

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE TEKSTİL TİPİ TANIMA

YÜKSEK LİSANS TEZİ

Gökhan GÜRGEN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

ARALIK 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE TEKSTİL TİPİ TANIMA

YÜKSEK LİSANS TEZİ

**Gökhan GÜRGEN
(504131227)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ece Olcay GÜNEŞ

ARALIK 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 504131227 numaralı Yüksek Lisans öğrencisi Gökhan GÜRGEN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “DERİN ÖĞRENME İLE TEKSTİL TİPİ TANIMA” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Ece Olcay GÜNEŞ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Mürvet KIRCI**
İstanbul Teknik Üniversitesi

Dr. Öğr. Üyesi Bülent BOLAT
Yıldız Teknik Üniversitesi

Teslim Tarihi : **14.11.2019**
Savunma Tarihi : **19.12.2019**





Aileme,



ÖNSÖZ

Öncelikle bu tez çalışmasını yapmama imkan sağlayan ve bu süreçte bana zaman ayırıp bilgisini esirgemeyen danışman hocam Prof. Dr. Ece Olcay GÜNEŞ'e desteği, anlayışı ve yardımı için teşekkürlerimi ve saygılarımı sunarım.

Hayatıma girdiği günden beri benim için en iyisini düşünen, desteğini, sevgisini ve bilgi birikimini hiçbir zaman esirgemeyen eşim Burcu GÜRGEN'e çok teşekkür ederim.

Hayatımın her anında bana yol gösteren, maddi ve manevi desteklerini esirgemeyen, her zamanda yanımda olan ve bugünlere gelmemde sonsuz emeği olan annem Zöhre GÜRGEN'e ve babam İbrahim Etem GÜRGEN'e çok teşekkür ederim.

Aralık 2019

Gökhan GÜRGEN
Elektronik Mühendisi



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xxi
1. GİRİŞ	1
1.1 Tezin Amacı	2
1.2 Literatür Özeti	3
1.3 Hipotez	7
2. DERİN ÖĞRENME VE CNN MİMARİLERİ	9
2.1 Özyinelemeli Sinir Ağları	14
2.2 Evrişimsel Sinir Ağları.....	17
2.2.1 VGGNet	21
2.2.2 AlexNet	22
2.2.3 ResNet.....	23
2.2.4 GoogleNet	24
2.2.5 MobileNet	26
2.2.6 Inception-ResNet	27
3. EVRİŞİMSSEL SİNİR AĞLARI İLE TEKSTİL TİPİ TANIMA	29
3.1 Veri Kümesinin Değerlendirilmesi ve Hazırlanması	29
3.2 Keras Kütüphanesi	31
3.3 Bilenen Bazı Evrişimsel Sinir Ağları Mimarileri ile Tekstil Tipi Tanıma Çalışmaları	32
3.3.1 VGG16 ve VGG19 ile tekstil tipi tanıma çalışmaları	32
3.3.2 MobileNetV2 ile tekstil tipi tanıma çalışmaları.....	35
3.3.3 ResNet50 ile tekstil tipi tanıma çalışmaları	37
3.3.4 InceptionV3 ile tekstil tipi tanıma çalışmaları	39
3.3.5 InceptionResNetV2 ile tekstil tipi tanıma çalışmaları	40
3.3.6 Sonuçların değerlendirilmesi	42
3.4 Önerilen CNN mimarisi ve uygulaması	44
3.4.1 Önerilen CNN mimarisi sonuçlarının analizi.....	47
3.4.2 Önerilen CNN mimarisi ve VGG16 mimarisinin birlikte kullanılması	48
3.5 Çalışmanın Prototiplenmesi	52
4. SONUÇ.....	55
KAYNAKLAR	59
ÖZGEÇMİŞ.....	63



KISALTMALAR

CNN	: Convolutional Neural Network
RNN	: Recurrent Neural Network
GPU	: Graphics Processing Unit
SVM	: Support Vector Machine
PMML	: Predictive Model Markup Language
SIFT	: Scale Invariant Feature Transform
FC	: Fully Connected
FV	: Fisher Vector
ILSVRC	: ImageNet Large Scale Visual Recognition Challenge
VGG	: Visual Geometry Group
RELU	: Rectified Linear Unit



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Bazı mimarilerin performans parametreleri.....	21
Çizelge 2.2 : VGGNet yapılandırmaları.	22
Çizelge 2.3 : 34 Katmanlı ResNet Mimarisi.	24
Çizelge 2.4 : GoogleNet Katmanları.	25
Çizelge 3.1 : Sınıf bazlı veri kümesi.	31
Çizelge 3.2 : Mevcut mimariler için sonuç tablosu.....	43
Çizelge 3.3 : Önerilen mimariye ait katmanlar.	45
Çizelge 3.4 : Özet sonuç tablosu.	48
Çizelge 3.5 : 6-kat çarpaz doğrulama sonuçları.....	50
Çizelge 4.1 : Sonuç özet tablo.....	57



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Dokunsal Sensörün Alt ve Üst Yüzeyleri [5].	4
Şekil 1.2 : Sensör, uygulama ve algılanan yüzeye ait sensor çıktısı [6].	4
Şekil 1.3 : Orijinal tekstil görüntüsü ve Kinect görüntüsü [6].	4
Şekil 1.4 : Tekstil ürünü ve ikili düzen görüntüsü [8].	5
Şekil 1.5 : Yatay, dikey ve noktasal olacak şekilde dokuma hataları [9].	5
Şekil 1.6 : Veri kümesi örnek görüntüleri [3].	6
Şekil 1.7 : Tekstilden görüntü alma sistemi [3].	6
Şekil 1.8 : Kamera ve aydınlatma düzeneği [3].	7
Şekil 1.9 : Raspberry Pi 3 ve Logitech C170 Web-Cam.	8
Şekil 2.1 : Yapay zeka, makine öğrenmesi ve derin öğrenme [11].	9
Şekil 2.2 : Yıllara göre ILSVRC sınıflandırma hata oranları.	11
Şekil 2.3 : Örnek RNN Modeli 1 [13].	15
Şekil 2.4 : Örnek RNN Modeli 2 [13].	16
Şekil 2.5 : Örnek RNN Modeli 3 [13].	16
Şekil 2.6 : CNN katmanları [18].	18
Şekil 2.7 : Görüntü ve konvolüsyon işlemi sonrası özellik haritası [18].	18
Şekil 2.8 : Maksimum pooling işlemi [17].	19
Şekil 2.9 : AlexNet Mimarisi [25].	23
Şekil 2.10 : Inception Modülü [27].	25
Şekil 2.11 : Konvolüsyon filtreleri [29].	26
Şekil 2.12 : (a) Derinlemesine ve (b) noktasal konvolüsyon filtreleri [29].	26
Şekil 3.1 : Veri kümesine ait örnek görüntüler.	30
Şekil 3.2 : VGG16 için tam bağlantı katmanı.	33
Şekil 3.3 : VGG16 için eğitim ve test doğruluk oranları.	33
Şekil 3.4 : VGG16 için hata matrisi.	34
Şekil 3.5 : VGG19 için eğitim ve test doğruluk oranları.	34
Şekil 3.6 : VGG19 için hata matrisi.	35
Şekil 3.7 : MobileNetV2 için tam bağlantı katmanı.	36
Şekil 3.8 : MobileNetV2 için eğitim ve test doğruluk oranları.	36
Şekil 3.9 : MobileNetV2 için hata matrisi.	37
Şekil 3.10 : ResNet50 için tam bağlantı katmanı.	37
Şekil 3.11 : ResNet50 için eğitim ve test doğruluk oranları.	38
Şekil 3.12 : ResNet50 için hata matrisi.	38
Şekil 3.13 : InceptionV3 için tam bağlantı katmanı.	39
Şekil 3.14 : InceptionV3 için eğitim ve test doğruluk oranları.	39
Şekil 3.15 : InceptionV3 için hata matrisi.	40
Şekil 3.16 : InceptionResNetV2 için tam bağlantı katmanı.	41
Şekil 3.17 : InceptionResNetV2 için eğitim ve test doğruluk oranları.	41
Şekil 3.18 : InceptionResNetV2 için hata matrisi.	42
Şekil 3.19 : Modellerin sınıf bazlı performansı.	43
Şekil 3.20 : 3. Konvolüsyonel katman sonrası pamuk sınıfına ait özellik haritası.	46
Şekil 3.21 : 3. Konvolüsyonel katman sonrası yün sınıfına ait özellik haritası.	46

Şekil 3.22 : CNN mimarisine ait filtreler.	46
Şekil 3.23 : Önerilen mimariye ait doğruluk oranı.....	47
Şekil 3.24 : Önerilen mimariye ait hata matrisi.	47
Şekil 3.25 : VGG16 hata matrisi.	49
Şekil 3.26 : Önerilen mimariya ait hata matrisi.	49
Şekil 3.27 : VGG16 ve önerilen CNN mimarisine ait hata matrisi.....	51
Şekil 3.28 : Modellerin sınıf bazlı performansı.....	51
Şekil 3.29 : Modellerin sınıf bazlı performansı.....	52



DERİN ÖĞRENME İLE TEKSTİL TİPİ TANIMA

ÖZET

Donanım teknolojisinin ilerlemesi ve işlem gücü yüksek cihazların günlük hayatın her alanına girmesiyle birlikte yazılım temelli uygulamalar sadece bilgisayar veya gömülü sistem uygulamalarında kalmayıp günlük hayatımızda kullandığımız birçok elektronik cihazda görülmeye başlanmıştır. Bununla beraber hemen hemen her yeni geliştirilen ürün veya ekipmanın internete erişebilir olmasıyla birlikte de her konuda çok ciddi boyutlarda veri toplama imkanı yaratılmaktadır. İşlem gücü yüksek sistemlerin gelişmesi, elde edilebilen verilerin artması, internete bağlanabilen cihazların yaygınlaşmasıyla birlikte makine öğrenmesi, yapay zeka gibi konular günlük hayatta kullanılan beyaz eşya, televizyon, küçük ev aletleri gibi ürün gruplarında da uygulanabilir hale gelmiştir. Bu sayede bu tarz ürünlerde de kişiselleştirilmiş kullanıcı deneyiminin artırılmasına ve sistemin en uygun performansta çalışmasını sağlayacak gerekli kararların yazılım tarafından verilebileceği algoritmaların gelişmesine olanak sağlanmıştır.

Uygun kararları verebilen ve bunları uygulayabilen otonom yapıların gelişmesiyle birlikte malzeme tanıma konusu da bu sistemler için önemli bir problem haline gelmiştir. Robotik uygulamalar, özel üretim süreçleri gibi konulara da girdi sağlayan malzeme tanımlaması üzerinde farklı yöntemler kullanılarak çok sayıda çalışma yapılmıştır. Cam, plastik, metal, tahta gibi temel ayrımların yanı sıra probleme özgü daha alt seviyede sınıflandırmaya odaklanan çalışmaların sayısı da gün geçtikçe artmaktadır.

Tekstil ürünlerine temas eden çamaşır kurutma makinesi, çamaşır yıkama makinesi, ütü gibi ürünler için tekstil tipinin türünün bilinmesi, tekstil ürününe zarar vermeden en uygun performans ile çalışma görevini yerine getirmek açısından oldukça önemlidir. Geleneksel uygulamalarda en uygun performans, kullanıcının seçebileceği tekstil tipine uygun programlarla ya da ayarlayabileceği sıcaklık, süre veya ürüne özgü diğer ayarlarla sağlanmaya çalışılmaktadır. Ancak bu seçim veya ayarlamalarda yapılacak yanlışlıklar, ürünü kullanan kişilerin özellikle de çok değer verdiği giyim ürünleri için istenmeyen sonuçların oluşmasına neden olabilmektedir.

Tekstil ile ilgili olarak lif yapısını tanıma, dokuma tipinin belirlenmesi ya da endüstriyel üretim uygulamalarında dokuma hatalarını tanıma gibi konularda çok sayıda çalışmalar yapılmıştır. Ancak literatür taraması yapıldığında tekstil tipinin tanınması ile ilgili olarak sınırlı sayıda çalışmanın bulunduğu görülmektedir.

Bu tez çalışmasıyla birlikte tekstil tipi tanıma probleminin karmaşık yapılar, sensör tabanlı sistemler kullanılmadan herkesin kolaylıkla ulaşabileceği herhangi bir kamera ile elde edilecek fotoğraf ile çözülmesi amaçlanmıştır. Bu sayede oluşturulacak model ile hem daha az maliyetli hem de ürüne farklı yollardan girdi sağlayabilecek bir sistem oluşturulabilecektir. Bu sistem doğrudan ürüne entegre edilebileceği gibi, ürün ile haberleşen uzaktan kontrol sağlayan ürün uygulamasının bir özelliği ya da tamamen bağımsız bir uygulama da olabilir. Ancak bu çalışmayla birlikte uygulama yönteminden daha çok kamera ile tekstil parçalarından çok yakından alınacak fotoğraflar ile günlük hayatımızda çok sık kullandığımız tekstil tiplerini sınıflandırma problemi üzerinde durulmuştur.

Çalışma, derin öğrenme tekniklerinden evrimsel sinir ağları (*CNN-Convolutional Neural Networks*) yapılarıyla gerçekleştirilmiştir. Sınırlı sayıda bulunan benzer bir çalışma için daha önceden hazırlanmış veri seti kullanılmıştır. Bu veri seti öncelikle literatüre girmiş ve çok büyük veri setleriyle eğretilmiş VGG16, VGG19, InceptionResNet, MobileNet, ResNet algoritmaları ile sık kullanılan pamuklu, polyester, kot, yünlü gibi tekstil tiplerini yüksek doğruluk oranlarıyla sınıflandırabildiği görülmüştür.

Bu amaçla, VGG16, VGG19, InceptionResNetV2, MobileNetV2, ResNet50 ve InceptionV3 CNN mimarileri kullanılarak detaylı sonuçlar elde edilmiş ve birbirileriyle karşılaştırılmıştır. ResNet50 ve InceptionResNetV2 ile istenilen sonuçlar elde edilememiş ancak diğer 4 mimari ile test veri kümesi ile yüksek doğruluk oranları elde edilebilmiştir.

VGG16 ve VGG19 için sırasıyla %94,21 ve %94,19 doğruluk oranları elde edilmiştir. InceptionV3 ile %95,16 doğruluk oranı elde edilirken MobileNetV2 ile ise %93,51 doğruluk oranı sağlanmıştır ve tüm çalışmalar için sınıf bazlı doğruluk oranlarını net görebilmek için hata matrisleri detaylı olarak değerlendirilmiştir.

Literatüre girmiş mevcut CNN mimarilerinin yanı sıra probleme özgü bir CNN mimarisi tasarlamak için çalışma yapılmıştır ve test veri kümesi ile elde edilen doğruluk oranı %84,30'tur. ImageNet ile eğitilmiş olan CNN mimarileri ile kıyaslandığı zaman daha düşük başarı oranı olan bu mimarinin, VGG16 ile birlikte kullanıldığı durumda genel performansı, yaklaşık %1 seviyelerinde arttırdığı görülmüştür. Toplam doğruluk oranı %95,51 olarak elde edilmiştir.

Önerilen CNN mimarisi ve VGG16 mimarisinin birlikte kullanılmasıyla elde edilen modeller ve ağırlıkları kaydedilerek taşınabilir ve daha düşük işlem gücü olan bir modül üzerine alınmıştır. Böylece, elde edilen çalışma ucuz ve hızlı bir şekilde prototiplenerek standart bir kamera aracılığıyla çalışmada kullanılan 9 farklı tekstil tipi sınıfı için karar verebilen bir uygulama geliştirilmiştir. Bu uygulama için standart bir Web Cam ile birlikte Raspberry Pi 3 modülü kullanılmıştır.

Bu çalışma için yüksek işlem gücü gerekliliğinden dolayı Google Colab'ın sunduğu ücretsiz GPU hizmeti kullanılmıştır. Geliştirmeler ve çalışmalar Python programlama dili ile yapılmış olup, derin öğrenme katmanları ve işlemleri için TensorFlow alt yapısını kullanan Keras kütüphanesi kullanılmıştır.

Elde edilen bu veriler ile birlikte, hem alışmanın yapıldığı CNN mimarileri sonuçları hem de literatürde yer alan benzer alışma sonuçlarına göre karşılaştırma yapılarak tekstilden standart bir kamera ile alınacak görüntü ile tekstil tipi tanıma ve sınıflandırmasının yapılabileceğı gösterilmiştir. Aynı zamanda özel bir ekipman ya da uygulama gerektirmemesi nedeniyle ucuz ve hızlıca devreye alınabilir, ürün veya özel alışmalara entegre edilebilir bir sistemin oluşturulabileceğı gösterilmiştir.





FABRIC CLASSIFICATION BY USING DEEP LEARNING

SUMMARY

Software based applications are not seen in only computer based and embedded systems, but also in many electronic devices that are used in our daily lives owing to advancement of hardware technology and high performance devices. Along with the fact that almost every newly developed product or equipment is connected to the internet, it is possible to collect huge amount of data from every area. Because of high performance systems, collecting data and many devices that can be connected to Internet, algorithms and applications that are developed for machine learning and artificial intelligence can be applied to the devices that are in use in our daily lives such as white goods, household appliances, televisions. In this way, it is possible to increase personalized user experience and to develop algorithms that ensure that the system operates at optimum performance.

With the development of autonomous structures that are capable to make accurate decisions and to perform these decisions, material recognition has become an important problem for these systems. There are many studies that have been carried out different methods on classification of materials. These studies provide progress for applications such as robotic applications and special production process. In addition to basic classification such as glass, metal, plastic, wool, number of studies that are focusing on the lower level of specific classification is increasing day by day.

Information of type of textile is very important for textile contacting products such as tumble dryers, washing machines, irons in order to perform its task with optimum performance without any damage on textile product. In traditional applications, system performance is tried to provided by user programme selection or adjusting the features such as temperature, time or any other product-specific settings according to textile type. However, wrong or inattentive selections or settings may lead to undesirable consequences for valuable textiles.

Many studies have been carried out in the field of textile such as fiber structure, texture or recognition of defects on fabrics in industrial applications. However, there are limited number of studies on recognition of textile types.

In this thesis, the aim is to solve textile type recognition problem with a simple camera that can be easily accessed by anyone without using complex structures and sensor based specific systems. In this way, designed model may lead cheaper and more effective solution that can provide different applications for products. This system can be directly integrated into the product, it can be a feature of the product application that provides remote operation or it can be completely independent 3rd party application. However, this is focused more on the problem of classification textile types that we use very frequently in our daily lives by taking photo very closely from textile parts.

The study was implemented with the structure of convolutional neural networks. Data set that is prepared for one study of limited number of studies on textile classification is used. Firstly, previously learned with huge amount data set algorithms such as VGG16, VGG19, InceptionResNet, MobileNet and ResNet was implemented with data set of this study. According to high accuracy rate of these methods, textiles can be classified as frequently used textile materials such as cotton, polyester, denim, wool etc.

For this purpose, VGG16, VGG19, InceptionResNetV2, MobileNetV2, ResNet50 and InceptionV3 CNN architectures were implemented and detailed results of these studies were obtained. Results of these architectures were compared with each other and results of InceptionResNetV2 and ResNet50 are not sufficient. However, remaining 4 architectures has the results with high accuracies on test data sets.

VGG16 and VGG19 which are very similar, have the results as 94.21% and 94.19% accuracies on test data sets. The results are also similar. However, implementation speed of VGG16 is much more than VGG19.

InceptionV3 has the result as 95.16%. This is the highest result for architectures that were developed for ImageNet and used in this study.

MobileNetV2 has the result as 93.51% and the architecture has the lowest parameter number than all architectures in this study.

In order to clearly see the results, class-based accuracy rates and confusion matrices were generated for all models. Also, comparison tables were shared for accuracies, parameter numbers, implementation time etc.

In addition to the existing CNN architectures in the literature, a study has been carried out to design a problem specific CNN architecture. The proposed architecture has basically 4 convolutional layers and 3 fully connected layers. For convolutional layers, “relu” activation function is used and for last fully connected layer “softmax” is used as activation function. Despite of small architecture, parameter number is around 1 million. Obtained result from test data for this proposed model is 84.30%. Compared to ImageNet-trained CNN architectures, it has a lower success rate and it is not suitable to use it instead of other architectures. After investigating of class based accuracies and general model, it has higher accuracies than VGG16 for some classes. With this information, accuracy of ensemble usage of proposed architecture and VGG16 is around 1% more than accuracy of VGG16. Total accuracy rate is 95.51% for ensemble usage of proposed model and VGG16.

Decision method of ensemble method is that if the classification results of both models are not same, a new probability table is generated for all classes from probabilities of models which is max. According to new probability table, classification result is obtained.

