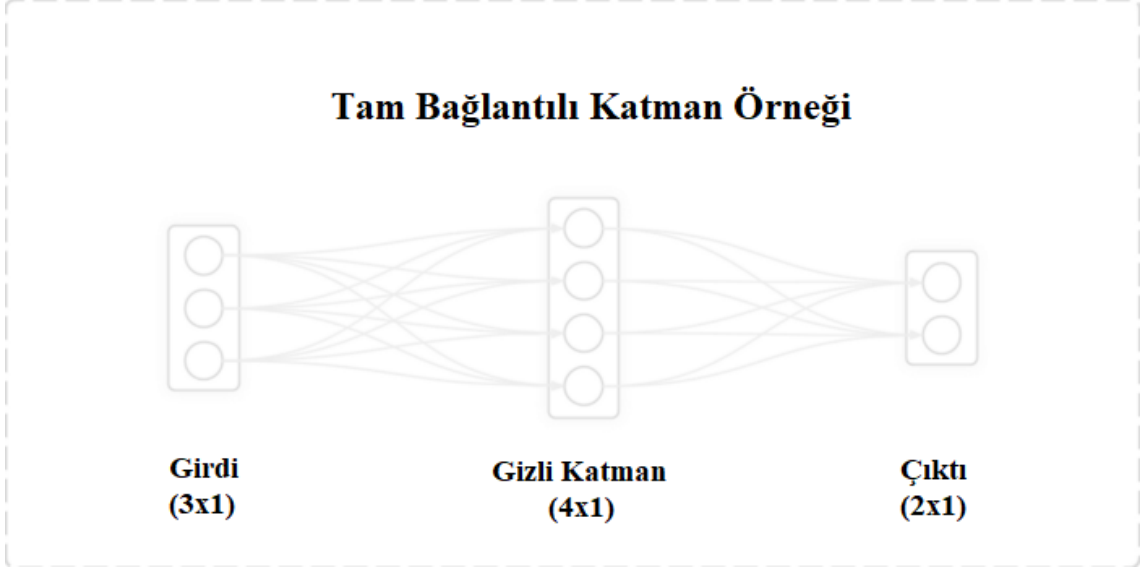
yüksek değerli imge noktasını olarak 3x3x3'lük özellik haritasını oluşturmuştur. Böylece 6x6x3'lük girdinin, 3x3x3 şeklinde aşağı örneklenmesi gerçekleştirilmiştir.



Şekil 2.3 Havuzlama ile Alt Örnekleme

Alexnet derin sinir ağı, maksimum havuzlama katmanlarında 3x3'lük süzgeçleri, 2 imge noktası kadar aralık değeriyle kaydırarak üstüste binen maksimum havuzlama işlemi gerçekleştirmiştir. Bu sayede havuzlama katmanının çıktı boyutlarının azaltılması sağlanarak, üstüste binmeyen maksimum havuzlama işlemiyle kıyaslandığında top-1 hata oranında %0,4 , top-5 hata oranında %0,3'lük bir iyileştirme sağlamıştır [39].

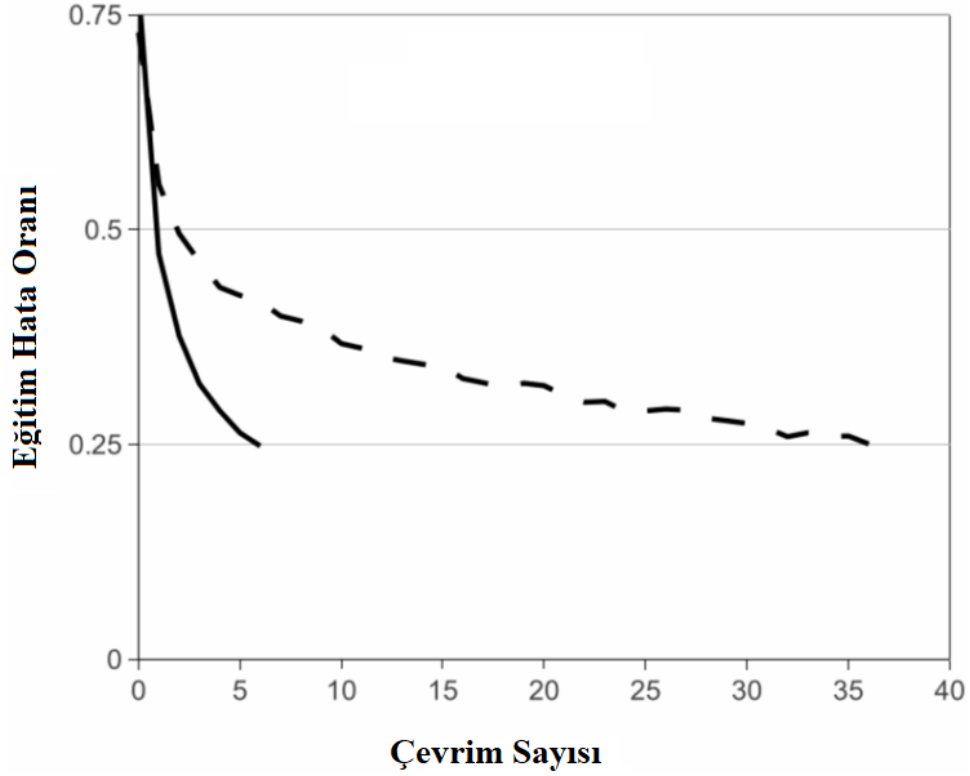
Tam bağlantılı katmanlar, sadece evrişimli sinir ağlarına özgü olmamasına rağmen, genellikle birçok evrişimli sinir ağının son birkaç katmanında yer almaktadırlar. Derin sinir ağı içerisinde bazı evrişim katmanları ve alt örneklemeye katmanlarından sonra gerçekleştirilmektedirler. Tam bağlantılı katmanlar, genellikle içerisinde bir ya da daha fazla gizli katman içeren bağımsız sinir ağlarıdır. Tam bağlantılı katmanın girdileri, eğitilebilir bir ağırlık vektörüyle çarpılmakta ve yanlı sinir hücrelerinde bulunan sabit bir değer bu sonuca eklenerek çıktılar oluşturulmaktadır. Tam bağlantılı katmanın çıktıları, tıpkı evrişimli katman çıktıları gibi etkilenim işlevinden geçirilerek bir sonraki katmana aktarılmaktadır. Şekil 2.4, 3x1 boyutundaki bir girdinin, 4x1 boyutundaki gizli katmanları olan bir tam bağlantılı katmandan geçerek 2x1 boyutunda çıktılar oluşturan bir tam bağlantılı katman örneğini göstermektedir [42].



Şekil 2.4 Tam Bağlantılı Katman

Alexnet derin sinir ağının en önemli özelliklerinden biri de, etkilenim işlevi olarak geleneksel olarak kullanılan tanh ya da sigmoid işlevlerinin yerine, doğrultulmuş doğrusal birim işlevini kullanmasıdır. Krizhevsky, etkilenim işlevi olarak doğrultulmuş doğrusal birim kullanımıyla, derin evrişimli sinir ağlarının, tanh ya da sigmoid işlevlerinin doyum bölgesini kullanan derin evrişimli sinir ağlarından çok daha hızlı bir şekilde eğitilebileceğini ortaya koymuştur [39]. Şekil 2.5, CIFAR-10 veri kümesiyle eğitilen Alexnet ve aynı veri kümesiyle eğitilen ancak tanh etkilenim işlevi kullanan benzer bir ağ karşısındaki eğitim hızlarının grafiğini göstermektedir [39]. Alexnet, doğrultulmuş doğrusal birim etkilenim işlevi sayesinde, %25 eğitim hata oranına 6 kat daha hızlı şekilde ulaşmaktadır.

Derin sinir ağlarının eğitimi sırasında, algoritmanın eğitim verisindeki özellikleri öğrenmek yerine ezberlemesiyle aşırı uyum problemi görülebilmektedir. Bu durum, tıpkı bir öğrencinin derste çözülen soruları ezberleyip aynı soruyla karşılaştığında ezberden çözebilmesi ama çözüm için aynı bilgileri gerektiren farklı tarz bir soru gördüğünde çözememesine benzetilebilir. Benzer şekilde evrişimli sinir ağlarında da ağın eğitim verisinin özelliklerini öğrenmek yerine ezberlemesi, eğitim kümesinin doğrulama verisindeki resimler için başarılı sonuçlar gösterip, aynı sınıfa ait farklı bir resim gösterildiğinde başarısız olmasına neden olmaktadır.

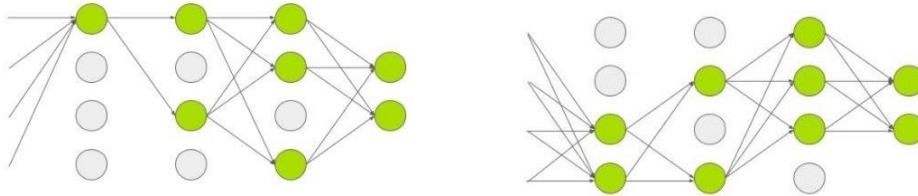


Şekil 2.5 Doğrultulmuş Doğrusal Birim İşlevinin Eğitim Hızına Etkisi

Alexnet evrişimli sinir ağı, aşırı uyum problemini önlemek için veri artırma ve düşürme yöntemlerini kullanmaktadır [39,43]. Veri artırma yöntemi, ağın büyüklüğünü ve dolayısıyla öğrenme kapasitesini artırmak için kullanılmaktadır. Bir evrişimli sinir ağının eğitim kümesinde aynı görüntünün döndürülmüş ya da aynalanmış versiyonlarını kullanmak, girdi boyutundaki alana denk gelen imge noktalarının yerlerini değiştirdiği için, ağı ezberlemek yerine öğrenmeye yönlendirir. Eğitim veri kümesinde bulunan bir köpek resminin aynalanmasıyla, aynı köpeğe ait ancak imge noktası değerleri değiştirilmiş bir köpek resmi elde edilmektedir. Böylece veri kümesinin basitçe iki katına çıkarılması mümkün olmaktadır. Bununla birlikte, Alexnet evrişimli sinir ağı mimarisinde, 256x256 boyutundaki resimlerden, rastgele kırılarak elde edilen 227x227 boyutundaki resimler de veri artırma yöntemi olarak kullanılmıştır. Bu sayede ana resimle oldukça benzer ancak imge noktası değerleri farklılaşmış yeni resimler oluşturularak veri kümesinin 2048 kat büyütülmesi mümkün olmuştur [39].

Aşırı uyum problemini önlemek için Alexnet evrişimli sinir ağında kullanılan bir diğer yöntem de düşürme tekniğidir [39]. Ağ içerisinde bulunan her bir gizli sinir hücresinin çıktısının %50 olasılıkla 0 olmasını sağlayarak, ağdan düşürülmesiyle

gerçekleştirilmektedir [43]. Bu teknikle ağdan düşürülen sinir hücrelerinin, o anki girdi için ileri ya da geri yayılıma etkileri sıfırlanmış olur. Böylece her bir girdinin ağı mimarisi içerisinde birbirinden farklı yolları takip etmesi sağlanarak aşırı uyum probleminin önüne geçilmektedir. Ek olarak, düşürme tekniğinin sağladığı diğer bir avantaj da her seferinde farklı sinir hücrelerinin ağdan düşürülmesi sayesinde, girdinin ait olduğu sınıfın en baskın özelliklerinin öğrenilmesidir. Şekil 2.6'da düşürme tekniğine bir örnek gösterilmektedir [40].



Şekil 2.6 Düşürme Tekniğinin Gösterimi

Alexnet algoritması birinci evrişimli katmanında $227 \times 227 \times 3$ boyutundaki resimleri girdi olarak kabul etmektedir. Bu katmanda her bir resim, aralık değeri 4, dolgu değeri 2 olan 11×11 boyutundaki 96 adet süzgeç kullanılarak evrişim işleminden geçmektedir. Birinci katmandaki evrişim işlemleri sonucunda $55 \times 55 \times 96$ boyutunda çıktılar üretilmektedir. Üretilen çıktılar doğrultulmuş doğrusal birim işlevinden geçerek bir sonraki katmana aktarılmaktadır.

Alexnet derin sinir ağı'nın ikinci katmanında ilk maksimum havuzlama katmanı bulunmaktadır. Bu katman, birinci evrişimli katmandan gelen $55 \times 55 \times 96$ 'lık çıktıları girdi olarak alıp, dolgu değeri 0, aralık değeri 2 olan 3×3 'lük süzgeçleri kullanarak üstüste binen maksimum havuzlama işlemiyle alt örnekleme işlemini gerçekleştirmektedir. Bu işlem sonucunda her bir girdiyi $27 \times 27 \times 96$ boyutuna düşürerek bir sonraki katmana iletmektedir.

Ağı üçüncü katmanında ikinci evrişimli katman yer almaktadır. Bu katman, ilk maksimum havuzlama katmanının ürettiği $27 \times 27 \times 96$ boyutundaki çıktıları girdi olarak kabul etmektedir. Üçüncü katmanda dolgu değeri 2, aralık değeri 1 olan 256 adet 5×5 boyutlarında süzgeçlerle evrişim işlemleri gerçekleştirilmektedir. Yapılan evrişim işlemleri sonucunda $27 \times 27 \times 256$ boyutlarında çıktılar üretilmektedir. Üretilen çıktılar,

doğrultulmuş doğrusal birim etkilenim işlevi uygulanarak dördüncü katmana iletilmektedir.

Dördüncü katmanda ikinci maksimum havuzlama katmanı bulunmaktadır. Üçüncü katmandaki evrişim işlemleri sonucu oluşan $27 \times 27 \times 256$ 'lık çıktılar, dördüncü katmanın girdilerini oluşturmaktadır. Bu katmanda da yine birinci maksimum havuzlama katmanı ile benzer şekilde dolgu değeri 0, aralık değeri 2 olan 3×3 'lük süzgeçler kullanılarak alt örnekleme işlemi gerçekleştirilmektedir. Gerçekleştirilen alt örnekleme işlemi sonucunda üretilen $13 \times 13 \times 256$ boyutundaki çıktılar doğrultulmuş doğrusal birim işlevinden geçirilerek beşinci katmana iletilmektedir.

Ağın beşinci katmanında üçüncü evrişim katmanı yer almaktadır. Dördüncü katmandan gelen $13 \times 13 \times 256$ boyutundaki çıktıları girdi olarak kabul eden bu katmanda, dolgu ve aralık değeri 1 olan, 3×3 boyutundaki 384 adet süzgeçle evrişim işlemleri gerçekleştirilmektedir. Evrişim işlemleri sonucunda oluşan $13 \times 13 \times 384$ boyutundaki çıktılar, bir kez daha doğrultulmuş doğrusal birim etkilenim işlevinden geçirilerek altıncı katmana aktarılmaktadır.

Altıncı katmanda dördüncü evrişim katmanı bulunmaktadır. Bu katmanda beşinci katmandan iletilen $13 \times 13 \times 384$ boyutundaki çıktılar, bir kez daha beşinci katmanda yer alan, dolgu ve aralık değeri 1 olan 3×3 boyutundaki süzgeçlerle evrişim işlemine tabi tutulurlar. Bu evrişim işlemi sonucunda girdi ve çıktılar arasında boyut değişikliği meydana gelmeden üretilen çıktılar, etkilenim işlevinden geçirilerek yedinci katmana iletilmektedirler.

Alexnet'in yedinci katmanında beşinci ve son evrişimli katman yer almaktadır. Yedinci katmanın girdilerini, altıncı katmandan gelen $13 \times 13 \times 384$ 'lük çıktılar oluşturmaktadır. Bu katmanda evrişim işlemleri, dolgu ve aralık değeri 1 olan 256 adet süzgeç üzerinden gerçekleştirilmekte ve işlem sonucunda $13 \times 13 \times 256$ boyutlarında çıktılar oluşturulmaktadır. Oluşturulan çıktılar, etkilenim işlevinden geçirilerek sekizinci katmana aktarılmaktadır.

Sekizinci katmanda üçüncü maksimum havuzlama katmanı yer almaktadır. İkinci ve dördüncü katmanlarda yer alan maksimum havuzlama katmanlarında olduğu

gibi, bu katmanda da dolgu deęeri 0, aralık deęeri 2 olan 3x3'lük süzgeçler kullanılarak, 6x6x256 boyutlarında çıktıları üretilip, etkilenim işlevinden geçirilerek dokuzuncu katmana aktarılmaktadır.

Dokuzuncu katmanda birinci tam bağlantılı katman bulunmaktadır. Bu katman, sekizinci katmandan gelen 6x6x256 boyutundaki çıktıları girdi olarak alıp, bu girdilere yanlılık deęerini de ekleyerek bir ağırlıklı toplama işlemi gerçekleştirmektedir. Gerçekleştirilen işlem sonucunda ürettięi 4096x1 boyutundaki çıktıları etkilenim işlevinden geçirilerek onuncu katmana aktarmaktadır.

Onuncu katmanda ikinci tam bağlantılı katman yer almaktadır. Bu katmanda dokuzuncu katmandan iletilen 4096x1 boyutundaki çıktıları yine ilk tam bağlantılı katmanda olduęu gibi yanlılık deęeri eklenerek ağırlıklı toplama işlemleri gerçekleştirilmekte ve yine 4096x1 boyutundaki çıktıları, etkilenim işlevi uygulanarak son katmana aktarılmaktadır.

Alexnet derin sinir aęının on birinci ve son katmanında üçüncü tam bağlantılı katman yer almaktadır. Üçüncü tam bağlantılı katman, onuncu katmandan gelen 4096x1 boyutundaki çıktıları kendisine girdi olarak kabul ederek yaptıęı ağırlıklı toplama işlemleriyle 1000x1 boyutunda bir çıktı meydana getirmektedir. Bu çıktıya eşiksiz en büyük etkilenim işlevinin uygulanması sonucunda, aęın birinci katmanına giren 227x227 boyutundaki resme dair sınıflandırma tahmin daęılımını içeren nihai çıktısı üretilmektedir.

Alexnet evrişimli sinir aęı, yukarıda bahsedilen mimari yapısı ve derin sinir aęları alanında getirdięi yenilikçi yaklaşımlarla literatürde çok çeşitli uygulamalarda tercih edilmiştir. Zamri, videolardan alınan anlık görüntüler üzerinden, insanların yaptıęı yürüme, koşma, zıplama, el çırpma vb. hareketleri %90'ın üzerinde isabetlilikle sınıflandırmıştır [1]. Sağlık sektöründe yapılan çalışmalarda, Mashrur, elektrokardiyografi görüntüleriyle Alexnet algoritmasını aktarım öğrenme yöntemiyle eğitmiş ve %97,9 isabetlilikle kalp atışı aykırılıklarını tespit edebilmiştir [2]. Benzer bir çalışmada David, mamografi görüntüleri üzerinden tümör tespitini %95,7 başarıyla gerçekleştirmiştir [14]. Gong, Alexnet algoritmasını avuçiçi tanıma amacıyla eğitmiş ve %99'a varan başarı oranları elde etmiştir [12].

Alexnet algoritmasının işbirlikçi robotlarla birlikte kullanımına dair literatür araştırılması yapılmış, ancak bu konudaki çalışmaların oldukça kısıtlı olduğu görülmüştür. Abbas, ev ortamında bulunan çeşitli nesnelerin sınıflandırılması ve Scorbob-ER5 işbirlikçi robot yardımıyla tut ve yerleştir uygulaması gerçekleştirmiştir [13]. Çalışmanın ana konusunu, robotun nesnelere tutabilmesi, ileri ve geri kinematik hesaplamaları oluşturmuştur.

2.3. Googlenet

Googlenet, 2014 yılında Google'da çalışan bir araştırmacı ekip tarafından geliştirilmiş derin evrişimli sinir ağıdır [44]. Havuzlama katmanları hariç 22, havuzlama katmanlarıyla birlikte toplam 27 katmandan oluşmaktadır. 2014 yılında düzenlenen ILSVRC Imagenet Geniş Çaplı Görsel Tanıma Yarışması'nda %6,67 top-5 hata oranıyla yarışmanın birincisi olmuştur [45]. Aynı zamanda ILSVRC 2012 şampiyonu Alexnet ve 2013 şampiyonu ZF-NET'i de top-5 hata oranı başarısında geçerek o güne kadarki en başarılı sonucu elde etmiştir [46].

Derin sinir ağlarında başarımların artışı sağlamanın bilinen en yaygın yöntemlerinden biri ağı boyutunu artırmaktır. Ağı boyutlarını artırmak iki şekilde mümkün olmaktadır. Birincisi ağı derinliğini, yani içerdiği katman sayısını artırmakla, ikincisi de ağı genişliğini yani her bir katmandaki sinir hücresi sayısını artırmakla sağlanmaktadır. Özellikle yeterli miktarda tanımlı eğitim verisi mevcutsa, yüksek başarımlı bir ağı model eğitmek için ağı derinliğinin ve genişliğinin artırılması tercih edilebilmektedir. Buna karşılık, derinliği ve genişliği artan evrişimli sinir ağlarında iki büyük problem gözlenmektedir. İlk olarak, ağı boyutlarının artmasıyla birlikte parametre sayısı artacağı için, ağı boyutları büyüdükçe aşırı uyum problemi göstermeye yatkınlaşmaktadır [44]. İkinci problem ise, ağı boyutlarının artışının getireceği ekstra hesaplama yüküdür. Örneğin, iki evrişimli katmanın art arda geldiği durumlarda evrişim süzgecindeki doğrusal bir artış, hesaplama yükünde üstel bir artışa neden olmaktadır. Ağı boyutlarının artmasıyla elde edilen kapasitenin ağı mimarisinde verimsiz kullanılması durumunda yapılan birçok hesaplama boşa gidebilmektedir. Googlenet evrişimli sinir ağı, ağı derinliğini ve genişliğini artırırken aynı zamanda bu iki problemin üstesinden gelme motivasyonu ile ortaya çıkmıştır [44].

Googlenet, mimarisi içerisinde bulunan başlangıç modülleriyle, ağın derinliğini artırarak başarımlarını sağlarken, parametre sayısını düşürerek hesaplama yükünün verimli bir şekilde kullanılmasını sağlamıştır. Googlenet mimarisi katmanları Çizelge 2.1’de gösterilmiştir [44].

Katmanlar içerisinde yer alan tüm evrişim işlemleri, başlangıç modülleri içerisindeki evrişimler de dâhil olmak üzere, doğrultulmuş doğrusal birim etkilenim işlevini kullanmaktadır [44]. “#3x3 azaltma” ve “#5x5 azaltma” sütunlarında yer alan sayılar, ilgili katmanda gerçekleştirilecek 3x3 ve 5x5’lik evrişim işlemleri öncesinde boyut azaltmak amacıyla gerçekleştirilecek 1x1’lik evrişim operasyonu sayısını ifade etmektedir.

