

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



**KAPSÜL AĞLAR İLE YÜZ VERİLERİNİN
SINIFLANDIRILMASI**

Ayşe ÇOBAN

Yüksek Lisans Tezi

YAZILIM MÜHENDİSİĞİ ANABİLİM DALI

TEMMUZ 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Yazılım Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

KAPSÜL AĞLAR İLE YÜZ VERİLERİNİN SINIFLANDIRILMASI

Tez Yazarı
Ayşe ÇOBAN

Danışman
Doç. Dr. Fatih ÖZYURT

TEMMUZ 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Yazılım Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Başlığı: Kapsül Ağlar ile Yüz Verilerinin Sınıflandırılması
Yazarı: Ayşe ÇOBAN
İlk Teslim Tarihi: 28.07.2022
Savunma Tarihi: 04.07.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Doç. Dr. Fatih ÖZYURT Fırat Üniversitesi, Mühendislik Fakültesi	<i>İmza</i> Onayladım
Başkan:	Dr. Öğr. Üyesi Sinem AKYOL Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Harun BİNGÖL Malatya Turgut Özal Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../2022 tarihli toplantısında tescillenmiştir.

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Kapsül Ağlar ile Yüz Verilerinin Sınıflandırılması” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

04/07/2022

Ayşe ÇOBAN

ÖNSÖZ

Son yıllarda teknolojinin de gelişmesi ile birlikte yüz verilerini sınıflandırma problemi, üzerinde devamlı olarak çalışmaların yapıldığı önemli bir örüntü tanıma problemi haline gelmiştir. Yüz verilerini sınıflandırma işlemi, yüz görüntüsünü girdi olarak alarak, ilgili görüntünün sahibine ilişkin sınıflama yapılması olarak tanımlanabilir. Bu konuda evrişimli sinir ağları, tercih edilen derin öğrenme uygulamalarının başında gelmektedir. Buna karşın evrişimli sinir ağlarının da kendi içinde bazı eksiklikleri bulunmaktadır. Bu duruma çözüm olarak kapsül ağları, önerilen derin öğrenme yöntemlerindendir. Bu çalışmada yüz verilerini sınıflandırma problemi üzerine kapsül ağlar ve bazı evrişimli sinir ağı modelleri karşılaştırılmıştır.

Çalışmamı yürütürken bana destek olan danışmanım Doç. Dr. Fatih Özyurt'a sabrı, yönlendirmeleri ve teşvikleri için teşekkürlerimi sunarım. Bu tez çalışması, Fırat Üniversitesi Bilimsel Araştırma Projeleri Koordinasyonu Birimi (FÜBAP) tarafından MF.21.54 protokol numaralı proje ile desteklenmiştir.

Ayşe ÇOBAN

ELAZIĞ, 2022

İÇİNDEKİLER

	Sayfa
ÖNSÖZ	iv
İÇİNDEKİLER	v
ÖZET	vi
ABSTRACT	vii
ŞEKİLLER LİSTESİ	viii
TABLolar LİSTESİ	ix
SİMGELER VE KISALTMALAR	x
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	4
2.1. Makine Öğrenmesi İle Yüz Verilerinin Sınıflandırılması	4
2.2. Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması	5
3. KAPSÜL AĞLAR	7
3.1. Yapay Sinir Ağları	7
3.2. Evrişimli Sinir Ağları	10
3.3. Hafif Siklet Evrişimli Sinir Ağları	12
3.4. Kapsül Ağlar	15
4. MATERYAL VE METOT	20
4.1. Kapsül Ağlar ve Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması	22
4.2. Hafif Siklet Evrişimli Sinir Ağları ile Kapsül Ağların Yüz Verilerini Sınıflandırma Performanslarının Karşılaştırılması	23
4.3. Kapsül Ağların Farklı Boyutlardaki Veri Kümelerinde Sınıflandırma Durumlarının İncelenmesi	26
5. SONUÇLAR	28
5.1. Kapsül Ağlar ve Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması Çalışması Sonuçları	28
5.2. Hafif Siklet Evrişimli Sinir Ağları ile Kapsül Ağların Yüz Verilerini Sınıflandırma Performanslarının Karşılaştırılması Çalışması Sonuçları	31
5.3. Kapsül Ağların Farklı Boyutlardaki Veri Kümelerinde Sınıflandırma Durumlarının İncelenmesi Çalışması Sonuçları	32
ÖNERİLER	37
KAYNAKLAR	38
ÖZGEÇMİŞ	41

ÖZET

Kapsül Ağlar ile Yüz Verilerinin Sınıflandırılması

Ayşe ÇOBAN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisiği Anabilim Dalı
Temmuz 2022, Sayfa: xii+40

Yüz verilerini sınıflandırma işlemi, üzerinde çok sayıda çalışma yapılan derin öğrenme konularından birisidir. Bu konuda evrişimli sinir ağları, kolay uygulanabilir olmaları ve başarılı sonuçlar vermeleri nedenlerinden ötürü tercih edilen derin öğrenme uygulamalarının başında gelmektedir. Buna karşın evrişimli sinir ağlarında bulunan havuzlama katmanı verilerde bilgi kaybına neden olmaktadır. Ayrıca evrişimli sinir ağları, verideki bileşenlerin birbirine göre durumlarını göz ardı ederek eğitim işlemini gerçekleştirmektedir. Bu duruma çözüm olarak kapsül ağları önerilen derin öğrenme yöntemlerindendir.

Bu tez çalışmasında üç farklı çalışma yapılarak sonuçlar başarımlık ve verimlilik açısından incelenmiştir.

Anahtar Kelimeler: Kapsül Ağlar, Yüz Tanıma, Örüntü Tanıma.