Model informations and model weights for all layers were obtained from the ensemble usage of proposed CNN architecture and VGG16 had recorded as “.json” and “.h5” file. In order to implement this model on portable device which has lower processing power, these recorded files were used. Thus, the study was prototyped cheaply and quickly on a portable device with standard camera unit. A small application software was implemented and the application is able to recognize and classify the fabrics according to 9 different fabric types by using standard camera unit. 9 different fabric types that is used in this study as cotton, denim, fleece, nylon, polyester, silk, terrycloth, wiscose and wool. A Raspberry Pi 3 module and a web cam were used for this application.

Due to high processing power requirement for deep learning algorithms and this study, Google Colab with GPU which is freely offered by Google, is used. Developments and studies are made with Python programming language. In order to develop deep learning algorithms and computational requirements for these architectures, Keras library is used. Keras is a open source library that was developed for Python and it is available to use such kind of low level libraries for deep learning Theano, TensorFlow etc. In this study, TensorFlow is used as backend for Keras library.

Finally, according to obtained results from both proposed CNN architecture and architectures that are implemented for ImageNet, it was shown that fabric recognition and classification can be made by using standard cameras that is take picture from textile units. At the same time, it has been shown that a system or model can be implemented very quickly with low cost and it can be integrated into products or special studies because of not requiring special equipment or application.

1. GİRİŞ

Günümüzde gelişen donanım teknolojisi ile birlikte yazılım tabanlı uygulamalar, profesyonel kullanımlar dışında günlük hayatta da kullanılan birçok cihaz ve ekipmanda yoğun bir şekilde yer edinmiştir. Bununla birlikte, donanım teknolojisindeki ilerleme aynı zamanda yapay zeka alanında da ciddi gelişmelerin önünü açmıştır. Bu gelişmeler, mevcut sistemlerin klasik yöntemleri bir kenara bırakarak değişen ve gelişen bu alanda bir dönüşüm geçirmelerini kaçınılmaz hale getirmiştir.

Yapay zeka, klasik kontrol teorisini, bulanık mantığı, sinir ağlarını da kapsayacak şekilde sürekli gelişme halinde olan bir konudur. Ancak donanım tasarımındaki yenilikler, çok çekirdekli işlemcili grafik kartları ve GPU (Graphic Processor Unit)'lardaki gelişmelerle birlikte, yapay zeka algoritmaları çok ciddi seviyelerde gelişim göstermiş ve derin öğrenme gibi yeni algoritmaların geliştirilmesinin önü açılmıştır [1]. Bu algoritmaların gelişmesiyle birlikte de otonom sistemler üzerine yapılan çalışmalar hız kazanarak kısa sürede birçok konuda çalışmalar ve ilerlemeler kaydedilmiştir. Daha güvenli, güvenilir, en uygun performansı verimli bir şekilde sağlayabileceği düşünülen otonom sistemler, birçok alan için araştırma ve çalışma konusu olmaktadır. Üretim teknolojilerinden, uzay teknolojisine, yer altı veya su altı araçlardan günümüz araba teknolojisine kadar bir çok alanda otonom sistemler üzerinde çalışmalar devam etmektedir [2]. Otonom sistemler, günümüzde özellikle otomobil üretici firmaların otonom araç teknolojilerindeki ilerlemesiyle birlikte artık herkes tarafından daha da bilinir bir konu haline gelmiştir.

Otonom sistemlerin ve bu konuda yapılan çalışmaların artmasıyla birlikte bu sistemlere girdi sağlayacak birçok alt başlıkta da çalışmalar yapılmaktadır. Malzeme tanıma ile ilgili yapılan çalışmalar da bu grupta yer almaktadır. Örnek olarak bir robot uygulamasında farklı malzeme taşıma işlemi yaptırılacaksa eğer tuttuğu malzemenin sınıflandırmasının doğru bir şekilde yapılabilmesi önemli bir konudur. Plastik, cam ya da daha hassas malzemeden yapılan bir cismi taşımak için tutma sırasında uygulanması gereken kuvvet farklılık gösterecektir. Aynı şekilde farklı zeminlerde hareket etmesi gereken bir uygulamada da zeminin yapısının algılanması önemli bir konu olacaktır [3].

Otonom sistemlerin yaygınlaşmasıyla birlikte şu anda kullanmış olduğumuz bir çok ev aleti de bu değişime ayak uyduracak ve uygulama özelinde bir alt çalışma ile desteklenmesi gerekecektir. Tekstil tipinin algılanması da tekstil ile temas eden ev aletleri ya da bu konuda yapılacak farklı herhangi bir çalışma için önemli bir konudur. Örnek olarak çamaşır kurutma makinesi, çamaşır yıkama makinesi, ütü gibi tekstile temas eden ev aletleri için en uygun çalışma koşulları tekstil tiplerine göre özel olarak ayarlanmaktadır. Geleneksel uygulamalarda kullanıcının tekstil tiplerine göre en uygun programı seçmesi ve ona göre önceden programa özel belirlenmiş parametreler ile çevrimin veya operasyonun tamamlanması beklenir. Bu sistemlerin otonom hale getirilmesiyle birlikte kullanıcı seçiminin yerini uygun tekstil tipini algılayıp ona göre en uygun program parametrelerine göre çalışan yazılımların sistemlere entegre olması gerekecektir. Bu amaçla, tekstil tipinin algılanması ile ilgili sensörler, kamera gibi farklı ekipmanlar aracılığıyla farklı yöntemlerle algılama çalışmalarının sayısı artmaya başlamıştır.

1.1 Tezin Amacı

Bu tez çalışması ile birlikte geliştirilen sistemin öncelikli amacı giysilerden yakından alınacak fotoğraflar ile giyim sektöründe daha sık kullanılan ve tercih edilen tekstil tiplerinin sınıflandırılmasının yapılması ve tanınmasıdır.

Bu sistemde aynı zamanda herhangi bir sensör tabanlı karmaşık bir sistem ya da belirli amaçlarla üretilmiş özel kameralar kullanılmayacaktır. Bu yaklaşımla, herkesin kolaylıkla ulaşabileceği cep telefonu kamerası, genel amaçlı fotoğraf makineleri, kamera modülleri gibi standart kameralar ile de sistemin modellenebilir ve doğru sonuçlar verebilir olması amaçlanmıştır.

Bu çalışmayla birlikte temel çalışma prensibi olarak giysilere temas eden çamaşır kurutma makinesi, çamaşır yıkama makinesi, ütü gibi ev aletlerinin sistemlerine ya da genel uygulamalarına hızlı ve kolayca entegre olabilecek bir yöntemin oluşması sağlanabilir. Aynı zamanda elde edilecek model, tekstil tipinin hızlı ve kolay bir şekilde tanımlanmasını gerektirecek bir uygulamada kullanılmak üzere taşınabilir, düşük maliyetli ve sınıflandırma performansı yüksek bir ürün olarak değerlendirilebilir.

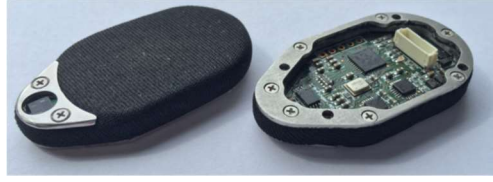
Tekstil üzerinde yapılan benzer çalışmalarda dokuma hatalarının yakalanması, dokuma tipi ve lif yapısını tanıma konularının daha çok ele alındığı görülmektedir. Bu çalışma, az sayıda çalışılmış olan tekstil tipi tanıma konusuna eğitim ve sınıflandırmada derin öğrenme yöntemleri kullanılarak elde edilen sonuçlar ile literatüre katkı sağlamaktadır.

1.2 Literatür Özeti

Tekstil tipinin sınıflandırılması ve tanınmasıyla ilgili bu zamana kadar yapılan çalışmalar incelendiğinde farklı yöntemler ve yaklaşımlar kullanıldığı görülmektedir. Ön plana çıkan dokunsal sensörler ve kamera ile yapılan çalışmalar yöntem, yaklaşım ve sonuçlar açısından irdelenerek ele alınmıştır.

Tekstil tipi sınıflandırma ve tanıma ile ilgili çalışmalara ek olarak 2 tane de tekstil üzerindeki hataları sınıflandırabilen ve tanıyabilen çalışmaya da literatür özeti kısmında yer verilmiştir. Özellikle üretim süreçleri ve endüstriyel uygulamalarda daha çok yer bulan tekstil üzerindeki hataların tanınması ile ilgili çalışmaların sayısı oldukça fazladır. Görüntü bazlı olarak tekstil üzerinde bu hataların tanınması için derin öğrenme yöntemlerinin kullanılması ve bu yaklaşımla çalışmaların yapılması tekstil ile ilgili yapılacak çalışmalar için kaynak oluşturmaktadır.

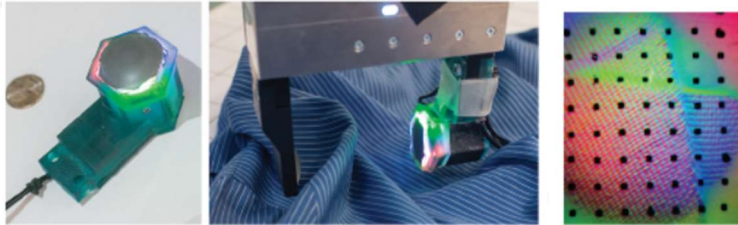
Khan ve arkadaşları, dokunsal bir sensör kullanarak 17 farklı tekstil tipi için öğrenme ve sınıflandırma konusunda bir çalışma yapmışlardır [4]. Çalışmalarında dokunsal sensör olarak Denei ve arkadaşlarının endüstriyel uygulamalarda kullanılmak üzere tekstil manipülasyonu ve sınıflandırılması için geliştirmiş oldukları sensörü referans olarak kullanmışlardır [5]. Şekil 1.1’de gösterilen bu sensör temel olarak basınç, akustik ve yakınlık sensörü olmak üzere 3 farklı temel algılama modeli içermektedir.



Şekil 1.1 : Dokunsal Sensörün Alt ve Üst Yüzeyleri [5].

Bu çalışmada, sensör yardımıyla farklı basınç değerleri ve iki farklı yönde sensörün tekstil üzerindeki hareketiyle elde edilen veri kümeleri destek vektör makineleri yaklaşımıyla eğitilerek sonuçlar elde edilmiştir. Problem ikili gruplara indirilerek bire bir SVM (Support Vector Machine) yaklaşımı ile sonuçlar elde edilmiştir. 2 farklı hareket yönü için %98 ve %99 olacak şekilde sınıflandırma doğruluk oranları elde edilmiştir [4].

Bir diğer çalışmada ise Yuan ve arkadaşları, tekstil tipini tanımak için sensör üreticisi bir firma tarafından tasarlanıp satışa sunulan yüksek çözünürlükle temas ettiği yüzeyin geometrisini ve dokuma şeklini algılayabilen Şekil 1.2’de çıktısıyla birlikte gösterilen bir sensör kullanılmıştır. Ayrıca aynı çalışma içerisinde derin görüntü alma imkanı tanıyan Şekil 1.3’te gösterilen Kinect sensör de kullanılarak sınıflandırma yapılmıştır [6].



Şekil 1.2 : Sensör, uygulama ve algılanan yüzeye ait sensor çıktısı [6].



Şekil 1.3 : Orijinal tekstil görüntüsü ve Kinect görüntüsü [6].

Bu çalışmada elde edilen veri kümeleriyle sadece tekstil tipi değil aynı zamanda kalınlık, yumuşaklık, esneklik, dayanıklılık, yıkama metodu gibi toplamda 11 farklı özellik için CNN (Convolutional Neural Networks) kullanılarak sınıflandırma

yapılmıştır. Gelsight sensörden alınan veriler için VGG19, Kinect sensörden alınan veriler için ise VGG16 kullanılmıştır. Ele alınan 11 farklı özellikten tekstil tipine ait doğruluk oranları ise modelin eğitilmesinde kullanılan görüntüler için %85 seviyesinde iken modelin görmediği görüntüler için %44'tür [6].

Wang ve arkadaşlarının yaptığı çalışmada da Gelsight sensör kullanılarak tekstil tipleri sınıflandırılmıştır. Tekstilin düz, katlanmış ve bu iki durumun birlikte ele alınarak sonuçların elde edildiği bu çalışmada CNN kullanılmıştır ve kayıp fonksiyonunda λ hata parametresi yer almaktadır. Bu parametrenin değerlerine göre sonuçlar elde edilmiş ve en yüksek doğruluk oranı $\lambda = 0.1$ için üç koşulun ortalaması olacak şekilde %78.48 olarak belirtilmiştir [7].

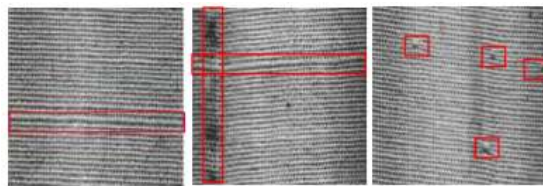
Zhaou ve Zhou'nun 2018 yılında yaptığı çalışma ise tekstil ürününü tamamen kapsayacak şekilde alınan görüntüler üzerinden yapılan kırışıklık analizine dayanmaktadır [8]. Şekil 1.4'te çalışmada kullanılan ikili düzene çevrilmiş bir örnek görüntü gösterilmiştir.



Şekil 1.4 : Tekstil ürünü ve ikili düzen görüntüsü [8].

Yapılan bu çalışmada herhangi bir yapay zeka algoritması kullanılmadan daha çok kırışıklık analizi ve oranlar üzerinden tekstil tipi tanımaya gidebileceği gösterilmiştir. Az sayıda görüntü kullanılarak kırışıklıkların boy ve en, alan ve tekstilin başlangıç noktası, alan ve yükseklik oranlarına bakarak pamuk, rami ve ipek olacak şekilde sınıflandırma yapılmıştır.

Ouyang ve arkadaşlarının kumaş üzerindeki hataları algıladıkları çalışmada istatistiksel hata analizi ve CNN mimarisi birlikte kullanılarak kumaş üzerindeki Şekil 1.5'te gösterilen hatalar %95 doğruluk oranıyla yakalanabilmektedir.



Şekil 1.5 : Yatay, dikey ve noktasal olacak şekilde dokuma hataları [9].

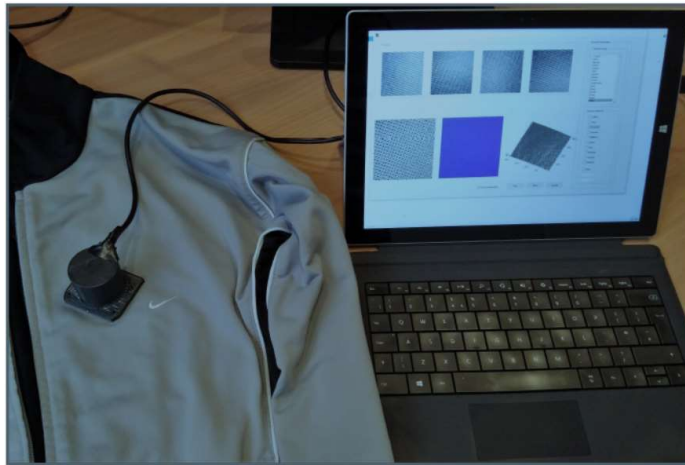
Kumaş üzerindeki hataların algılanması ile ilgili yapılan bir diğer çalışmada ise CNN AlexNet mimarisi kullanarak %75 doğruluk oranıyla hatalar tespit edilmiştir. Ancak önceden eğitilmiş AlexNet mimarisi kullanılarak sadece FC (Fully Connected) ağırlıklarının yeniden eğitilmesiyle bu oran %98'lere çıkartılabilmektedir [10].

Kampouris ve çalışma arkadaşlarının yaptığı çalışmada ise standart bir kamera aracılığıyla tekstil ürünlerinden yakından alınan fotoğraflar ve aynı zamanda onların 3 boyutlu modellemesiyle hem tekstil tipi sınıflandırma hem de giyim türü sınıflandırması yapılmıştır [3]. Şekil 1.6'da elde edilen veri kümesine ait örnekler gösterilmektedir.



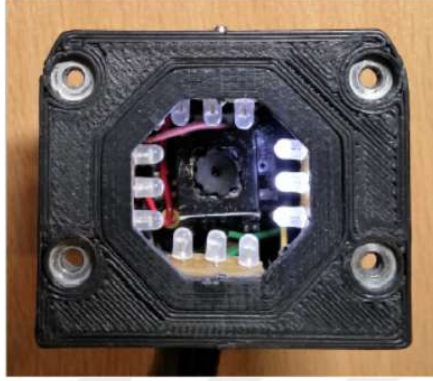
Şekil 1.6 : Veri kümesi örnek görüntüleri [3].

Çalışma için kamera ile geliştirdikleri sistem ile 26 farklı tekstil tipi için veri toplanmış ancak çalışma en çok kullanılan ve en çok veri toplanmış olan 9 farklı tekstil tipi için yapılmıştır. Veri toplama sistemine ait düzenek Şekil 1.7'de gösterilmektedir.



Şekil 1.7 : Tekstilden görüntü alma sistemi [3].

Kullandıkları sistemde kamera için 4 kenardan aydınlatma imkanıyla beraber her numuneden 4 farklı fotoğraf alınmıştır ve alınan bu fotoğrafları daha sonra 3 boyutlu modelleme sırasında da kullanmışlardır. Kullanılan kamera çözünürlüğü 640x480'dir. Kullanılan kamera ve aydınlatma düzeneği Şekil 1.8'de gösterilmektedir.



Şekil 1.8 : Kamera ve aydınlatma düzeneği [3].

Tekstil tiplerinden toplanan görüntüler ve bu görüntülerden elde edilen farklı formasyonlarla hem Scale Invariant Feature Transform (SIFT) hem de CNN yöntemleri kullanılarak sonuçlar elde edilmiştir. Ancak bu çalışmaya referans olacak şekilde doğrudan tekstil tipinden alınan görüntüler kullanıldığında CNN VGG mimarisi FC (Fully Connected) + FV (Fisher Vector) kullanımı ile birlikte elde edilen doğruluk oranı %71.7'dir.

1.3 Hipotez

Bu çalışmayla tekstil ürünlerinden yakından alınacak fotoğraflar ile tekstil tipi sınıflandırması yapabilen bir model geliştirilecektir. Kullanılacak görüntüler için Kampouris ve çalışma arkadaşlarının yapmış yaptıkları çalışmada oluşturdukları görüntü veritabanı kullanılacaktır [3].

Model görüntü bazlı olarak ele alınacağı için derin öğrenme yöntemlerinde CNN ile geliştirme yapılacaktır. Önerilen modelle birlikte eldeki veri seti kullanılarak en yüksek doğruluk oranına ulaşılması hedeflenmiştir. Bununla beraber literatüre girmiş ve kendini ispatlamış, çok büyük veri setleriyle eğitilmiş VGG16, VGG19, InceptionResNet gibi algoritmalar da çalışma kapsamında ele alınacak ve detaylı olarak incelenecektir. Yüksek doğruluk oranına ulaşabilmek için literatürde yer alan bu algoritmalar ve önerilen model, bu çalışmada kullanılacak veri seti ile eğitilerek çıktılar değerlendirilecek ve en uygun modelin gerçekleştirilmesi sağlanacaktır.

İstenilen doğruluk oranının yakalandığı modelin gerçekleşmesi sırasında izlenen süreç, yapılan çalışmalar, değerlendirmeler ve bu doğruluk oranının elde edilmesini sağlayan etkenler ve yaklaşımlar ayrıntılı olarak ele alınacaktır.

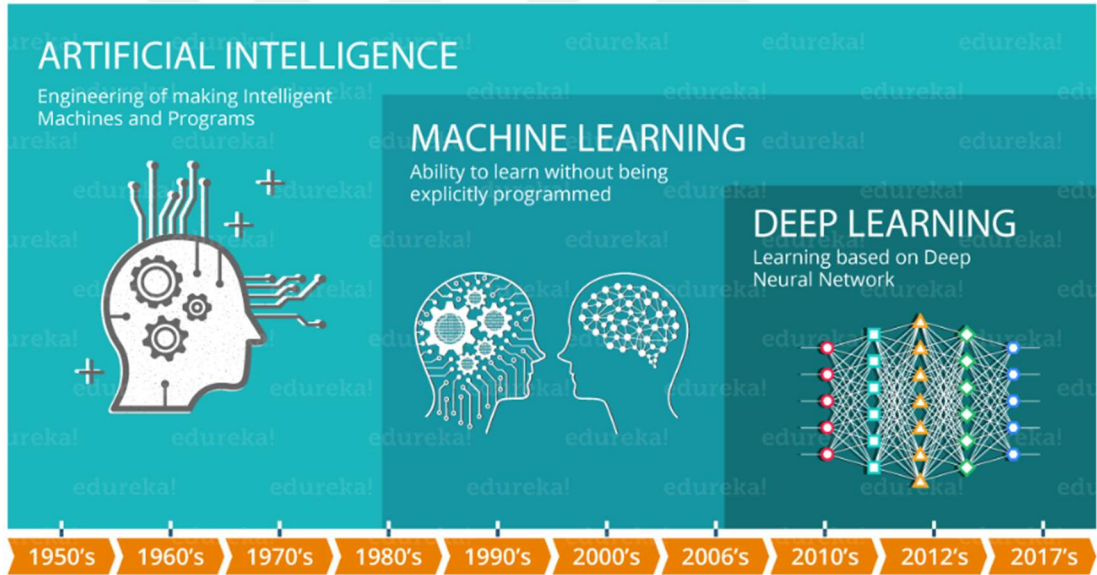
En uygun modelin geliştirilmesiyle, taşınabilir, kolay ulaşılabilir ve kullanılabilir bir sistem tasarlanacaktır. Elde edilen model PMML (Predictive Model Markup Language) dosyasına çevrilerek hızlı ve pratik bir şekilde modelleme yapılacaktır. Bu amaçla Raspberry Pi 3 ve Logitech C170 Web Cam kullanılacaktır. Kullanılacak olan modüller Şekil 1.9’da gösterilmektedir.