Googlenet evrişimli sinir ağı mimarisi, girdilere dair kritik bilgilerin ağın içerisinde kayba uğramadan hesaplama yükünü olabildiğince düşürmek üzere tasarlanmıştır. Bu amaçla 1x1 boyutundaki evrişimler ve global ortalama havuzlama gibi o güne kadar evrişimli sinir ağlarında pek görülmemiş farklı teknikler kullanılmıştır. Ağın ilk katmanında aralık değeri 2 olan 7x7 boyutundaki süzgeçlerle evrişim işlemi gerçekleştirilerek 224x224 ölçülerindeki girdi boyutunun bir an önce düşürülmesi amaçlanmıştır. Girdi boyutu; birinci evrişim katmanı sonunda 112x112x64’e, sonrasında aralık değeri 2 olan 3x3’lük süzgeçlerle uygulanan maksimum havuzlama katmanından geçtiğinde ise 56x56x64’e düşmüştür.

İkinci evrişim katmanı, aralık değeri 1 olan 3x3’lük süzgeçlerle uygulanmıştır. Bu katmanda gerçekleşen evrişim işlemi öncesinde 64 adet 1x1’lik süzgeçle evrişim işlemi gerçekleştirildiği için derinlik değeri 2 olarak belirtilmiştir. İkinci evrişim katmanından sonra da, birinci evrişim katmanı sonrasında olduğu gibi 2 aralıklı 3x3’lük süzgeçler kullanılarak maksimum havuzlama işlemi yapılmış ve girdi boyutları 28x28x192 değerine düşürülmüştür. Böylece girdilerin, başlangıç katmanına girmeden önce genişlik ve yükseklik değerleri 8’er kat düşürülerek parametre sayısı ve hesaplama yükünün azaltılması amaçlanmıştır [45].

Googlenet derin sinir ağının en önemli özelliği, Çizelge 2.2’de görülen başlangıç modülüdür. Başlangıç modülü geliştirilirken, Lin ve Arora tarafından önceki yıllarda yapılan çalışmalardan etkilenilmiştir [44,47,48].

Çizelge 2.1 Googlenet Katmanları

Katman	Süzgeç Boyutu/ Aralık Değeri	Çıktı Boyutu	Derinlik	#1x1	#3x3 Azaltma	#3x3	#5x5 Azaltma	#5x5	#1x1 Havuzlama	#Parametre	#işlem
Evrışım	7x7/2	112x112x64	1							2,7K	34M
Maks. Havuzlama	3x3/2	56x56x64	0								
Evrışım	3x3/1	56x56x192	2		64	192				112K	360M
Maks. Havuzlama	3x3/2	28x28x192	0								
Başlangıç (3a)		28x28x256	2	64	96	128	16	32	32	159K	128M
Başlangıç (3b)		28x28x480	2	128	128	192	32	96	64	380K	304M
Maks. Havuzlama	3x3/2	14x14x480	0								
Başlangıç (4a)		14x14x512	2	192	96	208	16	48	64	364K	73M
Başlangıç (4b)		14x14x512	2	160	112	224	24	64	64	437K	88M
Başlangıç (4c)		14x14x512	2	128	128	256	24	64	64	463K	100M
Başlangıç (4d)		14x14x528	2	112	144	288	32	64	64	580K	119M
Başlangıç (4e)		14x14x832	2	256	160	320	32	128	128	840K	170M
Maks. Havuzlama	3x3/2	7x7x832	0								
Başlangıç (5a)		7x7x832	2	256	160	320	32	128	128	1072K	54M
Başlangıç (5b)		7x7x1024	2	384	192	384	48	128	128	1388K	71M
Ort. Havuzlama	7x7/1	1x1x1024	0								
Düşürme (%40)		1x1x1024	0								
Doğrusal		1x1x1000	1							1000K	1M
Eşiksiz En Yüksek Etkilerim		1x1x1000	0								

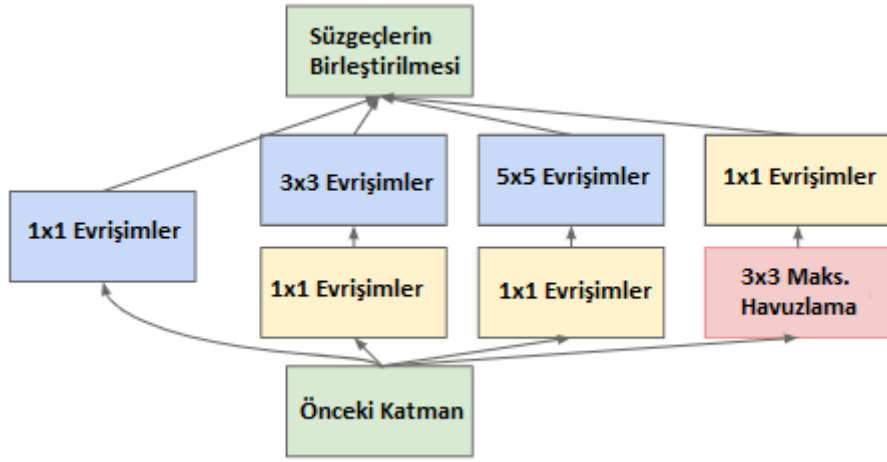
Çizelge 2.2 Googlenet Başlangıç Modülü

Başlangıç (3a)
Başlangıç (3b)
Maks. Havuzlama
Başlangıç (4a)
Başlangıç (4b)
Başlangıç (4c)
Başlangıç (4d)
Başlangıç (4e)
Maks. Havuzlama
Başlangıç (5a)
Başlangıç (5b)

Başlangıç modülü, toplamda 9 adet başlangıç katmanı ve 2 adet maksimum havuzlama katmanından oluşmaktadır. Üçüncü seviyedeki başlangıç katmanlarından dördüncü seviyeye ve dördüncü seviyedeki başlangıç katmanlarından beşinci seviyeye geçmeden önce maksimum havuzlama katmanlarının yer aldığı görülmektedir. Farklı seviyedeki başlangıç katmanlarına geçmeden önce maksimum havuzlama işlemi yapılarak, mevcut girdinin yükseklik ve genişlik imge noktası değerleri düşürülmektedir. Bu sayede girdinin ağ içerisinde ilerlerken, alt örnek sayısının ve dolayısıyla hesaplama yükünün azaltılması mümkün olmaktadır.

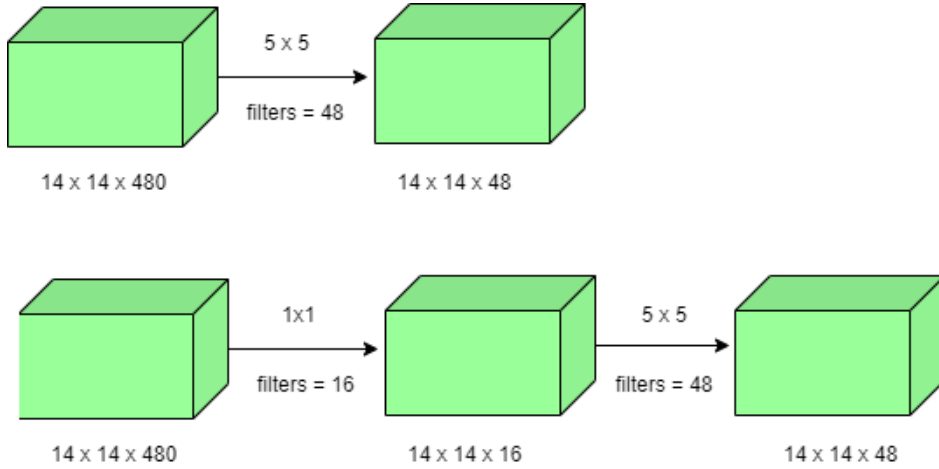
Başlangıç katmanı içerisinde gerçekleştirilen işlemler, Şekil 2.9'da verilmiştir [44]. Başlangıç katmanında, 1x1, 3x3, 5x5 boyutundaki evrişim işlemleri ve 3x3'lük maksimum havuzlama işlemleri paralel olarak gerçekleştirildikten sonra biraraya getirilip, tek bir nihai çıktı üzerinde birleştirilmektedir. Farklı boyutlardaki evrişim süzgeçlerinin kullanılma amacı, girdi resminde bulunan nesnelerin, çoklu ölçeğe daha iyi bir şekilde ele alınmasıdır [46]. Hesaplama yükünün verimli kullanımı amaçlandığından, 3x3'lük ve 5x5'lik evrişim işlemlerinin öncesine, 3x3'lük maksimum havuzlama işleminin ise sonrasına eklenen 1x1 boyutundaki evrişim süzgeçleriyle, hem

girdi boyutlarının düşürülmesi, hem de doğrultulmuş doğrusal birim etkilenim işlevi kullanımı sağlanmıştır



Şekil 2.7 Başlangıç Katmanı

. 1x1 boyutundaki evrişim süzgeçlerinin, girdi boyutlarını ve hesaplama yükünü düşürmeye yönelik etkisi Şekil 2.10'da gösterilmiştir [46].



Şekil 2.8 1x1 Evrişim Süzgeçleriyle Boyut Düşürme

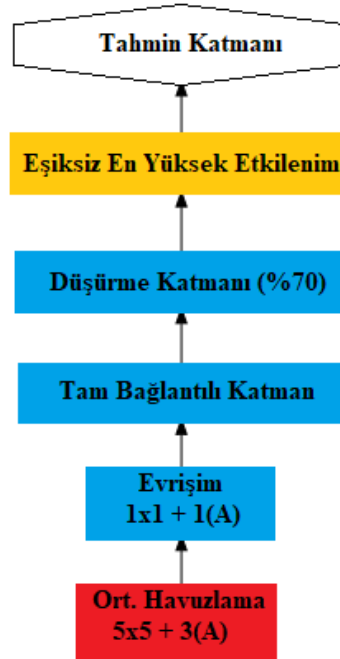
Şekil 2.10'da üst tarafta gösterilen 14x14x480 boyutundaki girdiye 5x5 boyutundaki 48 adet evrişim süzgeci uygulandığında toplam hesaplama yükü;

$$(14 \times 14 \times 48) \times (5 \times 5 \times 480) = 112.9 \text{ M olmaktadır.}$$

5x5'lik evrişim işlemi öncesinde 1x1 boyutunda 16 adet evrişim süzgeci eklendiğinde ise toplam hesaplama yükü;

$(14 \times 14 \times 16) \times (1 \times 1 \times 480) + (14 \times 14 \times 48) \times (5 \times 5 \times 16) = 1.5M + 3.8M = 5.3M$ olmaktadır. Araya eklenen 1×1 'lik evrişim süzgeçleri sayesinde toplam hesaplama yükünün yaklaşık $112.9M$ 'dan $5.3M$ 'a düşerek yaklaşık 24 kat azaltıldığı görülmektedir.

Googlenet derin sinir ağı, sahip olduğu derin mimarisi dolayısıyla yaşanabilecek kaybolan gradyan problemi ve aşırı uyuma yatkınlık problemini önlemek için yardımcı sınıflandırıcılar kullanmaktadır [44]. Bu yardımcı sınıflandırıcılar, mimarinin ortalarında, başlangıç modülü içerisindeki üçüncü (başlangıç 4a) ve altıncı (başlangıç 4d) başlangıç katmanlarında bulunmaktadır. Geri yayılım sürecinde ağırlıkların güncellenmesi sırasında, ağı derin katmanlarındaki özelliklerin ihmal edilebilir seviyede değişmesiyle ağı öğrenme kapasitesinin durma noktasına gelmesi ve aşırı uyum problemi görülebilmektedir [39,43,44,48]. Bu problemi önlemek amacıyla kullanılan yardımcı sınıflandırıcıların genel mimarisi Şekil 2.11'de verilmiştir [44].



Şekil 2.9 Yardımcı Sınıflandırıcı Mimarisi

Yardımcı sınıflandırıcı mimarisinin ilk katmanında, aralık değeri 3 olan 5×5 boyutundaki süzgeçlerin kullanıldığı bir ortalama havuzlama katmanı bulunmaktadır. İkinci katmanda boyut düşürme ve doğrultulmuş doğrusal birim etkilenimi amacıyla 128 adet 1×1 'lik evrişim süzgeçinden oluşan bir evrişim katmanı yer almaktadır.

Evrişim katmanı çıktıları, sırasıyla 1024 çıktısı olan bir tam bağlantılı katman ve %70 düşürme oranına sahip bir düşürme katmanından geçmektedir. Son olarak, düşürme katmanı çıktılarına eşiksiz en büyük etkilenim işlevi uygulanmakta ve çıktılar tahmin katmanına iletilmektedir. Tahmin katmanında, Googlenet mimarisi son adımında yer alan ana sınıflandırıcıyla benzer şekilde 1000 adet sınıf içeren bir sınıflandırma işlemi gerçekleştirilmektedir. Yardımcı sınıflandırıcılar, ağın sadece eğitimi sırasında kullanılmaktadır. Ağın orta bölümlerinde bir sınıflandırma işlemi gerçekleştirerek, eğitim sırasında hesaplanan kaybı, ağın toplam kaybına geri eklemek amacıyla kullanılmaktadırlar. Yardımcı sınıflandırıcıların kaybı, 0.3 ağırlık değeriyle çarpılarak eniyelenmiştir [44].

Başlangıç modülünün bitiminde, aralık değeri 1 olan 7×7 'lik imge noktası alanları içerisindeki, o ana kadar üretilmiş olan tüm özellik haritasının ortalama imge noktası değerinin alınmasını sağlayan ortalama havuzlama katmanı bulunmaktadır. Ortalama havuzlama katmanı, girdinin genişlik ve yükseklik boyutunu 7×7 'den 1×1 'e indirilmesini sağlamaktadır. Tam bağlantılı katman yerine ortalama havuzlama katmanı kullanılarak, ağın, ilk nesne tahmininde isabetlilik oranında %0.6 oranında iyileşme gözlenmiştir [44]. Ortalama havuzlama katmanından sonra, %40 düşürme oranına sahip bir düşürme katmanı yer almaktadır. Eğitim sırasında bu katmanda yer alan her bir sinir hücresinin %40 ihtimalle ağdan düşürülmesi sağlanarak ağın aşırı uyum problemi yaşamamasına karşı önlem alınmaktadır. Bir sonraki katmanda, içerisinde imajnet veri kümesinde bulunan 1000 sınıfı temsil eden 1000 adet gizli birim bulunan bir doğrusal katman bulunmaktadır. Aynı zamanda bu doğrusal katman, aktarım öğrenme yöntemiyle yeni veri kümelerini eğitip, yeni sınıflara ait tahmin yapılmasını kolaylaştırmaktadır [44]. Ağın son katmanında ise eşiksiz en yüksek etkilenim işlevini kullanan bir katman bulunmaktadır. Bu katman, girdinin, tanımlı nesne sınıflarından hangisine ait olabileceğinin olasılıksal dağılımını içeren bir çıktı vektörü üretmektedir. Bu nedenle girdinin, tanımlı sınıflar içerisinde hangi sınıfa ait olabileceğine dair olasılıkların toplamı her zaman 1'e eşit olmaktadır.

Googlenet evrişimli derin sinir ağı, özellikle içinde barındırdığı başlangıç katmanları sayesinde kaybolan gradyan ve aşırı uyum problemlerini aşarak oluşturduğu derin ağ mimarisiyle literatürde birçok farklı alanda kullanılmıştır. Han, hava ve uzay araçlarından çekilen uzaktan algılanan resimlerin sınıflandırılması için 5 farklı

algoritma içerisinde en başarılı sonuçları Googlenet algoritmasıyla almıştır [15]. Googlenet algoritması içerisindeki başlangıç katmanına 7x7'lik evrişim süzgeçleri eklemiş, 3 farklı uzaktan algılamalı veri kümesiyle yaptığı çalışmalarda %95'in üzerinde başarı oranı elde etmiştir [15]. Zhong, Alexnet ve Googlenet algoritmalarının, Çince el yazısı tanıma başarımlarını karşılaştırmış ve Googlenet'in top-1, top-2, top-5 ve top-10 isabetlilik oranlarının hepsinde Alexnet'ten daha yüksek başarımlar gösterdiğini tespit etmiştir [16]. Googlenet algoritmasını farklı özellik çıkarma işlemleriyle biraraya getirerek, %96'nın üzerinde ilk tahminde başarımlar elde etmiştir [16]. Fang, Alexnet ve Googlenet derin sinir ağlarının, bilgisayarlı tomografi görüntülerinden akciğer kanseri tespiti başarımlarını incelemiş, Googlenet'in iki boyutlu tomografi görüntülerinde %1.1, üç boyutlu görüntülerde ise %2 daha başarılı sonuç verdiğini göstermiştir [17]. Salavati, otonom sürüş ve navigasyon sistemleri için engel tespiti çalışmasında, özellik çıkarımları için Googlenet başlangıç katmanını kullanmış ve %94,81 isabetlilikle başarımlarına ulaşmıştır [18]. Ma, filizlenen patateslerin tespiti amacıyla LeNet, Alexnet ve Googlenet derin sinir ağlarının başarımlarını araştırmış, en başarılı tahmin yüzdesinin %80,56 oranıyla Googlenet tarafından gerçekleştirildiğini gözlemlemiştir [19].

Googlenet algoritmasının, robotlarla birlikte kullanımına dair literatür araştırması yapılmış ve bu alanda yapılmış birkaç çalışmaya rastlanmıştır. Massah, kivi meyvesinin hasat verimliliği tahmininde bulunmak amacıyla tekerlekli ve kameralı bir robotla bitkilerin fotoğraflarını çekerek, fotoğraf üzerindeki kivi meyvelerini tespit etmeye yarayan bir makine görü sistemi geliştirmiştir [3]. Massah, çalışmasında önerdiği yöntemle %96 başarımla meyve sayılarını doğru tahmin ederken, Googlenet algoritmasının %90 başarımla çalıştığını gözlemlemiştir [3]. Zhang, robot süpürgeler gibi iç mekân robotları için hareketli ve sabit engelleri tespit edip güzergâh planlaması yapma çalışmasında engelleri tespit amacıyla Googlenet algoritmasından yararlanmıştır [4]. Chrzanowski, küçük evcil hayvanların ev ortamında karşılaşılabileceği kazalardan onları korumak amacıyla takip eden ve tehlike durumunda uyarı veren tekerlekli kameralı mobil bir robot tasarlamıştır [5]. Robotun ilgili evcil hayvanı ve ona yaklaşan insanları tespit edebilmesi için Googlenet evrişimli sinir ağı kullanılmıştır [5]. Googlenet algoritmasının robotlarla birlikte kullanımını içeren bu çalışmaların [3–5] ortak özelliği, nesne tespiti ve sınıflandırma işlemlerinin yazılımsal sınıflandırma işlemiyle sınırlı kalması, fiziksel bir sınıflandırma işlemi yapılmamasıdır. Bu tez

çalışmasında yapılacak olan nesne tespiti ve sınıflandırma işlemi yazılımsal olarak yapılan sınıflandırmayla kalmayıp, robot yardımıyla yazılımdan gelen bilgileri kullanarak nesnelere fiziksel olarak da donanım tarafında sınıflandırılmasını sağlayacaktır.

2.4. Squeezenet

Squeezenet derin sinir ağı, 2016 yılında geliştirilmiştir. Maksimum tahmin isabetliliği sağlamaya çalışmak yerine, bilinen ve gelişmiş bir derin sinir ağı olan Alexnet ile aynı isabetliliği, çok daha düşük parametre ve model boyutuyla gerçekleştirme motivasyonu ile ortaya çıkmıştır [49]. Aynı isabetliliği sağlayan daha küçük bir evrişimli sinir ağının kullanılmasının üç temel avantajı bulunmaktadır. Bunlar;

- Model boyutu ve parametre sayısının düşüklüğü dolayısıyla ağın eğitim süresinin daha kısa olması. Daha az parametre içeren bir modelin hesaplama yükü daha düşük olacağı için ağın transfer öğrenimi ile eğitime süresi daha kısa olmaktadır.
- Düşük model boyutu sayesinde modelin güncellenmesi için gereken veri aktarımının daha az olması sayesinde güncelleme kolaylığı. Bu sayede otonom sürüş sistemlerine sahip otomobillerde engel tanıma, çarpışma önleme, şerit takibi vb. amaçlarla kullanılan modellerin, iyileştirme amaçlı güncellemelerinin daha kolay yapılması ve daha sık güncelleme alabilmesi mümkün olmaktadır.
- FPGA gibi sınırlı hafızaya sahip gömülü sistemler üzerinde modelin çalıştırılabilmesi. Bu sayede geliştirilen modelin, düşük belleğe sahip platformlar üzerinde dahi çalıştırılması mümkün kılınmıştır.

Squeezenet evrişimli sinir ağı, bu avantajların sağlandığı bir derin sinir ağını gerçekleştirmek üzere tasarlanmıştır [49].

Squeezenet mimarisi tasarlanırken, Alexnet modeli seviyesindeki isabetliliği daha düşük parametre sayısı ile gerçekleştirmek için üç temel strateji izlenmiştir. İlk olarak, 3x3 boyutundaki evrişim süzgeçleri yerine, ağın genelinde çoğunlukla 1x1 boyutundaki evrişim süzgeçleri kullanılmıştır. 1x1'lik süzgeçlerin, 3x3 boyutundaki süzgeçlere göre 9 kat daha az parametre oluşturması nedeniyle bu yöntem tercih

edilmiştir. İkinci olarak, ağıın içerisinde yer alan 3x3 boyutundaki evrişim süzgeçlerinin girdi boyutlarının küçültülmesi amaçlanmıştır. 3x3 boyutunda bir evrişim süzgeci kullanıldığında toplam parametre sayısı; girdi boyutları, uygulanan süzgeç sayısı ve süzgecin boyutlarının çarpımıyla doğru orantılı olduğu için, 3x3'lük evrişim süzgeçlerinin uygulanacağı girdilerin boyutlarının azaltılması hedeflenmiştir. SqueezeNet mimarisi, içerisinde bulunan sıkıştırma katmanlarıyla, 3x3'lük evrişim süzgeçleri uygulanmadan önce boyutların düşürülmesini sağlamaktadır [49,50]. Son olarak da evrişim katmanlarının daha büyük özellik haritaları oluşturabilmeleri için, alt örnekleme ağıın daha ileri katmanlarında yapılması sağlanmıştır. Özellik haritalarının boyutları, evrişim işlemine giren girdi boyutu ve alt örnekleme ağıın hangi evrişim katmanlarında yapılacağıın seçimiyle kontrol edilmektedir. Evrişim ya da havuzlama katmanlarında aralık değeri 1'den büyük ayarlanmasıyla alt örnekleme, yani girdi boyutlarından daha düşük boyutta bir çıktı üretilmesi, literatürde yaygın olarak kullanılmaktadır [39,44]. Ağıın ilk katmanlarında boşluk değeri 1'den büyük süzgeçler kullanıldığında, katmanların büyük çoğunluğunda küçük özellik haritaları oluşmaktadır. Tersine şekilde, ağıın büyük çoğunluğunda süzgeçlerin aralık değeri 1 olup, 1'den büyük aralık değeri olan süzgeçler ağıın ileri katmanlarında kullanıldığında ise katmanların çoğunluğunda daha büyük özellik haritaları oluşmaktadır. He ve Sun, alt örnekleme geciktirilmesinin dört farklı derin sinir ağı mimarisindeki etkilerini incelemiş, tamamının daha yüksek sınıflandırma isabetliliği sağladığını göstermiştir [51]. SqueezeNet derin sinir ağı mimarisi de tahmin isabetliliğini artırmak için alt örnekleme geciktirilmesi stratejisini benimsemiştir [49]. Birinci ve ikinci stratejilerin parametre sayısını azaltmaya, üçüncü stratejinin ise tahmin isabetliliğini artırmaya odaklandığı görülmektedir.