ABSTRACT

Classification of Facial Data with Capsule Networks

Ayşe ÇOBAN

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

July 2022, Page: xii+40

The process of classifying face data is one of the deep learning topics that has been studied extensively. In this regard, convolutional neural networks are among the most preferred deep learning applications because they are easy to implement and give successful results. On the other hand, the pooling layer in convolutional neural networks causes information loss in the data. In addition, convolutional neural networks perform the training process by ignoring the relative states of the components in the data. As a solution to this situation, capsule networks are one of the recommended deep learning methods.

In this thesis study, three different studies were conducted and the results were examined in terms of performance and efficiency.

Keywords: Capsule Networks, Face Recognition, Pattern Recognition.

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1.	Bir sınıfta bulunabilecek farklı özelliklere sahip örnek görüntü verileri 1
Şekil 1.2.	Evrişimli sinir ağı katmanları 1
Şekil 1.3.	Yüz bileşenlerinin birbirine göre durumları 2
Şekil 1.4.	Kapsül ağı katmanları 3
Şekil 3.1.	YSA bileşenleri 8
Şekil 3.2.	ÇKA modeli 9
Şekil 3.3.	ESA yapısı 11
Şekil 3.4.	Maksimum havuzlama işlemi 11
Şekil 3.5.	Tam bağlantılı katman düzleştirme işlemi 12
Şekil 3.6.	Geniştirilmiş evrişim işlemi 13
Şekil 3.7.	Deforme edilebilir evrişim işlemi 13
Şekil 3.8.	Grup evrişimi 14
Şekil 3.9.	Derinlemesine ayrılabilir evrişim 14
Şekil 3.10.	Kapsül giriş ve çıkışları 16
Şekil 4.1.	Kapsül ağlar ve evrişimli sinir ağları ile yüz verilerinin sınıflandırılması çalışması veri kümesi örneği 22
Şekil 4.2.	AlexNet modeli yapısı 23
Şekil 4.3.	Hafif siklet evrişimli sinir ağları ve kapsül ağlar ile yapılan çalışmada kullanılan veri kümesi örneği 24
Şekil 4.4.	MobileNetV2 modelinin yapısı 25
Şekil 4.5.	SqueezeNet yangın modülü yapısı 25
Şekil 4.6.	EfficientNetB0 modeli yapısı 26
Şekil 4.7.	Kapsül ağların farklı boyutlarda başarımının incelenmesi için hazırlanan veri kümesi örneği 27
Şekil 5.1.	AlexNet modeli eğitim ve doğrulama grafiği 28
Şekil 5.2.	Kapsül ağlar modeli eğitim ve doğrulama grafiği 29
Şekil 5.3.	Kapsül ağlar modeli karmaşıklık matrisi 29
Şekil 5.4.	Kapsül ağlar modeli tahminleme örnekleri 30
Şekil 5.5.	MobileNet modeli eğitim ve doğrulama grafiği 31
Şekil 5.6.	Kapsül ağlar modeli eğitim ve doğrulama grafiği 32
Şekil 5.7.	Modelin ilk veri seti ile elde edilen eğitim ve doğrulama süreci grafiği 33
Şekil 5.8.	Modelin ilk veri kümesi ile elde edilen karmaşıklık matrisi 33
Şekil 5.9.	Modelin ikinci veri seti ile elde edilen eğitim ve doğrulama süreci grafiği 34
Şekil 5.10.	Modelin ikinci veri kümesi ile elde edilen karmaşıklık matrisi 34
Şekil 5.11.	Modelin üçüncü veri seti ile elde edilen eğitim ve doğrulama süreci grafiği 35
Şekil 5.12.	Modelin üçüncü veri kümesi ile elde edilen karmaşıklık matrisi 35
Şekil 5.13.	Kapsül ağlar modelinin tahminleme örnekleri 36

TABLÖLAR LİSTESİ

	Sayfa
Tablo 3.1. Dinamik yönlendirme algoritması [2]	17
Tablo 4.1. Evrişim katmanı parametreleri	20
Tablo 4.2. Özellik kapsülü katmanı parametreleri	20
Tablo 4.3. Model katmanları ve parametreleri	21
Tablo 5.1. Evrişimli sinir ağı modelleri sonuçları	28
Tablo 5.2. Modellerin eğitim işlemi tamamlama süreleri	30
Tablo 5.3. Hafif siklet evrişimli sinir ağı modelleri sonuçları	31
Tablo 5.4. Modellerin eğitim işlemi tamamlama süreleri	32
Tablo 5.5. Kapsül ağı modelinin eğitim işlemleri tamamlama süreleri	36

SİMGELER VE KISALTMALAR

Simgeler

b	: Bias değeri
b_{ij}	: Benzerlik katsayısı
c_{ij}	: Yönlendirme ağırlıkları
exp	: Üstel fonksiyon
f	: Aktivasyon fonksiyonu
L_c	: Kayıp değeri
max	: Maksimum fonksiyonu
s_j	: Kapsül ağırlıklı toplam değeri
T_c	: Sınıfın doğruluk değeri
u_i	: Kapsülün girişleri
$\hat{u}_{i j}$	: Tahmin vektörleri
v_c	: Elde edilen tahmin değeri
v_j	: Kapsül çıkışları
w_i	: Ağırlık değerleri
W_{ij}	: Ağırlık matrisi
x_i	: Giriş değerleri
y_j	: Çıktı değeri
$\sum$	: Toplama fonksiyonu

Kısaltmalar

ÇKA	: Çok katmanlı ağlar
DVM	: Destek vektör makineleri
ESA	: Evrişimli sinir ağları
K-EYK	: K en yakın komşu
ReLU	: Rectified linear unit
YSA	: Yapay sinir ağları

1. GİRİŞ

Yüz verilerini sınıflandırma işlemi; bir grup insanın yüz görüntülerinden oluşan verilerin belirlenen sınıflara göre sınıflandırılmasıdır. Yüz verilerini sınıflandırma işlemi örüntü tanımanın ilgilendiği önemli problemlerden birisidir. Yüz verileri günümüzde yoğun olarak evrişimli sinir ağları kullanılarak sınıflandırılmaktadır.