Şekil 1.9 : Raspberry Pi 3 ve Logitech C170 Web-Cam

2. DERİN ÖĞRENME VE CNN MİMARİLERİ

Derin öğrenme konusunun gelişme sürecini anlayabilmek ve bu konuda yapılan çalışmaları anlamlandırabilmek için öncelikle yapay zeka ve makine öğrenmesi konusundaki süreci incelemek gerekmektedir. Bu üç konunun kronolojik olarak gelişme süreci ayrıntılı olarak Şekil 2.1’de gösterilmektedir. Temelde yapay zeka konusundaki çalışmaların artmasıyla birlikte verilerden öğrenen ve tahminde bulunabilen makine öğrenmesi yöntemlerinin geliştiği görülmektedir. Bu gelişmelerin ardından daha büyük boyutlu veriler ile öğrenme yapabilen derin öğrenme gelmektedir [11].



Şekil 2.1 : Yapay zeka, makine öğrenmesi ve derin öğrenme [11].

Yapay zeka konusundaki çalışmalar çok daha önceden başlamış olsa da terim olarak literatüre girmesi ve “Yapay Zeka” teriminin oluşması 1956 yılında John McCarthy ve arkadaşları tarafından sunulmasıyla birlikte olmuştur. Yapay zeka konusunun temelinde yatan düşünce kendi kararlarını kendi alabilen, insan gibi hatta daha da ötesinde akıllı makinelerin tasarlanmasıdır.

Makine öğrenmesi, yapay zeka çalışmalarının bir uygulaması olarak verilerden öğrenen ve önceki deneyimlere dayalı karar verebilen yaklaşımdır. Yapay zeka konusunda yapılan ilk çalışmalarda sistem, uyması gereken bir çok kurala göre daha önceden belirlenmiş eylemler listesine göre hareket edip kararlar alabilmektedir. Ancak makine öğrenmesi ile birlikte önceden belirlenmiş kurallar ve kararlar mekanizmasından öte büyük veri kümeleri kullanılarak öğrenme yapılabilmektedir ve bu öğrenme sonucunda kararlar verilebilmektedir [12].

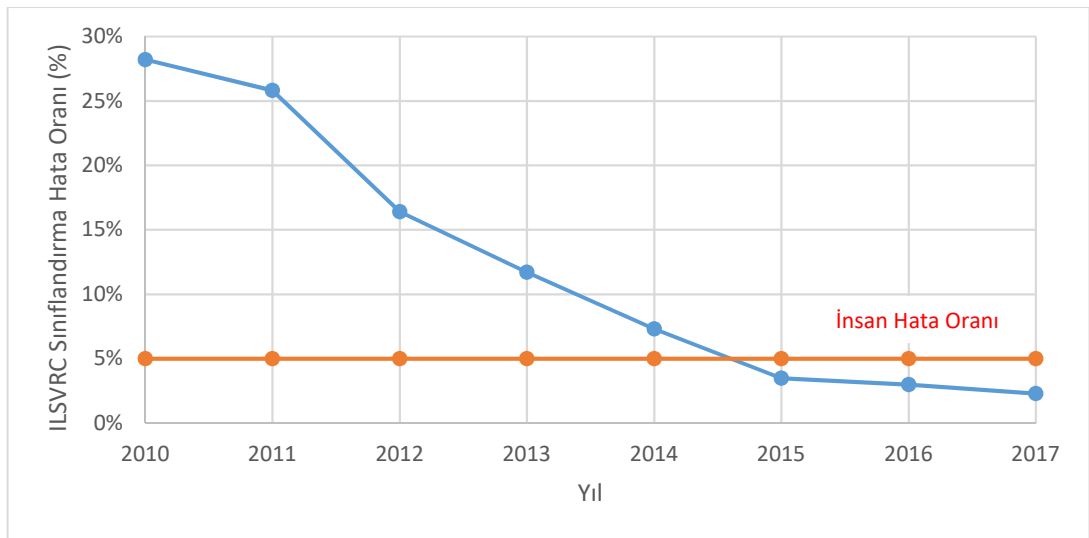
Derin öğrenme isminin literatüre girmesi ve bu haliyle çalışmaları Şekil 2.1’de gösterildiği gibi 2006 sonrasında olsa da bu konuda yapılan çalışmalar çok daha eskiye dayanmaktadır. Derin öğrenme tarihçesi incelenmek istendiğinde 1940 – 1960, 1980 – 1990 ve 2006 sonrası olmak üzere üç farklı dönemde yaşanan gelişmeleri üç farklı dalga olarak yorumlamak ve değerlendirmek gerekir [13].

1940 – 1960 ve 1980 – 1990 yılları arasındaki yapay sinir ağları konusunda ciddi araştırmalar yapılmıştır. İlk gelişim dalgası biyolojik öğrenme, tek bir nöronun eğitilmesi ve algılayıcı yapıları üzerindeki çalışmalarla birlikte başlamıştır. İkinci dalga ise bir ya da daha fazla gizli katmanlı sinir ağlarının eğitilmesi, geri yayılım gibi konularındaki gelişmelerle birlikte başlamıştır. 2006 sonrası gelişen son dalga ise çoklu katmanların öğrenilmesi ile ilgili yapılan çalışmalarla birlikte başlamıştır ve derin öğrenme kavramı oluşmuştur.

Derin öğrenme konusunun geçmişten günümüze olan sürecini çok kısa bir şekilde özetlemek gerekirse; derin öğrenmenin uzun ve zengin bir tarihinin mevcut olduğu, fakat farklı yaklaşımları yansıtan bir çok isim tarafından üzerinde çalışıldığı ve zaman zaman popülerliği artarken, bazı dönemlerde de azaldığı görülmüştür. Sistemlerin eğitimi için kullanabilecek veri kümelerinin artmasıyla birlikte derin öğrenme yaklaşımları daha kullanışlı hale gelerek popülerliği artmıştır. Aynı zamanda hem yazılım hem de donanım olarak bilgisayar teknoloji alt yapısının gelişmesiyle birlikte derin öğrenme modellerinde daha da fazla ilerleme sağlanabilmektedir. Veri kümelerinin artması, öğretilmek istenilen sistemlerin büyümesi gibi konular giderek problemlerin mevcut uygulamalar ile çözülmesini güç hale getirmektedir. Bu nedenle günümüz karmaşık uygulamaları, yapılan çalışmalarla beraber zamanla artan doğruluk oranlarıyla derin öğrenme yöntemleri sayesinde çözüme kavuşmaktadır [13].

Derin öğrenme beraberinde yapay zeka tarihi ve çalışmaları çok eskilere dayanmasına rağmen son yıllarda bu konuda yapılan çalışmalar ve uygulamaların sayısı her geçen gün artmaktadır. Özellikle 2006 sonrası yapılan çalışmalar ele alınacak olursa, ilk olarak ImageNet veritabanının oluşturulmasına bakılabilir. Stanford Üniversitesi ve Princeton Üniversitesi ekipleriyle kurulan ImageNet, arama motorları mantığıyla internette bulunan görüntüleri arayıp onları sınıflandıran ve tüm dünyada akademik çalışmalarda kullanılmak üzere bunları bir araya getiren bir veritabanıdır. Yaklaşık 15 milyon resmin yer aldığı bu veritabanı, görüntü işleme ve derin öğrenme gibi araştırma çalışmalarında kullanılmak üzere iyi araştırma iyi kaynakla olur mantığıyla oluşturulmuştur [14].

ImageNet'in 2010 yılından 2017 yılına kadar düzenlemiş oldukları yarışmalarda bir çok farklı ekip ImageNet veritabanını kullanarak görsel tanıma yapacak hem yazılım hem de donanım konusunda çalışmalar yapmıştır. Bu yarışmalarda veritabanında yer alan 1000 farklı görüntü sınıfı için sınıflandırmalar yapılmış ve sınıflandırma doğruluk oranlarını en üst seviyeye çıkarmak üzere günümüzde literatüre de girmiş çalışmalar ve mimariler oluşmuştur. Bu çalışmalarla ortaya çıkan AlexNet, GoogleNet, VGGNet, ResNet ve bunlar gibi bir çok mimari günümüz problemlerinin çözümü için de sıklıkla kullanılmaktadır. ILSVRC (ImageNet Large Scale Visual Recognition Competition) olarak adlandırılan bu yarışmalarda 2010 ve 2017 yılları arasında yapılan çalışmalarda, sınıflandırma hata oranının yıllara göre değişimi Şekil 2.2'de gösterilmiştir. İnsanların bu konudaki hata oranının yaklaşık %5 olduğu göz önüne alındığında çalışma sonuçlarının her geçen yıl çok daha iyi seviyelere geldiği görülmektedir.



Şekil 2.2 : Yıllara göre ILSVRC sınıflandırma hata oranları.

Sürücüsüz araç, yüz tanıma, el yazısı tanıma, nesne tanıma, Siri, Alexa gibi asistan sistemler, duygu tanıma gibi çalışmalara ek olarak makinelerin ve robotların insanların yapabildiği oyun oynama, yazı yazma, resim yapma, müzik besteleme gibi becerilerini de yapabilmesini sağlayacak çalışmalar 2006 sonrasındaki gelişmelerle birlikte yüksek bir ivme ile artmaktadır.

Yapılan çalışmaların içeriği ve niteliğinin artmasıyla birlikte problemlerin parametre sayısı da her geçen gün artmaktadır. Bununla beraber bu modellerin eğitilebileceği sistemlerin de donanım olarak gelişmesi önemlidir. Modern grafik işlemciler, çok çekirdekli işlemcilerin yetersiz kaldığı noktada çok daha iyi performans göstermektedir ve bu sayede GPU kullanarak 100 milyon parametrelili bir problem çözülebilmektedir [15]. 2009 yılında yapılan bu çalışmayla birlikte GPU'lar, derin öğrenme yöntemleri için önemli bir yer edinmiştir. Donanım performansının öneminin artmasıyla birlikte de derin öğrenme gelişimi sadece model ve algoritma geliştirme olarak değil, aynı zamanda donanım geliştirme tarafında da devam etmektedir. Nvidia, Intel, Amd gibi bir çok entegre üreticisi firma konunun donanım tarafı ile ilgili de ciddi çalışmalarda bulunmaktadır.

Derin öğrenme, Google, Microsoft, Netflix, Apple, IBM, Facebook, NVidia, NEC gibi dünyanın önde gelen bir çok teknoloji şirketi tarafından kullanılmaktadır ve aynı zamanda bu şirketler bu konuda yapmış oldukları çalışmalar ile derin öğrenme konusunun gelişimine katkı sağlamaktadır. Bununla beraber derin öğrenme konusunun gelişimi için Theano, Torch, TensorFlow, PyLearn gibi bir çok yazılım kütüphanesi de geliştirilmiş ve bu konuya katkı sağlamıştır [13].

Bu çalışma içerisinde, derin öğrenme yöntemlerinden özyinelemeli (Recurrent) ve evrimsel (Convolutional) sinir ağları konularına yer verilecektir. Ancak, çalışma içerisinde farklı CNN mimarilerinden faydalandığı için evrimsel sinir ağları konusunda literatüre girmiş mimarilere de ayrıca yer verilecektir. Ayrıca derin öğrenme yöntemlerinin temellerinin yapay sinir ağlarına dayanmasından dolayı aktivasyon fonksiyonlarının bu yöntemlerde önemi büyüktür. Bu nedenler, derin öğrenme mimarilerinden önce yaygın kullanılan aktivasyon fonksiyonlarına da yer verilecektir.

Aktivasyon fonksiyonları, giriş bilgisine göre bir kural ya da eşik değerine göre ilgili nöronların aktif veya deaktif olmasını sağlayan matematiksel kapılardır [16]. Günümüz problemlerinde yaygın olarak softmax, sigmoid, tanh, relu gibi aktivasyon fonksiyonları kullanılmaktadır.

Softmax, olasılık dağılımı olarak çıktı veren bir fonksiyondur ve sıklıkla son tam bağlantı katmanında sınıflandırmanın tamamlandığı katmanda kullanılmaktadır. Bu nedenle çok sınıflı problemlerin çözümünde kullanılmak için oldukça uygundur. Softmax formülü Denklem 2.1’de gösterilmektedir.

$$\sigma(x)_k = \frac{e^{x_k}}{\sum_{k=1}^K e^{x_k}}, k = 1, \dots, K \quad (2.1)$$

Sigmoid aktivasyonu, (0,1) aralığında çalışmaktadır ve matematiksel alt yapısı nedeniyle giriş değerindeki farklılık çıktının 0’a ya da 1’e hızlı yakınsamasını sağlayarak çıkış tahminini net olarak vermektedir. Denklem 2.2’de sigmoid aktivasyon fonksiyonu gösterilmektedir. Hesaplama maliyeti, gradyanların kaybolması modelin kararsızlığa gitme riski gibi dezavantajları bulunmaktadır.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

Sıfır merkezli bir aktivasyon fonksiyonu olup (-1, 1) aralığında çalışır ve bu nedenle olasılık çıktıları daha net olur. Tanh aktivasyon fonksiyonu Denklem 2.3’te gösterilmiştir. Sigmoid ile benzer dezavantajları taşımaktadır.

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.3)$$

Relu aktivasyon fonksiyonu Denklem 2.4’te gösterildiği üzere pozitif değerler için lineerken negatif değerler için sıfır değeri alır. Hesaplama kolaylığı, hızlı yakınsama, tanh ve sigmoid fonksiyonlarında olduğu gibi gradyanların kaybolması problemlerinin olmaması gibi avantajları bulunmaktadır. Genellikle gizli katmanlarda kullanılmaktadırlar.

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.4)$$

Avantajlarının yanı sıra bir nöron beklenmedik şekilde negatif alırsa sonucun sıfır olmasından dolayı nöron geri dönüşü olmayacak şekilde kaybedilecektir. Bu problemin önüne geçmek için Leaky ReLU diye adlandırılan başka bir aktivasyon fonksiyonu da bulunmaktadır. Denklem 2.5'te gösterildiği üzere, negatif değerler için sıfır değeri yerine bir a katsayısı ile çarpılmaktadır. Bu a kat sayısı genelde 1'den düşük 0'a yakın olacak şekilde seçilmektedir.

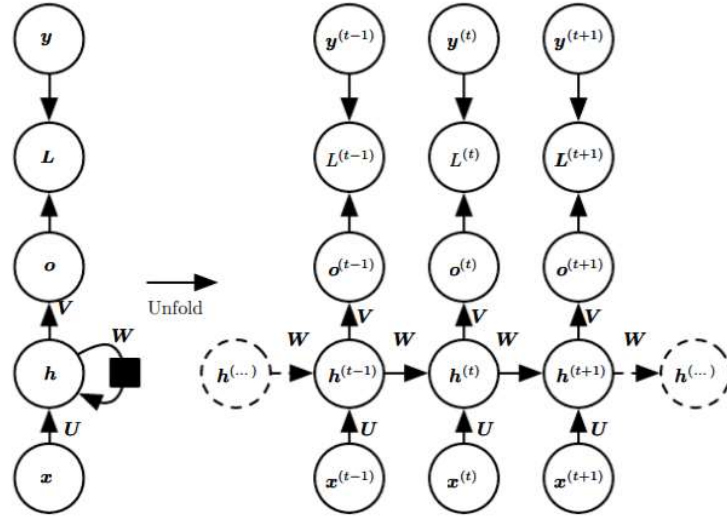
$$ReLU(x) = \begin{cases} a \cdot x, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.5)$$

Bununla birlikte eksponansiyel ve parametrik olmak üzere farklı relu aktivasyonları da bulunmaktadır.

2.1 Özyinelemeli Sinir Ağları

Geleneksel sinir ağları ileri beslemeli yapıdadır ve modelin girdisi bir ağdan geçirilerek bir çıktı elde edilir. Bu yapılarda girdi ve çıktı birbirinden tamamen bağımsızdır. Ancak özyinelemeli sinir ağlarında çıktılar aynı zamanda modelin girdisini oluşturduğu için bir sonraki aşamada elde edilecek çıktı bir öncekine bağlıdır. Önceki çıktıların tekrardan modeli etkilemesi sebebiyle RNN (Recurrent Neural Networks) yapıları aynı zamanda belleğe sahiptir. Bu nedenle yazı, konuşma, zamana bağlı veriler gibi önceki çıktıların da önem arz ettiği problemlerin çözümünde RNN kullanılmaktadır.

RNN, farklı şekilde tasarlanabilir. Önemli RNN tasarımlarına kısaca değinilecek olursa ilk model Şekil 2.3'te gösterilmiştir. Bu ağ yapısına göre, her bir zaman biriminde bir çıktı elde edilmektedir ve gizli katmanlar arasında yinelemeli bağlantı bulunmaktadır.



Şekil 2.3 : Örnek RNN Modeli 1 [13].

Şekil 2.3'te gösterilen RNN modeline ait ileri yayılım eşitlikleri, b ve c sapma vektörü, U,V ve W ağırlık matrisleri olacak şekilde Denklem 2.6, Denklem 2.7, Denklem 2.8 ve Denklem 2.9'da gösterilmiştir.

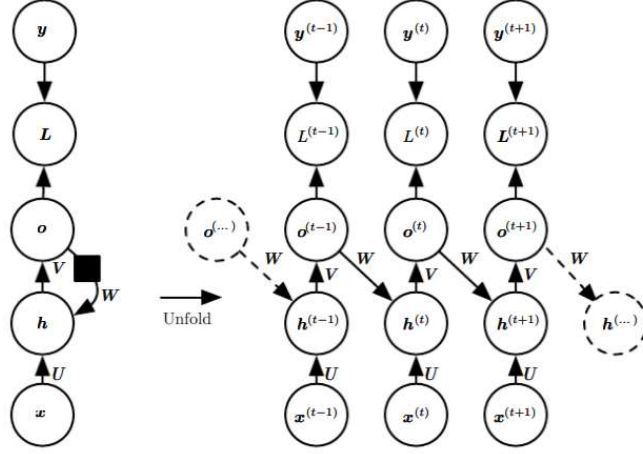
$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \quad (2.6)$$

$$h^{(t)} = \tanh(a^{(t)}) \quad (2.7)$$

$$o^{(t)} = c + Vh^{(t)} \quad (2.8)$$

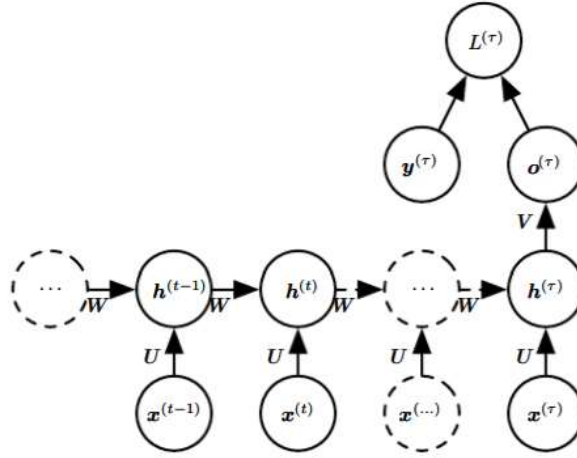
$$\hat{y}^{(t)} = \text{softmax}(o^{(t)}) \quad (2.9)$$

Bir diğer önemli modelde ise yine her bir zaman biriminde bir çıktı elde edilmektedir. Ancak bu modelde yinelemeli bağlantı bir zaman dilimine ait çıktıdan bir sonraki gizli zaman dilimine ait gizli katmana doğrudur. Bu yapıda, doğrudan gizli katmanlar arasında bağlantı olmaması, çıktının çok yüksek boyutlu olmadıkça önemli bilgileri geçmişten taşıyamaması gibi nedenlerden dolayı model başarısı daha düşüktür. Modeli eğitme sırasında zaman dilimlerinin birbirinden bağımsız olması nedeniyle paralel işleme imkan tanımaktadır. Model Şekil 2.4'te gösterilmiştir.



Şekil 2.4 : Örnek RNN Modeli 2 [13].

Bir diğer model ise Şekil 2.5'te gösterilmiştir. Bu modelde gizli katmanlar arasında bağlantı bulunmakla beraber, modelin çıktı üretmesi için tüm serinin okunması gerekmektedir. Tüm seri okunduktan sonra model tek bir çıktı üretmektedir. Seriyi analiz ederek bir sonraki süreç için sabit boyutlu bir çıktı elde edilmesi gereken uygulamalarda bu ağ yapısının kullanılması uygun olacaktır.