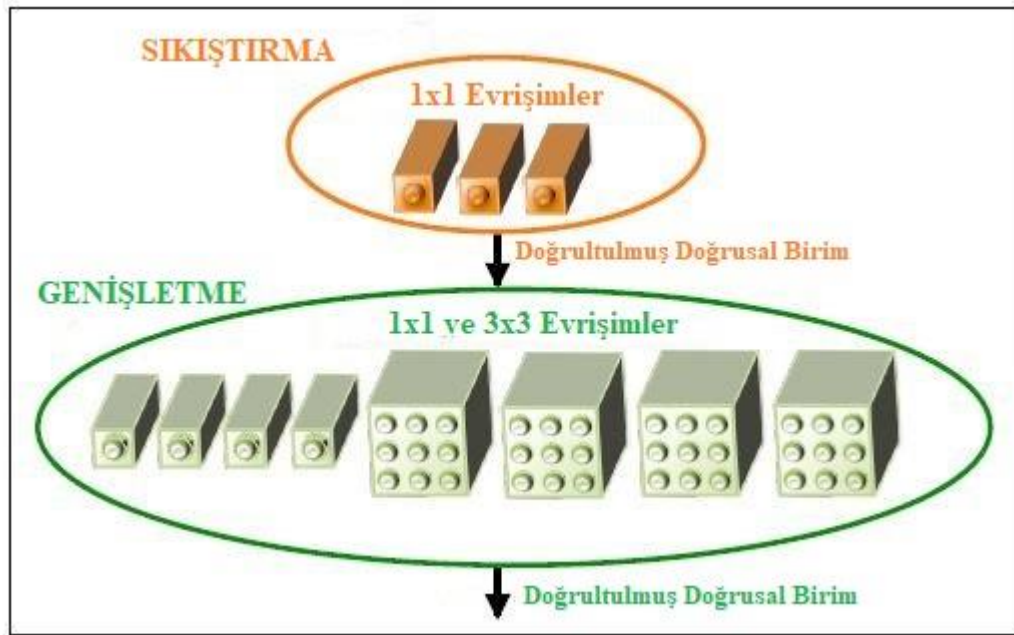
SqueezeNet ağıın genel mimarisi ve katman yapısı Çizelge 2.3'te gösterilmiştir [49]. SqueezeNet derin sinir ağı, girdi olarak 224x224x3 boyutundaki resimleri kabul etmektedir. Mimarinin ilk katmanında aralık değeri 2 olan 7x7 boyutundaki 96 adet evrişim süzgeci bulunan bir evrişimli katman bulunmaktadır. Ağıın başında ve sonunda yer alan 2 adet evrişimli katmanın arasında, SqueezeNet mimarisinin temelini oluşturan 9 adet ateş modülü yer almaktadır. Ağıın başından sonuna doğru ilerledikçe, ateş modülü başına kullanılan süzgeç sayısı katlanarak artmaktadır. Birinci evrişim katmanı ile dördüncü ve sekizinci ateş modüllerinden sonra, boşluk değeri 2 olan 3x3 boyutundaki süzgeçlerin kullanıldığı maksimum havuzlama katmanları bulunmaktadır.

Çizelge 2.3 SqueezeNet Katmanları

Katman İsmi/Tipi	Çıktı Boyutu	Süzgeç Boyutu/ Aralık Değeri	Derinlik	S1x1 (#1x1 Sıkıştırma)	E1x1 (#1x1 Genişletme)	E3x3 (#3x3 Genişletme)	S1x1 Seyreklik	E1x1 Seyreklik	E3x3 Seyreklik	Bit Sayısı	#Budama Öncesi Parametre	#Budama Sonrası Parametre
Girdi Resmi	224x224x3											
Evişim 1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6 bit	14.208	14.208
Maks. Havuzlama 1	55x55x96	3x3/2	0									
Ateş 2	55x55x128		2	16	64	64	100%	100%	33%	6 bit	11.920	5.746
Ateş 3	55x55x128		2	16	64	64	100%	100%	33%	6 bit	12.432	6.258
Ateş 4	55x55x128		2	32	128	128	100%	100%	33%	6 bit	45.344	20.646
Maks. Havuzlama 4	27x27x256	3x3/2	0									
Ateş 5	27x27x256		2	32	128	128	100%	100%	33%	6 bit	49.440	24.742
Ateş 6	27x27x384		2	48	192	192	100%	50%	33%	6 bit	104.880	44.700
Ateş 7	27x27x384		2	48	192	192	50%	100%	33%	6 bit	111.024	46.236
Ateş 8	27x27x512		2	64	256	256	100%	50%	33%	6 bit	188.992	77.581
Maks. Havuzlama 8	13x12x512	3x3/2	0									
Ateş 9	13x13x512		2	64	256	256	50%	100%	30%	6 bit	197.184	77.581
Evişim 10	13x13x1000	1x1/1 (x1000)	1				20% (3x3)			6 bit	513.000	103.400
Ort. Havuzlama 10	1x1x1000	13x13/1	0									
Etkilemler											1.248.424 (Toplam)	421.098 (Toplam)
Parametreler												
Sıkıştırma Bilgisi												

İkinci evrişim katmanının arkasında yer alan ağın son katmanında ise ortalama havuzlama katmanı yer almaktadır. Havuzlama katmanlarının ağırlıklı olarak ağın son katmanlarında yer almasının, büyük özellik haritaları oluşturmayı hedefleyen üçüncü stratejiyle uyumlu olduğu görülmektedir.

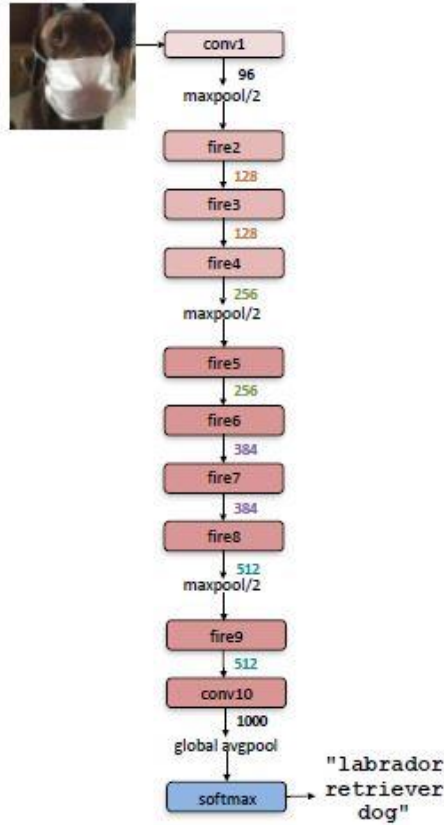
Ateş modülleri, SqueezeNet evrişimli sinir ağının temelini oluşturmaktadır. Sadece 1x1 boyutundaki evrişim süzgeçlerinin kullanıldığı bir sıkıştırma katmanı ile, 1x1 ve 3x3 boyutundaki evrişim süzgeçlerinin kullanıldığı genişletme katmanlarının birleşiminden oluşmaktadır. Ateş modülü genel yapısı Şekil 2.13'te gösterilmektedir [49].



Şekil 2.10 Ateş Modülü

Ateş modüllerinin sıkıştırma bölümünde, birinci stratejiyle uygun olarak sadece 1x1 boyutundaki evrişim süzgeçleri kullanılmaktadır. Genişletme bölümünde ise 1x1 ve 3x3 boyutundaki evrişim süzgeçleri birlikte yer almaktadır. Şekil 2.12'de yer alan "S1x1" sütunu, ateşleme modülünün sıkıştırma bölümünde yer alan 1x1'lik süzgeç sayısını, "E1x1" sütunu ateşleme modülünün genişletme bölümünde yer alan 1x1'lik süzgeç sayısını, "E3x3" sütunu ise ateşleme modülünün genişletme bölümünde yer alan 3x3'lük süzgeç sayısını ifade etmektedir. Ağı katmanları arasında ilerledikçe artan derinlikle birlikte daha çok süzgece ihtiyaç duyulduğu ve buna bağlı olarak da kullanılan süzgeç sayılarının artış gösterdiği görülmektedir. Birinci stratejiye uygun

olarak, sıkıştırma bölümünde yer alan 1×1 'lik süzgeç sayısı, genişletme bölümünde yer alan 1×1 ve 3×3 'lük süzgeç sayılarının toplamından daha düşük olacak şekilde ayarlanmıştır [49]. Sıkıştırma ve genişletme katmanlarındaki tüm evrişim işlemlerinde etkilenim işlevi olarak doğrultulmuş doğrusal birim işlevleri [52] kullanılmıştır. Dokuzuncu ateşleme modülünün sonrasında, düşürme oranı %50 olacak şekilde bir düşürme işlevi [43] kullanılmıştır. Sınıflandırma için eşiksiz en yüksek etkilenim işlevi kullanılmıştır. Sınıflandırma öncesinde tam bağlantılı katman yerine, evrişimli katman ve ortalama havuzlama katmanı yer almaktadır [44,48,49]. SqueezeNet algoritmasının, $224 \times 224 \times 3$ boyutlarındaki bir resmi alıp bu resme ait tahmin çıktısını ürettiği işlemlerin özeti Şekil 2.14'te gösterilmiştir [49].



Şekil 2.11 SqueezeNet Algoritması Genel Akışı

SqueezeNet derin sinir ağı, 4.8MB'lık model boyutuyla, 240MB'lık model boyutuna sahip olan Alexnet algoritmasından 50 kat daha düşük bir model boyutuna sahip olmasına karşılık top-1 isabetlilik oranında daha yüksek ve top-5 isabetlilik oranlarında Alexnet algoritmasıyla aynı isabetlilik oranlarını yakalamıştır. Ayrıca derin sıkıştırma teknikleri [53] kullanılarak SqueezeNet model boyutunun 0.47MB'a

düşürülmesi mümkün olmuştur. Böylece Alexnet algoritması seviyesinde bir isabetlilik oranı, 510 kat daha düşük bir model boyutuyla gerçekleştirilmiştir [49]. Squeezenet evrişimli derin sinir ağının, Alexnet ile model boyutu ve isabetlilik oranı karşılaştırmaları, Çizelge 2.4’te verilmiştir [49].

Çizelge 2.4 Squeezenet ve Alexnet Model Boyutu ve İsbetlilik Karşılaştırması

CNN architecture	Compression Approach	Data Type	Original → Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
AlexNet	None (baseline)	32 bit	240MB	1x	57.2%	80.3%
AlexNet	SVD (Denton et al., 2014)	32 bit	240MB → 48MB	5x	56.0%	79.4%
AlexNet	Network Pruning (Han et al., 2015b)	32 bit	240MB → 27MB	9x	57.2%	80.3%
AlexNet	Deep Compression (Han et al., 2015a)	5-8 bit	240MB → 6.9MB	35x	57.2%	80.3%
SqueezeNet (ours)	None	32 bit	4.8MB	50x	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	8 bit	4.8MB → 0.66MB	363x	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	6 bit	4.8MB → 0.47MB	510x	57.5%	80.3%

Squeezenet algoritması, düşük model boyutuna karşılık sağladığı isabetlilik oranıyla literatürde çeşitli alanlardaki uygulamalarda tercih edilmiştir. Akpınar, göğüs – x-ray filmlerinden aykırılık tespiti çalışmasında Squeezenet algoritmasını kullanmış ve normal ve anormal olarak sınıflandırdığı x-ray görüntülerinde %90.95 isabetlilik oranı elde etmiştir [6]. Nakamichi, mikroskop görüntüleri içerisinde tümör hücresi sınıflandırması çalışmasında Squeezenet algoritmasını kullanarak %89,86 isabetlilikle doğru pozitif oranı elde etmiştir [7]. Durmuş, domates yapraklarında hastalık tespiti çalışmasında Alexnet ve Squeezenet algoritmalarının başarımlarını karşılaştırmış, Squeezenet algoritmasıyla %94,3; Alexnet algoritmasıyla ise %95,6 isabetlilik oranı gözlemlemiştir [8]. Hassanpour, dokümanların sayfa resimleri üzerinden yazının başlığı, tablo ve şekiller, doküman üzerindeki el yazıları vb. özelliklerin tespiti çalışmasında 5 farklı algoritmanın başarımlarını kıyaslamış, Squeezenet algoritmasıyla %74,40 isabetlilik başarımları elde etmiştir [9]. Wang, akıllı trafik gözetim sistemleri için otomobil, otobüs, kamyon gibi motorlu araçların tespiti çalışmasında Squeezenet algoritmasını modifiye ederek kullanmış ve %97’nin üzerinde başarılı isabetlilik oranları elde etmiştir [10].

Squeezenet derin sinir ağının robotlarla birlikte kullanımına dair yapılan literatür taramasında bu alanda oldukça sınırlı çalışma olduğu görülmüştür. Jiang, ev ortamında bulunabilen kumanda, makas, spatula gibi çeşitli nesnelerin sınıflandırılması için

SqueezeNet algoritmasını kullanmış, algoritmanın yaptığı nesne tahminine göre robotun nesneyi doğru şekilde kavramasını hedeflemiştir [11]. Yapılan çalışmada, SqueezeNet algoritmasının nesne tahminine göre robot kolun uç işlevcisinin nesneyi farklı şekillerde kavrayabileceği bir yazılım hazırlanmış ve %88.6 isabetlilikle nesnelerin doğru tespit edildiği gözlemlenmiştir [11].

3. DERİN SİNİR AĞLARINDA AKTARIM ÖĞRENİMİ YÖNTEMİYLE SINIFLANDIRMA

3.1. Aktarım Öğrenme Yazılımının Hazırlanması

İnsan beyni, günlük hayatta karşılaştığı problemlerin çözümünde daha önce edindiği tecrübelerden yararlanır. Aktarım öğrenmesi ise tıpkı insanların daha önce edindikleri bilgileri başka bir durumda kullanabilmesi gibi, makine öğrenmesi yöntemlerinin bir problemi çözerken elde ettiği bilgileri, farklı ancak benzer başka bir problemin çözümünde kullanabilmesini sağlar.

Aktarım öğrenme tekniğiyle bir derin öğrenme modelinin istenilen problemi çözebilmesi için yeniden eğitilmesinin, derin öğrenme modelinin en baştan oluşturulmasına göre önemli birkaç avantajı vardır. Öncelikle bir derin öğrenme modelinin sıfırdan oluşturulması için çok büyük bir veri kümesine ihtiyaç duyulmaktadır. Bir nesne sınıflandırma problemine yüksek başarımla tahminler oluşturan bir derin öğrenme modelinin eğitilebilmesi için ilgili her bir nesnenin yüz binlerce görüntüsünden oluşan bir veri kümesine ihtiyaç duyulmaktadır [36,54]. Ayrıca, gerekli veri kümesi oluşturulsa bile, modelin eğitilmesi en güncel ve güçlü donanımlı bilgisayarlarla bile günlerce sürebilmektedir. Aktarım öğrenme yönteminde ise önceden eğitilmiş modellerin ince ayar yapılarak yeniden kullanılması mümkün olduğu için çok daha az miktarda veriye ihtiyaç duyulmaktadır. Modelin sıfırdan eğitilmesi için milyonlarca veri kümesi gerekirken, aktarım öğrenme yönteminde birkaç yüz adetlik veri kümesleriyle dahi yüksek başarımlı sonuçlar alınabilmektedir. Bununla birlikte, aktarım öğrenmesinde, önceden eğitilmiş modelin genellikle köşeler ve renkler gibi düşük seviye özelliklerin öğrenildiği ilk katmanları değiştirilmeden kullanıldığı için, çok daha hızlı bir şekilde eğitilebilmektedir.

Bu çalışmanın konusu olan nesne sınıflandırma probleminin derin öğrenme teknikleriyle çözümü için, daha az miktarda veri kümesine ihtiyaç duyulması, modelin çok daha hızlı bir şekilde eğitilebilmesi ve önceden eğitilmiş modele tam bağlantılı katmanlar eklenerek başarımların artırılabilmesi gibi avantajlarından dolayı

aktarım öğrenme yöntemi kullanılmıştır. Tez çalışması kapsamında Alexnet, Googlenet ve SqueezeNet derin öğrenme algoritmalarının başarımları kıyaslanacağı ve bu nedenle yapılan her bir ince ayarın sonuçlarını gözlemlemek için defalarca modelin yeniden eğitilmesi ihtiyacı, bu tercihte etkili olmuştur.

Aktarım öğrenme yazılımının geliştirileceği ortam için, hâlihazırda sinir ağları ve derin öğrenmeyle ilgili kütüphanelerinin bulunması ve geliştirme kolaylığı göz önünde bulundurularak MATLAB programlama dili ve geliştirme ortamı seçilmiştir. MATLAB geliştirme ortamının Çizelge 3.1’de bulunan evrişimli sinir ağlarını ön eğitilmiş şekilde bulundurmasıyla kullanım kolaylığı avantajı bulunmaktadır [55].

Çizelge 3.1 MATLAB’da Tanımlı Ön Eğitilmiş Evrişimli Sinir Ağları

Network
squeezenet
googlenet
inceptionv3
densenet201
mobilenetv2
resnet18
resnet50
resnet101
xception
inceptionresnetv2
shufflenet
nasnetmobile
nasnetlarge
darknet19
darknet53
efficientnetb0
alexnet
vgg16
vgg19

Bu çalışma kapsamında Çizelge 3.1’deki evrişimli sinir ağlarından, literatürde de yaygın olarak kullanılan Alexnet, Googlenet ve SqueezeNet derin sinir ağlarının, tezin konusu olan vida, pul ve somun nesnelere aktarım öğrenme yöntemiyle

sınıflandırabilme becerileri ve başarımları incelenmiştir. Aktarım öğrenme yazılımı hazırlanırken, incelenen her bir evrişimli ağ için farklı bir aktarım öğrenme yazılımı hazırlamak yerine, küçük modifikasyonlarla üç evrişimli sinir ağı için de ortak kullanılabilir bir yazılım geliştirilmesi esas alınmıştır.

Bu noktada karşılaşılan problemlerden biri, incelenen evrişimli sinir ağlarının eğitilmesi için farklı giriş boyutlarında veri kümelerine ihtiyaç duyulması olmuştur. Alexnet ve Squeezenet ağları 227x227x3 boyutunda girişlerden oluşan bir veri kümesine ihtiyaç duymaktayken, Googlenet ağı 224x224x3 boyutunda girişlerden oluşan bir veri kümesine ihtiyaç duymaktadır. Bu problemin ön hazırlık aşamasında üstesinden gelmek için, veri kümesindeki nesne resimlerini, evrişimli sinir ağının ihtiyaç duyduğu boyutlara dönüştürmek amacıyla MATLAB geliştirme ortamında küçük bir program hazırlanmıştır. Hazırlanan program, boyutu değiştirilecek veri kümesini kaynak dizinden alıp, veri kümesindeki resim sayısını hesaplamakta, sonrasında bir döngü içerisinde tüm nesnelere bitene kadar her bir nesne resmini teker teker alıp, resmin boyutunu istenilen şekilde değiştirerek hedef dizine kaydetmektedir. Bu program vasıtasıyla Alexnet ve Squeezenet evrişimli sinir ağlarının eğitilmesi için kaynak veri kümesindeki tüm resimler 227x227x3 boyutuna; Googlenet evrişimli sinir ağı için ise 224x224x3 boyutuna dönüştürülmüştür. Bu sayede bu çalışmada kullanılacak 3 derin sinir ağının da aktarım öğrenmesi için ihtiyaç duyduğu uygun veri kümelerinin oluşturulması mümkün kılınmıştır.

Aktarım öğrenme yazılımı koşturulduğunda, ilk olarak ağın eğitileceği kaynak veri kümesinin adresi tariflenir. Veri kümesindeki fotoğrafların ne kadarının eğitim amaçlı, ne kadarının test amaçlı olarak kullanılacağı yüzdesel olarak belirtilir. Bu çalışmada kullanılan veri kümesinin %70'i eğitim verisi olarak, %30'u ise doğrulama verisi olarak ayrılmıştır. Eğitim verisi oranı %70'in altına düştüğü zaman algoritmaların nesne tanıma başarımlarında düşüş görülmesi, doğrulama verisinin %30'un aşağısında ayarlandığında ise benzetim test sonuçlarının daha büyük sapmalar göstermesi göz önüne alınarak veri kümesindeki eğitim verisi ve doğrulama verisi yüzdeleri belirlenmiştir. Hangi fotoğrafların eğitim amaçlı, hangilerinin test amaçlı kullanılacağı yazılım tarafından rastgele seçilmesi sağlanmıştır. Devamında, aktarım öğrenme yöntemiyle yeniden eğitilecek ön eğitimli derin sinir ağı yazılım tarafından çağırılır. Bu aşamada eğer öncesinde yüklenen veri kümesinin boyutlarıyla çağırılan

ağın eğitilmesi için gerekli olan veri kümesinin boyutları birbirinden farklıysa yazılım tarafından hata mesajı gösterilir ve yazılımın çalışması sonlanır. Eğer yüklenen ağ ve kaynak veri kümesi uyumluysa yazılım çalışmaya devam eder ve kullanıcıya yüklenen ağın genel mimarisini, mevcut katmanlarını ve bu katmanların yaptığı işlemlerle ilgili detaylı bilgileri görsel olarak ekranda gösterir.

Ön eğitilmiş ağın evrişimli katmanları, veri kümesinde bulunan fotoğraflardan özellik çıkarımında bulunmakta, sonrasında ise tam bağlantılı katmanında tanımlı bulunan nesnelere içerisinden, çıkarılan özelliklere göre nesne tanıma tahminini gerçekleştirmektedir. Ön eğitilmiş bir derin ağın, istenilen yeni nesnelere sınıflandırılması için, son öğrenilebilir katmanı ve son sınıflandırma katmanının değiştirilmesi gerekmektedir. Son öğrenilebilir katman, ağın özelliklerine göre değişmekle birlikte genellikle tam bağlantılı katman olmaktadır. Ön eğitilmiş sinir ağının, yeni veri kümesindeki nesnelere göre nesne tanıma tahmini yapabilmesi için, bu katmanların yenileriyle değiştirilmesi gerekmektedir. Bu katmanların kaçınılmaz olarak katman olduğu, çalışılan ağın özellikleri incelenerek manuel olarak tespit edilebileceği gibi, yazılımla otomatik olarak da tespit edilebilmektedir. Bu çalışmada hazırlanan aktarım öğrenme yazılımında, her bir ön eğitilmiş ağ için farklı bir aktarım öğrenme yazılımı kullanma ihtiyacını ortadan kaldırmak ve verimliliği artırmak amacıyla son öğrenilebilir katman ve son sınıflandırma katmanının yazılım tarafından otomatik çağırılması ve yeni nesnelere tanıyabilmek için bu katmanların değiştirilmesi sağlanmıştır.