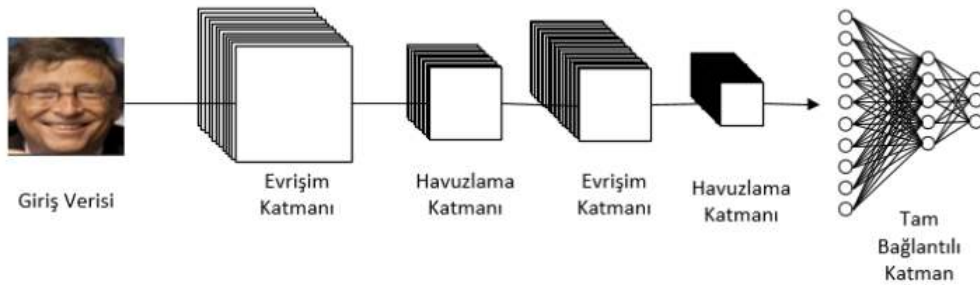
ESA'ların yüz verilerini sınıflandırma işleminde başarımları yüksektir. Fakat bu başarımla ilgili verilerin farklı perspektiflerden örneklerinin alınması durumunda düşmektedir. Bu sorunun başlıca nedeni evrişimli sinir ağlarında kullanılan havuzlama katmanının verilerde bilgi kaybına neden olmasıdır. Bunun bir sonucu olarak ESA farklı açı, poz, durum gibi özelliklere sahip örneklerin yeterli sayıda olmadığı veri kümelerinde başarılı sınıflandırmalar yapamayabilir.



Şekil 1.1. Bir sınıfta bulunabilecek farklı özelliklere sahip örnek görüntü verileri

Şekil 1.1'de bir veri kümesinde bulunabilecek belirli bir sınıfa ait farklı açılardan, farklı durumlardan alınan görüntü örnekleri verilmiştir. Her farklı örnek için, yeterli sayıda benzer örnek bulunmaması ESA'nın başarımlarını etkilemektedir. Bu sorunların çözümü olarak kapsül ağları tercih edilen yöntemlerdendir. Kapsül ağlar verileri skaler olarak değil vektörel olarak alırlar. Vektörün parametreleri verilerin durumu hakkında bilgiler içerir. Dolayısıyla kapsül ağlar; verilerdeki bileşenlerin birbirine göre konum, açı, yönelim gibi özelliklerini göz önüne alarak eğitim işlemi yapan ağlardır.

ESA'lar temel olarak Şekil 1.2'de gösterildiği gibi; evrişim katmanı ardından havuzlama katmanı ve tam bağlantılı katman dediğimiz katmanlardan oluşur [1].



Şekil 1.2. Evrişimli sinir ağı katmanları

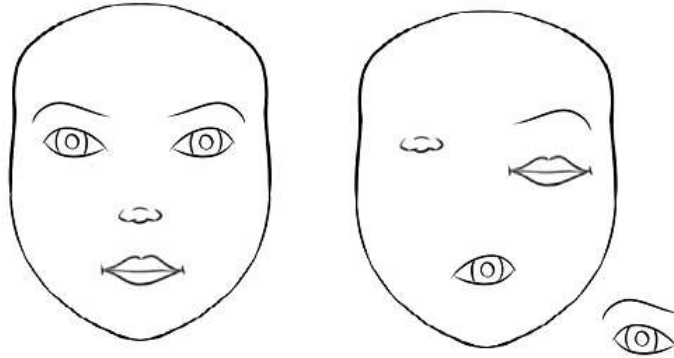
Evrişim katmanı giriş olarak alınan görüntülerin belirlenen boyutlarda ve sayılarda filtrelerden geçirildiği katmandır. Bu katman sayesinde görüntünün belirli sayıda filtreden geçirilmiş formları elde edilir. Bu katman bir veya birden fazla kez ard arda uygulanabilir.

ESA'nın mimarisinde evrişim katmanından sonra havuzlama katmanı gelmektedir. Havuzlama katmanında evrişim katmanından alınan görüntülere farklı çeşitleri bulunan havuzlama işlemlerinden (maksimum, minimum, ortalama havuzlama) seçilen bir havuzlama yöntemi uygulanır. Burada amaç görüntüde istenmeyen özniteliklerin boyut olarak sistemi verimsiz hale getirmemesi için en aza indirilmesidir. Bu katman sonucunda havuzlama katmanının parametrelerine göre boyutları azaltılan ve istenilen özniteliklere sahip görüntüler elde edilir. Bu katman da modelin yapısına göre bir veya birden fazla kez tekrarlanabilir.

Son olarak tam bağlantılı katman gelmektedir. Bu katman basit anlamda bir yapay sinir ağıdır ve elde edilen görüntü matrisleri bir düzleştirme işleminden geçerek bu katmana iletilir. Bu katman sonunda bazı doğrusal olmayan aktivasyon fonksiyonları aracılığıyla sınıflandırma işlemi gerçekleştirilir.