Şekil 2.5 : Örnek RNN Modeli 3 [13].

RNN'nin belleğinin olması ve zaman birimleri boyunca önceki verileri de kullanması nedeniyle bir zaman serisine göre çıktı vermesi gereken uygulamalarda kullanılabilmesi ve evrişimsel sinir ağları ile birlikte kullanılabilir olması, bu yaklaşımın avantajlarıdır. Bununla birlikte modelin dezavantajları da bulunmaktadır. Eğer sistemde uzun vadeli bağlılık varsa genellikle gradyanlar kaybolur ancak bazı durumlarda da optimizasyonu bozacak şekilde değerler alarak modelin kararsızlaşmasına neden olabilir ve modelin eğitilmesi mümkün olmayabilir. Bununla birlikte RNN'nin eğitilmesinin zor olması ve aktivasyon fonksiyonu tanh ya da relu olduğu zaman büyük serileri işleyememesi sistemin diğer dezavantajlarından [17].

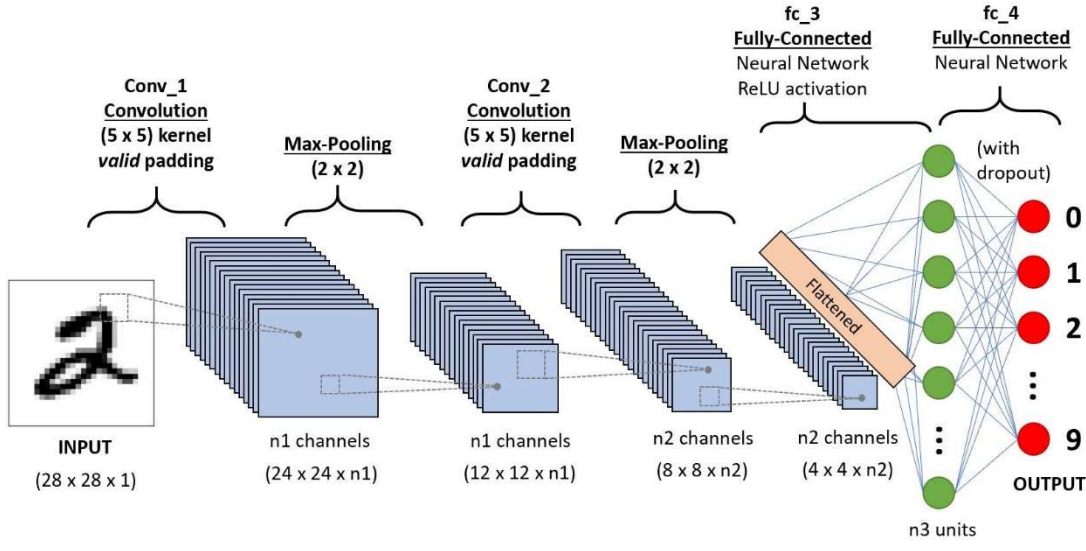
RNN mimarilerinde aktivasyon fonksiyonu olarak daha çok tanh tercih edilir. Bununla beraber tanh aktivasyon fonksiyonunun gerçekleşmesi için yüksek işlem gücünün gerekmesi ve özellikle geri yayılım optimizasyonu sırasında gradyanların kaybolma sorunları RNN mimarisinin dezavantajlarının sebeplerinden biridir [16].

2.2 Evrişimsel Sinir Ağları

Evrişimsel sinir ağları, temel olarak çok sayıda sinir ağları katmanı içererek özellikle görüntü ve video tanıma, sınıflandırma, analiz gibi konularda çözüm sağlamaktadır. Bir CNN (Convolutional Neural Networks) mimarisi, insan beynindeki nöronların bağlantı modeline benzer şekilde görsel korteksin organizasyonundan ilham alınarak tasarlanmıştır [18].

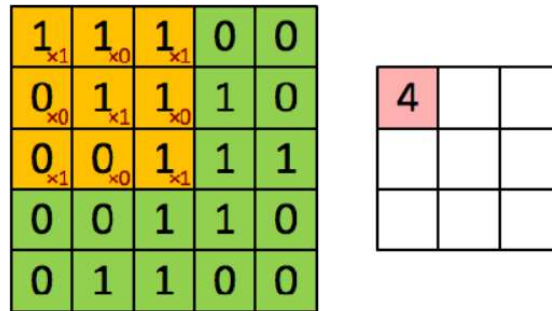
CNN'yi ileri beslemeli sinir ağlarından ayıran en büyük özelliği büyük boyutlu görüntüler için çok sayıda parametreye çözüm üretebilir olmasıdır. Düşük boyutlu bir görüntü için ileri beslemeli sinir ağları çözüm üretebilir durumdayken, günümüz teknolojisi ile elde edilmiş yüksek çözünürlüklü RGB görüntüler için yapılacak çalışmalarda çözüm üretemez durumdadır. Bu nedenle, görüntü ve video üzerine yapılan çalışmalarda uygun CNN mimarileri kullanılarak çözüm üretilebilmektedir [18].

CNN, temel olarak 3 ana katmandan oluşmaktadır. Bunlar, konvolüsyonel katman, pooling katmanı ve tam bağlantı (Fully Connected) katmanı. Temel olarak konvolüsyonel ve pooling katmanları modelin öğrenmesini sağlarken, tam bağlantı katmanı ise sınıflandırmayı sağlamaktadır [19]. Bu katmanlara örnek olması açısından el yazısı rakamları sınıflandırmak için oluşturulmuş bir CNN yapısı örneği Şekil 2.6’da gösterilmektedir.



Şekil 2.6 : CNN katmanları [18].

Konvolüsyonel katman, CNN mimarilerinin ana parçasıdır. Bu katmanda girdilerin özelliklerinin öğrenilmesi sağlanmaktadır. Girdi görüntüdeki yüksek ve düşük seviyeli filtreler uygulanarak özellik haritası oluşturulmuş olur. Genel olarak bu katmanda sigmoid, relu ya da tanh fonksiyonları aktivasyon fonksiyonu olarak kullanılabilir [20]. Konvolüsyonun görüntü üzerine nasıl uygulandığı ve elde edilen özellik haritası Şekil 2.7’de gösterilmiştir.

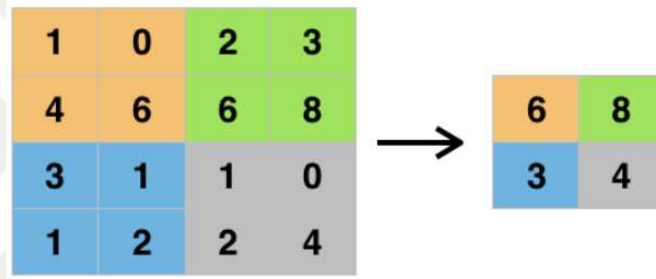


Şekil 2.7 : Görüntü ve konvolüsyon işlemi sonrası özellik haritası [18].

Konvolüsyon işlemine ait matematiksel eşitlik ise Denklem 2.6’da verilmiştir. Denklemde yer alan m ve n kernel mxn boyutlarını belirtirken, i ve j ise konvolüsyonun hesaplanacağı matris koordinatını belirtmektedir.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.10)$$

Pooling katmanı ise genellikle konvolüsyonel katmanlar arasında uygulanır. Temel amacı, özellik haritasının boyutunun düşürülmesiyle modelin işlenmesi için gereken hesaplama gücünü azaltmaktır. Ayrıca değişmez ve baskın olan özellikleri model dışına çıkartarak modelin etkili bir şekilde eğitilmesini sağlamaktadır. Bir çok pooling işlemi olmasına rağmen en yaygın olarak maksimum ve ortalama pooling katmanları kullanılmaktadır. Örnek olarak maksimum pooling işlemi Şekil 2.8’de gösterilmiştir.



Şekil 2.8 : Maksimum pooling işlemi [17].

Tam bağlantı katmanında ise bir önceki katmandan alınan tüm nöronlar bu katmandaki her bir nörona bağlanır. CNN mimarisinin yapısına göre tam bağlantı katmanı bir ya da daha fazla olabilmektedir. Son tam bağlantı katmanından sonra ise çıktı katmanı gelir. Sınıflandırma amaçlı çalışmalarda bu aşamada sıklıkla softmax regresyon kullanılarak çıktı sınıfları için olasılık dağılımları elde edilerek çıktı verilir [19].

CNN mimarilerinde temel katmanlar dışında normalizasyon ve dropout katmanları da bulunmaktadır. Dropout katmanındada yapılan işlem, eğitim sırasında aşırı uyma problemlerinin önüne geçmek için bazı nöron çıkışlarının belirli bir olasılık değerine göre 0 olarak ayarlanmasıdır. Genel olarak tam bağlantı katmanlarından sonra kullanılmaktadır. Konvolüsyonel katmanlardan sonra kullanılması durumunda genellikle test hata oranlarını yükseltmektedir [20].

Normalizasyon katmanı ise giriş özelliklerinin, eğitim sırasında anlık ve beklenmedik değişimlerinin önüne geçmek için kullanılan bir yöntemdir. Hem konvolüsyonel hem de tam bağlantı katmanlarında kullanılabilir ancak, kullanılacağı yer konusunda farklı öneriler bulunmaktadır. Örnek olarak, konvolüsyonel ya da tam bağlantı, normalizasyon, aktivasyon ve dropout olacak şekilde uygulanabilir [20].

CNN mimarilerinde ve genel olarak tüm makine öğrenmesi, derin öğrenme yöntemlerinde doğru sonuca ulaşmak için ilgili fonksiyonunun minimumuna ya da maksimumuna ulaşmak gerekir. Kayıp fonksiyonları da bu nedenle önemli olup kayıp fonksiyonunu minimize etmeye çalışarak kurulan modelin başarısı belirlenebilir. Bu çalışma içerisinde de kullanılmış olan Cross-Entropy kayıp fonksiyonu Denklem 2.11’de gösterilmektedir.

$$L(x, y) = -[y \cdot \log(z) + (1 - y) \cdot \log(1 - x)] \quad (2.11)$$

Bu bölümde ayrıca, özellikle ImageNet’in düzenlemiş olduğu ILSVRC yarışmalarına katılıp yüksek performans göstererek literatüre girmiş CNN mimarileri hakkında kısa bilgiler verilmiştir. Bu mimariler VGGNet, AlexNet, Resnet, GoogleNet ve Inception-ResNet’tir.

Bu çalışma sırasında da kullanılan ve Keras kütüphanesinde de yer alan mimarilerden VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2 ve MobileNetV2 mimarilerinin ImageNet doğrulama (validation) veri kümesine ait performans parametreleri Çizelge 2.1’de gösterilmiştir [21].

Çizelge 2.1 : Bazı mimarilerin performans parametreleri.

Model	Boyut	Top-1 Doğruluk	Top-5 Doğruluk	Parametre Sayısı
VGG16	528 MB	0.713	0.901	138,357,544
VGG19	549 MB	0.713	0.900	143,667,240
ResNet50	98 MB	0.760	0.930	25,613,800
InceptionV3	92 MB	0.779	0.937	23,851,784
InceptionResNetV2	215 MB	0.803	0.953	55,873,736
MobileNetV2	14 MB	0.750	0.901	3,538,984

2.2.1 VGGNet

VGGNet, ImageNet'in düzenlemiş olduğu ILSVRC 2014 yarışmasında sınıflandırma görevinde ikinci olmuş bir CNN mimarisidir. Oxford Üniversitesi'nden VGG (Visual Geometry Group) tarafından geliştirilmiştir.

VGGNet, girdi görüntüsü olarak 224x224x3 boyutunu kabul eder, daha yüksek boyutları bu boyuta kırparak işlemi yapabilir. Ağırlık katmanı 11'den 19'a kadar olacak şekilde farklı versiyonları mevcut olmakla birlikte, hepsinde 3x3 filtreler kullanılmaktadır. Böylelikle parametre sayısı azaltılmış ve lineer olmayan 3 farklı düzeltme katmanı kullanılmıştır [23].

Pooling katmanı için maksimum pooling kullanılmıştır ve 3 tane tam bağlantı katmanı içermektedir. Bu çalışmada da sonraki bölümlerde kullanılmış olan VGGNet yapılarından 16 ve 19 katmanlı olan VGG16 ve VGG19 mimarisinin katman detayları Çizelge 2.2'de gösterilmektedir.

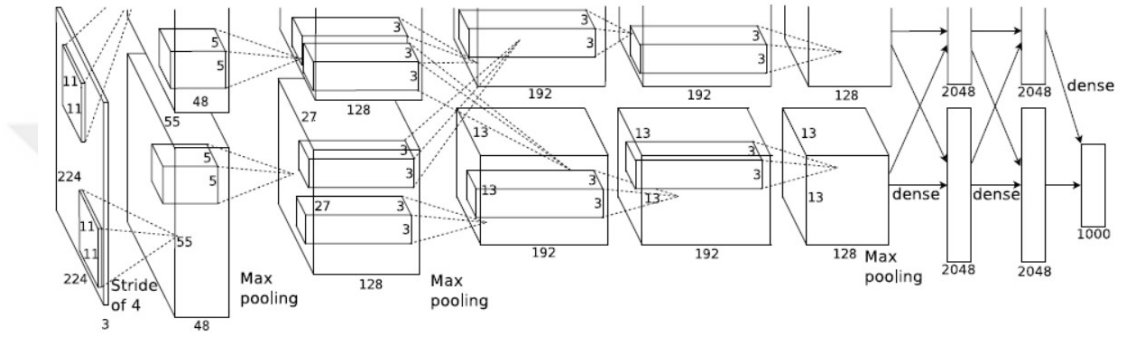
Çizelge 2.2 : VGGNet yapılandırmaları.

VGG16	VGG19
224x224 RGB Görüntü	
2 x Konvolüsyonel Katman (3x3, 64)	2 x Konvolüsyonel Katman (3x3, 64)
Maksimum Pooling	
2 x Konvolüsyonel Katman (3x3, 128)	2 x Konvolüsyonel Katman (3x3, 128)
Maksimum Pooling	
3 x Konvolüsyonel Katman (3x3, 256)	4 x Konvolüsyonel Katman (3x3, 256)
Maksimum Pooling	
3 x Konvolüsyonel Katman (3x3, 512)	4 x Konvolüsyonel Katman (3x3, 512)
Maksimum Pooling	
3 x Konvolüsyonel Katman (3x3, 512)	4 x Konvolüsyonel Katman (3x3, 512)
Maksimum Pooling	
Tam Bağlantı - 4096	
Tam Bağlantı - 4096	
Tam Bağlantı - 1000	
Softmax Aktivasyonu	

2.2.2 AlexNet

AlexNet, ImageNet'in düzenlemiş olduğu ILSVRC 2012 yarışmasında birinci olmuş bir CNN mimarisidir. ILSVRC 2010 yarışmasına katılmış oldukları modelin geliştirilmesiyle AlexNet CNN mimarisi son halini almıştır.

AlexNet, girdi görüntüsü olarak 224x224x3 boyutunu kabul etmektedir ve 8 katmandan oluşmaktadır. Birinci konvolüsyonel katmanda 11x11, ikinci konvolüsyonel katmanda 5x5 ve geri kalan konvolüsyonel katmanlarda ise 3x3 kernel olarak filtre uygulanmıştır. Daha önceki uygulamalarda aktivasyon fonksiyonu olarak genellikle tanh kullanılmasına karşı bu mimaride relu aktivasyon fonksiyonu kullanılmıştır. Tanh fonksiyonuna göre çok %25 hata oranına ulaşması 6 kat daha hızlıdır [24]. Aynı zamanda Şekil 2.9’da görüleceği üzere mimari iki parçaya ayrılarak işleme alındığı için iki farklı GPU kullanımına imkan tanımaktadır.



Şekil 2.9 : AlexNet Mimarisi [25].

2.2.3 ResNet

ResNet, Microsoft ekibinin ImageNet’in düzenlemiş olduğu ILSVRC yarışması için geliştirmiş olduğu çok sayıda artık blokların kullanıldığı bir CNN mimarisidir. ImageNet için tasarlanmış 18, 34, 50, 152 gibi farklı sayılarda katmanları olan ResNet mimarileri bulunmaktadır. Çizelge 2.3’te 34 katmanlı ResNet mimarisi gösterilmektedir.

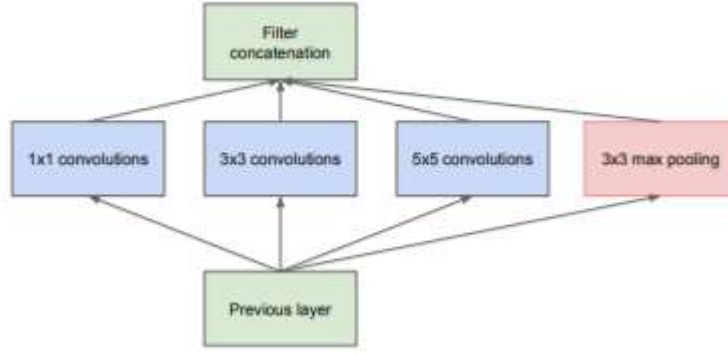
Çizelge 2.3 : 34 Katmanlı ResNet Mimarisi.

224x224 RGB Görüntü
Konvolüsyonel Katman, (7x7, 64), Stride=2
Maksimum Pooling
2 x Konvolüsyonel Katman, (3x3, 64) 2 x Konvolüsyonel Katman, (3x3, 64) 2 x Konvolüsyonel Katman, (3x3, 64)
2 x Konvolüsyonel Katman, (3x3, 128), Stride=2 2 x Konvolüsyonel Katman, (3x3, 128) 2 x Konvolüsyonel Katman, (3x3, 128) 2 x Konvolüsyonel Katman, (3x3, 128)
2 x Konvolüsyonel Katman, (3x3, 256), Stride=2 2 x Konvolüsyonel Katman, (3x3, 256) 2 x Konvolüsyonel Katman, (3x3, 256) 2 x Konvolüsyonel Katman, (3x3, 256) 2 x Konvolüsyonel Katman, (3x3, 256) 2 x Konvolüsyonel Katman, (3x3, 256)
2 x Konvolüsyonel Katman, (3x3, 512), Stride=2 2 x Konvolüsyonel Katman, (3x3, 512) 2 x Konvolüsyonel Katman, (3x3, 512)
Ortalama Pooling
Tam Bağlantı - 1000

2.2.4 GoogleNet

GoogleNet, Google ekibinin geliştirdiği bir CNN mimarisidir ve 2014 ILSVRC’de sınıflandırma doğruluk oranı olarak VGGNet ile yakın sonuçlar vermiştir. Ancak lokalizasyon konusundaki başarısıyla birlikte 2014 ILSVRC’de birinci olmuş bir mimaridir.

Mimarinin temelleri “Inception” adı verilen modüllere dayanmaktadır ve şu anda 4. versiyon inception modülü geliştirilmiştir. Temel “Inception” modülü Şekil 2.10’da gösterilmiştir.



Şekil 2.10 : Inception Modülü [27].

GoogleNet mimarisi, 224x224x3 boyutuyla görüntüleri işlemektedir ve her konvolüsyonel katman inception modülü içermektedir. Bu mimari hesaplama verimliliği ön planda tutularak tasarlanmış ve özellikle hafıza olacak şekilde sınırlı kaynaklarla çalışabilmesi hedeflenmiştir [27]. Örnek olarak, AlexNet mimarisi eğitim sırasında 60 milyon, VGGNet yaklaşık olarak bunun 3 katı kadar parametre kullanırken GoogleNet sadece 5 milyon parametre ile sonuca ulaşmaktadır [28].

GoogleNet mimarisine ait katmanlar Çizelge 2.4’te gösterilmiştir.

Çizelge 2.4 : GoogleNet Katmanları.

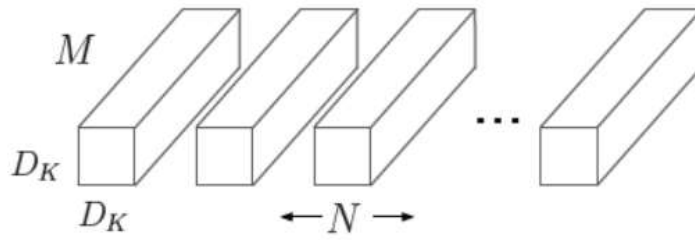
224x224 RGB Görüntü
Konvolüsyonel Katman, (7x7, 64), Stride=2
Maksimum Pooling
Konvolüsyonel Katman, (3x3, 64)
Maksimum Pooling
2 x Inception
Maksimum Pooling
5 x Inception
Maksimum Pooling
2 x Inception
Ortalama Pooling
Dropout(0.4)
Tam Bağlantı - 1000
Softmax Aktivasyonu

2.2.5 MobileNet

MobileNet, gömülü sistemlerde ve mobil uygulamalarda görüntü işleme ve sınıflandırma çalışmalarında kullanılmak üzere geliştirilmiş bir mimaridir.