Hazırlanan aktarım öğrenme yazılımı tarafından, son öğrenilebilir katmanın tam bağlantılı katman olduğu derin ağlarda, bu tam bağlantılı katmanın yerine, çıktı sayısı yeni veri kümesindeki sınıf sayısına eşit olan yeni bir tam bağlantılı katman eklenmektedir. Son öğrenilebilir katmanın evrişimli katman olduğu durumda ise, bu evrişimli katmanın yerine, süzgeç sayısı, yeni veri kümesindeki sınıf sayısına eşit olan yeni bir evrişimli katman eklenir. Yeni eklenen katmanların, aktarım edilen katmanlardan daha hızlı eğitilebilmesi için, bu katmanların öğrenim oranı faktörleri ince ayar yapılarak artırılır. Sınıflandırma katmanında ise gelen görüntünün özelliklerine göre, ön eğitilmiş ağların kendi kütüphanesinde bulunan sınıf kategorileri içerisinden hangisine en yakın özellikleri gösteriyorsa ona göre bir sınıflandırma yapılmaktadır. Ön eğitilmiş sınıflandırma katmanındaki nesnelere yerine sadece istenilen nesnelere sınıflandırmasının yapılabilmesi için yeni bir sınıflandırma katmanı oluşturularak yeni

sınıflandırma katmanı ile değiştirilir. Oluşturulan yeni sınıflandırma katmanının, ağı aktarım öğrenmesiyle eğitilmesi sırasında nesne tahmini oluşturacağı sınıf bilgilerini, veri kümesindeki nesnelerin klasör isimlerinden otomatik olarak alması sağlandığı için, ayrıca sınıf etiket bilgileri oluşturulmamaktadır. Yeni eklenen katmanların doğru katmanların yerine eklenip eklenmediği, tekrar ağı analiz ederek kontrol edilir ve doğrulanır.

Bundan sonraki aşamada ön eğitilmiş ağı yeni veri kümesini kullanarak aktarım öğrenmesi yöntemiyle yeniden eğitmeye hazır olmasına karşın, bu tez çalışması süresince yapılacak tüm kontrollü deneylerin sonucunu görmek için aktarım öğrenme yazılımı defalarca kez tekrar koşturulacağı için, aktarım öğrenme süresini eniyilemek ve iyileştirmek adına birtakım ince ayarlar yapılmıştır. Bu amaçla yapılan eniyilenmelerden biri de, ön eğitilmiş ağı yeniden eğitilmesinden önce, ağı herhangi bir değişiklik yapılmayan ilk aşamalarındaki katmanlarının öğrenme oranı ağırlıklarının sıfır olarak ayarlanarak dondurulmasıdır. Bu işlem, yazılım üzerinden ağı tüm katmanlarının açılması, dondurulacak katmanların öğrenme oranı faktörlerinin sıfır olarak ayarlanması ve sonrasında ağı katmanlarının yeniden birbirine bağlanmasıyla gerçekleştirilmiştir [55]. Bu sayede aktarım öğrenme yazılımı ağı yeniden eğitirken, dondurulan katmanlara ait parametrelerin yeniden hesaplanmaması sağlanmıştır. Dondurulan katmanlarla ilgili parametreler yeniden hesaplanmadığı için, bu katmanlardaki öğrenme oranı faktörünün sıfıra çekilmesiyle ağı eğitim süresinde önemli iyileştirme sağlanmıştır. Ayrıca yine bu işlem sayesinde, küçük bir veri kümesiyle eğitilen ağlarda görülebilen aşırı uyum probleminin de önüne geçilmiştir.

Aktarım öğrenme yazılımının olabilecek minimum sürede eğitilip maksimum isabetlilikle tahmin geliştirebilmesi için yapılan bir diğer çalışma ise eğitim özellikleriyle ilgili ince ayarlar olmuştur. Tüm veri kümesinin 1 kez üzerinden geçen bir tam eğitim çevrimini tanımlayan döngü sayısının belli bir sayıdan sonra artmasının sadece eğitim süresini uzattığı, buna karşılık tahmin isabetliliğinin ise ihmal edilebilir düzeyde çok küçük artış gösterdiği görülmüştür. Bu tez kapsamında yapılan kontrollü deneylerle, döngü sayısı 6'dan daha fazla artırıldığında eğitim süresinin, eğitilen ağa göre en az 5 dakika artış gösterdiği, buna karşılık en fazla isabetlilik artışının ise %0.03 oranında gerçekleştiği görüldüğü için, aktarım öğrenme yazılımının eğitim sırasında kullandığı döngü sayısı 6 çevrim yapacak şekilde ayarlanmıştır. Ayrıca her eğitim

çevrimi sırasında veri kümesindeki resimlerin %70'inin eğitim, %30'unun doğrulama testi için ayrılması ve her döngüde bu verilerin rastgele bir şekilde eğitim veya doğrulama verisi olarak kullanılması sağlanmıştır. Bu sayede, yazılımın veri kümesinin rastlantısal bir %70'iyle aktarım öğrenme eğitimini tamamladıktan sonra, veri kümesinin %30'unda bulunan rastgele resimler üzerinden öğrendiği özelliklere göre yaptığı nesne sınıfı tahmin bilgileriyle, nesnenin ait olduğu gerçek sınıf karşılaştırılarak yapılan benzetim isabetlilik yüzdesinin, Bölüm 3.4'te anlatılan uygulama ortamında yapılacak testler öncesinde gerçekçiliğinin artırılması hedeflenmiştir.

3.2. Aktarım Öğrenme Veri Kümesinin Hazırlanması

Bölüm 3.1'de anlatılan aktarım öğrenme yazılımında kullanılacak veri kümelerinin oluşturulması için, aktarım öğrenme yazılımının tahmin çıktılarını oluşturan nesne sınıflarında bulunan her bir nesnenin belirli sayıda fotoğraflarına ihtiyaç vardır. Aktarım öğrenme yönteminde ön eğitilmiş ağlar kullanıldığı için bu sayı, ağı sıfırdan modellenmesine göre çok daha düşüktür. Literatürde, her bir nesne sınıfı için kullanılan resim sayısının, nesne tanıma başarımına etkisini inceleyen çalışmalar mevcuttur [56]. Genel kabul görmüş kural, her bir nesne için yaklaşık 1000 fotoğraf olarak geçmektedir [36,56]. Bu tez çalışmasında da genel kabul görmüş olan nesne başına 1000 fotoğraf kuralına uyularak veri kümeleri oluşturulmuştur. Veri kümeleri oluşturulurken çekilen tüm fotoğraflar, Apple marka Iphone 11 model cep telefonunun arka kamerasıyla çekilmiştir. Vida, pul ve somun nesne sınıflarını oluşturmak için, Şekil 3.2'de gösterilen farklı uzunluk, çap ve renklerdeki 5'er adet vida, pul ve somunun fotoğrafları çekilerek veri kümesi oluşturulmuştur. Ayrıca tahmin sınıflarının sadece bu üç nesneden oluşması durumunda, Bölüm 4.3'de detaylı anlatılan robot kontrol yazılımına, kamerada bu üç nesneden biri olmasa dahi sürekli bu üç nesneden birinin bilgisinin gitmesinin, robot kontrol yazılımının robotu kontrol etmesini zorlaştıracağından, herhangi bir nesne olmayan durumları tanımlamak amacıyla, herhangi bir nesne içermeyen 1000 adet fotoğraf çekilmiş ve bu fotoğraflardan oluşan "Nesne Yok" sınıfı tanımlanmıştır. Veri kümelerini oluştururken, farklı boyutlarda ve fiziksel özelliklerde olan 5'er adet vida, pul ve somun kullanıldığından, 1000 fotoğraftan oluşan bir nesne sınıfı oluşturmak için Şekil 3.3'te görülen her bir nesnenin ayrı ayrı 200'er adet fotoğrafı çekilmiştir. Böylelikle vida nesnesi için 1000 adet vida fotoğrafı, somun nesnesi için 1000 adet somun fotoğrafı ve pul nesnesi için de 1000

adet pul fotoğrafı elde edilmiştir. Ayrıca, test düzeneğinde herhangi bir nesnenin mevcut olmadığı durumları sınıflandırabilmek adına, düzenekte herhangi bir nesne mevcut değilken 1000 adet fotoğraf çekilmiş ve bu fotoğrafları kullanarak “Nesne Yok” isminde bir sınıf oluşturulmuştur.



Şekil 3.1 Veri Kümesinde Kullanılan Nesnelere

Böyle bir sınıfa neden ihtiyaç duyulduğıyla ilgili gerekçeler, Bölüm 4.3’te detaylandırılmıştır. Fotoğrafların çekimi esnasında herhangi bir otomasyon düzeneğı kullanılmamıştır. Çekimler esnasında nesnelerin kadradaki yeri, kamera ile nesne arasındaki uzaklık ve çekim açıları elle ayarlanmıştır. Her bir nesnenin fotoğrafı çekilirken, nesnenin üzerinde belirlenen mesafeden hem dikey, hem de çeşitli eğik açılardan nesne etrafında 360 derecelik bir tur tamamlarken fotoğraflar çekilmiştir. Bölüm 3.2.1 ve Bölüm 3.2.2’de, aynı nesnelere kullanılarak oluşturulan veri kümelerinde, çekilen fotoğrafların nesneye olan mesafesinin ve fotoğraf çekimi sırasındaki ortam ışığının, aktarım öğrenme yazılımının başarımına olan etkileri incelenerek anlatılmıştır.

3.2.1. Veri Kümesinin Hazırlanmasında Mesafe Etkisi

Veri kümesi oluşturulurken çekilen fotoğrafların mesafesi, nesnenin kadrajdaki konumu ve fotoğrafın bütünü üzerinde nesnenin ne kadarlık bir alanı kapladığıyla doğrudan ilgili olduğu için aktarım öğrenme yazılımının eğitim sonrası göstereceği nesne tanıma başarımını etkilemektedir. Bu bölümde, veri kümesi oluşturulurken çekilen fotoğrafların mesafesinin, aktarım öğrenme yazılımının başarımına nasıl etkileri olduğu incelenecektir.

Veri kümesi oluşturma çalışmaları sırasında ilk olarak tüm fotoğraflar, Şekil 3.3'te bir örneği görüldüğü üzere nesne ile kamera lensi arasındaki uzaklık 20cm olacak şekilde ayarlanarak çekilmiştir.

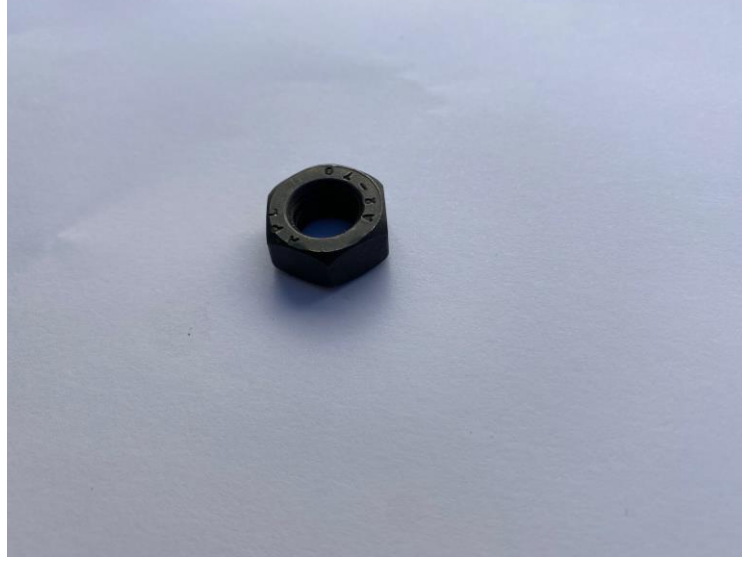


Şekil 3.2 20cm Mesafeden Fotoğraflanan Nesne Örneği

Bu şekilde 20cm mesafeden çekilen fotoğraflarda, nesnelerin resmin bütünü içerisinde kapladığı alan, en küçük nesne için %1,6, en büyük nesne için %6,3 oranında olduğu görülmüştür. Çekilen fotoğraflar, aktarım öğrenme yazılımıyla eğitilecek algoritmalara uygun imge noktası sayısı yeniden boyutlandırılmıştır (Alexnet ve Squeezetnet için 227x227, Googlenet için 224x224). Her 3 algoritma da kendilerine uygun boyutlara dönüştürülen veri kümesiyle Bölüm 3.1'de anlatılan aktarım öğrenme yazılımı kullanılarak eğitilmiştir. Eğitim sonrasında bu veri kümesiyle eğitilen her 3 algoritmanın da, eğitim sonunda veri kümesindeki nesnelerin rastgele seçilen %30'uyla

yapılan benzetim başarımlarını kıyaslamasında başarımlarının %75'in altında kaldığı görülmüştür. Test bilgisayarına bağlanan bir USB kamerayla algoritmaların nesne tanıma başarımları incelenmiş, ancak her 3 algoritmanın da yaptıkları nesne tahminlerinin anlık olarak sürekli değişim gösterdiği, zamanla değişmeyen tek bir nesne tahmini çıktısı üretmedikleri görülmüştür.

Bunun üzerine nesne ile kamera lensi arasındaki uzaklık 10cm'e ayarlanarak çekilen fotoğraflardan oluşan yeni bir veri kümesi oluşturulmuştur. Şekil 3.4, 10cm mesafeden fotoğraflanan bir nesne örneğini göstermektedir.



Şekil 3.3 10cm Mesafeden Fotoğraflanan Nesne Örneği

10cm mesafeden çekilen fotoğraflarda nesnenin, resmin bütününde kapladığı alan, en büyük nesne için %22,68 en küçük nesne için %5,76 olmuştur. Oluşturulan bu veri kümesiyle çekilen fotoğraflar da algoritmaların ihtiyaç duyduğu boyutlara dönüştürülmüş ve algoritmalar eğitilmiştir. Eğitimin sonunda yapılan benzetim başarımlarını kıyaslamasında 3 algoritmanın da başarımlarında, 20cm'den çekilen fotoğraflarla oluşturulan veri kümesine göre iyileşme olduğu görülmüştür. Buna karşılık, 3 algoritmanın da veri kümesindeki nesnelerin rastgele %30'u kullanılarak yapılan başarımlarını kıyaslama testinde, başarımlarının %84'ün üzerine çıkamadığı görülmüştür. USB kamera üzerinden test ortamında yapılan donanımsal incelemelerde ise, özellikle vida nesnelerinin hem görece daha büyük olması, hem de fiziksel şekil itibarıyla somun ve puldan daha farklı olması sayesinde 3 algoritma tarafından da sorunsuz bir şekilde tespit edilebildiği görülmüştür. Somun ve pul nesnelerinde ise bu nesnelerin görece

daha küçük ve birbirine benzer geometrik şekilleri dolayısıyla eğitilmiş algoritmalar tarafından yaptıkları tahminlerin sıklıkla karıştığı, kamera ve görüntü sabitken kadrajdaki nesneye algoritmanın yaptığı tahminin pul ve somun arasında sıklıkla değişim gösterdiği gözlenmiştir.

Nesneler, resmin kadrajında daha büyük bir alan sağladıkça algoritmaların nesne tahmin isabetliliğinin arttığının gözlemlenmesi üzerine, nesnelerin fotoğraflarının 5cm'den çekilerek yeni bir veri kümesi oluşturulmak istenmiş, ancak çalışmada kullanılan kamerayla bu mesafeden çekilen fotoğraflarda kameranın odak sorunu yaşadığı görüldüğü için yapılamamıştır. O nedenle nesnelerin fotoğraf kadrajında kapladığı alanı daha fazla büyütmek için, kameranın yaklaşma özelliği kullanılarak 2.5x yaklaşma yapılarak 10cm'den çekilen fotoğraflarla yeni bir veri kümesi oluşturulmuştur. Şekil 3.5, bu yöntemle oluşturulan veri kümesinden bir örneği göstermektedir.



Şekil 3.4 10cm Mesafeden 2.5x Yakınlaşma ile Fotoğraflanan Nesne Örneği

2.5x yaklaşma ile 10cm'den çekilen fotoğraflarla oluşturulan veri kümesi sayesinde, fotoğraflanan nesnelerin kadrajın tamamına oranı en küçük nesne için %14,4, en büyük nesne için %56,7 olmuştur. Önceki veri kümelerinde yapılan fotoğraf imge noktası boyutlarının algoritmalara uygun şekilde yeniden boyutlandırılması işlemi bu veri kümesi için de tekrarlanmıştır. Sonrasında algoritmalar sırasıyla bu veri kümesiyle eğitilerek incelenmişlerdir. Algoritmaların eğitimini tamamlaması sonrası veri kümesinin test için ayrılan bölümündeki fotoğraflarla yapılan benzetim başarımlarını kıyaslaması testlerinde tüm algoritmalar %90'ın üzerinde tahmin isabetliliği başarımları

göstermişlerdir. Bu algoritalardan Alexnet algoritması, eğitim sonucunda %91,80 benzetim isabetlilik oranı gösterirken, bu oran Googlenet Algoritması için %93,54 ve Squeezenet için %96,76 olarak gerçekleşmiştir. USB kamera üzerinden aktarılan gerçek zamanlı nesne görüntülerine yapılan tahminler incelendiğinde ise, Alexnet algoritmasının, pul ve somun nesnelere ayırt etmekte çok başarılı olamadığı, kamera ve nesne sabit olmasına rağmen zaman zaman yaptığı tahmini değiştirdiği ve zamanla değişmeyen tek bir tahmin çıktısı üretecek şekilde çalışmadığı görülmüştür. Squeezenet ve Googlenet algoritmalarının ise kameranın gösterdiği nesnelere için doğru tahmin bilgilerini üretebildiği görülmüştür. Her iki algoritmanın da nesne ve kamera görüntüsü sabitken, kadraj içerisinde yer alan nesneyi doğru tahmin ettiği ve tahminini zaman içerisinde değiştirmeyerek sabit nesnelere söz konusu olduğunda tek bir tahmin çıktısı ürettiği ve bu tahmini değiştirmeden çalıştıkları görülmüştür.

Veri kümesi oluşturulurken çekilen fotoğrafların nesnelere olan uzaklığı ve kadrajda kapladığı alanın, nesne tanıma başarımlarına olan etkisi, Çizelge 3.2’de özetlenmiştir.

Çizelge 3.2 Veri Kümesinin Hazırlanmasında Mesafe Etkisi

Mesafe	En Küçük Nesnenin Kadrajda Kapladığı Alan	En Büyük Nesnenin Kadrajda Kapladığı Alan	Alexnet	Googlenet	Squeezenet
20cm	1,60%	6,30%	69,47%	72,57%	74,80%
10cm	5,76%	22,68%	76,55%	80,38%	83,73%
10cm + 2.5x Yakınlaşma	14,40%	56,70%	91,80%	93,24%	96,76%

3.2.2. Veri Kümesinin Hazırlanmasında Ortam Işığının Etkisi

Veri kümesinin hazırlanmasında mesafenin ve buna bağlı olarak nesnelere kadrajda kapladığı alanın aktarım öğrenme yazılımına etkileri Bölüm 3.2.1’de

incelenmiştir. Bu bölümde ise veri kümesi oluşturulurken çekilen fotoğrafların çekildiği andaki ortam ışığının, aktarım öğrenme yazılımının başarımını nasıl etkilediği incelenmiştir. Bu çalışma için, diğer koşullar ve değişkenler sabit tutularak, gün ışığında herhangi bir ek aydınlatma (ampul, led, fotoğraf flaşı vb.) kullanılmadan çekilen fotoğraflarla birinci veri kümesi oluşturulmuştur. Buna karşılık, hiç gün ışığında çekilmeyen, tamamen ortamda bulunan elektrik ampullerinin yaydığı sarı ışık altında çekilen fotoğraflar kullanılarak ikinci veri kümesi oluşturulmuştur. Son olarak, her iki veri kümesiyle de algoritmalar eğitilmiş ve algoritmaların eğitimi tamamladıktan sonraki başarımları, benzetim ve uygulama olarak incelenmiştir. Her iki veri kümesi de, Bölüm 3.2.1’de edinilen tecrübe doğrultusunda 10cm mesafeden, 2.5x yakınlaştırma kullanılarak çekilen fotoğraflardan oluşturulmuştur. İfade karmaşasını önlemek adına birinci veri kümesinden gün ışığı veri kümesi; ikinci veri kümesinden ise ortam ışığı veri kümesi olarak bahsedilecektir. Bu bölümde gün ışığında ve ortam ışığında çekilen fotoğraflarla hazırlanan veri kümelerinin, algoritmalar eğitildikten sonraki başarımlarına olan olumlu ya da olumsuz etkileri üzerinde durulacaktır. Algoritmaların başarımlarıyla ilgili detaylı bilgiler, Bölüm 3.4 ve Bölüm 3.5’te yapılacaktır.

Gün ışığı veri kümesi kullanılarak eğitilen algoritmaların, gün ışığındaki başarımları, bir önceki kısımda mesafe etkisi incelenirken 10cm mesafeden 2.5x yakınlaşma kullanılarak oluşturulan veri kümesiyle eğitilen algoritmaların başarımlarıyla paralellik göstermiştir. Mesafe etkisi araştırılırken oluşturulan tüm veri kümeleri, gün ışığında çekilen fotoğraflarla oluşturulduğu için, aynı koşullar altında yapılan deneyle aynı sonuca ulaşılmıştır. Gün ışığı veri kümesiyle eğitilen algoritmaların, ortam ışığı altındaki başarımları incelendiğinde ise, USB kamera üzerinden görüntülenen vida, pul ve somun nesnelere, vida nesnesiyle ilgili Alexnet, Googlenet ve Squeezenet algoritmalarının tamamının vida nesnelere başarılı bir şekilde tanıyıp sınıflandırma yaptığı, nesne sabit bir şekilde kamera kadrajında kaldığı süre boyunca, gerçek zamanlı üretilen tahmin sınıfı bilgisinin değişiklik göstermediği gözlenmiştir. Pul ve somun nesnelere kamera kadrajında olduğu durumda ise algoritmaların ortam ışığındaki başarımlarının, gün ışığındaki başarımlarından daha kötü olduğu görülmüştür. Alexnet algoritmasının hem pul ve somun nesnelere hatalı tahmin etme sıklığı, hem de kadrajda sabit bir nesne olmasına rağmen üretilen sınıf tahmininin değişkenliği artış göstermiştir. Gün ışığı veri kümesiyle eğitilen Squeezenet ve Googlenet algoritmalarının, ortam ışığındaki başarımları

incelendiğinde ise, zaman zaman kadrajda sabit bir nesne olmasına rağmen sınıflandırma tahmininin anlık sapmalarla değiştiği görülmüştür. Gerçekleşme sıklığı çok yüksek olmamasına rağmen, Squeezenet ve Googlenet algoritmalarında nadiren gözlenen bu durum, bu algoritmaların gün ışığındaki başarımlarıyla ortam ışığı altındaki başarımları arasındaki en büyük farklılık olarak göze çarpmıştır.