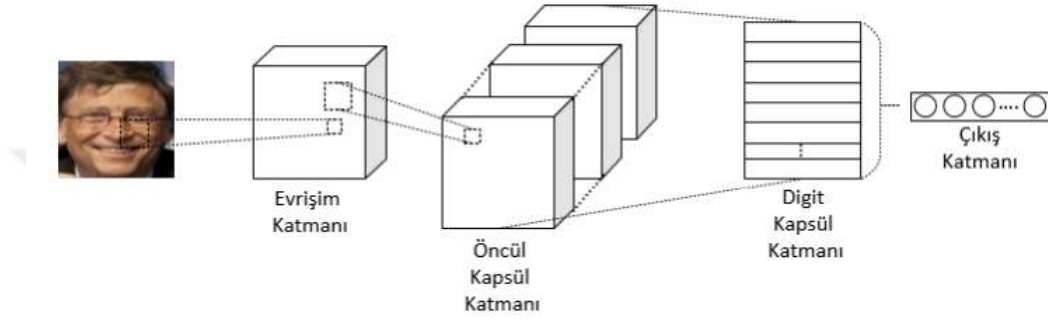
ESA, yüz verilerinin sınıflandırılması probleminde yüksek başarımlar elde etmektedir. Fakat söz konusu görüntülere başka perspektiflerden bakıldığında ESA aynı başarımları gösterememektedir. ESA'da, evrişim işlemlerinden sonra uygulanan havuzlama katmanları görüntüde bilgi kaybına neden olmaktadır [2]. Bu kayıp, yüz verilerinin sınıflandırılmasında, sınıflandırma işlemine etkisi olabilecek bazı özniteliklerin de kaybının olabileceği anlamına gelmektedir.

Ayrıca ESA görüntüyü anlamlandıran nesnelerin birbiriyle olan durumlarını göz önüne almadan eğitilmektedir. Örneğin görüntüde yüzü oluşturabilmesi için gözler, burun, ağız gibi bileşenlerin belirli bir konumu vardır. Bunların birbirlerine göre Şekil 1.3'de gösterildiği gibi yanlış konumlarda bulunması halinde söz konusu görüntü bir yüzü tanımlayamaz. Bu durumda bileşenlerin birbirine göre konum, durum, yönelim gibi özelliklerinin göz ardı edilmemesi gerekmektedir.



Şekil 1.3. Yüz bileşenlerinin birbirine göre durumları

Bu problemlere bir çözüm olarak 2017 yılında Geoffrey Hinton, Sara Sabour ve diğerleri tarafından kapsül ağlar ve dinamik yönlendirme algoritması önerilmiştir [2]. Kapsül ağlar, mimarisinde evrişim katmanları kullanılması yönüyle ESA'ya benzemektedir. Kapsül ağının ilk aşamasında kullanılan evrişim katmanı ESA'dakiyle aynıdır. Fakat bu evrişim katmanından sonra gelen evrişim katmanında ve diğer katmanlarda dinamik yönlendirme algoritması uygulanmaktadır. ESA'lardaki nöronların yerine kapsül ağlarda kapsüller kullanılır. Kapsül ağı katmanları temel olarak Şekil 1.4'de gösterildiği gibidir.



Şekil 1.4. Kapsül ağı katmanları

Kapsül ağların ilk katmanı ESA'da olduğu gibi evrişim katmanıdır. Evrişim işleminden geçen veriler öncül kapsül katmanı olarak adlandırılan ikinci katmana iletilir. Öncül kapsül katmanında veriler yeniden boyutlandırılır ve aktivasyon işleminden geçirilir. Bu işlemler sonunda verilerden DigitCaps adı verilen vektörler elde edilmektedir. DigitCaps vektörlerinin sayısı sınıflandırma işlemi için belirlenen sınıf sayısı kadardır.

Kapsül ağları giriş olarak aldığı verileri özelliklerine göre parametrelendirerek vektörel olarak ele alır, dinamik yönlendirme algoritması sayesinde de özelliklerine göre birbirine bağlı kapsüllerle eğitim gerçekleştirir. Kapsül ağların bu özellikleri sayesinde, veri kümesindeki görüntülerin farklı perspektiflerden alındıkları durumlarda dahi başarılı olarak sınıflanabilmeleri beklenmektedir.

Kapsül ağlar dolayısıyla evrişimli sinir ağlarına göre veriyi oluşturan parçalardan daha fazla bilgi alır. Dinamik yönlendirme algoritması ile de girişleri benzer kapsüllerin birbiri ile bağlantılı aktive olmalarını sağlar. Bu durumun bir sonucu olarak kapsül ağların küçük veri kümelerinde sınıflandırma başarımının, ESA'lara göre daha az değişmesi beklenmektedir.

2. LİTERATÜR TARAMASI

Yüz verilerinin sınıflandırılması problemi 1990'lı yılların başından itibaren üzerinde çalışmalar yapılan bir konudur. Bu konu hakkında yapılan literatür taraması; makine öğrenmesi ile yüz verilerinin sınıflandırılması ve evrimsel sinir ağları ile yüz verilerinin sınıflandırılması olarak iki grupta incelenmiştir.

2.1. Makine Öğrenmesi İle Yüz Verilerinin Sınıflandırılması

Makine öğrenmesi; standart algoritmalar ile çözümü olmayan problemlerin söz konusu olduğu durumlarda kullanılan bir yaklaşımdır [3]. Makine öğrenmesi yöntemleri ile görüntü verilerini sınıflandırma işlemi yapılmak istendiğinde, bu verilerin bir ön işlemden geçmesi gerekir. Bu ön işleme süreci verilerden, istenmeyen bazı özellikleri kaldırmak ve bu verilerin istenilen öz niteliklerini daha belirgin hale getirmek amacıyla uygulanır. Ön işleme sürecinde eşikleme, gürültü kaldırma, döndürme, merkeze alma gibi teknikler kullanılabilir. Verilerden doğru öz niteliklerin çıkarılması için ön işleme aşaması oldukça önemlidir.