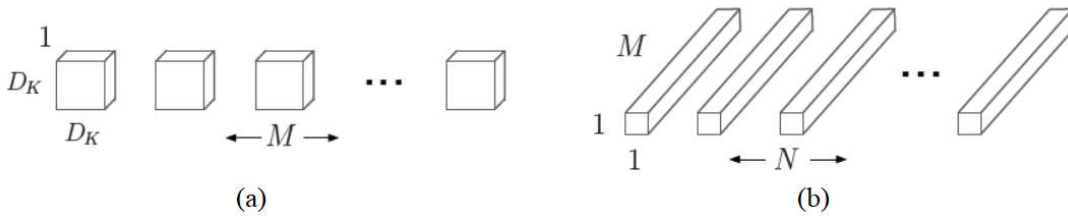
MobileNet mimarisi, ayrışabilen konvolüsyonlar kullanılarak daha verimli işlem yapabilen bir mimaridir. Bu sayede daha küçük ve hızlı bir ağ oluşturularak mobil cihazlarda gerçekleştirilebilirliği sağlanmıştır. Nesne tanıma, yüz özellikleri, büyük ölçekli coğrafi konumlandırma gibi farklı uygulamalarla da mimarinin doğruluk oranları elde edilerek mimari geliştirilmiştir [29].

Bu mimaride Şekil 2.11’de gösterilen standart konvolüsyon filtreleri yerine derinlemesine ve noktasal olacak şekilde ayrıştırılmış halde konvolüsyon işlemi uygulanmaktadır.



Şekil 2.11 : Konvolüsyon filtreleri [29].

Derinlemesine ve noktasal konvolüsyonel filtreler Şekil 2.12’de gösterilmiştir.



Şekil 2.12 : (a) Derinlemesine ve (b) noktasal konvolüsyon filtreleri [29].

Şekil 2.11 ve Şekil 2.12’de gösterilen filtrelerin boyutları ve uygulanacak filtre sayısı F olacak şekilde hesaplama maliyetleri standart konvolüsyon işlemi için Denklem 2.12’de, derinlemesine ve noktasal olarak ayrışan konvolüsyonel filtre için ise Denklem 2.13’de gösterilmiştir.

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2.12)$$

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (2.13)$$

Bu yaklaşımla, MobileNet mimarisi, 3x3 konvolüsyonlar kullandığından dolayı doğruluk oranında küçük bir kayıpla 8-9 kat daha hızlı işlem yapabilmektedir [29].

2.2.6 Inception-ResNet

Inception-ResNet mimarisi, Inception ve ResNet mimarilerinin birleştirilmesiyle oluşturulmuş hibrit bir mimaridir. Inception modüllerinin artıkların da dahil edildiği bağlantıların eklenmesiyle geliştirilmiştir.

Bu mimarinin iki farklı versiyonu bulunmaktadır. İlk versiyon Inception-v3 ile benzer işlem gücü gerektirirken, ikinci versiyon ise Inception-v4 ile benzer performans göstermesine rağmen daha yüksek işlem gücü gerektirmektedir [30].



3. EVRİŞİMSEL SİNİR AĞLARI İLE TEKSTİL TİPİ TANIMA

Bu bölümde, tekstil tipi sınıflandırma ve tanıma problemine çözüm önermek üzere standart kameralar ile alınan görüntülerin işleneceği bir evrişimsel sinir ağı tasarımı yapılacak, mevcut mimarilerin veri kümesi ile olan doğruluk oranları elde edilecek ve sonuçlar analiz edilerek yorumlanacaktır.

Bu amaçla, öncelikle VGG16, VGG19, MobileNetV2, InceptionResNetV2, ResNet50V2 ve InceptionV3 modelleri eldeki veri kümesi ile eğitilerek ayrı ayrı değerlendirilecektir. Ardından önerilen mimarinin ayrıntıları aktarılarak eğitim sonucu ve yapılan çalışmalar açıklanacaktır.

Çalışmada tekstil ürünlerine ait veri kümesi olarak, Kampouris ve arkadaşlarının 2016 yılında yaptıkları çalışma için toplamış oldukları tekstil görüntüleri kullanılmıştır [3].

Yazılım geliştirme dili olarak Python kullanılmıştır. Derin öğrenme fonksiyonlarını yerine getirmek amacıyla ise TensorFlow altyapısı ile çalışan Keras kütüphanesi kullanılmış olup, öncül çalışmalar için JetBrains PyCharm Community Edition kullanılmıştır. Önerilen CNN mimarisi ve diğer mimarilerin eğitilmeleri için GPU'ya ihtiyaç duyulması nedeniyle geliştirmeler Google Colab üzerinde yapılmıştır.

Çalışmanın prototiplenmesi için Raspberry Pi 3 ve Logitech C170 Web Cam kullanılmıştır.

3.1 Veri Kümesinin Değerlendirilmesi ve Hazırlanması

Veri kümesinde 26 farklı sınıfa ait görüntüler bulunmaktadır. Ancak çalışma kapsamına örnek sayısı çok az olanlar ve farklı tipteki tekstil ürünlerinin birlikte kullanılmasıyla elde edilen görüntüler dahil edilmemiştir. Çalışmada da 9 farklı sınıfa ait toplamda farklı ışıklandırma ile fotoğraf çekilmiş versiyonları ile birlikte toplamda 5092 görüntü bulunmaktadır ve bunların 4245 adedi eğitim verisi, 847 adedi ise test verisi olarak kullanılmıştır.

Veri kümesinde yer alan ve çalışmada kullanılacak 9 farklı sınıf pamuk, kot, koyun yünü, naylon, polyester, ipek, havlu kumaşı, viskon ve yün için örnek görüntüler Şekil 3.1’de gösterilmektedir.



Şekil 3.1 : Veri kümesine ait örnek görüntüler.

Toplamda 5082 adet görüntüden oluşan veri kümesi vardır. Bu veri kümesinin 4235 adedi eğitim verisi, 847 aded, ise test verisi olarak kullanılacaktır. Ayrıca çalışmada kullanılan bu veri kümesinde sınıflar dengeli dağılmamıştır. 2352 adet pamuk, 648 adet kot, 132 adet koyun yünü, 228 adet naylon, 904 adet polyester, 200 adet ipek, 120 adet havlu kumaşı, 148 adet viskon ve 360 adet yün sınıfına ait görüntü bulunmaktadır. Sınıf bazında veri sayısı Çizelge 3.1’de gösterilmiştir.

Çizelge 3.1 : Sınıf bazlı veri kümesi.

Tekstil Tipi	Eğitim Numune Sayısı	Test Numune Sayısı
Pamuk	1960	392
Kot	540	108
Koyun Yünü	110	22
Naylon	190	38
Polyester	754	150
İpek	167	33
Havlu Kumaşı	100	20
Viskon	124	24
Yün	300	60

Veri kümesinde bulunan görüntüler 400x400 çözünürlüğünde renkli görüntülerdir. Bu nedenle, çalışma sırasında kullanılan mimarilerin giriş boyutlarına göre görüntüler kırılıp, gri seviye dönüşümü yapılarak kullanılmıştır.

Veri kümesindeki görüntüler, tekstil ürününden çok yakından alındığı için, görüntünün tamamı tekstile aittir. Bu nedenle, CNN mimarilerinin ihtiyaç duyduğu boyutları elde etmek için herhangi bir veri kaybı sorunu olmadığından yeniden boyutlandırma yerine kırpma yöntemi tercih edilerek görüntü netliğinin korunması sağlanmıştır.

3.2 Keras Kütüphanesi

Keras, Python dilini kullanan ve TensorFlow, CNTK, Theano gibi temel derin öğrenme kütüphanelerinin üzerinde çalışan bir arabirim kütüphanesidir. Modüler, esnek ve kullanıcı dostu tasarımı nedeniyle kolay ve hızlı prototipleme imkanı tanımaktadır. Bununla birlikte, bu çalışmada sadece CNN kullanılmış olsa da RNN'yi destekleyebilmektedir ve hem CPU'da hem de GPU'da sorunsuz bir şekilde çalışabilmektedir [31].

Bu çalışmada, hızlı denemeler yapıp sonuçları görme, CNN katmanlarına ayrı ayrı hızlı ve esnek şekilde müdahale edebilme, literatüre girmiş CNN mimarilerine hızlı erişim ve sonuçları görme, Raspberry Pi için adaptasyonunun kolay olması ve daha bunlar gibi bu çalışmaya özgü bir çok artı özelliğinin olması nedeniyle Keras kütüphanesi tercih edilmiştir.

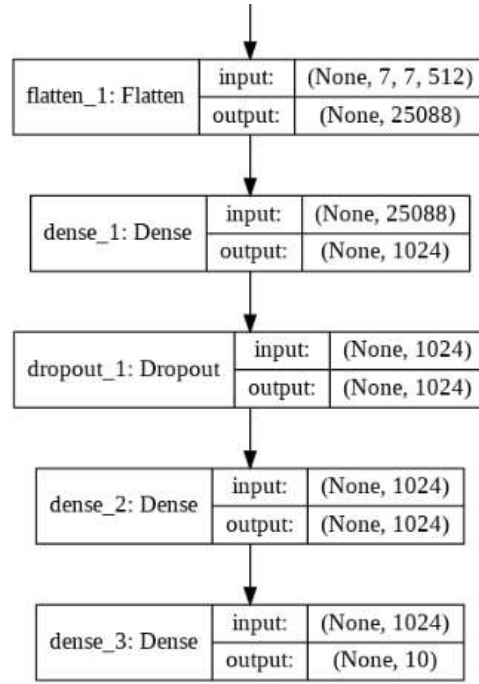
3.3 Bilenen Bazı Evrişimsel Sinir Ağları Mimarileri ile Tekstil Tipi Tanıma Çalışmaları

Bu bölümde, Keras kütüphanesinde de yer alan VGG16, VGG19, MobileNetV2, InceptionResNetV2, ResNet50 ve InceptionV3 mimarileri ile elde edilen veri seti kullanılarak tekstil tipi tanıma çalışması yapılmıştır.

Bu mimariler, ImageNet veri kümesi için ILSVRC yarışmasına yönelik geliştirildiğinden çok yüksek veri kümesi ve 1000 adetlik sınıfa göre tasarlanmıştır. Bu nedenle, tüm ağırlıklar sıfırlanarak eldeki küçük veri kümesi ile modeller eğitildiği zaman sonuç alınamamıştır. Bu mimariler ile sonuç alabilmek için ImageNet ile eğitilmiş olan bu modeller, bu çalışmaya transfer edilmiş ve ImageNet için hesaplanmış ağırlıklar kullanılmıştır. Tam bağlantı elemanları modele dahil edilmemiş ve bu çalışmanın konusuna uygun olacak şekilde tam bağlantı elemanları modele eklenerek sonuçlar elde edilmiştir.

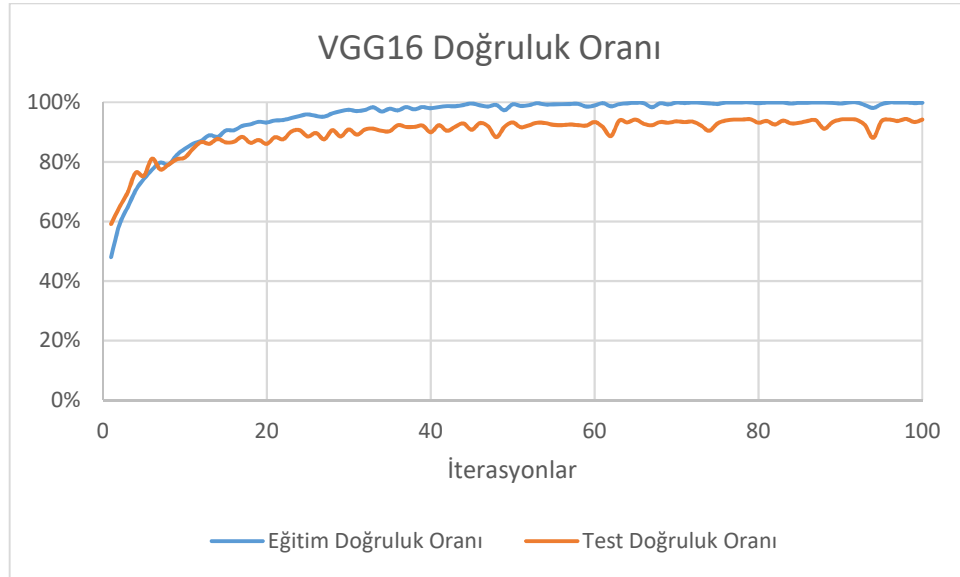
3.3.1 VGG16 ve VGG19 ile tekstil tipi tanıma çalışmaları

ImageNet ile eğitilmiş VGG16 mimarisinin ağırlıkları kullanılmıştır ve ilk 5 katmanın eğitilmeleri engellenmiştir. Tam bağlantı katmanları alınmayan model için sırasıyla düzleştirme, tam bağlantı, dropout ve tam 2 tane daha tam bağlantı katmanı eklenerek tekstil tipi tanıma problemine uygun sınıf sayısına göre çalışabilir hale getirilmiştir. Eklenen tam bağlantı katmanı Şekil 3.2’de gösterilmiştir.



Şekil 3.2 : VGG16 için tam bağlantı katmanı.

Modelin bu haliyle 41.465.674 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 GPU ile eğitim süresi 2509 saniyedir. Test veri kümesi için %94,21 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekilde 3.3'te gösterilmiştir.



Şekil 3.3 : VGG16 için eğitim ve test doğruluk oranları.

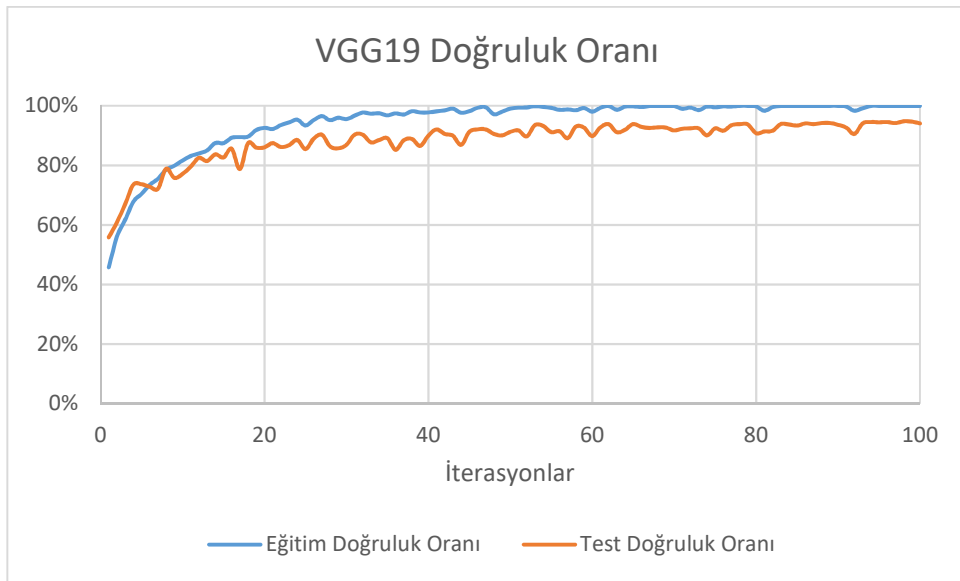
Modele ait hata matrisi ise Şekil 3.4'te gösterilmektedir.



Şekil 3.4 : VGG16 için hata matrisi.

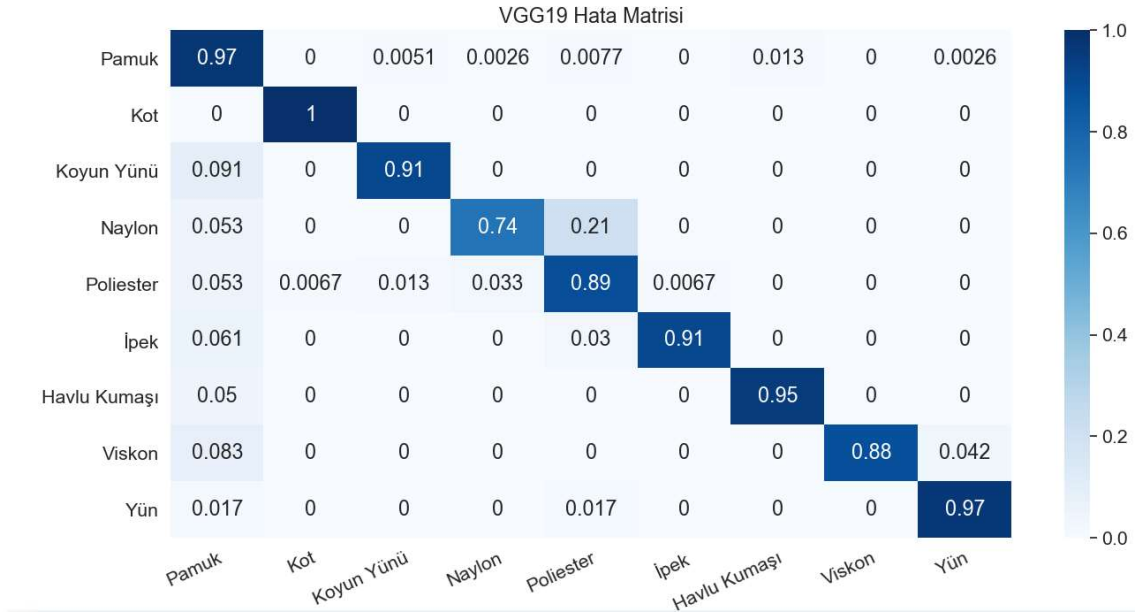
VGG19 mimarisi için de VGG19 mimarisinden kaynaklı farklarla birlikte VGG16 mimarisinde olduğu gibi ilerlenmiştir. Tam bağlantı katmanı VGG16 ile birebir aynıdır.

Modelin bu haliyle 46.662.794 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 GPU ile eğitim süresi 11214 saniyedir. Test veri kümesi için %94,19 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekilde 3.5'te gösterilmiştir.



Şekil 3.5 : VGG19 için eğitim ve test doğruluk oranları.

Modele ait hata matrisi ise Şekil 3.6'da gösterilmektedir.

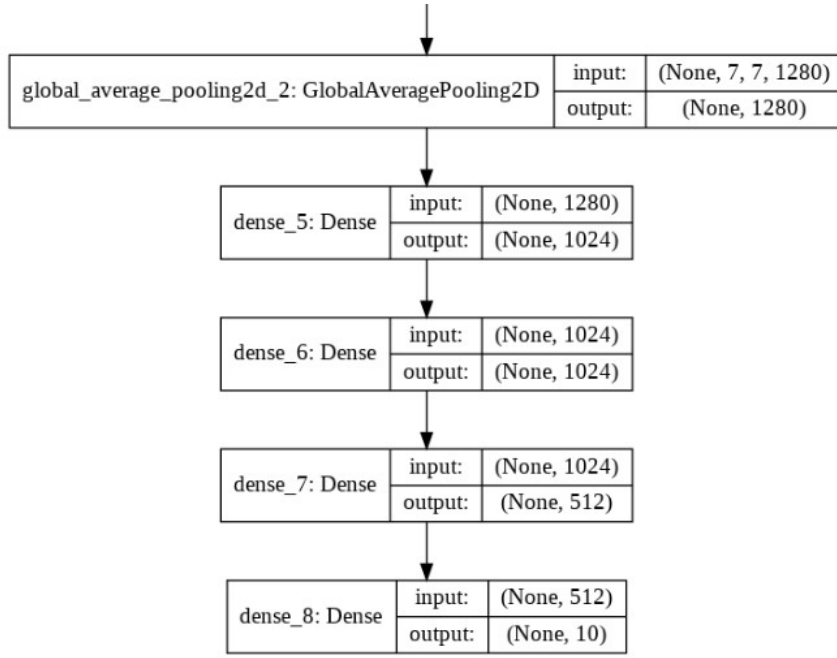


Şekil 3.6 : VGG19 için hata matrisi.

VGG16 ve VGG19 sonuçları değerlendirildiğinde %94,21 ve %94,19 ile benzer doğruluk oranları elde edilmiş ve eğitim aşamasının kararlı olduğu görülmüştür. Bununla birlikte 0,9 ve üzeri olarak değerlendirecek olursak, sınıf bazlı doğruluk oranlarının da VGG16 için naylon sınıfında, VGG19 için ise hem naylon hem de viskon sınıfında bu değerin altında kaldığı görülmektedir. Ancak genel olarak model performansı ele alındığında başarılı bir sonuç elde edildiği söylenebilir.