Işığın algoritma başarımlarına etkisiyle ilgili çalışmanın ikinci aşamasında ortam ışığı veri kümesi kullanılarak algoritmalar eğitilmiştir. Ortam ışığı veri kümesiyle eğitilen algoritmaların ilk olarak gün ışığındaki başarımları incelenmiştir. Ortam ışığı veri kümesiyle eğitilen algoritmaların, gün ışığında USB kamerasından gelen görüntülere dair nesne tanıma tahminlerinde Alexnet algoritmasının vida nesnelерinin ekran görüntüsüne yaptığı tahminlerde nadiren de olsa anlık tahmin değişikliği gösterdiği durumlar görülmüştür. Somun nesneleri kamera kadrajına gösterildiğinde Alexnet algoritmasının kararlı bir şekilde doğru tahmini oluşturduğu, pul nesneleri kameraya gösterildiğinde ise algoritmanın ürettiği tahminin sıklıkla pul ve somun arasında geçiş yaptığı görülmüştür. Googlenet ve Squeezenet algoritmalarının başarımları ise birbirine paralel şekilde gerçekleşmiştir. Her iki algoritmada vida nesnesini başarıyla tanıyıp tahmin etmiştir. Pul ve somun nesnelерinin tahminlerinde ise deneylerin büyük çoğunluğunda hatasız tahmin etmesine karşılık, nadiren de olsa pul nesnesi ekrandayken tahminin anlık olarak somuna kaydığı, somun nesnesi ekrandayken de tahminin anlık olarak pul şeklinde değiştiği durumlar gözlemlenmiştir.

Işığın algoritma başarımlarına etkisinin araştırıldığı çalışmanın son aşamasında, ortam ışığı veri kümesiyle eğitilen algoritmaların aynı ortam ışığındaki başarımları incelenmiştir. Eğitim veri kümesinin hazırlandığı ortam ve eğitilen algoritmaların test ortamı aynı olduğunda, algoritmaların gün ışığına göre daha yüksek başarımlar gösterdikleri görülmüştür. Squeezenet ve Googlenet algoritmalarının, USB kamera üzerinden vida, pul ve somun nesnelерinin her birini ayrı ayrı başarıyla sınıflandırabildiği görülmüştür. Alexnet algoritmasının ise vida ve somun nesnesini istikrarlı bir şekilde doğru sınıflandırdığı, ancak pul nesnesini sınıflandırırken ürettiği tahmin bilgisinin zaman zaman somun olarak değişiklik gösterebildiği gözlemlenmiştir.

Veri kümesinin oluşturulduğu ortam ve test ortamı arasındaki ilişki yapılan deneyler sonrasında incelendiğinde, gün ışığı veri kümesiyle eğitilen algoritmaların gün

ışığında daha başarılı olduğu, ortam ışığı veri kümesiyle eğitilen algoritmaların da ortam ışığında test edildiğinde daha yüksek isabetlilik ve başarımla çalıştığı gözlemlenmiştir. Bu tezin konusu olan UR5 robotla vida pul ve somun gibi farklı geometrik şekillere sahip nesnelerin sınıflandırılacağı test ortamında, hem gün ışığı hem de ortam ışığında çalışabileceği göz önüne alındığında, kullanılacak veri kümesinin hem ortam ışığında hem de gün ışığında çekilen fotoğrafları içermesi gerekliliği, yapılan deneylerin sonunda ortaya çıkan en önemli sonuç olmuştur.

3.3. Derin Öğrenme Algoritmalarının Yazılım Ortamında İncelenmesi

Derin öğrenme algoritmalarının nesne tanıma başarımlarının, algoritmaların eğitildiği veri kümesiyle doğrudan ilişkili olduğu, Bölüm 3.1 ve Bölüm 3.2’de yapılan kontrollü deneylerle ortaya konmuştur. Bu nedenle tez çalışması kapsamında incelenen derin sinir ağlarının, benzetim başarımları yapılmadan önce, bu ağların eğitileceği veri kümelerinin, ağın başarımını yükseltecek şekilde eniyilenmesi sağlanmıştır.

Bölüm 3.2.1’de veri kümesini oluşturmak için fotoğrafı çekilen nesnelere kullanılan fotoğraf makinesi lensi arasındaki mesafe incelenmiş ve 10cm mesafeden, 2.5x yakınlaşma ile çekilen fotoğraflarla oluşturulan veri kümesiyle eğitilen Alexnet, Googlenet ve Squeezenet derin sinir ağları en yüksek nesne tanıma başarımını göstermiştir.

Bölüm 3.2.2’de veri kümesini oluşturmak için çekilen fotoğrafların gün ışığında ve ortam ışığında çekilmesinin, algoritmaların nesne tanıma başarımlarına nasıl bir etki oluşturduğu incelenmiştir. Gün ışığında çekilen fotoğraflarla eğitilen algoritmaların, gün ışığında bulunan test ortamında daha iyi başarımlar gösterdiği; ortam ışığında çekilen fotoğraflarla eğitilen algoritmaların yine ortam aydınlatması altında test edilirken daha yüksek isabetlilikle çalıştığı görülmüştür. Bu tez çalışması kapsamındaki UR5 işbirlikçi robotun gerçekleştireceği nesne tanıma ve sınıflandırma işlemlerinin hem gün ışığında hem de ortam aydınlatması altında en iyi başarımları gösterebilmesi için, veri kümesinin hem ortam aydınlatmasında hem de gün ışığında çekilen fotoğrafları içermesi gerektiği ortaya çıkmıştır.

Bu bölümde algoritmaların benzetim başarımlarını kıyaslamaları ve sonuçlarından bahsedilecektir. Alexnet, Googlenet ve Squeezenet algoritmalarının benzetim başarımlarını kıyaslamaları için, algoritmaların eğitileceği veri kümeleri, 10 santimetre mesafeden 2.5x yaklaşma ile çekilen fotoğraflarla oluşturulmuştur. “Pul”, “vida”, “somun” ve “nesne yok” sınıflarının her biri için, 500 adet gün ışığında ve 500 adet ortam ışığında çekilen nesne fotoğraflarıyla her bir sınıf için 1000 adet eğitim verisi seçilerek algoritmalar aktarım öğrenme yazılımıyla eğitilmiştir. Tüm algoritmalar, Windows 10 işletim sisteminde, Intel i7-4720HQ dört çekirdekli işlemci, 16GB rastgele erişimli bellek ve NVIDIA GeForce GTX960M 4GB bellekli ekran kartına sahip bir test bilgisayarı kullanılarak MATLAB R2019B sürümü içerisinde eğitilmiştir.

İlk olarak Alexnet algoritması, yukarıda detayları açıklanan veri kümesi kullanılarak aktarım öğrenme yazılımıyla eğitilmiştir. Eğitim için veri kümesinde bulunan her bir nesne sınıfına ait fotoğrafların rastgele seçilen %70’i Alexnet algoritmasının eğitilmesi amacıyla kullanılmıştır. 6 döngüden oluşan eğitimin tamamlanması, 16 dakika 25 saniye sürmüştür. Eğitim tamamlandıktan sonra, aktarım öğrenme yazılımı, veri kümesindeki fotoğrafların, her bir nesne için geriye kalan %30’unu algoritmanın başarımlarını benzetim olarak ölçmek amacıyla kullanmıştır. Doğrulama verisi olarak ayrılan veri kümesinin bu %30’luk kısmındaki her bir fotoğrafa, eğitilen algoritma tarafından tahmin yaptırılmıştır. Şekil 3.6, algoritmaların, doğrulama verilerine yaptığı bir tahmin örneğini göstermektedir.

Alexnet algoritması, doğrulama verisi olarak ayrılan bu fotoğraflara, %91.80 isabetlilikle tahmin çıktıları üretmiştir. Şekil 3.7, Alexnet algoritmasının benzetim başarımlarını testi sonucunu göstermektedir.

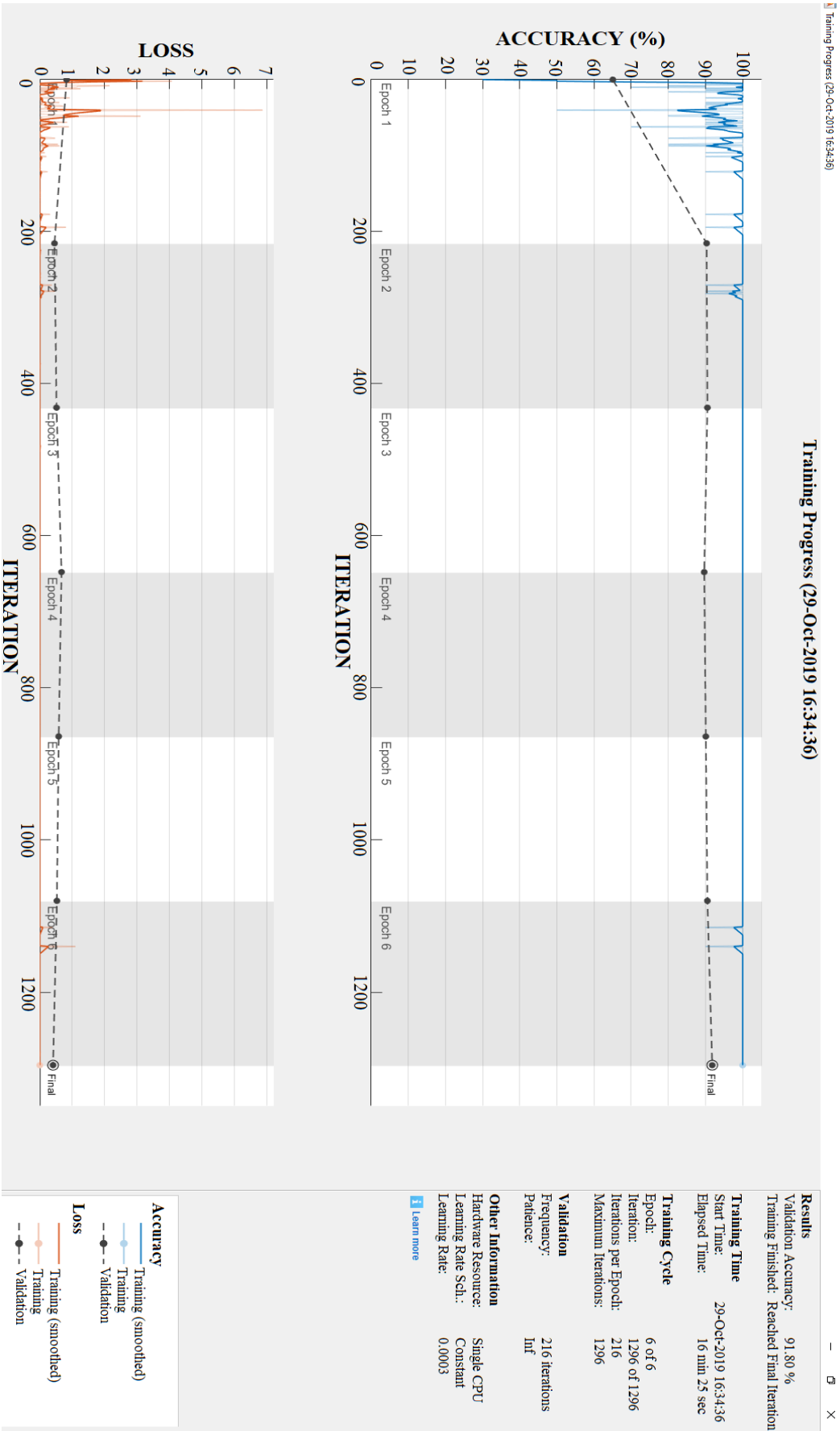


Şekil 3.5 Doğrulama Verisine Yapılan Tahmin Örneği

İkinci olarak Googlenet algoritması aynı veri kümesi kullanılarak eğitilmiştir. Veri kümesinin rastlantısal olarak seçilen %70'i eğitim amacıyla kullanılmıştır. Altı döngüden oluşan Googlenet algoritmasının eğitimi, 46 dakika 29 saniyede tamamlanmıştır. Eğitimi tamamlanan algoritma, veri kümesinin test için ayrılan %30'unda bulunan fotoğraflarda bulunan nesnelere sırasıyla tek tek sınıflandırma işlemi yapmıştır. Googlenet algoritması, doğrulama verisi olarak ayrılan bu fotoğraflara, %93,24 isabetlilikle doğru sınıflandırmıştır. Şekil 3.8 Googlenet algoritması benzetim başarımlarını göstermektedir.

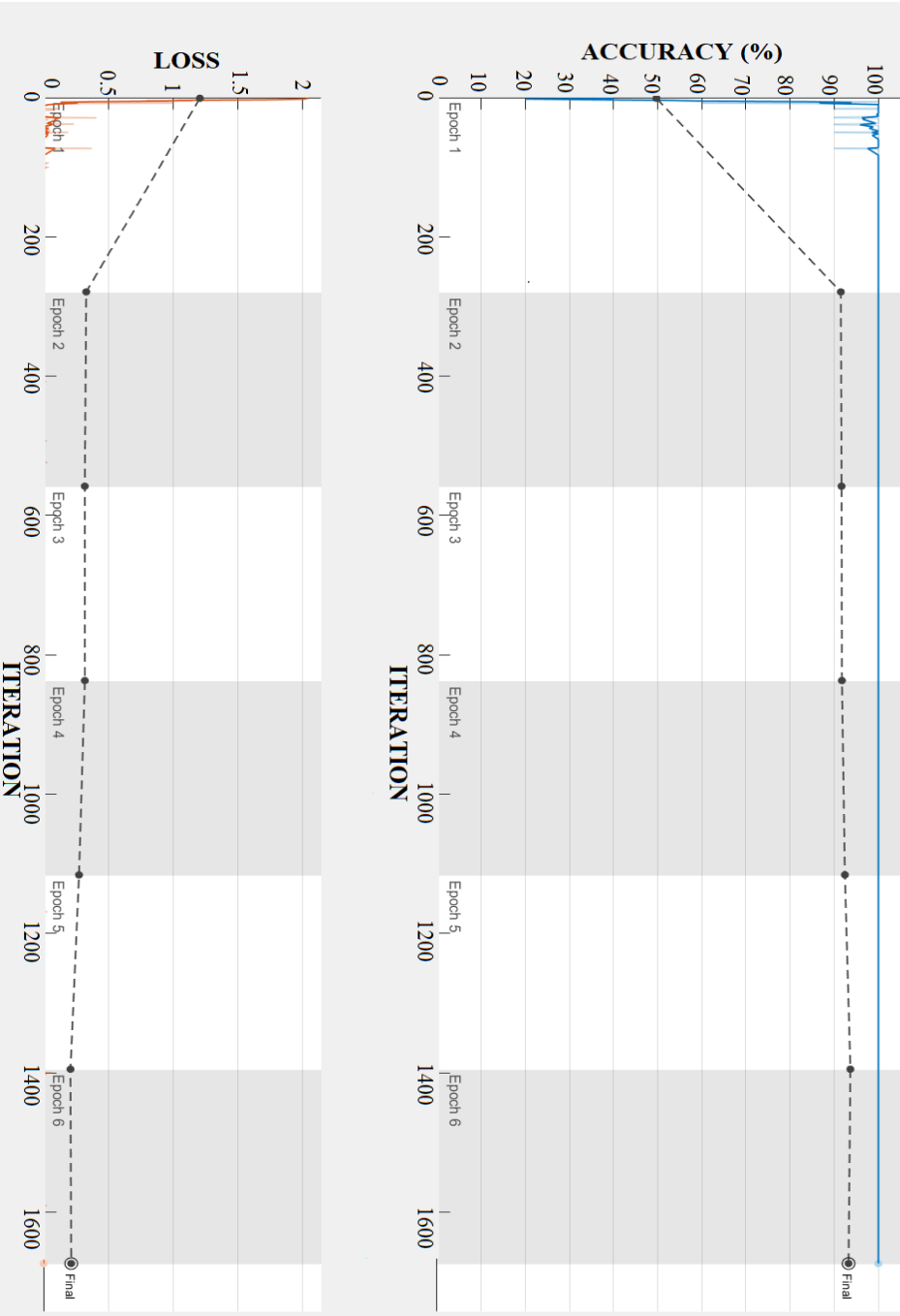
Son olarak Squeezenet algoritması, aynı veri kümesi kullanılarak aktarım öğrenme yazılımıyla eğitilmiştir. Eğitim sırasında diğer algoritmaların eğitimlerinde olduğu gibi, algoritmanın, veri kümesinin rastgele seçilen %70'i ile eğitilmesi sağlanmıştır. Squeezenet algoritmasının altı döngüden oluşan eğitimi tamamlama süresi 15 dakika 17 saniye olmuştur. Eğitim tamamlandıktan sonra algoritma, veri kümesinin test için ayrılan rastgele %30'luk bölümünde yer alan nesne fotoğraflarına yaptığı tahminlerle başarımlarını test edilmiştir. Squeezenet algoritması, doğrulama verisinde yer alan nesnelere %96,76 isabetlilikle doğru sınıflandırma başarımlarını göstermiştir. Şekil 3.9 Squeezenet algoritması benzetim başarımlarını göstermektedir.

Benzetim başarım kıyaslamaları testleri sonucunda tüm algoritmaların %90'ın üzerinde başarım ile nesnelere sınıflandırabildiği gözlemlenmiştir. Başarım yüzdeleri karşılaştırıldığında en yüksek başarım oranına Squeezenet algoritması ulaşırken, en düşük başarım oranının Alexnet algoritmasına ait olduğu görülmüştür. Diğer yandan algoritmaların eğitimi tamamlama süreleri kıyaslandığında Googlenet algoritmasının eğitim süresinin, Alexnet ve Squeezenet algoritmalarına göre 3 kat daha uzun sürdüğü görülmüştür. Benzetim başarım testi sonucunda Squeezenet algoritması, hem en yüksek başarım yüzdesine sahip olması, hem de aktarım öğrenme yazılımıyla en hızlı şekilde eğitimi tamamlayabilen algoritma olmasıyla dikkat çekmiştir.



Şekil 3.6 Alexnet Algoritması Benzetim Başarım Testi Sonucu

Training Progress (05-Jan-2020 16:33:24)



Results
 Validation Accuracy: 93.24 %
 Training Finished: Reached Final Iteration

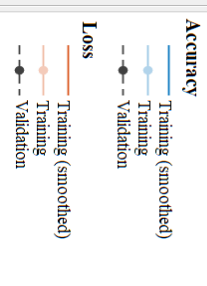
Training Time
 Start Time: 05-Jan-2020 16:33:24
 Elapsed Time: 46 min 29 sec

Training Cycle
 Epoch: 6 of 6
 Iteration: 1674 of 1674
 Iterations per Epoch: 279
 Maximum Iterations: 1674

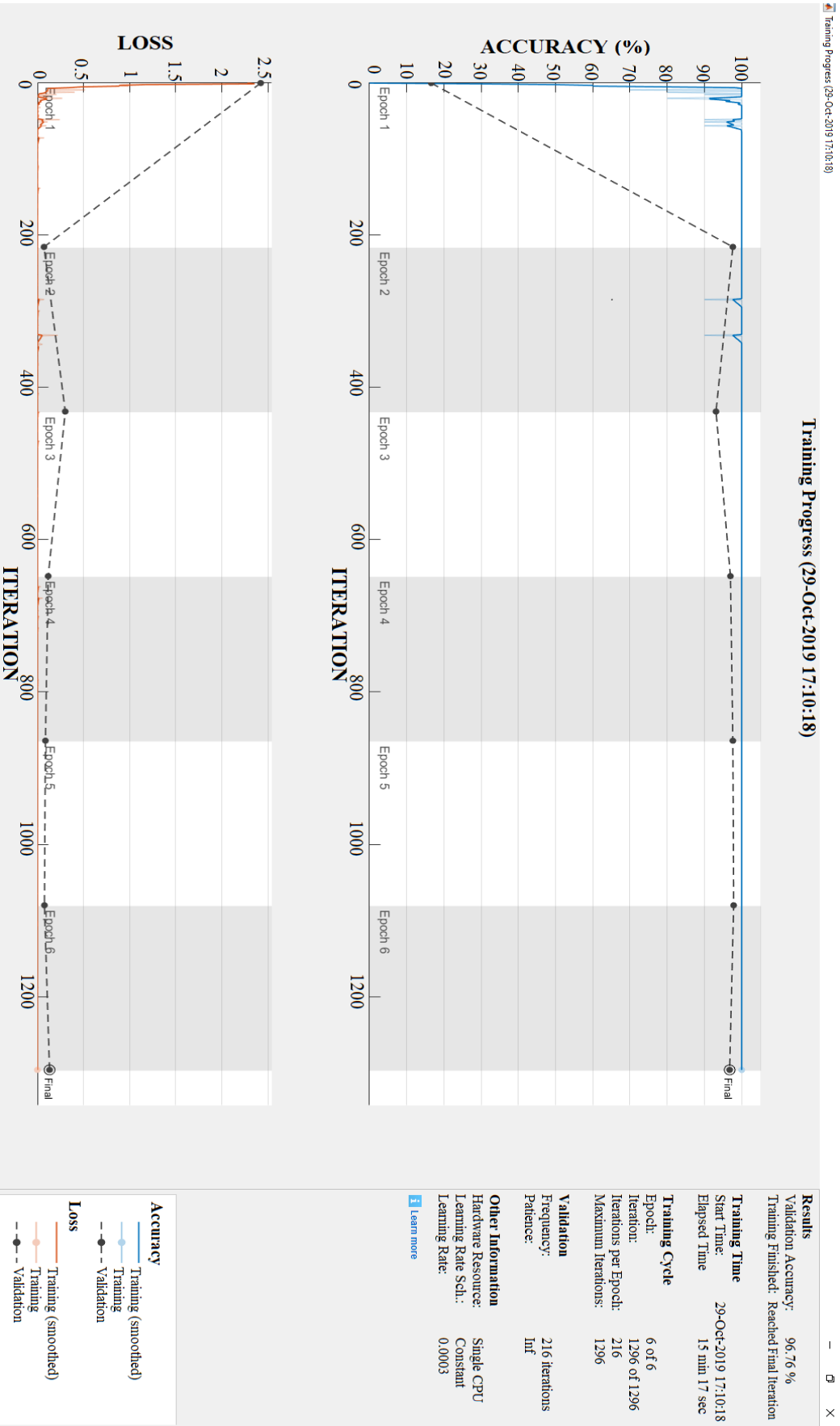
Validation
 Frequency: 279 iterations
 Patience: Inf

Other Information
 Hardware Resource: Single CPU
 Learning Rate Sch.: Constant
 Learning Rate: 0.0003

[Learn more](#)



Şekil 3.7 Googlet Net Algoritması Benzetim Başarım Testi Sonucu



Şekil 3.8 Squeezenet Algoritması Benzetim Başarım Testi Sonucu

3.4. Derin Öğrenme Algoritmalarının Uygulama Ortamında İncelenmesi

Çalışma kapsamında incelenen derin öğrenme algoritmalarının, benzetim başarımlarını kıyaslamaları testleri Bölüm 3.3’de gerçekleştirilmiştir. Bu bölümde aktarım öğrenme yazılımıyla tezin konusu olan vida, pul ve somun nesnelerini sınıflandırabilmek üzere eğitilmiş algoritmaların, uygulama ortamında gösterdikleri başarımları incelenecektir.