Ön işleme aşamasından sonra bu verilerden öz niteliklerin çıkarılması aşaması gelmektedir. Öz niteliklerin çıkarılması işlemi, bir sınıfta bulunup diğer sınıflarda bulunmayan bazı ayırt edici bilgilerin belirlenip çeşitli yöntemlerle elde edilmesidir. Görüntü verileri için kullanılan bir çok öz nitelik çıkarma yöntemi bulunmaktadır. Bunlardan bazıları; Temel Bileşen Analizi [4], Bağımsız Bileşen Analizi [5], Doğrusal Ayırıcılık Analizi [6], Gabor dalgacık dönüşümü [7], Ölçekten Bağımsız Öz nitelik Dönüşümü [8] olarak sıralanabilir. Farklı perspektiflerde ve farklı aydınlatmalarda bulunan yüz görüntü verilerini ayırt etmek oldukça zor olan sınıflandırma problemleridir. Bu gibi problemler için Temel Bileşen Analizi, Doğrusal Ayırıcılık Analizi gibi yöntemler önerilmektedir [6].

Özellik çıkarma işleminden geçen verilerin sınıflandırılması gerekmektedir. Sınıflandırma işlemlerinde sıkça kullanılan K-EYK, karar ağaçları, yalın Bayes, DVM, YSA gibi makine öğrenmesi teknikleri bulunmaktadır. Yüz verilerinin sınıflandırılması işlemi için bu makine öğrenmesi teknikleri ile çeşitli uygulamalar yapılmıştır.

Bu konuda ilk başarılı çalışmayı 1991'de Alex Pentland ve Matthew Turk istatistiksel temel bileşen analizi kullanarak özyüzler (Eigenfaces) yöntemi ile yayımlamışlardır [9].

1998 yılında P. J. Philips, DVM ile yüz tanıma üzerine bir çalışma yaparak % 77-78 oranında başarımla bir uygulama geliştirmiştir [10].

2000 yılında Guo ve diğerleri yüz tanıma ve sınıflandırma işlemi için DVM tabanlı bir model uygulamışlardır. Bu uygulamada 40 kişiden alınan toplamda 400 resimden oluşan Cambridge ORL veri seti kullanılmıştır. Bu uygulama %3-6 arasında bir hata oranıyla gerçekleştirilmiştir [11].

Literatüre bakıldığında yüz tanıma ve yüz verilerinin sınıflandırılmasıyla ilgili çok sayıda çalışmalar yapılmıştır. 1990'lı yılların sonlarına gelindiğinde ise makine

öğrenmesinin bir alt dalı olan derin öğrenme ile çalışmalar yapılmış ve derin öğrenmenin başlıca yöntemlerinden olan evrişimli sinir ağları görüntü verilerini sınıflandırma işleminde tercih edilen yöntemlerden olmaya başlamıştır.

2.2. Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması

ESA, günümüzde görüntü sınıflandırma problemi için sıkça tercih edilen derin öğrenme algoritmalarındandır. ESA'da görüntü sınıflandırma işlemi yapılırken görüntü verileri direkt olarak modele verilebilmektedir. Bu açıdan uygulanması kolay ve başarımı yüksek algoritmalarlardır.

ESA'nın ilk başarılı modeli 1998 yılında Yann LeCun ve diğerleri tarafından yayımlanan LeNet modelidir [12]. Bu çalışmadan sonra 2012 yılında Geoffrey Hinton, Ilya Sutskever ve Alex Krizhevsky tarafından AlexNet isimli 60 milyon parametreye sahip bir evrişimli sinir ağı modeli geliştirilmiştir [13].

Yüz verileri üzerine yapılan çalışmalara bakıldığında ise literatürde ilk olarak 1997 yılında Lawrence ve diğerleri çalışmalarında ESA kullanarak, bir yüz tanıma uygulaması gerçekleştirmişlerdir [14].

2014 yılında Taigman ve diğerleri tarafından Facebook araştırma laboratuvarında DeepFace adlı ilk yüksek başarılı model geliştirilmiştir [15]. Bu model, 4000 farklı kişiden alınan 4.4 milyon etiketli yüz görüntüsünden oluşan bir veri seti ile eğitilmiştir. DeepFace %97.35 başarımla sağlayarak o zamana kadar olan en yüksek başarıma sahip model olarak kabul görmüştür [16].

Yine 2014 yılında WebFace isimli 10 evrişim, 5 havuzlama ve 2 tam bağlantılı katmandan oluşan bir yüz tanıma uygulaması geliştirilmiştir. Bu çalışma LWF veri seti ile %97.73 başarımla göstermiştir [17].

2015 yılında Parkhi, Vedaldi ve Zisserman tarafından, Derin Yüz Tanıma isimli bir uygulama önerilmiştir. Derin yüz tanıma sistemi ise LFW veri setinde %98.95 gibi bir başarımla elde etmiştir [18].

Standart ESA'nın kullanımının maliyetli bir yol olması düşüncesinin üzerine daha az maliyetle başarılı sınıflandırma yapabilecek hafif siklet evrişimli sinir ağları üzerine de birçok çalışma yapılmıştır. Yapılan çalışmalardan bir tanesi 2016 yılında yayınlanan, mimarisinde yangın modülü olarak adlandırılan bir yapı bulunduran SqueezeNet algoritmasıdır [19].

2017 yılında ise derinlemesine ayrılabilir evrişimi benimseyen MobileNet olarak adlandırılan hafif siklet evrişimli sinir ağı yayımlanmıştır [20]. Yapılan bazı iyileştirmeler sonucu bu ağı yayımlanan ikinci versiyonu da bulunmaktadır [21].

2019 yılında modelde ölçeklendirilmenin yeniden düşünülmesi yaklaşımıyla doğan EfficientNet algoritması da başarılı hafif siklet evrişimli sinir ağlarına bir örnektir. Yapılan çalışmada söz konusu ağ, CIFAR-100 veri setinde %91.7 başarımla elde etmiştir [22].