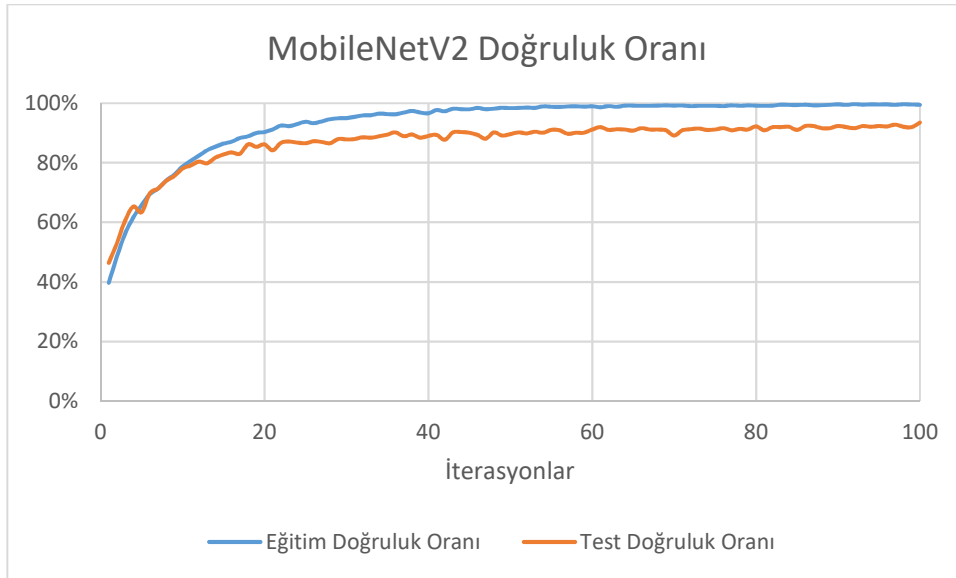
3.3.2 MobileNetV2 ile tekstil tipi tanıma çalışmaları

ImageNet ile eğitilmiş MobileNetV2 mimarisinin ağırlıkları kullanılmıştır ancak tekrardan eğitilmeleri engellenmemiştir. Tam bağlantı katmanları alınmayan model için ortalama pooling ve 3 tam bağlantı katmanı eklenerek tekstil tipi tanıma problemine uygun sınıf sayısına göre çalışabilir hale getirilmiştir. Eklenen tam bağlantı katmanı Şekil 3.7’de gösterilmiştir.



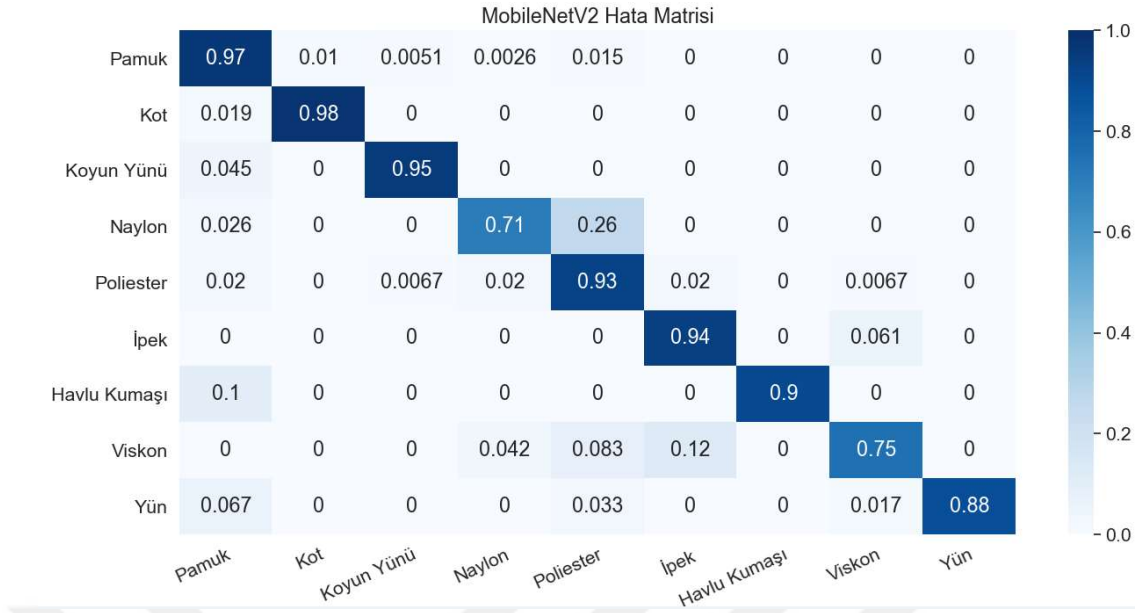
Şekil 3.7 : MobileNetV2 için tam bağlantı katmanı.

Modelin bu haliyle 5.115.146 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 GPU ile eğitim süresi 2509 saniyedir. Test veri kümesi için %93,51 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekilde 3.8’te gösterilmiştir.



Şekil 3.8 : MobileNetV2 için eğitim ve test doğruluk oranları.

Modele ait hata matrisi ise Şekil 3.9’da gösterilmektedir.

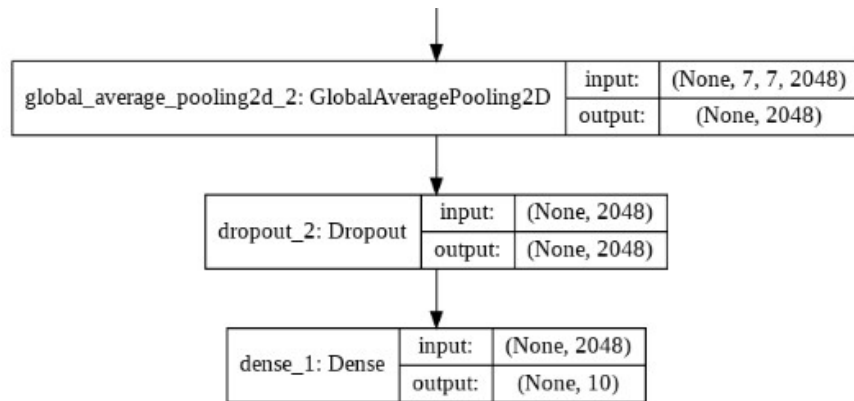


Şekil 3.9 : MobileNetV2 için hata matrisi.

Sınıf bazında hata oranları incelendiğinde viskon ve naylon için diğer sınıflara göre daha düşük performans göstermektedir. Ancak genel olarak model sonuçları değerlendirildiğinde test kümesi ile elde edilen %93,51’lik doğruluk oranı ve eğitimin kararlı bir şekilde gerçekleştirilmesi nedeniyle modelin genel olarak başarılı bir performans ortaya koyduğu söylenebilir.

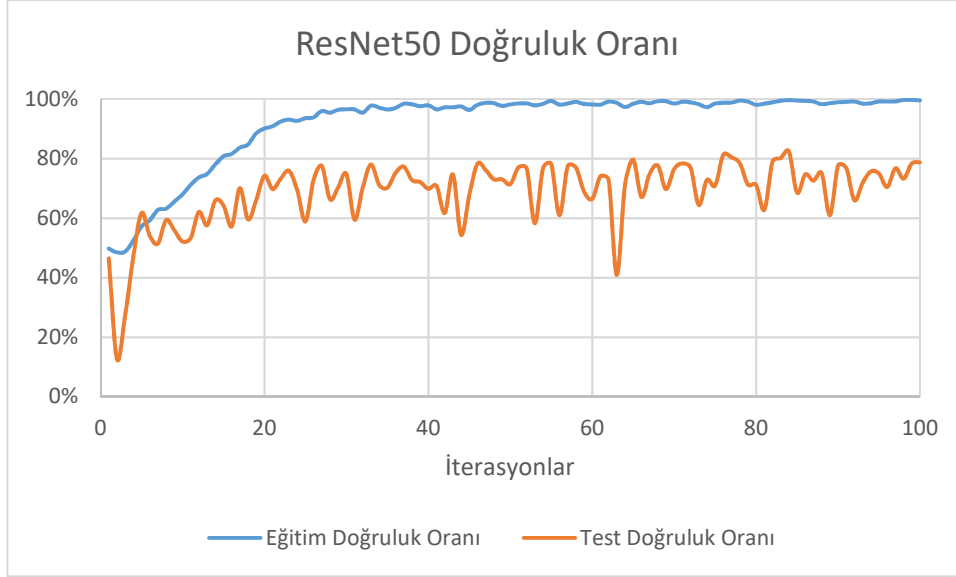
3.3.3 ResNet50 ile tekstil tipi tanıma çalışmaları

ImageNet ile eğitilmiş ResNet50 mimarisinin ağırlıkları kullanılmıştır ancak tekrardan eğitilmeleri engellenmemiştir. Tam bağlantı katmanları alınmayan model için ortalama pooling, dropout katmanı ve tam bağlantı katmanı eklenerek tekstil tipi tanıma problemine uygun sınıf sayısına göre çalışabilir hale getirilmiştir. Eklenen tam bağlantı katmanı Şekil 3.10’da gösterilmiştir.



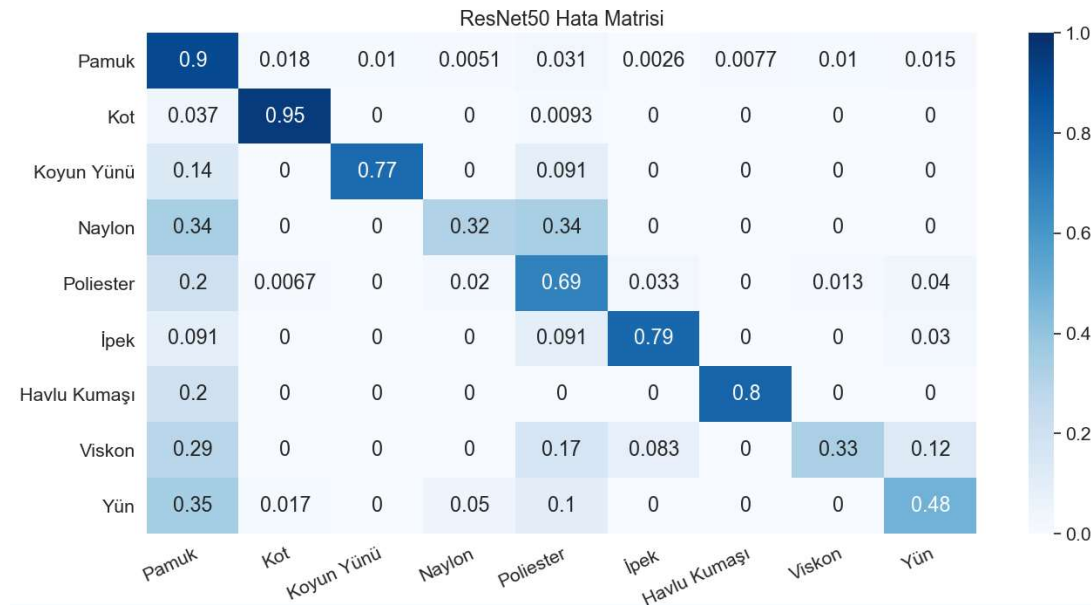
Şekil 3.10 : ResNet50 için tam bağlantı katmanı.

Modelin bu haliyle 23.534.592 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 ile eğitim süresi 3116 saniyedir. Test veri kümesi için %78,75 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekilde 3.11’de gösterilmiştir.



Şekil 3.11 : ResNet50 için eğitim ve test doğruluk oranları.

Modele ait hata matrisi ise Şekil 3.12’de gösterilmektedir.



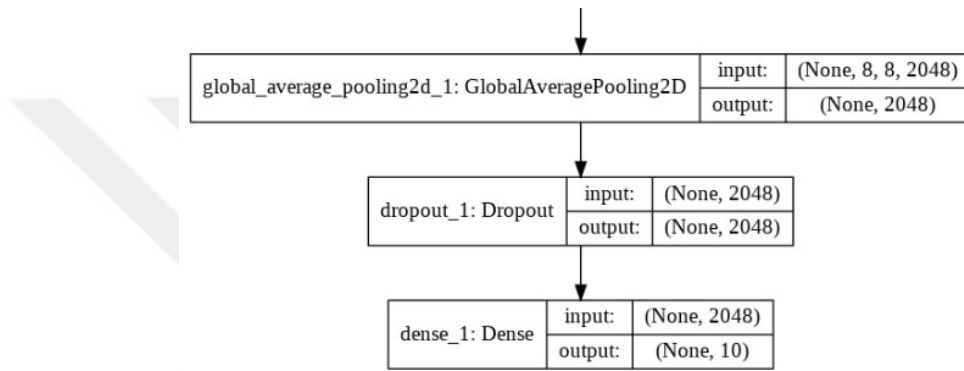
Şekil 3.12 : ResNet50 için hata matrisi.

ResNet50 ile elde edilen sonuçlar incelendiğinde hem genel performans, hem sınıf performansları hem de eğitim sürecindeki kararsızlık göz önüne alındığında model

performansının bu veri kümesi ile yetersiz sonuç verdiği görülmektedir. Eldeki veri kümesinin artıkların değerlendirilmesine uygun olmadığı yorumlanabilir.

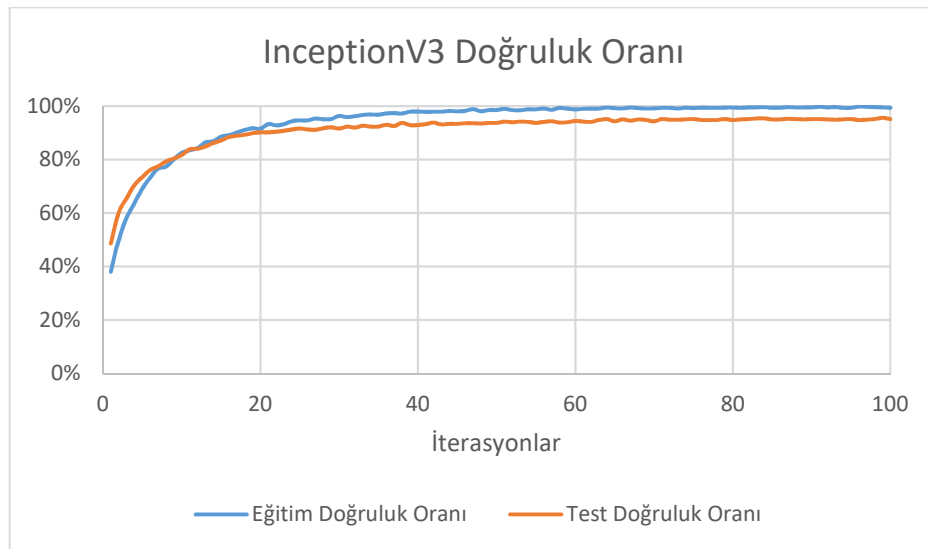
3.3.4 InceptionV3 ile tekstil tipi tanıma çalışmaları

ImageNet ile eğitilmiş InceptionV3 mimarisinin ağırlıkları kullanılmıştır ve tüm katmanların tekrardan eğitilmeleri engellenmiştir. Tam bağlantı katmanları alınmayan model için ortalama pooling, dropout katmanı ve tam bağlantı katmanı eklenerek tekstil tipi tanıma problemine uygun sınıf sayısına göre çalışabilir hale getirilmiştir. Eklenen tam bağlantı katmanı Şekil 3.13'te gösterilmiştir.



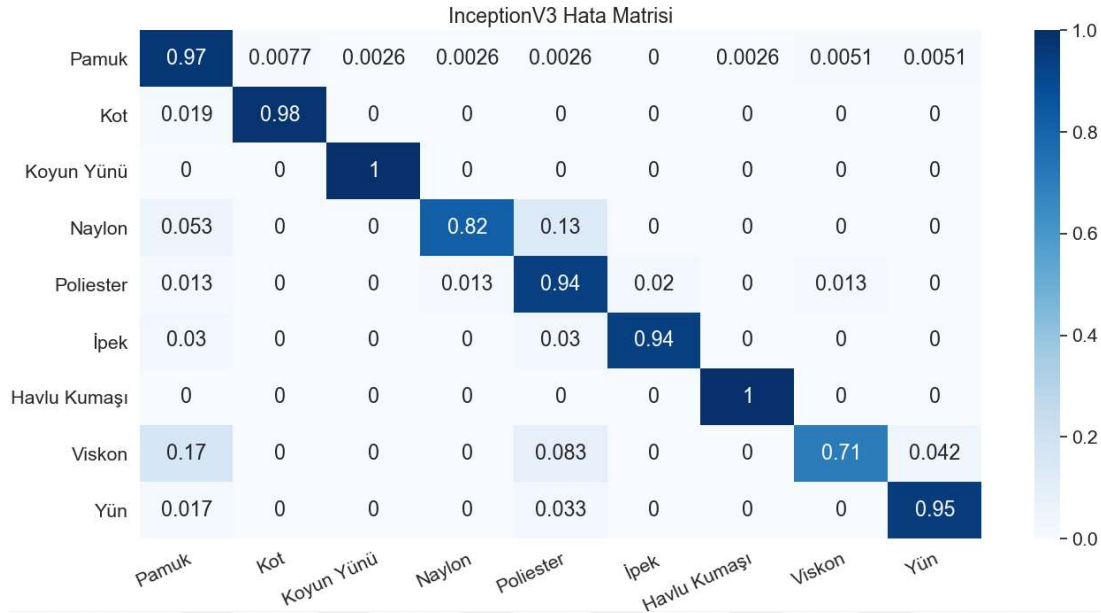
Şekil 3.13 : InceptionV3 için tam bağlantı katmanı.

Modelin bu haliyle 21.788.842 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 ile eğitim süresi 4218 saniyedir. Test veri kümesi için %95,16 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekil 3.14'te gösterilmiştir.



Şekil 3.14 : InceptionV3 için eğitim ve test doğruluk oranları.

Modele ait hata matrisi ise Şekil 3.15’te gösterilmektedir.

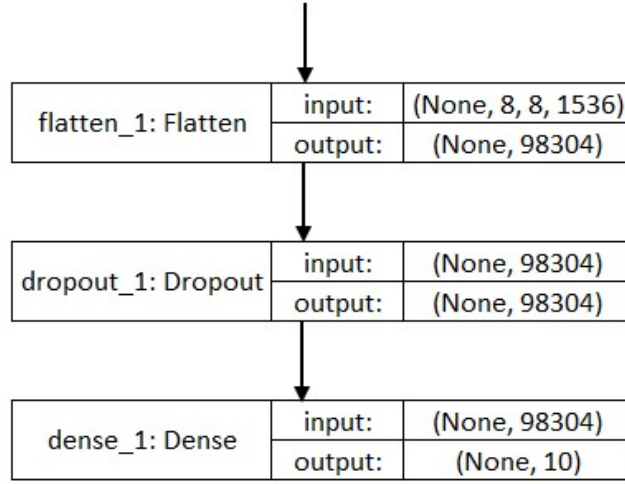


Şekil 3.15 : InceptionV3 için hata matrisi.

InceptionV3 mimarisi ile elde edilen sonuçlar incelendiğinde test kümesi ile %95,16 olacak şekilde oldukça yüksek doğruluk oranı elde edilmiştir. Sınıf bazlı olarak bakıldığında ise MobileNetV2’de olduğu gibi viskon ve naylon sınıflarında 0,9 değerinin altında kaldığı görülmektedir. 100 iterasyon için eğitim süreci değerlendirildiğinde de başarılı ve kararlı bir eğitim süreci olduğu görülmektedir.

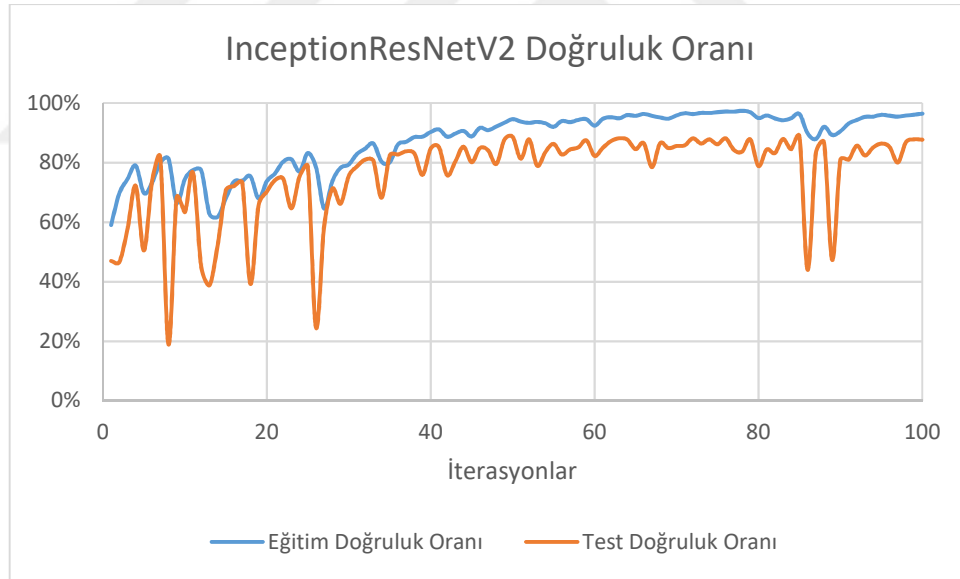
3.3.5 InceptionResNetV2 ile tekstil tipi tanıma çalışmaları

ImageNet ile eğitilmiş InceptionResnetV2 mimarisinin ağırlıkları kullanılmıştır. İlk 2 katmanının eğitilmesi engellenip geri kalan ağırlıkların çalışmaya ait veri kümesi ile eğitilmesi sağlanmıştır. Tam bağlantı katmanları alınmayan model için flattening katmanı, dropout katmanı ve tam bağlantı katmanı eklenerek tekstil tipi tanıma problemine uygun sınıf sayısına göre çalışabilir hale getirilmiştir. Eklenen tam bağlantı katmanı Şekil 3.16’da gösterilmiştir.