Algoritmaların uygulama ortamında göstereceği başarımlarını incelemek için, UR5 işbirlikçi robotla kurulacak olan test altyapısının bir benzeri, ev ortamında oluşturulmuştur. Bu amaçla, hız ayarlı step motor tarafından sürülen 1 metrelik bir kayar bant sistemi yapılmıştır. Test bilgisayarına bağlı bir adet USB kamera, bu kayar bantın orta bölümüne yukarıdan bakacak şekilde 10 cm mesafeyle yerleştirilmiştir. MATLAB üzerinden kamera erişimi açılarak, algoritmaların, kameradan gelen gerçek zamanlı anlık görüntülere sürekli bir şekilde sınıflandırma tahmini üretmesi sağlanmıştır. Vida, pul ve somun nesnelere, kayar bantın başlangıç noktasından elle yerleştirilmiş ve her bir algoritmanın, nesne kadrajda yokken, nesne kadraja girdiğinde ve nesne kadrajdan çıktığı süreç boyunca ürettiği tahmin çıktıları gözlenmiştir. Uygulama testlerini gerçekleştirmek üzere, vida pul ve somun nesnelere her biri için, farklı renk ve boyutlarda beşer adet olmak üzere toplam 15 nesne rastgele olarak seçilmiş ve üç derin öğrenme algoritmasının da uygulamalı başarımlarını testleri bu nesnelere gerçekleştirilmiştir. Testler süresince her bir nesnenin, kayar bant vasıtasıyla kamera önünden 20 kez geçmesi sağlanmıştır. Bu sayede, bu tez çalışması kapsamında incelenen Alexnet, Googlenet ve Squeezenet algoritmalarının, her bir nesne sınıfı için 100’er kez olmak üzere toplamda 300 kez nesne tanıma ve sınıflandırma başarımları test edilmiştir.

Uygulamalı test altyapısı oluşturulurken, kameradan gelen görüntünün gerçek zamanlı olarak değil gecikmeli bir şekilde ekrana geldiği, görüntüde kasma ve donmalar yaşandığı görülmüştür. Bu durum, yapılan tahminlerin isabetliliğini gözlemlemeyi zorlaştırmıştır. Yaşanan problemin, test bilgisayarının kameradan gelen gerçek zamanlı yüksek çözünürlüklü görüntüyü işlemek için başarımlarının yetmemesinden kaynaklandığı anlaşılmıştır. Problemin üstesinden gelmek için, USB kameranın 1280x720 imge noktası olan standart çözünürlük ayarı düşürülmüştür. Çeşitli çözünürlük ayarlarıyla

yapılan denemeler sonrası, hem nesneyi tanıyabilecek, hem de gerçek zamanlı görüntü akışını sağlayabilecek en iyi çözünürlüğün 640x480 imge noktası olduğu görülmüştür. Kamera çözünürlüğü bu ayara getirilerek, uygulama testleri sırasında gerçek zamanlı ve akıcı bir kamera görüntüsü elde edilmesi sağlanmıştır.

Uygulamalı kıyaslama testleri yapılırken, kayar bandın hızı saniyede 5cm ilerleyecek şekilde ayarlanmıştır. Bu sayede her bir nesnenin kamera kadrajı içerisinde en az 2 saniye boyunca bütün olarak yer alması sağlanmıştır. Her bir nesne sınıfı için yapılan 100 tekrarlı denemelerde, yapılan bir denemenin başarılı sayılması için nesne kadrajda yokken algoritma tarafından üretilen tahminin nesne yok olarak sabit kalması, nesne kadraja girdiği andan itibaren ilgili nesneye dair tahminin oluşması ve bu tahminin nesne kadrajdan çıkana kadar sabit kalması, nesne kadrajdan çıktıktan sonra bir sonraki nesne gelene kadar tahminin tekrar nesne yok olarak sabit kalması gerekmektedir. “Nesne yok – ilgili nesne tahmini – nesne yok” şeklinde oluşması beklenen bu akışı bozan anlık bir tahmin sapması olsa dahi, o deneme başarısız sayılmıştır.

Uygulama başarımları incelenen algoritmaların her biri; Bölüm 3.2.1 ve Bölüm 3.2.2’de incelenen ortam ışığı ve mesafe etkilerine göre oluşturulan, yarısı gün ışığında diğer yarısı ortam ışığında ve her biri 10cm mesafeden 2.5x yakınlaşma ile çekilen fotoğraflardan oluşan veri kümesiyle eğitilmiştir.

Bu koşullar altında gerçekleştirilen uygulama başarımları testlerinde ilk olarak Alexnet algoritmasının uygulama başarımları testleri yapılmıştır. Alexnet algoritmasıyla beş farklı vida nesnesinin 20 tekrarlı testleriyle yapılan toplam 100 testin 91 tanesinde nesne kadrajda yokken nesne yok tahmini, nesne kadraja girdiği andan kadraj dışına çıkana kadar vida tahmini, nesne kadrajdan çıktıktan bir sonraki nesne gelene kadar da nesne yok tahmini şeklinde başarılı bir tahmin akışı gözlemlenmiştir. 9 hatalı tahminin 4’ü nesnenin kadraja girişi esnasında, 3’ü nesne kadrajdan çıkarken, 2 tanesi de nesne kadrajdayken anlık tahmin sapması şeklinde gerçekleşmiştir. Alexnet algoritmasının somun nesnelere için yapılan uygulama testlerinde ise 100 testin 65’i başarılı, 35’i başarısız sonuçlanmıştır. Somun nesnelereindeki nesne tanıma başarımlarının düşmesinin temel sebebinin, nesne kamera kadrajı içerisindeyken tahminin pul olarak hatalı tahmin edilmesi olduğu gözlemlenmiştir. 35 başarısız testin 23’ü bu kapsamda somun

nesnesinin pul olarak hatalı tahmin edilmesiyle gerçekleşirken, 7 adedi nesne kadraja girdiği sırada, 5 adedi ise nesne kadrajdan çıkarken gerçekleşmiştir. Özellikle daha küçük boyutlu somun nesnelere test edilirken hatalı tahmin oranının arttığı tespit edilmiştir. 23 hatalı somun tahmininin 16'sı test edilen 5 farklı somun nesnesinden en küçük boyutlu iki nesnenin testleri sırada meydana gelmiştir. Alexnet algoritmasının pul nesnesi için yapılan uygulama testlerinde 81 test başarılı sonuçlanırken 19 testin başarısız olarak sonuçlandığı görülmüştür. Başarısız testlerin hata dağılımlarına bakıldığında 8 tanesinin pul nesnesi kadraja girerken, 5 tanesi nesnenin tamamı kadrajdayken, 6 tanesi de nesne kadrajdan çıktığı esnada üretilen tahmin çıktısının anlık sapmalar göstermesiyle meydana geldiği gözlemlenmiştir.

Uygulamalı başarımların testlerinin ikinci aşamasında Googlenet algoritmasının test ortamındaki başarımlarını incelenmiştir. Googlenet algoritmasının vida nesnelerini tanıma başarımlarını incelemek için yapılan testlerde, nesnenin kayar bandın başından sonuna kadar geçişi esnasında, kadrajda vida yokken nesne yok tahmini, vida kadraja girdikten kadrajdan çıkana kadar vida tahmini, vida kadrajdan çıktığında tekrar nesne yok tahmini üreterek başarılı bir akış ürettiği test sayısı 96 olmuştur. Başarısız olan 4 akışın 2 tanesi vidanın kadraja girdiği sırada, 2 tanesi de vidanın kadrajdan çıkması esnasında olduğu gözlemlenmiştir. Googlenet algoritmasının somun nesnesi için yapılan uygulamalı kıyaslama testlerinde “nesne yok-somun-nesne yok” şeklinde başarılı tahmin akışlarının gerçekleşme sayısı 76 olmuştur. Başarısız sonuçlanan 24 testin içerisinde 8 adedi somun nesnesi kamera kadrajına girerken, 7 tanesi somun nesnesi kadrajdayken tahminin pul olarak anlık değişmesi nedeniyle, 3 tanesi somun nesnesinin pul olarak hatalı tahmin edilmesiyle, 6 tanesi de nesne kadrajdan çıkarken oluşmuştur. Son olarak algoritmanın pul nesnesi tanıma başarımlarını incelenmiştir. Googlenet algoritması, 100 kez tekrar edilen testlerin 84'ünde “nesne yok-pul-nesne yok” şeklinde başarılı tahmin akışı üretirken, 16 testte en az 1 kez başarılı akışın dışında tahmin üreterek başarısız olmuştur. Başarısız testlerin 5'i pul nesnesi kamera kadrajına girerken, 4'ü kadrajda pul nesnesi varken tahminin anlık olarak somun olarak değişmesiyle, 2 tanesi nesne kadrajdayken somun olarak hatalı tahmin edilmesiyle, 5 tanesi de pul nesnesi kadrajdan çıkarken anlık tahmin sapması sonucunda başarısız olduğu görülmüştür.

Algoritmaların uygulamalı başarımların kıyaslamalarının son aşamasında Squeezenet algoritmasının nesne tanıma başarımlarını incelenmiştir. Algoritmanın vida nesnesi için

yapılan 100 tekrarlı testin 98 adedinde tahmin çıktılarını “nesne yok-vida-nesne yok” şeklinde üreterek başarılı olduğu görülürken, yalnızca 2 adet testten başarısız olduğu görülmüştür. Başarısız olan testlerin biri vida kadraja girerken, diğeri de vida kadrajdan çıktığı sırada nesne yok tahminine anlık sapma oluşmasıyla gerçekleştiği görülmüştür. Somun nesnesi için yapılan 100 tekrarlı testin 90’ında algoritma tarafından “nesne yok-somun-nesne yok” şeklinde başarılı tahmin akışı oluşturulurken 10 test başarısız sonuçlanmıştır. Başarısız testlerin dağılımı incelendiğinde, 3 tanesinin somun nesnesi kadraja girerken anlık sapmayla, 2 tanesinin nesne kamera kadrajı içerisindeyken tahminin pul olarak anlık sapma göstermesiyle, 5 tanesinin de somun nesnesinin kadrajdan çıktığı esnada anlık sapmayla oluştuğu gözlemlenmiştir. Pul nesnesi için gerçekleştirilen 100 testte ise algoritma 92 kez “nesne yok-pul-nesne yok” şeklinde başarılı tahmin akışı oluştururken, 8 testte başarısız olmuştur. Başarısız testlerin 4 tanesi pulun kamera kadrajına girişi esnasında anlık tahmin sapmasıyla, 1 tanesi pul kamera kadrajındayken anlık tahmin sapmasıyla, 3 tanesi de pulun kamera kadrajından çıkışı esnasında anlık tahmin sapmasıyla oluşmuştur.

3.5. Algoritma Başarımlarının Karşılaştırılması ve Uygun Algoritmanın Seçilmesi

Bölüm 3.3’te bu tez çalışması kapsamında incelenen Alexnet, Googlenet ve Squeezenet algoritmalarının pul, vida, somun ve nesne olmayan durumları tanıma başarımları benzetim başarımları testleri koşturularak incelenmiştir. Bölüm 3.4’te ise bu algoritmaların uygulamalı başarımları testleri yapılarak, her bir nesne sınıfı için 100 denemenin kaçında başarılı oldukları araştırılmıştır. Bu bölümde ise yapılan benzetim ve uygulama başarımları testlerinin sonuçları bir bütün olarak değerlendirilecek ve 4. Bölümde UR5 işbirlikçi robotla yapılacak sınıflandırma işleminde kullanılacak algoritmanın seçimi yapılacaktır.

Algoritmaların benzetim başarımları testlerinin sonuçları Çizelge 3.3’te, uygulama başarımlarının kıyaslama testi sonuçları ise Çizelge 3.4’te özetlenmiştir. Alexnet algoritması, eğitim süresi açısından Squeezenet algoritmasıyla birlikte en hızlı eğitimi tamamlayan algoritma olarak öne çıkmıştır.

Çizelge 3.3 Algoritmaların Benzetim Başarımları Kıyaslama Çizelgesi

Algoritma	Eğitim Süresi	Benzetim Başarım Yüzdesi
Alexnet	16 dakika 25 saniye	91,80%
Googlenet	46 dakika 29 saniye	93,24%
Squeezenet	15 dakika 17 saniye	96,76%

Çizelge 3.4 Algoritmaların Uygulama Başarımları Kıyaslama Çizelgesi

Algoritma	Vida	Somun	Pul	Uygulama Başarım Yüzdesi	Benzetim Başarımından Sapma
Alexnet	91 Başarılı 9 Başarısız	65 Başarılı 35 Başarısız	81 Başarılı 19 Başarısız	79,00%	(-) %12,80
Googlenet	96 Başarılı 4 Başarısız	76 Başarılı 24 Başarısız	84 Başarılı 16 Başarısız	85,00%	(-) %8,24
Squeezenet	98 Başarılı 2 Başarısız	90 Başarılı 10 Başarısız	92 Başarılı 8 Başarısız	93,33%	(-) %3,43

Eğitim süresinin kısa olması, aktarım öğrenme yazılımında iyileştirme amaçlı bir değişiklik yapıldığında bu değişikliğin sonuçlarını görmek için algoritmanın yeniden eğitilmesi gerekeceği için önem teşkil etmektedir. Buna karşılık, eğitim süresinin algoritmaların nesne tanıma başarımına doğrudan bir etkisi olmadığı için, tek başına tercih sebebi olmak için yeterli gelmemektedir. Alexnet algoritmasının benzetim başarım yüzdesi %91,80, uygulamalı testlerde ise %79 olarak ölçülmüştür. Uygulamalı

kıyaslama testlerinde, benzetim başarımları testi sonrası oluşan %91,80'lik başarımlar beklentisinden %12.80 sapma gerçekleşmiştir. Bu sapmanın en büyük nedeni de uygulamalı testler sırasında Alexnet algoritmasının somun nesnelere tahmin etme başarısının diğer algoritmalarla kıyasla çok düşük kalmasıdır. 100 adet somun testinin yalnızca 65'ini başarılı tahmin edebilen algoritma, özellikle başarısız olan 35 testin 23'ünde somun nesnesini pul olarak tanımlamasıyla düşük bir başarımlar göstermiştir. Hem benzetim hem de uygulamalı başarımlar kıyaslama testlerinde en kötü başarımlar göstermesi nedeniyle Alexnet algoritması, UR5 işbirlikçi robotla yapılacak nesne sınıflandırma işlemleri için tercih edilmemiştir.

Googlenet algoritmasının benzetim ve uygulama testleri başarımlar birlikte değerlendirildiğinde benzetim başarımlar testi, algoritmanın %93,24 isabetlilikle doğru tahminler üretebileceğini öngörmüştür. Bu oranıyla Alexnet algoritmasını geride bırakmış, ancak %96,76 benzetim başarımlar yüzdesine sahip Squeezenet algoritmasının gerisinde kalmıştır. Ayrıca 46 dakika 29 saniye süren eğitim süresiyle, Squeezenet algoritmasının 15 dakika 17 saniyelik eğitim süresine kıyasla 3 kat daha yavaş eğitilebilmesi de benzetim başarımlar testlerinde görülen bir diğer olumsuzluk olmuştur. Uygulamalı testlerde ise her bir nesne için Googlenet algoritmasının başarılı tahmin sayısı, Alexnet algoritmasını geçmiş ancak Squeezenet algoritmasının gerisinde kalmıştır. Uygulamalı testler sonucunda %85 başarımlar gösteren algoritma, benzetim başarımlar yüzdesi tahmini olan %93,24'ten %8,24'lük bir sapma göstermiştir. Googlenet algoritmasının uygulama başarımlar testleri sırasında görülen en büyük problem, somun testlerinde 3 kez, pul testlerinde ise 2 kez olmak üzere toplamda 5 kez nesne kamera kadrajının içerisindeyken hatalı sınıflandırma yapması olmuştur. Benzetim ve uygulama başarımlar kıyaslamaları testlerinin tamamında Squeezenet algoritmasının gerisinde kalması dolayısıyla Googlenet algoritması da tercih edilmemiştir.

Squeezenet algoritması, 15 dakika 17 saniye ile incelenen algoritmalar arasında en hızlı eğitilebilen algoritma olmuştur. Benzetim başarımlar testinde de %96,76 ile en yüksek benzetim başarımlar yüzdesini elde etmiştir. Uygulamalı başarımlar testlerinde her bir nesne için en yüksek başarılı test sayısına sahip algoritma olmayı başarmıştır. Uygulamalı başarımlar testlerinde gösterdiği %93,33 başarımlar yüzdesiyle, benzetim başarımlar testlerindeki %96,76'lık başarımlar tahmininden yalnızca %3,43 sapma göstermiş ve bu alanda da en düşük sapmayı gösteren algoritma olmuştur. Tüm bunlara ek olarak,

uygulamalı başarımlar testlerinde Alexnet algoritması 23 kez, GoogLeNet algoritması 5 kez kamera kadrajı içerisindeki nesneyi tamamen hatalı tahmin ederken, SqueezeNet algoritması uygulama testleri süresince hiç hatalı tahmin üretmemiştir. Uygulamalı testlerde başarısız sonuçlanan toplam 20 testin tamamında, ürettiği doğru tahminin anlamlı olarak sapma göstermesinden dolayı testi başarısız kabul edilmiştir.

Yukarıda açıklanan tüm sebepler birlikte değerlendirildiğinde, gerek benzetim gerek uygulamalı başarımlar kıyaslamaları testlerinde en yüksek başarımları gösteren SqueezeNet algoritması, UR5 işbirlikçi robotla yapılacak olan nesne sınıflandırma işleminde kullanılmak üzere seçilmiştir.

4. UR5 İŞBİRLİKÇİ ROBOTLA SINIFLANDIRMA

Çalışma kapsamında incelenen Alexnet, Googlenet ve Squeezenet evrişimli derin sinir ağlarının; vida, pul ve somun nesnelere sınıflandırma başarımları üçüncü bölümde detaylı olarak incelenmiş, yapılan benzetim ve uygulamalı başarımların testleri sonucunda en yüksek başarımları gösteren Squeezenet derin sinir ağı, UR5 işbirlikçi robotla gerçekleştirilecek sınıflandırma işlemi için seçilmiştir. Bu bölümde UR5 işbirlikçi robotun sahip olduğu genel yetenekler, UR5 robotuyla yapılan sınıflandırma işlemi için gerekli donanım altyapısının nasıl oluşturulduğu, robotun hareketlerini kontrol eden robot kontrol yazılımının hazırlık aşamaları ve MATLAB ortamında hazırlanan nesne sınıflandırma yazılımı ile UR5 robot arasındaki haberleşmenin nasıl sağlandığı anlatılacaktır.

4.1. UR5 İşbirlikçi Robot

Tez çalışması kapsamında sınıflandırma işlemlerini gerçekleştirecek işbirlikçi robot olarak Universal Robots firmasının UR5 model robotu kullanılmıştır. Model isminin sonunda yer alan 5 rakamı, robotun maksimum yük taşıma kapasitesi olan 5kg değerini ifade etmektedir. UR5 modeli, insan kolunun yapabildiği tüm hareketleri taklit edebilme ve gerçekleştirebilmek amacıyla 6 döner ekleme sahiptir. Sahip olduğu döner eklemler sayesinde insan kolunda bulunan omuz, dirsek ve bilek hareketlerini 360 derecelik rotasyonlarla gerçekleştirebilmektedir.

UR5 işbirlikçi robotun ağırlığı 18kg olarak ölçülmüştür. Robotun hareketleri sırasında üzerinde bulunan masanın sarsılmaması için, ayaklarında kilit mekanizması bulunan bir masa üzerine monte edilmiş ve masanın ayakları kilitlenmıştır. Robotun masa yüzeyine oturma alanı 14,9cm olarak ölçülmüştür. Robotun oturulduğu alan merkez olmak üzere, merkezden 85cm mesafeye kadar maksimum uzanabilme kapasitesi bulunmaktadır. Robot 1m/sn maksimum hareket hızına ve 180°/sn maksimum ekleme dönüş hızına sahiptir.

UR5 robotun aynı zamanda 475x423x268 mm boyutlarında bir kontrol kutusu ve 12" boyutunda dokunmatik ekrana sahip bir programlama kumandası bulunmaktadır. Programlama kumandası, Polyscope grafik kullanıcı arayüzünü içermektedir. Bölüm 4.3'de detayları anlatılan robot kontrol programı, bu arayüz kullanılarak tasarlanmıştır. Robotun kontrol kutusu üzerinde 16 adet sayısal giriş ve çıkış, 2 adet analog giriş ve çıkış bulunmaktadır. Robot kontrol kutusu üzerinden 24V, 2A; robot kolun üzerinden ise 12/24V, 0,6A güç çıkışı verebilmektedir. Bölüm 4.2'de bahsedilen uç işlevcinin güç ihtiyacı, kontrol kutusunun 24V,2A DC güç çıkışı kullanılarak karşılanmıştır. Robot; TCP/IP 100Mbit, IEEE802.3u, 100BASE-TX ve Ethernet yuvası & Modbus TCP iletişim protokollerini desteklemektedir [57]. UR5 işbirlikçi robotun genel görünümü Şekil 4.1'de verilmiştir.



Şekil 4.1 UR5 İşbirlikçi Robot

4.2. Donanım Hazırlıkları

UR5 işbirlikçi robot kullanılarak yapılan nesne sınıflandırma işlemlerinde, sınıflandırılacak malzemelerle robotu beslemek için frekans kontrollü bir doğrusal titreştirici kullanılmıştır. Malzemelerin düzgün bir şekilde ilerleyebilmesi için, 10cm genişliğinde ve 80cm uzunluğunda bir mekanik gereç, doğrusal titreştirici üzerine monte edilmiştir. Besleme hattının bitişinden 10cm geride yer alacak şekilde bir adet

yakınlık algılayıcı konumlandırılmıştır. Yakınlık algılayıcı, iki farklı amaçla tercih edilmiştir. Bunlardan birincisi, algılayıcının önüne malzeme geldiğinde doğrusal titreştiricinin titreşimi durdurmasını sağlamak üzere doğrusal titreştiriciye girdi sağlamaktır. İkincisi ise robota malzeme var bilgisini iletme üzere girdi oluşturmaktır. Bu sayede algılayıcı önüne malzeme geldiğinde doğrusal titreştirici durdurularak malzemenin algılayıcı hizasında robotun alması için beklemesi, aynı zamanda da robota malzeme var bilgisi sayısal girdi olarak iletilerek robotun malzeme almak üzere harekete geçmesi için gerekli ön şartlardan birinin sağlanması mümkün olmuştur. Hattan gelen vida, pul ve somunların yükseklik değerleri birbirlerinden farklı olduğu için, yakınlık algılayıcıdan çıkan lazer ışınının, yüksekliği en düşük malzemeyi de algılayabilmesi için, mekanik gerecin zemininin 1mm yukarısında zemine paralel olarak konumlanacak şekilde ayarlanmıştır. Robotun malzemeleri almasını kolaylaştırmak için, yakınlık algılayıcının 10cm gerisine, hattın genişliğini 10cm'den 6cm'e düşürecek şekilde 2 adet mekanik çubuk karşılıklı olarak monte edilmiştir. Hattan gelecek çapı en büyük olan malzeme 5cm çaplı bir pul olduğu için, çubuklar arası mesafe herhangi bir malzemenin geçişini engellemek adına 6cm olarak ayarlanmıştır.