Hafif siklet evrişimli sinir ağları ile yüz verileri üzerine yapılan çalışmalara da bakılacak olunursa;

2018 yılında yüz görüntüleri ile üç farklı hafif siklet evrişimli sinir ağının karşılaştırıldığı yaş ve cinsiyet sınıflandırması üzerine bir çalışma yapılmıştır [23].

2019 yılında yayımlanan bir çalışmada ise yüksek boyutlu yüz veri setleri üzerinde yüksek başarımlar elde edilmiştir [24].

2021 yılında hafif siklet evrişimli sinir ağları kullanılarak yapılan, yüz verileri ile cinsiyet sınıflandırılmasını konu alan çalışma CMU Multi-PIE veri seti üzerinde %94 ile %96 arası sınıflandırma başarımları elde etmiştir [25].

Yüksek başarımları nedeniyle tercih sebebi olan ESA'nın sınıflandırdıkları verilerin farklı açılardan örnekleri ile sınıflandırma işlemi yapılmak istenildiğinde ESA'ların aynı başarımları elde edemediği gözlemlenmiştir.

2017 yılında Geoffrey Hinton, Sara Sabour ve diğerleri, ESA'nın eksik yönlerine değinerek, bir alternatif olarak kapsül ağlar ve dinamik yönlendirme algoritmasını önermişlerdir [2]. Kapsül ağlar ile yapılan bu çalışmada MultiMNIST ve CIFAR10 veri setleri üzerinde yüksek başarımlı çalışmalar gerçekleştirmiştir. Bu çalışmadan sonra kapsül ağları hakkında çok sayıda uygulama yapılmıştır [26]. Yapılan uygulamalarda kapsül ağlarının ESA'lara göre daha az sayıda verilerle de daha iyi başarımlar elde edebileceğine dikkat çekilmiştir [27].

Bu çalışmada ise kapsül ağlarla ilgili üç ayrı çalışma yapılmıştır. Çalışmada kapsül ağların küçük boyutlu veriler üzerinde olan başarımlar ve verimlilikleri araştırılmıştır. Aynı zamanda yine küçük boyutlu veriler üzerinde, çeşitli ESA'lar ve ESA'ların geliştirilmesiyle ortaya çıkan hafif siklet evrişimli sinir ağlarının kapsül ağları ile karşılaştırılması yapılmıştır. Bu çalışmanın kapsül ağlar hakkında yapılabilecek daha farklı çalışmalar için bir örnek olması hedeflenmektedir.

3. KAPSÜL AĞLAR

Kapsül ağlarının temelleri ESA'ya dayanmaktadır. Kapsül ağlar, ESA gibi evrişim katmanına sahip olmalarına rağmen havuzlama katmanını kabul etmeyen, havuzlama katmanı yerine dinamik yönlendirme algoritmasını öneren bir yöntemdir.

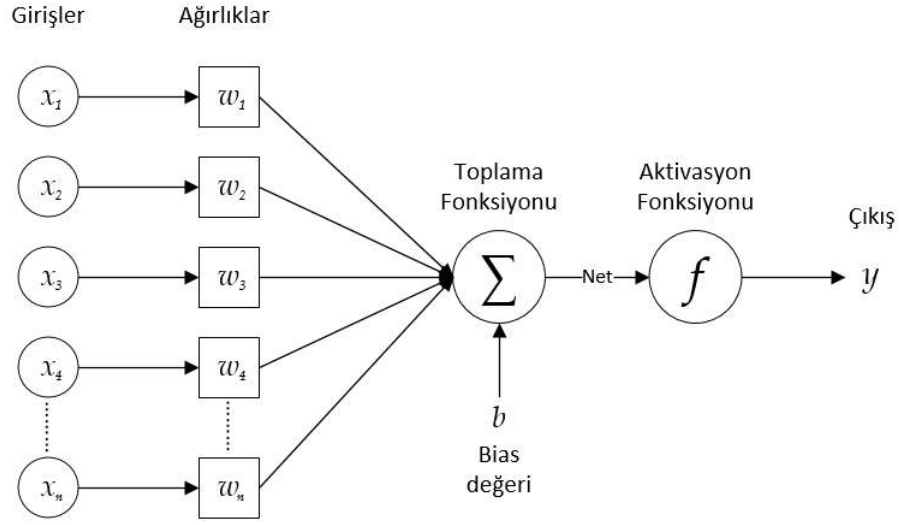
ESA'nın ise temelleri YSA'ya dayanmaktadır. En sade biçimde ifade edilecek olunursa ESA, YSA'nın katmanlarının daha da derinleştirilmesi ile ortaya çıkmış bir derin öğrenme algoritmasıdır. Bu bölümde; YSA, ESA ve kapsül ağlar algoritmalarının yapı ve özelliklerine değinilmiştir.

3.1. Yapay Sinir Ağları

Günümüzde kullanılan bilgisayarlar, çok çeşitli görevleri insanlardan daha hızlı ve güvenilir bir şekilde gerçekleştirebilmektedir. Bununla birlikte hiçbir örüntü tanıma sistemi, insanın çeşitli biçimler ve yönlerdeki nesneleri ayırt etme kapasitesi ile rekabet edememektedir. İnsanoğlu kesin olmayan veya karmaşık problemleri, deneyimlerinden edindiği bilgilerle, bilgisayarlara göre daha başarılı bir şekilde çözebilmektedir. Bu problemleri çözerken beynimiz bu hedeflere birbirine bağlı nöron adı verilen binlerce milyon basit hücre aracılığıyla ulaşır [28]. İnsan beyninin verilerden anlamlı bilgiler çıkarması, verileri sınıflandırması ve verilerden yola çıkarak tahminlemeler yapması yeteneği YSA için ilham kaynağı olmuştur.