Şekil 3.16 : InceptionResNetV2 için tam bağlantı katmanı.

Modelin bu haliyle 54.336.736 adet eğitilebilir parametresi bulunmaktadır ve 100 iterasyon için Google Colab Tesla K80 ile eğitim süresi 9557 saniyedir. Test veri kümesi için %87,72 olacak şekilde doğruluk oranı elde edilmiştir. Eğitim ve test doğruluk oranları Şekil 3.17’de gösterilmiştir.



Şekil 3.17 : InceptionResNetV2 için eğitim ve test doğruluk oranları.

Modele ait hata matrisi ise Şekil 3.18’de gösterilmektedir.



Şekil 3.18 : InceptionResNetV2 için hata matrisi.

InceptionResNetV2 mimarisine ait sonuçlar incelendiğinde ise ResNet50’de olduğu gibi eğitim sürecinin kararsız olduğu görülmektedir. Ayrıca test kümesi ile %87,72 doğruluk oranı elde edilmesine rağmen sınıf bazlı sonuçlar kontrol edildiğinde bir çok sınıfta problemler görülmektedir. InceptionV3 ile gayet başarılı sonuçlar alınırken ResNet50’de de görüldüğü üzere artıklardan gelen olumsuz etkiler InceptionResNetV2 hibrit model sonuçlarının da yetersiz kalmasına neden olmuştur.

3.3.6 Sonuçların değerlendirilmesi

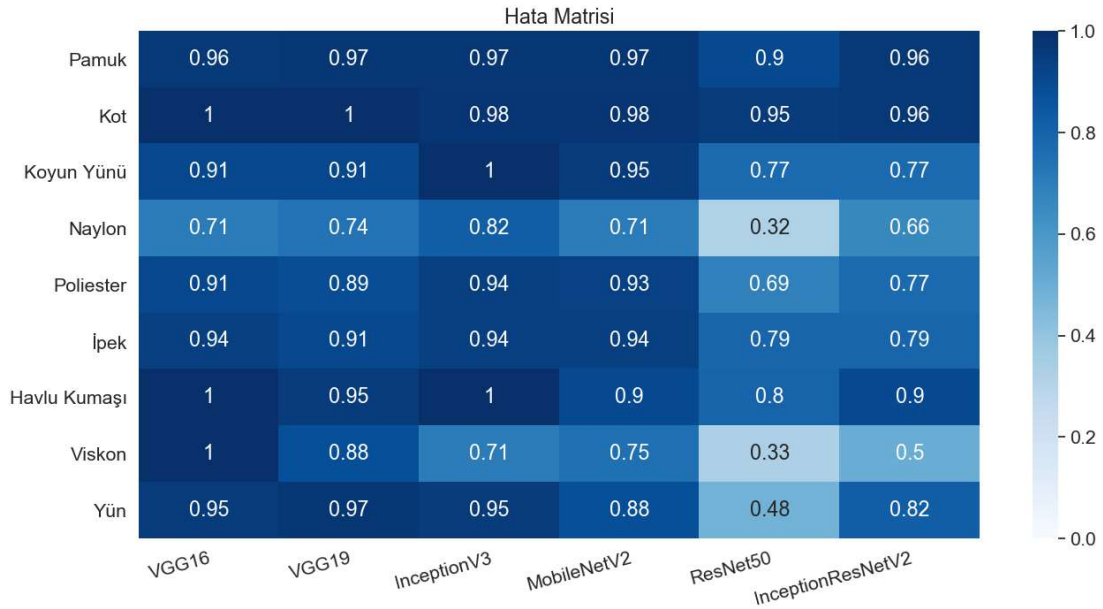
VGG16, VGG19, MobileNetV2, InceptionV3, ResNet50 ve InceptionResNetV2 için elde edilen sonuçlar Çizelge 3.2’de özet olarak paylaşılmıştır.

Çizelge 3.2 : Mevcut mimariler için sonuç tablosu.

Model	Test Doğruluğu(%)	Parametre Sayısı	Süre(sn)
VGG16	94,21	41.465.674	2509
VGG19	94,19	46.662.794	11214
InceptionV3	95,16	21.788.842	4218
MobileNetV2	93,51	5.115.146	2509
ResNet50	78,75	23.534.592	3116
InceptionResNetV2	87,72	54.336.736	9557

Çalışmada kullanılan veri kümesi ile VGG16, VGG19, InceptionV3 ve MobileNetV2 mimarileriyle elde edilen sonuçlar problem çözümü için kullanılabilir olduklarını göstermektedir.

Sınıf bazlı doğruluk oranları üzerinde çalışan 6 model için performanslar Şekil 3.19’da gösterilmiştir.



Şekil 3.19 : Modellerin sınıf bazlı performansı.

Sınıflandırma problemlerinde genel doğruluk oranının yüksek olması istenilen bir durumdur, ancak veri kümesi dengeli değilse, bu değer yanıltıcı olmaktadır. Bu

nedenle, sınıf bazlı doğruluk oranlarının da dikkate alınması gerekmektedir. Bu yüzden elde edilen sonuçlar değerlendirildiğinde 0.9 değeri limit olarak belirlenirse, VGG16 mimarisinin 9 sınıftan 8 tanesi için iyi sonuç vererek çalışmadaki en başarılı sonucu verdiği söylenebilir.

3.4 Önerilen CNN mimarisi ve uygulaması

Bir önceki bölümde literatüre girmiş ve ImageNet veri kümesi ile eğitilmiş mimarilerle yapılan çalışmalara ek olarak, bu bölümde tekstil tipi tanıma problemine özel oluşturulmuş ve tüm katmanların ağırlıklarının bu veri kümesi ile eğitildiği bir CNN mimarisi tasarlanmıştır. Bu kısımda, tekstilden yakından alınan görüntülerin sınıflandırmada fayda sağlayan özellikleri, konvolüsyonel katmanlardaki filtreler ayrıntılı olarak incelenmiştir.

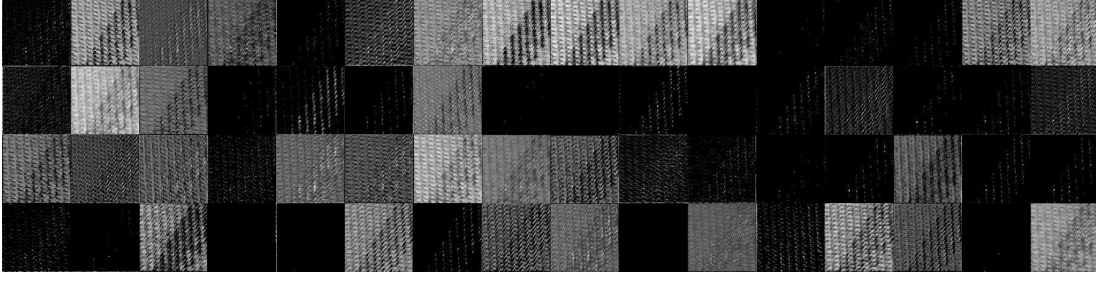
Önerilen CNN mimarisi, temel olarak 4 adet konvolüsyonel katman ve 3 adet tam bağlantı katmanından oluşmaktadır. Her katmanda normalizasyon yapılmış olup dropout sadece tam bağlantı katmanında yapılmıştır. Konvolüsyonel katmanlar için aktivasyon fonksiyonları “relu” olarak seçilmiştir. Keras kütüphanesinin çıktısı olarak önerilen CNN mimarisinin katmanları Çizelge 3.3’te gösterilmektedir.

Çizelge 3.3 : Önerilen mimariye ait katmanlar.

224x224 Gri Görüntü
Konvolüsyonel Katman, (3x3, 32)
Batch Normalizasyonu
Konvolüsyonel Katman, (3x3, 64)
Batch Normalizasyonu
Maksimum Pooling
Konvolüsyonel Katman, (3x3, 64)
Batch Normalizasyonu
Maksimum Pooling
Konvolüsyonel Katman, (3x3, 128)
Batch Normalizasyonu
Flatten Katmanı
Tam Bağlantı - 256
Dropout(0,2)
Tam Bağlantı – 128
Dropout(0,2)
Tam Bağlantı - 10

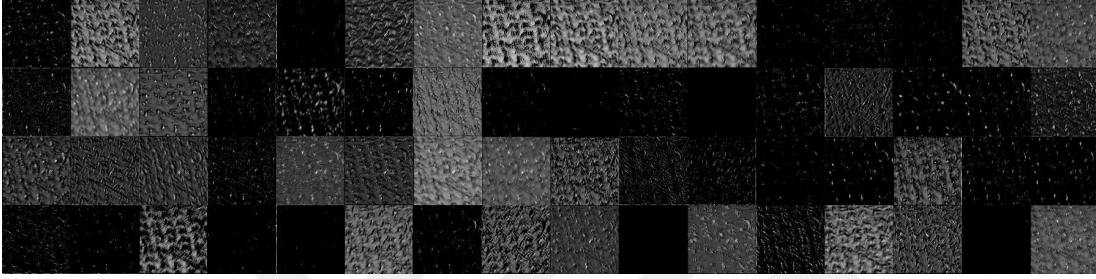
Tekstil tipine göre giyim ürünlerinin dokunmasındaki farklılıklar ve kalınlıklar değerlendirilerek tekstil tipi sınıflandırılması yapılmaktadır. Bu durum farklı tekstil tipi aynı dokuma yöntemi veya birebir aynı görüntüde olacak şekilde bir işlem yapıldığı anda sınıflandırma sonucunda yanlışlıklara sebep olabilecektir. Ancak veri kümesinin elde edilmesi ve CNN mimarileri ile gerçekleştirme sonuçları göz önüne alındığında şu anki uygulamalarda sınıflandırma yapılmasına engel bir durum bulunmamaktadır.

Örnek olarak Şekil 3.20’de, önerilen mimarinin üçüncü konvolüsyonel katman sonrasındaki özellik haritası gösterilmektedir.



Şekil 3.20 : 3. Konvolüsyonel katman sonrası pamuk sınıfına ait özellik haritası.

Yün sınıfından bir numuneye ait 3. Konvolüsyonel katman sonrası özellik haritası ise Şekil 3.21’de gösterilmiştir.



Şekil 3.21 : 3. Konvolüsyonel katman sonrası yün sınıfına ait özellik haritası.

Filtrelerin görselleştirilmiş hali ise Şekil 3.22’de paylaşılmaktadır.

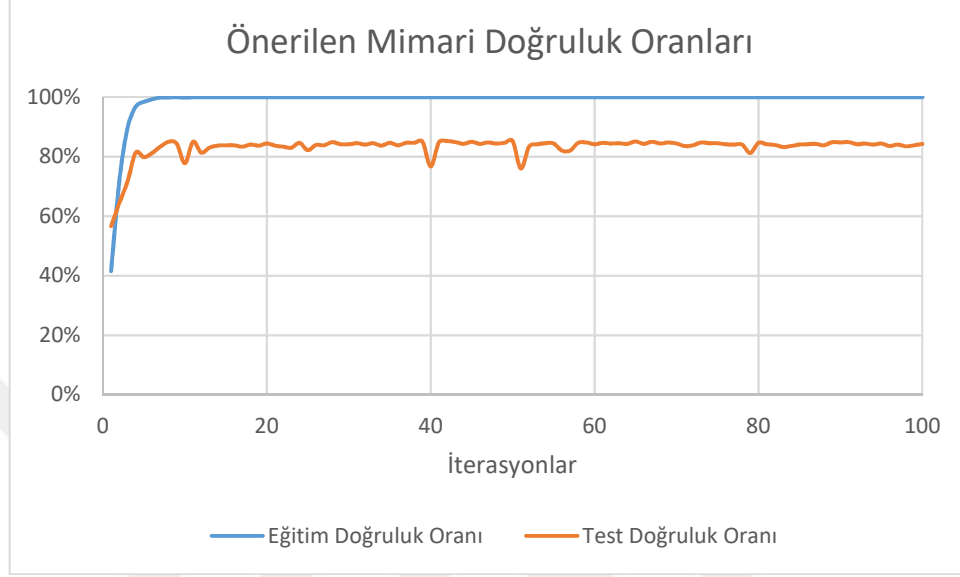


Şekil 3.22 : CNN mimarisine ait filtreler.

Bu mimari, diğer mimarilerde olduğu gibi 100 iterasyon, ve batch sayısı 32 olacak şekilde eğitilmiş ve sonuçlar değerlendirilmiştir.

3.4.1 Önerilen CNN mimarisi sonuçlarının analizi

Modelin 102.926.410 eğitilebilir parametresi bulunmaktadır. 100 iterasyon için Google Colab Tesla K80 ile eğitim süresi 2108 saniyedir. Eğitim ve test doğruluk oranları ile iterasyon sayısındaki ilişki grafiği Şekil 3.23'te gösterilmiştir.



Şekil 3.23 : Önerilen mimariye ait doğruluk oranı.

Test veri kümesi için doğruluk oranı %84,30'dur. Sonuç kabul edilebilir seviyelerde olmasına rağmen diğer CNN mimarileri sonuçlarıyla karşılaştırıldığında oldukça düşük seviyedir.

Hata matrisi ise Şekil 3.24'te gösterilmektedir.



Şekil 3.24 : Önerilen mimariye ait hata matrisi.

Çizelge 3.2’de paylaşılan özet tablosuna, önerilen CNN mimarisinin de bilgileri girilerek Çizelge 3.4 elde edilmiştir.

Çizelge 3.4 : Özet sonuç tablosu.

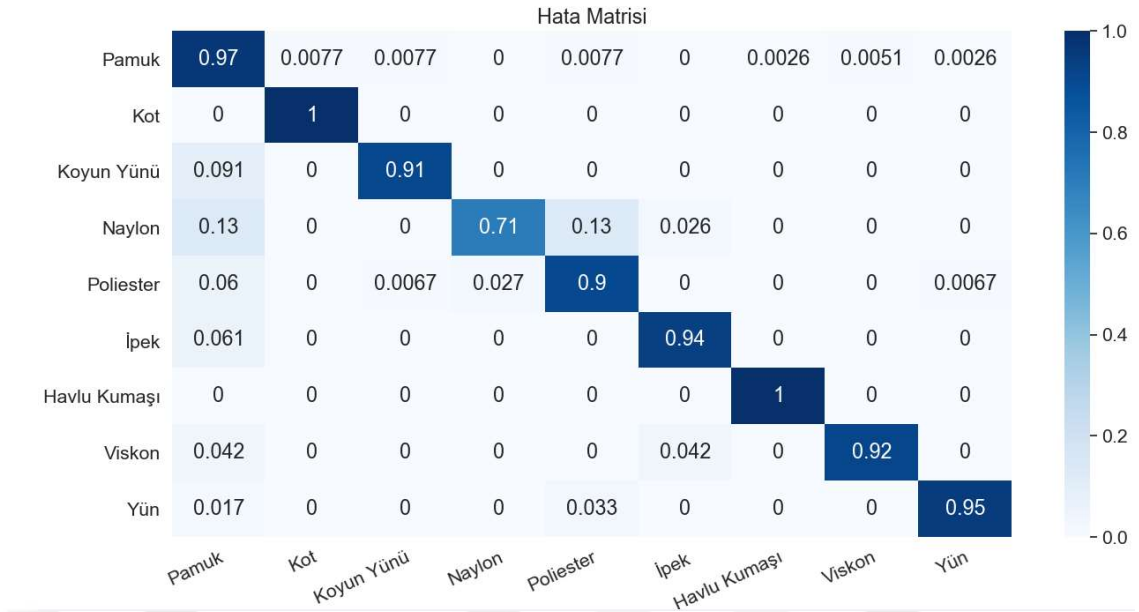
Model	Test Doğruluğu(%)	Parametre Sayısı	Süre(sn)
VGG16	94,21	41.465.674	2509
VGG19	94,19	46.662.794	11214
InceptionV3	95,16	21.788.842	4218
MobileNetV2	93,51	5.115.146	2509
ResNet50	78,75	23.534.592	3116
InceptionResNetV2	87,72	54.336.736	9557
Önerilen CNN	84,30	102.926.410	2108

Sonuçlar diğer mimariler ile karşılaştırıldığında yeterli olmasa da bazı sınıflardaki performansı kaynaklı toplam doğruluk oranına olumlu etkisi olma ihtimali üzerinde durulmuştur. Bu nedenle, VGG16 ile birlikte değerlendirilerek modeller tekrardan eğitilmiştir.

3.4.2 Önerilen CNN mimarisi ve VGG16 mimarisinin birlikte kullanılması

Derin öğrenme algoritmalarının çıktıları her denemede değiştiğinden ortak kullanım için önceden elde edilen sonuçlar yerine birlikte yeni bir modül yazılarak ilerlenmiş ve sonuçlar değerlendirilmiştir.

Öncelikle modül içerisinde her iki modelin de eğitilmesi gerekmektedir. VGG16 için yeniden elde edilen hata matrisi Şekil 3.25’de gösterilmektedir. Toplam doğruluk oranı %94,33’tür.



Şekil 3.25 : VGG16 hata matrisi.

Bu çalışmada önerilen mimariya ait hata matrisi ise Şekil 3.26'da gösterilmiştir. Toplam doğruluk oranı %83,47'dir.



Şekil 3.26 : Önerilen mimariya ait hata matrisi.

Sonuçlar incelendiğinde elde edilen verilerin tekil çalışmalarda elde edilen sonuçlar ile benzer olduğu görülmektedir. İki modeli bir arada kullanmak için aşağıdaki yöntemler denenmiştir.

- Test verisindeki her eleman için olasılıklar ortalaması alınarak karar verme

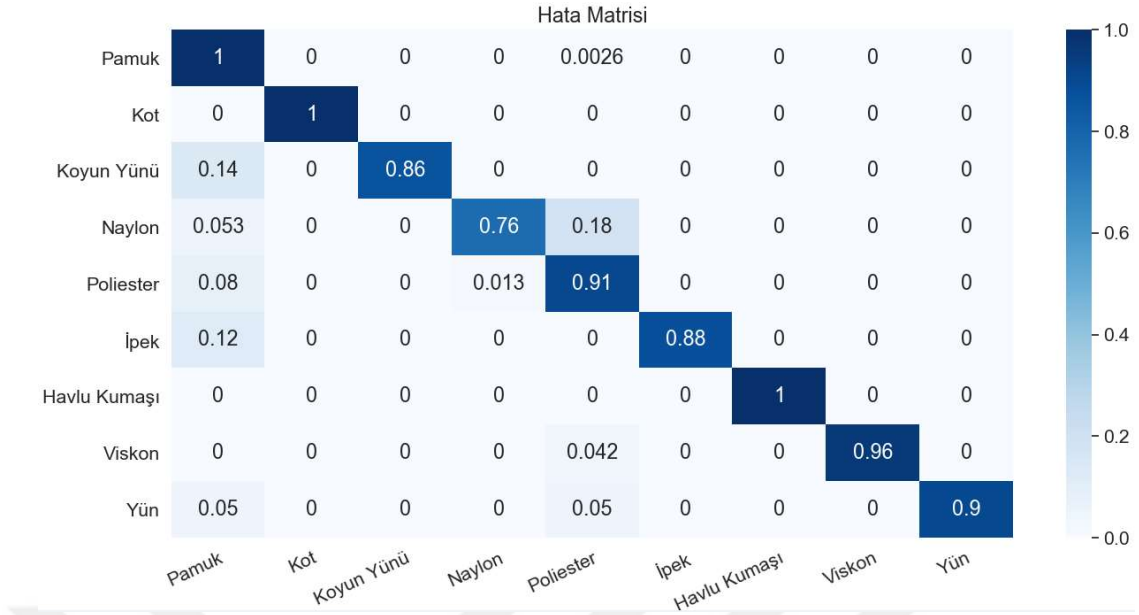
- İki modelden olasılık değerin büyük olanın belirlediği sınıfa göre karar verme
- Eğer iki modelin verdiği karar farklıysa, olasılık değeri büyük olanın belirlediği sınıfa göre karar verme
- Eğer iki modelin verdiği karar farklıysa, her iki modelin ilgili sınıfa ait olasılık değerlerinin ortalamaları alınarak yeni oluşan olasılık tablosundan sınıfa karar verme
- Eğer iki modelin verdiği karar farklıysa, her iki modelin ilgili sınıfa ait olasılık değerlerinin maksimumları alınarak yeni oluşan olasılık tablosundan sınıfa karar verme

En başarılı sonuçlar en son belirtilen modelin ilgili sınıflara ait olasılıklarının maksimumlarının alınmasıyla elde edilen tabloya göre karar verme sonucunda elde edilmiştir. Bu durumu doğrulamak için çarpaz doğrulama yapılarak sonuçlar elde edilmiştir. Toplam veri kümesinin 6’da 1’i test kümesi olarak kullanıldığından çarpaz doğrulama 6-kat olarak yapılmıştır. Sonuçlar Çizelge 3.5’te verilmiştir.