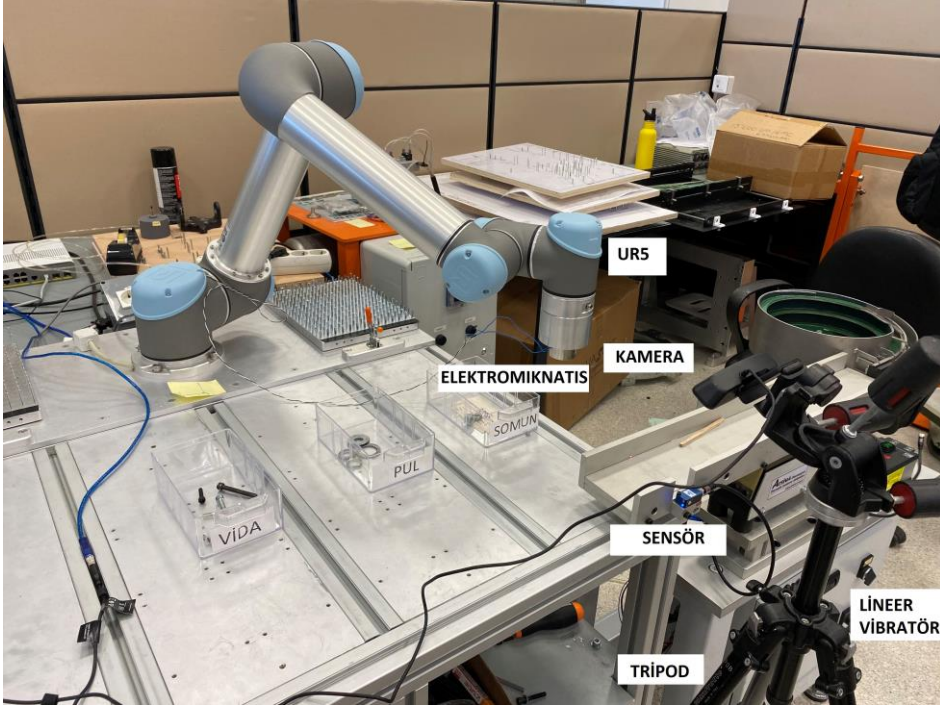
UR5 işbirlikçi robotun, malzemeleri alıp, ait oldukları kutulara götürüp bırakabilmesi için uygun bir uç işlevci seçimi oldukça önemlidir. Sınıflandırılan vida, somun ve pulların birbirinden farklı geometrik özelliklerde oluşu, aynı sınıfa ait nesnelere arasında dahi farklı uzunluk ve çaplarda malzemelerin olması, tüm malzemelerin robot tarafından yakalanıp ait olduğu kutuya atılmasını sağlayacak ortak bir uç işlevci seçimi yapılmasını zorlaştırmaktadır. Bu problemin üstesinden gelmek için, sınıflandırılacak tüm malzemelerin metalik ve mıknatıslanabilir özellikte olmasından dolayı elektromanyetik kontrollü bir mıknatıs kullanımı tercih edilmiştir. Bu sayede tüm malzemelerin, robot tarafından kavranmasına gerek kalmadan hattın üzerinden alınıp, ait olduğu nesne sınıfına ait kutulara yerleştirilmesi sağlanmıştır. Tercih edilen elektromanyetik mıknatıs 12V DC gerilimle çalışmaktadır. Mıknatısın güç besleme hatları, UR5 işbirlikçi robotun 24V,1A çıkış verebilen sayısal çıkışına, 24V-12V DC-DC çevirici üzerinden bağlanarak mıknatısa gerekli güç beslemesi sağlanmıştır. Mıknatısın, robotun uç kısmına monte edilebilmesi için, bir tarafı robot kolun ucuna monte edilebilen, diğer tarafı mıknatısın üzerinde bulunan metrik 8 ölçüsündeki vida girişine sahip bir ara bağlantı parçası kullanılarak elektromıknatısın robot kolun uç işlevcisi olarak kullanımı sağlanmıştır. Ara bağlantı parçası ve üzerine

monte edilen elektromanyetik mıknatıstan oluşan uç işlevcisi, Şekil 4.2’de gösterilmiştir.



Şekil 4.2 Uç İşlevci

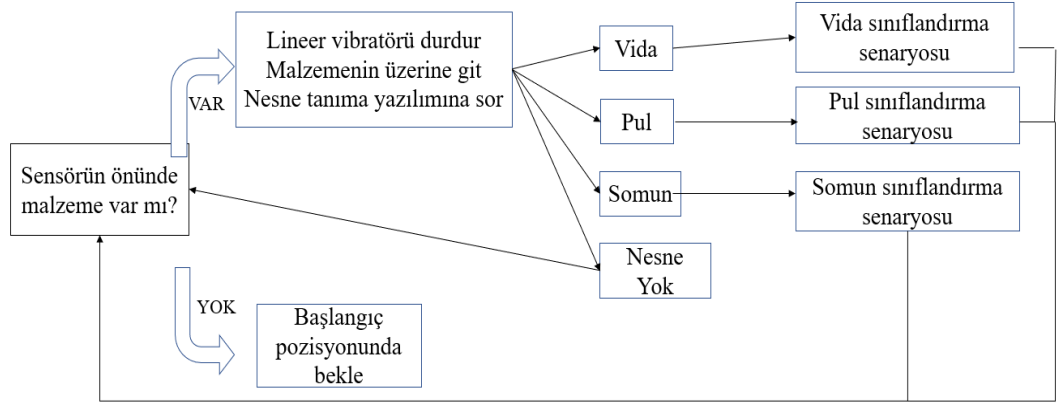
Hattan gelen malzemeleri gözlemleyecek olan USB kamera, tripod üzerine sabitlenerek, yakınlık algılayıcının önündeki alanı görecektir. Kamera konumu ayarlanırken, robotun, hattan gelen malzemeleri alma işlemi esnasında kameraya çarpmayacağı güvenli bir mesafeye yerleştirilmesine dikkat edilmiştir. Kamera açısı yakınlık algılayıcının önündeki alanı görecektir şekilde ayarlanıp, kadrajda sadece bu alan olacak şekilde sayısal yakınlaşma uygulanmıştır. Vida, pul ve somun malzemelerinin kendilerine ait olan kutularda sınıflandırılabilmesi için 3 adet kutu yanyana ve aralıklı olarak test masası üzerine yerleştirilmiştir. Hazırlanan donanım altyapısının genel görünümü Şekil 4.3’te verilmiştir.



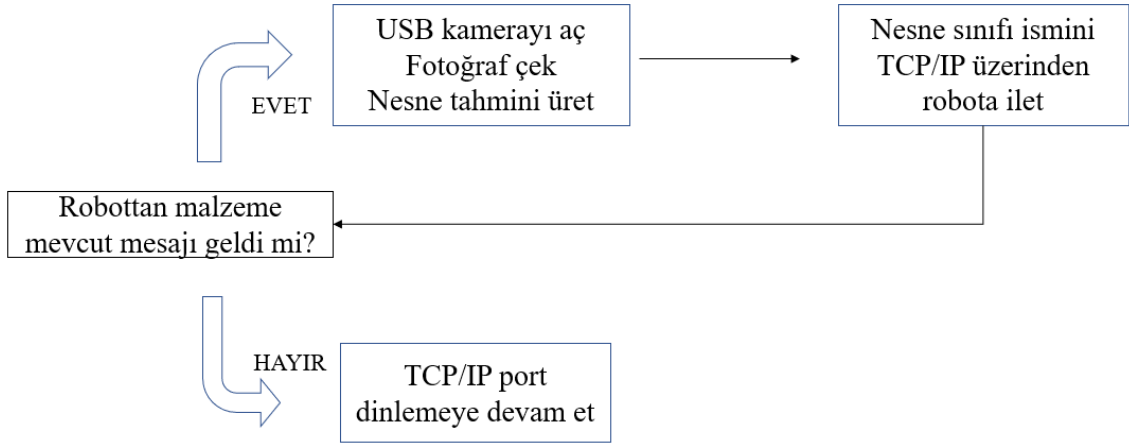
Şekil 4.3 Donanım Altyapısı Genel Görünümü

4.3. Robot Kontrol Yazılımının Hazırlanması

Bölüm 4.2’de detayları anlatılan donanım altyapısı kurulduktan sonra, ilgili donanımlarla, tezin konusu olan nesnelerin sınıflandırma işlemlerinin gerçekleştirilebilmesi için robot kontrol yazılımı geliştirilmiştir. Robot kontrol yazılımı, UR5 işbirlikçi robotun kendi arayüzünde bulunan Polyscope programlama arabirimiyle hazırlanmıştır. Bununla beraber, SqueezeNet algoritmasının, kameradan gelen görüntülere göre yaptığı nesne tahmin bilgilerini robota ileterek robotun, hattan gelen malzeme tipine göre ne yapacağını kontrol etmek üzere MATLAB geliştirme ortamında ayrı bir nesne tanıma yazılımı hazırlanmıştır. Hazırlanan nesne tanıma yazılımı ile UR5 robotun kullanıcı paneli üzerinde hazırlanan robot kontrol yazılımlarının birbirleriyle iletişimleri TCP/IP haberleşme protokolü üzerinden gerçekleştirilmiştir. TCP/IP haberleşmesi kurulurken, test bilgisayarı üzerinde koşturulan MATLAB yazılımı sunucu, UR5 kontrol paneli üzerinde koşturulan robot kontrol yazılımı ise istemci olarak ayarlanmıştır. Hazırlanan robot kontrol yazılımının akışı Şekil 4.4’te, nesne tanıma yazılımının akışı Şekil 4.5’te verilmiştir.



Şekil 4.4 Robot Kontrol Yazılımı Akışı



Şekil 4.5 Nesne Tanıma Yazılımı Akışı

Hazırlanan MATLAB programı koşturulduğunda, ilk olarak, Bölüm 3.1’de anlatıldığı şekilde hazırlanan aktarım öğrenme yazılımıyla eğitilmiş olan SqueezeNet algoritmasını çalışma alanına yükler. USB üzerinden bağlı bulunan kameranın yaklaşma, odak gibi ayarlarını yapılandırır. TCP/IP haberleşmesi için IP adresi ve port ayarlarını gerçekleştirdikten sonra ilgili port bağlantısını açarak portu dinlemeye başlar. Bu noktada robot kontrol yazılımı tarafından malzeme mevcut bilgisi iletilmediği sürece MATLAB yazılımı, robottan bu bilgiyi beklemeye devam eder. Robot kontrol yazılımı tarafından malzeme mevcut bilgisi geldiğinde ise kamerayı çalıştırır ve 10 adet fotoğraf çeker, son çekilen fotoğrafı SqueezeNet derin sinir ağına girdi olarak iletir. Bu işlem, hareket halindeki nesnelerin algılayıcının önüne gelip durduğu sırada kameranın çektiği fotoğrafın nesne henüz hareket halindeyken çekilmesini önlemek amacıyla yapılmaktadır. Kameradan gelen görüntüye göre nesne tahmini bilgisini oluşturup, üretilen tahmin çıktısına göre robot kontrol yazılımına TCP/IP üzerinden “vida”, “pul”,

“somon” ya da “nesne yok” şeklinde yazı dizisi gönderir. Ek olarak, kameranın çektiği fotoğrafı ve bu fotoğrafa yapılan nesne sınıflandırma tahminini ekrana çıktı olarak basarak kullanıcıya bildirir. Program sonlandırıldığında ise TCP/IP iletişimini durdurarak dinlediği portu kapatır.

Robot kontrol yazılımı ise koşturulduğunda ilk olarak robotun başlangıç konumu, malzemeleri almak üzere hattın üzerine gideceği ve alçalacağı nokta, gelen malzeme çeşidine göre malzemeleri aldıktan sonra bırakacağı noktalar gibi koordinat parametreleri ayarlanır. TCP/IP haberleşmesi başlatılıp ip adresi ve port ayarları yapılandırılarak robot istemci moduna geçer. Tüm parametre değerleri ayarlandığında robot Şekil 4.3’te görülen başlangıç konumuna gider ve beklemeye başlar.

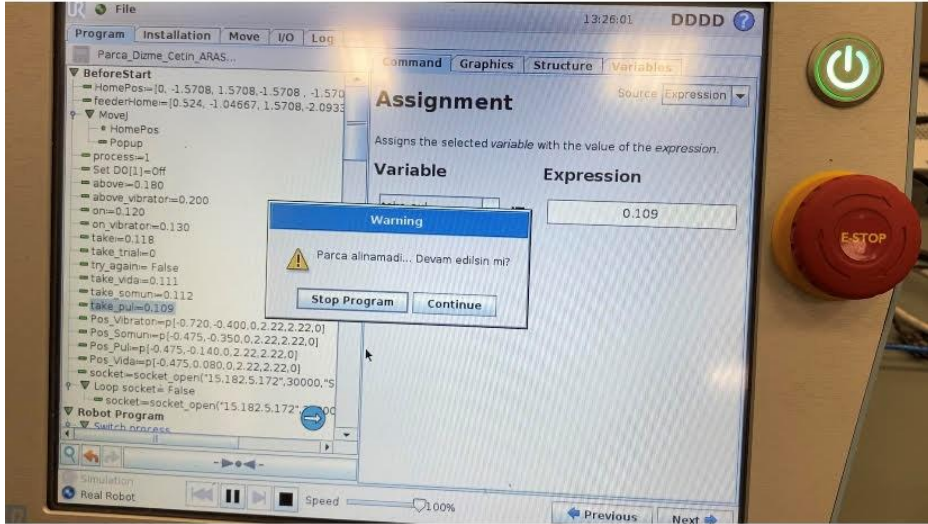
Robotun hattan gelen malzemeyi almak için harekete geçmesi iki adet ön koşula bağlıdır. Bunlardan ilki yakınlık algılayıcının önüne malzeme gelerek algılayıcıdan çıkan lazer ışınının önünü kesmesiyle algılayıcının mevcut durum bilgisinin değişmesidir. Yakınlık algılayıcı tarafından robota malzeme mevcut bilgisi iletilindiğinde, robot tarafından bu bilgi TCP/IP üzerinden “malzeme mevcut” yazı dizisiyle mesaj olarak MATLAB programına iletilerek, programın kamerayı açıp fotoğraf çekmesi ve görüntüye göre sınıflandırma çıktısını üretmesi sağlanır. Robotun malzeme almak üzere harekete geçmesinin ikinci ön koşulu ise, üretilen tahmin çıktısının pul, somun ve vida nesnelere ait olması; “nesne yok” tahmin çıktısının üretilmemiş olmamasıdır. Bu sayede robot harekete geçmeden önce hattın üzerinde gerçekten bir malzeme olduğu bilgisi hem yakınlık algılayıcı sayesinde donanımsal, hem de tahmin çıktısı kontrolüyle yazılımsal olarak çift aşamalı kontrol edilmektedir. Bu sayede algılayıcının bozularak malzeme olmadığı halde malzeme mevcut bilgisi iletilmesi ya da MATLAB yazılımının malzeme olmadığı halde hatalı tahmin yaparak bir malzeme mevcut olarak algılaması gibi istenmeyen durumlara karşı robotun harekete geçebilmesini önlemek adına çift taraflı güvenlik tedbiri alınmıştır.

Yakınlık algılayıcıdan gelen malzeme mevcut bilgisi robot kontrol yazılımı tarafından MATLAB programına iletilmekte ve malzemenin ne olduğu sorulmaktadır. MATLAB programı tarafında ise malzeme mevcut bilgisi geldiğinde kamera açılarak kadrajda bulunan görüntüye dair sınıflandırma tahmini çıktısı oluşturulup malzemenin sınıf bilgisi robot kontrol yazılımına cevap olarak dönmektedir.

Robot kontrol yazılımı, algılayıcının önüne hangi malzemenin geldiğine göre değişen alt senaryolara ayrılmıştır. Farklı nesne sınıfları için, robotun malzemeyi almak için alçalacağı yükseklik değerleri ile malzemeyi taşıyıp bırakacağı kutuların konum bilgileri değişkenlik göstereceği için, robot kontrol yazılımı, nesne sınıfına göre alt senaryolara ayrılmıştır. MATLAB programından gelen nesne sınıfı bilgisine göre robot kontrol yazılımı, ilgili nesne sınıfına ait senaryoyu çalıştırmaktadır. Bu senaryolar çalıştırıldığında robot ilk olarak malzeme akış hattının üzerine, yakınlık algılayıcının hizasına 20cm yükseklikten gelerek hattı ortalayıp sonrasında ilgili nesne sınıfına ait tanımlı yükseklik bilgisine göre alçalmaktadır. Malzemeyi alma noktasına ulaştığında ise sayısal çıkışı aktif hale getirerek elektromanyetik mıknatısı aktifleştirmektedir. Mıknatısın aktifleştirilmesi sonrasında önce hattın bir kenarına, sonra diğer kenarına, son olarak da tekrar ortaya doğru yatay ekseninde hareket ederek algılayıcının hizası boyunca hattı taramaktadır. Özellikle vidaların hattan akışı sırasında, doğrusal titreştiricinin daha hızlı bir akış sağlamak için yüksek frekanslarda çalıştırıldığı durumlarda algılayıcı tarafından hat durdurulduğunda yapısı gereği sağa sola yuvarlanma ihtimaline karşı malzemenin mıknatıs tarafından alınmasını sağlamak için robota yatay ekseninde bu hareket yaptırılmaktadır. Algılayıcı hizasında elektromıknatıs açıkken hattın yatay ekseninde taranması tamamlandıktan sonra malzemeyi alan robot hattın ortasına gelmekte ve oradan öncelikle olduğu yerde yükselip, devamında aldığı malzemeye ait kutunun üzerine gelmekte ve sayısal çıkışı kapatarak mıknatısı devredışı konuma getirmekte ve malzemenin ait olduğu sınıfın kutusuna bırakılması sağlanmaktadır. Bu aşamada, robot kontrol yazılımı hazırlanırken yapılan deneylerde, özellikle pul gibi çok hafif olan malzemelerde, mıknatıs kapatılmasına rağmen indüklenen manyetik alandan dolayı malzemenin kutuya düşmediği, uç işlevcisine yapışık kalmaya devam ettiği durumlar gözlemlenmiştir. Bu problemin üstesinden gelmek için, robot kolun malzemenin ait olduğu kutunun üstüne gelip mıknatıs kapatıldıktan sonra önce yukarı doğru 5cm, sonra aşağı doğru 5cm en yüksek ivmeyle hareket ettirilerek malzemenin kutu içerisine düşürülmesi sağlanmıştır. Malzeme bırakıldıktan sonra robot kontrol yazılımı tarafından robot kolu tekrar başlangıç konumuna gönderilip, bir sonraki malzeme gelene kadar bu konumda beklemeye devam etmektedir.

Elektromıknatisin çalışmaması veya algoritmanın malzemeyi hatalı sınıflandırmasından dolayı uç işlevcinin malzemeyi almak için malzemeye yeteri kadar yaklaşmaması gibi nedenlerle robotun malzemeyi alamadığı durumlara karşı, robotun malzemeyi almak için yapacağı tekrarlı deneme sayısı üç kez ile sınırlandırılmıştır. Bu sınırlandırma, robot kontrol yazılımında robotun her bir malzemeyi almak için kaç deneme yaptığını sayan bir sayaç üzerinden sağlanmıştır. Robot, malzemeyi alıp alamadığını, hattın üzerine gelip malzemeyi almak için algılayıcı hizasındaki yatay ekseninde hattı tarama hareketini yapıp yükseldiğinde, yakınlık algılayıcıdan malzeme mevcut bilgisinin gelip gelmemesinden anlamaktadır. Nesneyi almak için hareketini yapıp yükseldikten sonra yakınlık algılayıcıdan gelen bilgi “malzeme yok” olarak değiştiğinde aldığı malzemeyi ait olduğu sınıfa ait kutuya götürüp bırakmakta, algılayıcıdan malzeme mevcut bilgisi gelmeye devam ediyorsa nesneyi alamadığını anlayıp, malzeme mevcut bilgisini TCP/IP üzerinden MATLAB programına ileterek algoritmanın mevcut nesne için yeniden sınıflandırma çıktısı üretmesi istenmektedir. Bu işlem 3 kez tekrarlandıktan sonra malzeme alınmayıp yakınlık algılayıcı tarafından hala malzeme mevcut bilgisi geldiği durumda, robot kontrol paneli üzerinde Şekil 4.4’te gösterilen şekilde uyarı mesajı çıkmaktadır. Bu sayede operatöre malzeme alma işlemini durdurma ya da devam butonuna tıklayarak robotu bir kez daha malzeme almaya yönlendirme seçeneği sunulmuştur. Ek olarak, robotun malzeme almayı başaramadığı durumda sürekli denemeye devam etmesi engellenmiştir.

Robot kolunun malzemeyi almak üzere hattın üzerine gelirken ne kadar alçalacağını belirlemede, ilgili nesne sınıfına ait en yüksek malzeme esas alınarak o nesne sınıfına dair nesnelerin robot tarafından alınacağı yükseklikler belirlenmiştir.



Şekil 4.6 Parça Alınamadı Uyarısı

Vida pul ve somun nesne sınıfları içerisinde en ince malzemeler pul olduğu için, vida ve somun nesnelere SqueezeNet algoritması tarafından pul olarak algılanması durumunda, robotun pul almak için gideceği yükseklik seviyesine gitmeye çalışırken malzemeyi ezme, uç işlevcisine, hattın kendisine ve doğrusal titreştiriciye zarar verme riski ortaya çıkmaktadır. Bu durumun önüne geçmek için robot kontrol paneli üzerinden, robot kolunun uygulayabileceği maksimum güce sınırlandırılma getirilerek, robotun herhangi bir hareketi sırasında bir engelle karşılaştığı durumda güç uygulamaya devam etmesi engellenmiştir.

4.4. Sistem Başarım Testleri

Donanım hazırlıkları ve robot kontrol yazılımının tamamlanması sonrasında, kurulan sistemin tüm donanım ve yazılım arabirimleriyle birlikte nasıl çalıştığını incelemek üzere sistem başarım testleri gerçekleştirilmiştir. Bu bölümde sistem başarım testlerinin nasıl yapıldığı ve bu testlerin sonuçları anlatılacaktır.