İnsan beyninde bulunan nöronlar synapsler aracılığıyla iletişim kurarlar ve işledikleri bilgileri diğer nöronlara gönderirler. Benzer şekilde YSA'da bulunan, insan beynindeki nöronlardan esinlenerek oluşturulmuş olan nöron veya düğüm olarak adlandırılan basit işlem elemanları, dışarıdan gelen bilgileri bir toplama fonksiyonu ile toplar ve aktivasyon fonksiyonundan geçirerek çıktıyı üretilip ağırlık bağlantılarının üzerinden diğer nöronlara gönderir [29]. Her nöron, her biri ağırlık olarak bilinen ilişkili bir sayısal değere sahip bağlantılarla diğer nöronlara bağlanır. Bu ağırlıklar, birbirine bağlı nöronlar arasındaki etkinin gücünü belirler [30].

YSA, bir veya birden çok katmandan oluşabilir. Bir yapay sinir ağı sadece girdi ve çıktı katmanlarından oluşuyorsa tek katmanlı bir ağıdır. Tek katmanlı bir yapay sinir ağının bileşenleri Şekil 3.1'de gösterilmiştir.



Şekil 3.1. YSA bileşenleri

Girişler

Şekil 3.1’de $x_1, x_2, x_3, \dots, x_n$ değişkenleriyle gösterilen değerler yapay sinir ağına gönderilen giriş verilerini modelize etmektedir.

Ağırlıklar

Şekil 3.1’de $w_1, w_2, w_3, \dots, w_n$ değişkenleriyle gösterilen değerler yapay sinir ağındaki nöronların bağlantılarını ifade etmektedir. Her bir bağlantının bir ağırlık değeri mevcuttur. Ağırlık değerleri bağlantının yapıldığı nöronun etki gücünü ifade etmektedir.

Toplama fonksiyonu

Şekil 3.1’de toplam sembolüyle gösterilmiş olan toplama fonksiyonu her bir nöron ile nöronun bağlı olduğu ağırlıkların, çarpılıp toplanmasıyla elde edilir. Toplam fonksiyonunun ağı uygulanmasıyla ağı Net değeri hesaplanmış olur. YSA’larda sıkça kullanılan toplam fonksiyonu Denklem 3.1’de gösterildiği gibidir.

$$Net = \sum_{i=1}^n x_i w_i \quad (3.1)$$

Aktivasyon fonksiyonu

Şekil 3.1’de f sembolüyle gösterilen aktivasyon fonksiyonu elde edilen Net değerini skaler halden doğrusal olmayan hale dönüştürmek için kullanılır. Doğrusal olmayan aktivasyon fonksiyonlarının kullanılması Net değerlerinin türevlenebilir olmasını ve bu sayede YSA’ların karmaşık ve çok farklı problemlere uygulanabilmesini sağlar. YSA’larda sıkça kullanılan aktivasyon fonksiyonlarından Sigmoid fonksiyonu Denklem 3.2’de ve hiperbolik tanjant aktivasyon fonksiyonu ise Denklem 3.3’de gösterilmiştir.

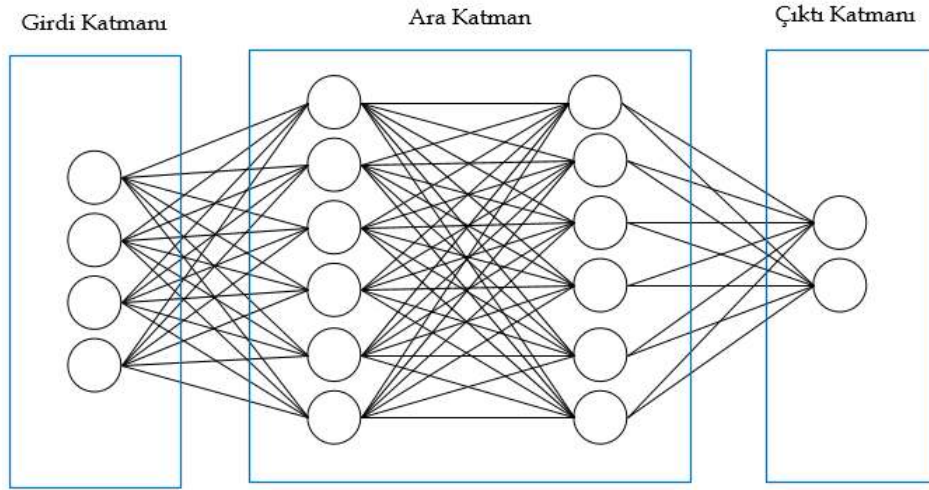
$$f(Net) = \frac{1}{1 + e^{(-Net)}} \quad (3.2)$$

$$f(Net) = \frac{e^{(Net)} - e^{(-Net)}}{e^{(Net)} + e^{(-Net)}} \quad (3.3)$$

Eşik değeri

Şekil 3.1’de b sembolüyle gösterilen değer, eşik değerini ifade etmektedir. Eşik değeri aktivasyon fonksiyonundan elde edilen eğrinin kaydırılmasını sağlayan sabit bir değeri ifade eder.

Sadece giriş ve çıkış katmanı olan tek katmanlı ağlar, karmaşık işlemleri hesaplama yeteneğinden yoksundurlar. Karmaşık işlevleri hesaplamak için en az bir ara katman olmalıdır [31]. Tek katmanlı YSA’nın karmaşık ve doğrusal olmayan problemlerde yeterli başarımlar gösterememesinin üzerine çok katmanlı YSA (ÇKA) geliştirilmiştir. Bir ÇKA modeli Şekil 3.2’de gösterildiği gibidir.



Şekil 3.2. ÇKA modeli

Girdi katmanı

Şekil 3.2’de gösterilen girdi katmanı sinir ağına sunulmak istenen bilgilerin (verilerin) bulunduğu katmandır.