Çizelge 3.5 : 6-kat çarpaz doğrulama sonuçları.

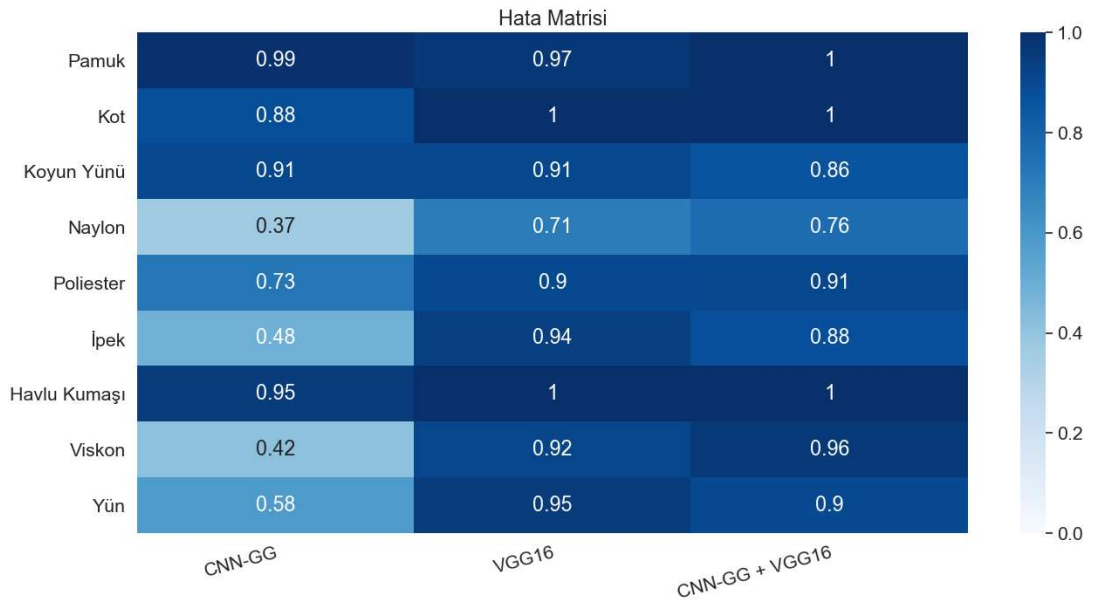
Model	1	2	3	4	5	6
Önerilen	%84,65	%83,94	%84,77	%83,59	%83,47	%85,83
VGG16	%94,57	%92,56	%89,49	%93,74	%92,09	%93,62
Önerilen + VGG16	%95,51	%94,21	%93,74	%94,8	%94,21	%95,39

VGG16 ve önerilen CNN mimarisinin birlikte kullanılmasıyla elde edilen doğruluk oranı %95,51’dir. Hata matrisi ise Şekil 3.27’de gösterilmiştir.



Şekil 3.27 : VGG16 ve önerilen CNN mimarisine ait hata matrisi.

Toplam doğruluk oranında %1'den fazla iyileşme olurken bazı sınıflarda VGG16'ya göre daha kötü sonuçlar elde edildiği görülmektedir. Şekil 3.28'de bu fark daha net bir şekilde gösterilmektedir.



Şekil 3.28 : Modellerin sınıf bazlı performansı.

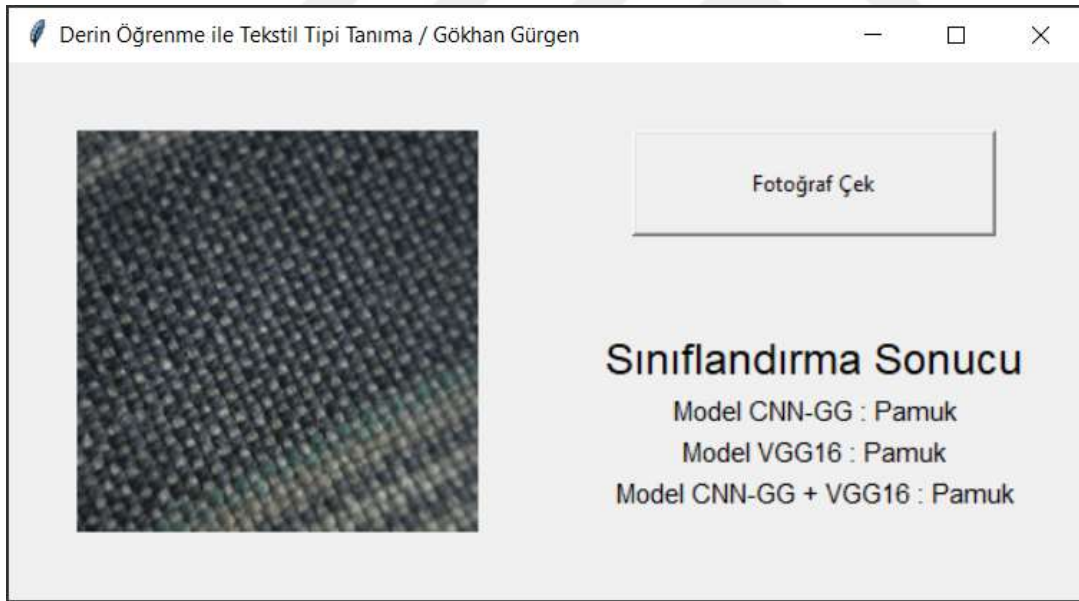
Birlikte kullanılan modelde, pamuk, viskon, naylon, polyester için bir miktar iyileştirme olurken, koyun yünü, ipek ve yün için ise bir miktar doğruluk oranı kötüleşmiştir. Doğruluk oranında düşüş görülen sınıfların son doğruluk oranlarının çok düşük olmadığı göz önünde bulundurulduğunda toplam doğruluk oranının da

artmasıyla birlikte önerilen CNN mimarisinin ve VGG16'nın birlikte kullanılmasıyla tekstil tipi tanıma doğruluk oranının bir miktar arttırılmış olduğu söylenebilir.

3.5 Çalışmanın Prototiplenmesi

Prptotip geliştirme için Raspberry Pi 3 kullanılmasına karar verilmiştir. Prototipin yapılacağı ortamın işlem gücü yetersiz olduğu için bu çalışmanın içeriği nedeniyle eğitim işlemini böyle bir cihazda gerçeklemek mümkün değildir. Bir önceki bölümde Google Colab'da GPU kullanımıyla eğitilip elde edilen önerilen CNN mimarisi ve VGG16 mimarisine ait katmanların ağırlıkları ve model bilgileri “.h5” ve “.json” dosyaları olarak kaydedilmiştir.

Prototiplemede kullanılacak Raspberry Pi 3 cihazı içerisinde CNN mimarilerinde elde edilen model dosyaları eklenerek tüm modelin çalışması sağlanmıştır. Basit bir arayüz hazırlanarak fotoğraf çektikten sonra çekilen fotoğrafa göre sınıflandırmasını yapan bir program geliştirilmiştir. Arayüz görüntüsü Şekil 3.29'da gösterilmektedir.



Şekil 3.29 : Modellerin sınıf bazlı performansı.

Kamera çözünürlük ayarı, veri kümesi 400x400x3 boyutlu resimlerden oluştuğu için farklılık oluşturmamak adına aynı şekilde ayarlanmıştır. 400x400x3 olarak çekilen bir fotoğraf 224x224 boyutuna kırpılıp gri seviye dönüşümü yapıldıktan sonra modele girdi olarak aktarılmaktadır.

İki modelin Raspberry Pi 3'e taşınmasıyla birlikte kameradan alınacak bir görüntüye karşı hem önerilen mimarinin, hem VGG16'nın hem de bu iki modelin ortak

kullanıldığı yapıya ilişkin sınıflandırma sonucu doğrudan elde edilmektedir. Bu üç sonuç da ayrı ayrı programda gösterilmektedir.





4. SONUÇ

Bu tez çalışmasında, derin öğrenme yöntemleri kullanılarak tekstil tipini tanıyabilecek mimariler üzerinde çalışılmış ve sadece tekstilden alınacak yaakin görüntü ile verilebilecek en iyi performans verilmeye çalışılmıştır. Elde edilen sonuçlar birbirleriyle karşılaştırılmış ve sınıf bazlı olarak sonuçlar ele edilmiştir. Ayrıca elde edilen başarılı modeller, taşınabilir bir cihaz ve kamera sistemine adapte edilerek çalışmanın hızlı ve ucuz gerçekleştirilebilirliği ve prototipleme imkanı olduğu gösterilmiştir.

Bu çalışmayla birlikte, ImageNet ile geliştirilmiş olan ve literatüre girmiş 6 farklı CNN mimarisi incelenerek tekstil tipi tanıma problemi için performansı değerlendirilmiştir. Bu mimariler VGG16, VGG19, MobileNetV2, InceptionV3, ResNet50 ve InceptionResNetV2'dir. Ayrıca, probleme özgü tasarlanmaya çalışılan başka bir CNN mimarisi de öneri olarak sunulup sonuçları elde edilmiştir.

VGG16 ve VGG19 mimarilerinin tam bağlantı katmanının tekstil tipi tanıma problemi için tekrardan oluşturulduktan sonra eğitim sonrası test veri kümesi ile sırasıyla %94,21 ve %94,19 doğruluk oranları elde edilmiştir. VGG16 eğitim süresi 2509 saniye sürerken, VGG19'da bu süre 11214 olarak hesaplanmıştır.

InceptionV3 mimarisi eğitildikten sonra test veri kümesi ile kontrol edildiğinde doğruluk oranının %95,16 olduğu görülmüştür.

MobileNetV2 mimarisi ise test veri kümesi kontrol edildiğinde %93.51 doğruluk oranı sağlamıştır. MobileNet mimarisinin az işlem ve az parametre ile mobil cihazlarda uygulanabilirliği yönündeki tasarım yaklaşımı nedeniyle de çalışmada kullanılan tüm mimariler arasında en az parametreye sahip olan mimaridir.

ResNet50 ve InceptionResNetV2 mimarileri eğitilip test edildiğinde ise doğruluk oranlarının diğer mimarilere göre daha düşük seviyelerde kaldığı görülmüştür. ResNet50 için doğruluk oranı %78,75, InceptionResNetV2 için ise %87.72'dir

Çalışmada önerilen mimarinin ise parametre sayısı oldukça yüksek olmasına rağmen işlem süresi kısadır. Bunda artıkların kullanılmaması, tek hat ile tasarım gerçekleştirilmesi

gibi etkenler önemli bir rol oynamıştır. Test veri kümesi ile doğruluk oranı kontrol edildiğinde ise %84,30 sonucu elde edilmiştir. VGG16, VGG19, InceptionV3 ve MobileNetV2 ile elde edilen yüksek doğruluk oranları nedeniyle bu haliyle doğrudan kullanılmasının bir amacı bulunmamaktadır. Ancak sınıf bazlı doğruluk oranları incelendiğinde bazı sınıflarda yüksek başarısı nedeniyle başka bir model ile birlikte kullanımı sırasında daha iyi sonuç verme ihtimali üzerinde durulmuş ve VGG16 ile birlikte gerçekleştirilerek sonuçlar elde edilmiştir.

VGG16 ve önerilen CNN mimarisinin birlikte kullanılmasındaki karar mekanizması görüntü bazlı olasılık değerlerini ele alarak çalışmaktadır. Eğer bir görüntü için iki modelin birbirinden bağımsız olarak verdiği karar farklıysa sınıflara ait olasılık değerleri iki modelin maksimumları alınarak yeni bir olasılık tablosu oluşturulur ve bu tabloya göre sınıf kararı verilir. Böylece, test veri kümesi ile yapılan değerlendirmede VGG16'ya göre bazı sınıflardaki doğruluk oranı düşse de bazılarında arttığı ve genel olarak ele alındığında toplam doğruluk oranının arttığı görülmüştür. VGG16 için doğruluk oranı %94,33'tür. Önerilen CNN mimarisi için ise elde edilen doğruluk oranı %84,37'dir. Bu iki modelin bir arada kullanılması ile birlikte ise %95,51 doğruluk oranı elde edilmiştir.

Elde edilen tüm sonuçlar özet olarak Çizelge 4.1'de paylaşılmıştır.

Çizelge 4.1 : Sonuç özet tablo.

Model	Test Doğruluğu(%)
VGG16	94,21
VGG19	94,19
InceptionV3	95,16
MobileNetV2	93,51
ResNet50	78,75
InceptionResNetV2	87,72
Önerilen CNN	84,30
Önerilen CNN + VGG16	95,51

Bu çalışmaya aynı zamanda referans olan ve veri kümesi kullanılan Kampouris ve çalışma arkadaşlarının yapmış olduğu çalışmada ise sadece görüntü kullanılarak %71,7 doğruluk oranı elde edilerek çalışmada önerdikleri diğer yöntemlerle de en yüksek %79,6 doğruluk oranı ile tekstil tipi sınıflandırması yapılmıştır [3].

Elde edilen sonuçlar ışığında tekstil tipinden elde edilecek standart kameradan alınmış fotoğraflar ile pamuk, yün, kot, ipek gibi sınıflandırmanın yapılabilirliği gösterilmiştir. Bu çalışma için tanıma algoritmalarının dayandığı temel nokta dokuma şekli, kalınlığı ve yönü gibi özellikler olduğundan, farklı tekstil tipi ile aynı dokuma yapılırsa eğer sistem sadece kamera ve görüntü bazlı çalışırken hatalı sınıflandırma yapacaktır. Ancak bu çalışmada kullanılan veri kümesi ve uygulama sırasında farklı tekstil ürünleri ile de yapılan kontroller ve incelemeler şu anda genel uygulama olarak kamera ile sınıflandırma yapmanın uygulanabilir ve mümkün olduğunu göstermektedir. Bu olası dezavantajın önüne geçmek için daha yüksek doğruluk oranları elde etmek ya da farklı koşulların üstesinden gelebilmek için sensör çözümüyle birlikte yapılan çalışmalar da değerlendirilebilir. Ancak bu çalışmayla birlikte standart bir kamera ve derin öğrenme yöntemleri ile tekstil tipi sınıflandırılmasının yüksek doğruluk oranlarıyla yapılabilirliği gösterilmiştir.

Genel başlık olarak malzeme tanıma konularına dahil olabilecek bu çalışma gerek amaşıır kurutma makinesi, amaşıır makinesi, ütü gibi ev aletlerine adaptasyonu, gerek tekstil tipine yönelik yapılacak bağımsız alıřmalar, endüstriyel üretim ya da sınıflandırma sistemleri gibi uygulamalara girdi oluřturabilecektir.



KAYNAKLAR

- [1] **Kim, J., Lee, D., Lee, S.** (2019). Conceptual design of autonomous emergency operation system for nuclear power plants and its prototype, *Nuclear Engineering and Technology*.
- [2] **Antsaklis, P. J., Passino, K. M., Wang, S. J.** (1991). An introduction to autonomous control system, *IEEE Control Systems Magazine*, Vol. 11, No. 4, pp. 5-13.
- [3] **Kampouris, C., Zafeiriou, S., Ghosh, A., Malassiotis, S.** (2016). Fine-Grained Material Classification Using Micro-geometry and Reflectance, *European Congerence on Computer Vision*, pp. 778-792.
- [4] **Khan, A. A., Khosravi, M., Denei, S., Maiolino, P., Kasprzak, W., Mastrogiovanni, F., Cannata, G.** (2016). A tactile-based fabric learning and classification architecture, *2016 IEEE International Conference on Information and Automation for Sustainability*, pp. 1-6.
- [5] **Denei, S., Maiolino, P., Baglini, E., Cannata, G.** (2015). On the Development of a Tactile Sensor for Fabric Manipulation and Classification for Industrial Applications, *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5081-5086.
- [6] **Yuan, W., Mo, Y., Wang, S., Adelson, E. H.** (2018). Active Clothing Material Perception Using Tactile Sensing and Deep Learning, *2018 IEEE International Conference on Robotics and Automation*, pp. 1-8.
- [7] **Wang, F., Liu, H., Sun, F., Pan, H.** (2019). Fabric Recognition Using Zero-Shot Learning, *Tsinghua Science and Technology*, Vol. 24, No. 6, pp. 645-653.
- [8] **Zhao, Z., Zhou, Y.** (2018). Clothing Fabric Automatic Recognition, *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 24-27.
- [9] **Ouyang, W., Xu, B., Yuan, X.** (2019). Fabric Defect Detection Using Activation Layer Embedded Convolutional Neural Network, *IEEE Access*, Vol. 7, pp. 70130-70140.
- [10] **Şeker, A.** (2018). Evaluation of Fabric Defect Detection Based on Transfer Learning with Pre-trained AlexNet, *2018 IEEE International Conference on Artificial Intelligence and Data Processing*, pp. 1-4.
- [11] **Bakshi, A.** (2019). *What is Deep Learning? Getting Started with Deep Learning*. Retrieved October 28, 2019, from <https://www.edureka.co/blog/what-is-deep-learning>

- [12] **Raicea, R.** (2017). *Want to know how Deep Learning works? Here's a quick guide for everyone*. Retrieved October 28, 2019, from <https://medium.com/free-code-camp/want-to-know-how-deep-learning-works-heres-a-quick-guide-for-everyone-1aedca88076>
- [13] **Goodfellow, I., Bengio, Y. & Courville, A..** (2017). *Deep Learning*. London, The MIT Press.
- [14] **Url-1** <<http://image-net.org/about-overview>>, erişim tarihi 28.10.2019.
- [15] **Rajat, R., Madhavan, A., Andrew, Y.** (2019). Large-scale deep unsupervised learning using graphic processors, *Proceedings of the 26th Annual International Conference on Machine Learning*.
- [16] **Jain, P.** (2019). *Complete Guid of Activation Functions*. Retrieved October 28, 2019, from <https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044>
- [17] **Url-2** <<https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>>, erişim tarihi 28.10.2019.
- [18] **Saha, S.** (2018). *Comprehensive Guide to Convolutional Neural Networks*. Retrieved October 28, 2019, from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [19] **Guo, T., Dong, J., Li, H., Gao, Y.** (2017). Simple convolutional neural network on image classification, *2017 IEEE 2nd International Conference on Big Data Analysis*, pp. 721-724.
- [20] **Thoma, M.** (2017). *Analysis and Optimizatio of Convolutional Neural Network Architectures* (Master Thesis). Karlsruhe Institute of Technology, Department of Computer Science, Karlsruhe.
- [21] **Url-3** <<https://keras.io/applications/>>, erişim tarihi 28.10.2019.
- [22] **Nair, V., Hinton, G. E.** (2010). Rectified Linear Units Improve Restricted Boltzmann Machines, *27th International Conference on Machine Learning*, pp. 807-814.
- [23] **Simonyan, K., Zisserman, A.** (2014). Very Deep Convolutional Networks For Large-Scale Image Recognition, *2015 ICLR*.
- [24] **Tsang, S.** (2018). *Review: AlexNet, CaffeNet — Winner of ILSVRC 2012 (Image Classification)*. Retrieved October 28, 2019, from <https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160>
- [25] **Krizhevsky, A., Sutskever, I., Hinton, G. E.** (2012). ImageNet classification with deep convolutional neural networks, *25th International Conference on Neural Information Processing Systems*, pp. 1097-1105.
- [26] **He, K., Zhang, X., Ren, S. Sun, J.** (2016). Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

- [27] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovic, A. (2015). Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.
- [28] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. (2016). Rethinking the Inception Architecture for Computer Vision, *2015 IEEE Conference on Computer Vision and Pattern Recognition*.
- [29] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *IEEE Conference on Computer Vision and Pattern Recognition*.
- [30] Szegedy, C., Vanhoucke, V., Ioffe, S. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning, *Conference on Artificial Intelligence*, pp. 4278-4284.
- [31] Url-4 <<https://keras.io/>>, erişim tarihi 28.10.2019.



ÖZGEÇMİŞ



Ad-Soyad : Gökhan GÜRGEN
Doğum Tarihi ve Yeri : 19.03.1990 / Amasya
E-posta : gokhangurgenn@gmail.com

ÖĞRENİM DURUMU:

Lisans : 2013, İstanbul Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Elektronik Mühendisliği Programı

MESLEKİ DENEYİM VE ÖDÜLLER:

- **Kasım 2016 –** : Arçelik A.Ş. Kurutma Makinesi İşletmesi
*Elektronik ve Kontrol Sistemleri Takımı,
Arge Mühendisi.*
- **Temmuz 2014 – Kasım 2016** : Arçelik A.Ş. Kurutma Makinesi İşletmesi
Kalite Güvence Mühendisi
- **Eylül 2013 – Temmuz 2014** : Elektro Otomasyon Ltd. Şti.
Gömülü Sistem Yazılım Mühendisi