Hazırlanan sistemin başarımını ölçmek üzere test bilgisayarında bulunan MATLAB tabanlı nesne tanıma yazılımı ve UR5 işbirlikçi robotun kontrol paneli üzerinde kayıtlı olan robot kontrol yazılımları çalıştırılmıştır. Doğrusal titreştiricinin gücü açılarak malzemelerin ilerleyeceği hatta titreşim verilmiştir. Malzemelerin hattan akışları, belirli bir sıralamayı takip etmeden tamamen rastlantısal bir şekilde malzemeler seçilerek hattın üzerine yerleştirilmiştir. Başarım testleri, Bölüm 3.4'te tariflenen yöntemle uyumlu olarak, her bir malzeme sınıfı için hat üzerinden 100 kez o sınıfa ait

bir malzeme geçene kadar sistem çalıştırılmış ve sonuçlar not edilmiştir. Her bir malzeme için, nesne sınıflandırma yazılımının ilk seferde doğru sınıflandırması ve robotun ilk seferde malzemeyi alarak doğru nesne kutusuna bırakması halinde test başarılı, diğer tüm koşullarda test başarısız kabul edilmiştir. Testler sırasında 2 ayrı malzeme kümesi kullanılmıştır. İlk başarı testi, aktarım öğrenme yazılımının veri kümesinde kullanılan malzemelerle gerçekleştirilmiştir. Veri kümesinde bulunan vidalar hattan toplamda 100 kez geçirilmiş ve tamamının başarılı bir şekilde MATLAB tabanlı nesne sınıflandırma yazılımı tarafından doğru sınıflandırılarak UR5 robot tarafından alındığı ve vida kutusuna yerleştirildiği gözlemlenmiştir. Veri kümesinde kullanılan somunlar, malzeme besleme hattından toplamda 100 kez geçirildiğinde ise 98 kez sınıflandırma operasyonunun başarıyla sonuçlandığı, 2 kez ise başarısız olduğu görülmüştür. Başarısız iki işlemde de nesne sınıflandırma yazılımı tarafından somun yerine pul tahmini üretildiği, bu nedenle de robotun malzemeyi alarak somun kutusu yerine pul kutusuna bıraktığı gözlemlenmiştir. Veri kümesindeki pul nesnelere hat üzerinden 100 kez geçirildiğinde ise 98 sınıflandırma işleminin başarıyla gerçekleştirildiği, 2 işlemin ise başarısız olduğu görülmüştür. Bu iki işlemde de nesne sınıflandırma yazılımının ilk seferde pul yerine somun tahmini ürettiği, bu nedenle robotun somun malzemeleri için tanımlı yükseklikte durduğu ve puldan uzak kaldığı için pulu alamadığı gözlemlenmiştir. Malzeme alınmadığı için robot kontrol yazılımı 2. kez malzemeyi almayı denemiş, bu deneme sırasında nesne sınıflandırma yazılımı bu kez doğru tahmini ürettiği için pullar hattan alınıp pul kutusuna bırakılabilmıştır. Hattan geçen 300 malzeme için gerçekleştirilen işlemlerin 296'sı başarılı 4'ü başarısız olmuş, sistemin genel başarı oranı %98,66 olarak kaydedilmiştir.

Başarı testi ikinci aşamasında, veri kümesinde yer alan malzemeler yerine, veri kümesinde hiç bulunmayan ve SqueezeNet derin sinir ağının aktarım öğrenme yöntemiyle eğitilmesi sırasında kullanılmayan vida, pul ve somunlar kullanılarak testler gerçekleştirilmiştir. Oluşturulan sistemin, aynı sınıfa ait ancak eğitim verisinde hiç kullanılmamış olan nesnelere sınıflandırma başarımlarını incelemek amacıyla bu başarı testi gerçekleştirilmiştir. Kullanılan nesnelere değiştirilmesi dışındaki diğer tüm test koşulları, ilk başarı testiyle aynı tutulmuştur.

Yeni malzemelerle gerçekleştirilen başarı testi sırasında, hattan geçen toplam 100 adet vidanın tamamı doğru sınıflandırılıp doğru kutuya bırakılarak başarılı olmuştur.

Somun testlerinde ise malzeme akış hattından gelen toplam 100 adet somundan 95 tanesi başarılı olarak sınıflandırma işleminden geçerken, 5 işlem başarısız olmuştur. Başarısız olan 5 adet işlemin tamamında, nesne sınıflandırma yazılımının malzemeyi pul olarak sınıflandırdığı ve robotun somunu alıp pul kutusuna attığı gözlemlenmiştir. Pul sınıflandırmasında ise hattan geçen 100 adet pulun 97 tanesinin başarılı olarak sınıflandırılması tamamlanmıştır. Başarısız olan 3 sınıflandırma işleminden 2 tanesinde nesne sınıflandırma yazılımı malzemeyi ilk denemede somun olarak tanımlamış, malzeme alınamadıktan sonra gerçekleşen ikinci denemede malzemeyi doğru tanımlayarak malzemenin alınıp pul kutusuna atılmasını sağlamıştır. Başarısız olan son sınıflandırma işleminde ise her üç malzeme alma denemesinde de pul nesnesi sınıflandırma yazılımı tarafından somun olarak algılandığı için malzeme alınamamış ve robot kontrol arayüzünde “Parça alınamadı” uyarı mesajı çıktıktan sonra işlem sonlandırılmıştır. Yeni malzemelerle toplamda gerçekleştirilen 300 sınıflandırma operasyonunun 292 tanesi başarıyla sonuçlanmış ve ilk seferde başarı oranı %97,33 olarak kaydedilmiştir. Başarım testlerinin sonuçları Çizelge 4.1’de verilmiştir.

Çizelge 4.1 Sistem Başarım Testi Sonuçları

	Vida	Somun	Pul	İlk Denemede Başarım Yüzdesi
Veri Kümesindeki Malzemeler	100 Başarılı 0 Başarısız	98 Başarılı 2 Başarısız	98 Başarılı 2 Başarısız	98,66%
Yeni Malzemeler	100 Başarılı 0 Başarısız	95 Başarılı 5 Başarısız	97 Başarılı 3 Başarısız	97,33%

Veri kümesindeki malzemelerle yapılan sistem başarım testlerindeki başarım yüzdesi, Bölüm 3.4’de Squeezenet derin sinir ağıyla yapılan uygulamalı başarım testlerinden daha yüksek olduğu görülmüştür. Bölüm 3.4’de gerçekleştirilen testlerde, malzeme akışı sırasında nesnelerin kamera kadrajına giriş ve kadrajdan çıkış anlarında yaşanan tahmin sapmaları da başarısız olarak kaydedilirken, sistem başarım testlerinde kullanılan algılayıcı sayesinde sadece malzemeler kamera kadrajının içerisinde sabitken çekilen fotoğraflara göre sınıflandırma tahminleri üretilmiştir. Veri kümesindeki malzemelerle gerçekleştirilen testlerde, nesnenin sadece kamera kadrajı içerisinde yer aldığı durumlar kıyaslandığında Squeezenet derin sinir ağının Bölüm 3.4’de yapılan uygulamalı testlerde 3 kez başarısız sınıflandırma işlemi kaydedilirken, Bölüm 4.4’de gerçekleştirilen sistem başarım testlerinde 4 kez başarısız sınıflandırma

gözlemlenmiştir. Bu durum, oluşturulan sistem başarımının, Bölüm 3.4’de yapılan SqueezeNet algoritması uygulamalı başarım testleri sonuçlarıyla uyumlu olduğunu göstermektedir.

Ek olarak, yeni malzemelerle yapılan sistem başarım testi sonuçlarının, veri kümesindeki malzemelerle yapılan başarım testi sonuçlarına oldukça yakın olduğu görülmüştür. Bu durumda, eğitilen SqueezeNet derin sinir ağının malzeme özelliklerini ezberleyerek aşırı uyum problemine yatkınlık göstermediğini, aynı sınıfa ait farklı ve yeni malzemelerle karşılaştığında malzemelerin özelliklerini tanıyarak doğru nesne sınıfı tahmini özelliklerini başarıyla üretebildiği görülmüştür.

5. SONUÇLAR VE DEĞERLENDİRMELER

Evrişimli sinir ağları, endüstride çeşitli uygulamalarda kullanım alanları bularak giderek yaygınlaşan bir derin öğrenme tekniğidir. Gelişmiş grafik işleme birimlerinin üretilmesiyle büyük verilerin daha hızlı şekilde işlenebilmesi mümkün olmuş ve evrişimli derin sinir ağlarının başarımı yükselmiştir. Nesne tanıma, nesne tespiti, hareket tanıma ve hareket takibi konularında yüksek başarımlı gösteren evrişimli derin sinir ağları, endüstride çeşitli sivil ve askeri uygulamalarda tercih edilebilmektedir.

Aktarım öğrenme tekniği, evrişimli sinir ağlarının eğitilmesinde sıklıkla tercih edilmektedir. Aktarım öğrenmesi yönteminin önemli bir avantajı, bir evrişimli sinir ağının en baştan eğitilmesine kıyasla çok daha az veriyle ve çok daha hızlı bir şekilde eğitilebilmesidir. Nesne tanıma amaçlı tasarlanmış bir evrişimli sinir ağının sınıflandırma çözüm kümesine, aktarım öğrenme yöntemiyle yeni nesnelere tanıma yeteneği eklenebilmekte ya da sınıflandırma çözüm kümesi sadece istenilen yeni nesnelere içerecek şekilde değiştirilebilmektedir.

Tez kapsamında ilk olarak, farklı geometrik özelliklere sahip vida, pul ve somun nesnelere, çeşitli evrişimli sinir ağlarına aktarım öğrenme yöntemi uygulanarak sınıflandırma başarımları incelenmek istenmiştir. Bu amaçla Alexnet, Googlenet ve Squeezenet evrişimli derin sinir ağlarını aktarım öğrenme yöntemiyle eğitmek için bir aktarım öğrenme yazılımı MATLAB geliştirme ortamında hazırlanmıştır. Aktarım öğrenme yazılımında kullanılan eğitim veri kümesi özelliklerinin, aktarım öğrenmesi sonrası nesne tanıma başarımına etkilerini araştırmak üzere çeşitli veri kümeleri elle hazırlanmıştır. Veri kümelerinin oluşturulduğu ortam ışığının ve veri kümesindeki nesnelere fotoğraf kadrajında kapladığı alanın, nesne tanıma başarımına olan etkileri incelenmiştir. En yüksek nesne tanıma başarımının elde edildiği veri kümesi kullanılarak, Alexnet, Googlenet ve Squeezenet algoritmaları hazırlanan aktarım öğrenme yazılımıyla eğitilmiştir. Eğitilen algoritmaların vida, pul, somun nesnelere ve bu nesnelere kamera kadrajında bulunmadığı durumları sınıflandırma başarımları, benzetim ve uygulama ortamında gerçekleştirilen başarımlı kıyaslama testleriyle analiz edilmiştir. Karşılaştırmalı testler sonrasında Squeezenet evrişimli derin sinir ağının, tez

kapsamındaki nesnelere tanımda %93,33 ile en yüksek başarıyı gösteren algoritma olduğu görülmüştür.

Tezin bir sonraki aşamasında, aktarım öğrenme yazılımıyla eğitilen SqueezeNet algoritmasının nesne sınıflandırma tahmin çıktıları kullanılarak, çeşitli vida, pul ve somun nesnelere UR5 işbirlikçi robotla fiziksel olarak sınıflandırılacağı bir altyapı oluşturulması hedeflenmiştir. Malzemelerin doğrusal bir hat üzerinde akışını sağlamak için doğrusal titreştirici, malzeme geldiğinde hattı durdurmak ve robota malzeme mevcut bilgisini iletmek üzere bir yakınlık algılayıcı kullanılmıştır. Malzemelerin hat üzerinden alınıp ait oldukları kutulara yerleştirilmesini sağlamak amacıyla robot kolun uç işlevcisi olarak, malzemelerin metalik özelliğe sahip olması nedeniyle elektromanyetik mıknatıs tercih edilmiştir. MATLAB tabanlı nesne sınıflandırma yazılımından gelen nesne bilgilerine göre robotun ilgili malzemeyi alıp ait olduğu nesne kutusuna malzemeyi bırakmasını sağlamak amacıyla, UR5 işbirlikçi robotun Polyscope programlama arayüzü kullanılarak bir robot kontrol yazılımı hazırlanmıştır. Robot kontrol yazılımı ile nesne sınıflandırma yazılımı arasındaki iletişim TCP/IP haberleşme protokolü üzerinden gerçekleştirilmiştir. Robotun çalışması sırasında malzemeyi alamaması, nesne sınıflandırma yazılımının hatalı tahmin üretmesi gibi istenmeyen senaryolara karşı alınabilecek önlemler araştırılmış, donanımsal ve yazılımsal güvenlik tedbirleri sisteme entegre edilmiştir. Nihayetinde, oluşturulan altyapının tüm donanım ve yazılım arabirimleriyle birlikte çalışma başarıyı test edilmiştir. Sistem başarı testlerinde, nesne sınıflandırma yazılımının tahmin çıktılarını kullanarak UR5 robot tarafından fiziksel olarak gerçekleştirilen nesne sınıflandırma işlemlerinde %97,33'lük başarı oranı elde edilmiştir.

Gelecek çalışmalarda malzemelerin ait olduğu sınıf bilgisiyle beraber metrik ölçülerini de tahmin etmeye çalışan bir sınıflandırma yazılımı geliştirilebilir. Veri kümesi genişletilerek aynı metrik ölçüye sahip nesnelere yeterli miktardaki görselleri kullanılarak aktarım öğrenme yöntemiyle nesnelere metrik ölçülerine göre sınıflandırma başarıları incelenebilir. Metalik özelliğe sahip olmayan farklı geometrik özelliklere sahip nesnelere sınıflandırılmasında kullanılacak uygun uç işlevci tasarımı ve kavrama yöntemleri üzerine araştırmalar yapılabilir. Ek olarak, kızılötesi kamera kullanımıyla malzemenin hangi hammaddeden (pirinç, bakır, çelik vb.) üretildiğini tahmin etmeye yönelik çalışmalar yürütülebilir.

KAYNAKLAR

- [1] N.M. Zamri, G.F. Ling, P.Y. Han, O.S. Yin, Vision-based Human Action Recognition on Pre-trained AlexNet, içinde: 2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2019: ss. 1–5.
- [2] F.R. Mashrur, Automatic Identification of Arrhythmia from ECG Using AlexNet Convolutional Neural Network, içinde: 2019 4th International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2019: ss. 20–22.
- [3] J. Massah, K. Asefpour Vakilian, M. Shabaniyan, S.M. Shariatmadari, Design, Development, and Performance Evaluation of a Robot for Yield Estimation of Kiwifruit, *Computers and Electronics in Agriculture*. 185 (2021) 106–132.
- [4] L. Zhang, Y. Zhang, Y. Li, Path Planning for Indoor Mobile Robot Based on Deep Learning, *Optik*. 219 (2020) 165096.
- [5] A. Chrzanowski, P. Detko, T.P. Stefański, Intelligent Autonomous Robot Supporting Small Pets in Domestic Environment, içinde: IFAC-PapersOnLine, 2019: ss. 446–451.
- [6] K.N. Akpınar, S. Genc, S. Karagol, Chest X-Ray Abnormality Detection Based on SqueezeNet, içinde: 2nd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2020, 2020: ss. 12–13.
- [7] K. Nakamichi, H. Lu, H. Kim, K. Yoneda, F. Tanaka, Classification of Circulating Tumor Cells in Fluorescence Microscopy Images Based on SqueezeNet, içinde: International Conference on Control, Automation and Systems, 2019: ss. 1042–1045.
- [8] H. Durmus, E.O. Gunes, M. Kirci, Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning, içinde: 2017 6th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2017, 2017: ss. 1–5.
- [9] M. Hassanpour, H. Malek, Document Image Classification Using SqueezeNet Convolutional Neural Network, içinde: 5th Iranian Conference on Signal Processing and Intelligent Systems, ICSPIS 2019, 2019: ss. 18–19.
- [10] Z. Wang, J. Huang, N.N. Xiong, X. Zhou, X. Lin, T.L. Ward, A Robust Vehicle Detection Scheme for Intelligent Traffic Surveillance Systems in Smart Cities, *IEEE Access*. 8 (2020) 139299–139312.
- [11] Y. Jiang, X. Li, M. Yu, Z. Bai, Robotic Grasp Detection Using Light-weight CNN Model, içinde: Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020, 2020: ss. 1034–1038.

- [12] W. Gong, X. Zhang, B. Deng, X. Xu, Palmprint Recognition Based on Convolutional Neural Network-Alexnet, içinde: 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), 2019: ss. 313–316.
- [13] M. Abbas, J. Narayan, S. Banerjee, T. Nadu-, S.K. Dwivedy, AlexNet based Real-Time Detection and Segregation of Household Objects using Scorbot, içinde: 2020 4th International Conference on Computational Intelligence and Networks (CINE), 2020: ss. 1–6.
- [14] M. David, Breast Cancer : Tumor Detection in Mammogram Images Using Modified AlexNet Deep Convolution Neural Network, içinde: 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 2017.
- [15] Y. Han, J. Wang, Application of Convolutional Neural Networks in Remote Sensing Image Classification, içinde: Proceedings - 2019 2nd International Conference on Safety Produce Informatization, IICSPI 2019, 2019: ss. 279–282.
- [16] Z. Zhong, L. Jin, Z. Xie, High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps, içinde: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, IEEE, 2015: ss. 846–850.
- [17] T. Fang, A Novel Computer-Aided Lung Cancer Detection Method Based on Transfer Learning from GoogLeNet and Median Intensity Projections, içinde: 2018 IEEE International Conference on Computer and Communication Engineering Technology, CCET 2018, IEEE, 2018: ss. 286–290.
- [18] P. Salavati, H.M. Mohammadi, Obstacle detection using GoogleNet, içinde: 2018 8th International Conference on Computer and Knowledge Engineering, ICCKE 2018, IEEE, 2018: ss. 326–332.
- [19] J. Ma, J. Rao, Y. Qiao, W. Liu, Sprouting Potato Recognition Based on Deep Neural Network GoogLeNet, içinde: Proceedings of 2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things, CCIOT 2018, IEEE, 2018: ss. 502–505.
- [20] W.S. McCulloch, W.H. Pitts, A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics. 5 (1943) 115–133.
- [21] H.J. Kelley, Gradient Theory of Optimal Flight Paths, ARS Journal. 30 (1960) 947–954.
- [22] S. Dreyfus, The Numerical Solution of Variational Problems, Journal of Mathematical Analysis and Applications. 5 (1962) 30–45.
- [23] A.G. Ivakhnenko, V.G. Lapa, U. States., J.P.R. Service., Cybernetic Predicting Devices, CCM Information Corp., New York, 1965.
- [24] D. Crevier, AI: The Tumultuous History of the Search for Artificial Intelligence, içinde: 1993: ss. 100–144.

- [25] K. Fukushima, Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*. 36(4) (1980) 193–202.
- [26] S. Linnainmaa, Taylor Expansion of the Accumulated Rounding Error, *BIT*. 16(2) (1976) 146–160.
- [27] D.E. Rumelhart, G.E. Hinton, Learning Representations by Back-propagating Errors, *Nature*,323(6088). (1986) 533–536.
- [28] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten Digit Recognition with a Back-Propagation Network, içinde: *NIPS*, 1989: ss. 396–404.
- [29] C. Cortes, V. Vapnik, Support-Vector Networks, *Machine Learning*. 20 (1995) 273–297.
- [30] S. Hochreiter, J. Jürgen Schmidhuber, Long Shortterm Memory, *Neural Computation*. 9 (1997) 1735–1780.
- [31] J.F. Kolen, S.C. Kremer, Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies, içinde: *A Field Guide to Dynamical Recurrent Networks*, 2001: ss. 237–243.
- [32] S.Z. Seyyedsalehi, S.A. Seyyedsalehi, Why Dose Layer-by-Layer Pre-training Improve Deep Neural Networks Learning?, içinde: *Handbook of Deep Learning Applications*, 2019: ss. 293–318.
- [33] X. Glorot, A. Bordes, Y. Bengio, Deep Sparse Rectifier Neural Networks, *Journal of Machine Learning Research*. 15 (2011) 315–323.
- [34] K. He, J. Sun, Deep Residual Learning for Image Recognition, içinde: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015: ss. 5353–5360.
- [35] D. Laney, 3D Data Management: Controlling Data Volume, Velocity and Variety, *META Group Research Note*. 949 (2001) 4.
- [36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: A Large-scale Hierarchical Image Database, içinde: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009: ss. 248–255.
- [37] K. Nguyen, C. Fookes, S. Member, A. Ross, S. Member, S. Sridharan, S. Member, Iris Recognition with Off-the-Shelf CNN Features : A Deep Learning Perspective, *IEEE Access*. 6 (2018) 18848–18855.
- [38] Q. V Le, M.A. Ranzato, M. Devin, G.S. Corrado, A.Y. Ng, Building High-level Features Using Large Scale Unsupervised Learning, içinde: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013: ss. 8595–8598.
- [39] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep

- Convolutional Neural Networks, içinde: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012: ss. 1097–1105.
- [40] S. Nayak, Understanding AlexNet | Learn OpenCV, (y.y.). <https://learnopencv.com/understanding-alexnet/> (erişim 21 Mart 2021).
- [41] A.O. Durmaz, Geçmişten Geleceğe Bias., (y.y.). <https://medium.com/datarunner/geçmişten-geleceğe-bias-9ed5176be2e0> (erişim 21 Mart 2021).
- [42] A. Daniel, Understanding AlexNet: A Detailed Walkthrough, (y.y.). <https://towardsdatascience.com/understanding-alexnet-a-detailed-walkthrough-20cd68a490aa> (erişim 21 Mart 2021).
- [43] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-adaptation of Feature Detectors, ArXiv. 1207.0580 (2012).
- [44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 07-12-June (2015) 1–9.
- [45] R. Alake, Deep Learning: GoogLeNet Explained, (y.y.). <https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765> (erişim 28 Mart 2021).
- [46] Anonim, Understanding GoogLeNet Model - CNN Architecture, (y.y.). <https://www.geeksforgeeks.org/understanding-googlenet-model-cnn-architecture/> (erişim 28 Mart 2021).
- [47] S. Arora, A. Bhaskara, R. Ge, T. Ma, Provable Bounds for Learning Some Deep Representations, CoRR, abs/1310.6343. (2013).
- [48] M. Lin, Q. Chen, S. Yan, Network in network, içinde: 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, 2014.
- [49] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, arxiv. 1602.07360 (2016).
- [50] S.-H. Tsang, Review: SqueezeNet (Image Classification), (y.y.). <https://towardsdatascience.com/review-squeezenet-image-classification-e7414825581a> (erişim 10 Nisan 2021).
- [51] K. He, J. Sun, Convolutional Neural Networks at Constrained Time Cost, içinde: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: ss. 5353–5360.
- [52] V. Nair, G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann

- Machines, içinde: ICML 2010 - Proceedings, 27th International Conference on Machine Learning, 2010: ss. 807–814.
- [53] S. Han, H. Mao, W.J. Dally, Deep compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, içinde: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 2016: ss. 1–14.
- [54] K. He, R. Girshick, P. Dollar, Rethinking ImageNet Pre-Training, içinde: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: ss. 4917–4926.
- [55] Anonim, Pretrained Deep Neural Networks - MATLAB & Simulink, (y.y.). https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html#mw_45a8c0b2-26fa-48e9-905a-a7ed7b87bfc8 (erişim 20 Mart 2021).
- [56] D. Soekhoe, P. van der Putten, A. Plaat, On the Impact of Data Set Size in Transfer Learning Using Deep Neural Networks, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 9897 LNCS (2016) 50–60.
- [57] Anonim, UR5 Robot Teknik Özellikleri, (y.y.). https://www.universal-robots.com/media/50630/ur5_tr.pdf (erişim 23 Mayıs 2021).