Ara katman

Şekil 3.2’de gösterilen ara katman girdi katmanından aldığı bilgileri işleyerek çıktı katmanına ileten katmandır. Ara katman bir veya daha fazla katmandan oluşabilir.

Çıktı katmanı

Şekil 3.2’de gösterilen çıktı katmanı, ara katmandan aldığı işlenmiş bilgileri sınıflandırarak çıktı olarak sunma işlemini gerçekleştiren katmandır.

ÇKA’dan alınan çıktı ile beklenen çıktı arasındaki fark ÇKA’nın yaptığı tahminleme işleminde ne kadar hata yaptığını ifade eden değerdir. Bu hata değerini en aza indirmek için ÇKA’da ileri doğru hesaplama ve geriye doğru hesaplama işlemleri gerçekleştirilir. İleri doğru hesaplama safhasında, ağırlık çıktısı ve hata değeri hesaplanır. Geriye doğru hesaplamada ise hesaplanan hata değerinin en aza indirilmesi amacıyla katmanlar arası ağırlık değerleri ve eşik değerleri güncellenir. İleri yayılımı elde edilen çıktının hatası, geri yayılım algoritması kullanarak, minimize edilir ve bu sayede ağırlıkların doğruluğunun artırılması amaçlanır.

İleri doğru hesaplama

Girdi katmanından ÇKA’ya $(x_1, x_2, x_3, \dots, x_n)$ girdi değerlerinin sunulmasıyla işlem başlar. Girdi değerleri ve $(w_{1j}, w_{2j}, w_{3j}, \dots, w_{nj})$ ağırlıklara toplam fonksiyonu uygulanarak Denklem 3.4’deki gibi Net değerleri hesaplanır. Hesaplanan bu Net değerleri ara katman nöronlarına iletilir.

$$Net_j = \sum_{i=1}^n x_i w_{ij} \quad (3.4)$$

Hesaplanan net değerlerine Sigmoid aktivasyon fonksiyonunun uygulandığını ve eşik değerinin de b olduğunu kabul edersek çıktı değeri y , Denklem 3.5’de gösterildiği gibi hesaplanır.

$$y_j = \frac{1}{1 + e^{-(Net_j + b_j)}} \quad (3.5)$$

Bu değerler ara katmandaki her bir nöron için hesaplanıp, çıktı değerleri elde edilir. Bu işlemlerin yapılması ileri yönlü hesaplama işlemini gerçekleştirir.

Geriye doğru hesaplama

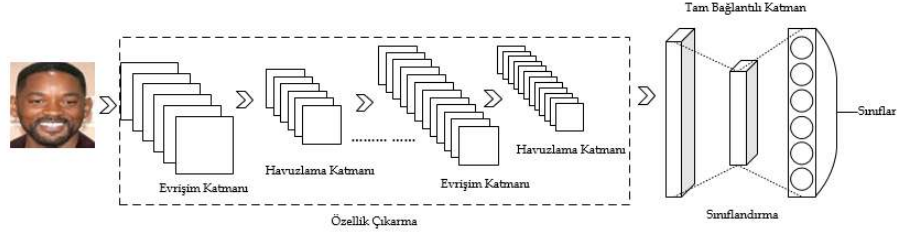
Geri yayımlı sinir ağlarında, en az bir nöron, sonraki katmanlardaki nöronlarca beslenir. Veri akışının sadece ileriye doğru olarak değil, geriye doğru da olabileceği YSA’dır [32]. Burada ileri doğru hesaplama ile elde edilen çıkış değeri, beklenen değer ile karşılaştırılır. Beklenen çıktı değeri ile üretilen çıktı değeri arasındaki fark, hata olarak kabul edilir. Geri yayılım algoritması her seferinde, hatanın türevini alarak ağırlıkları günceller. Böylece daha yüksek başarımlar için en optimize ağırlıkların elde edilmesi sağlanır.

3.2. Evrişimli Sinir Ağları

ESA; çok katmanlı algılayıcıların bir türüdür. ESA’da tıpkı YSA gibi zeki canlıların görme tanıma ve sınıflandırma gibi özelliklerinden esinlenilerek geliştirilmiştir. Son yıllarda

görüntü işlemede en iyi sonuçları elde eden ESA, ses işleme, doğal dil işleme, sinyal işleme gibi birçok alanda sıkça kullanılmaktadır [33].

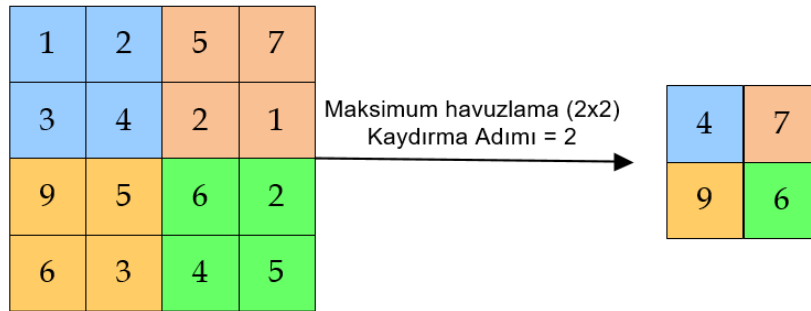
ESA; bir veya birden fazla art arda evrişim ve havuzlama katmanlarından oluşur. Evrişim ve havuzlama katmanları, görüntüden özellik çıkarımı işlemi yapılan katmanlardır. Evrişim ve havuzlama katmanlarından sonra tam bağlantılı katman gelmektedir. Tam bağlantılı katman ise sınıflandırma işleminin yapıldığı katmandır. ESA'nın yapısı Şekil 3.3'de gösterildiği gibidir.



Şekil 3.3. ESA yapısı

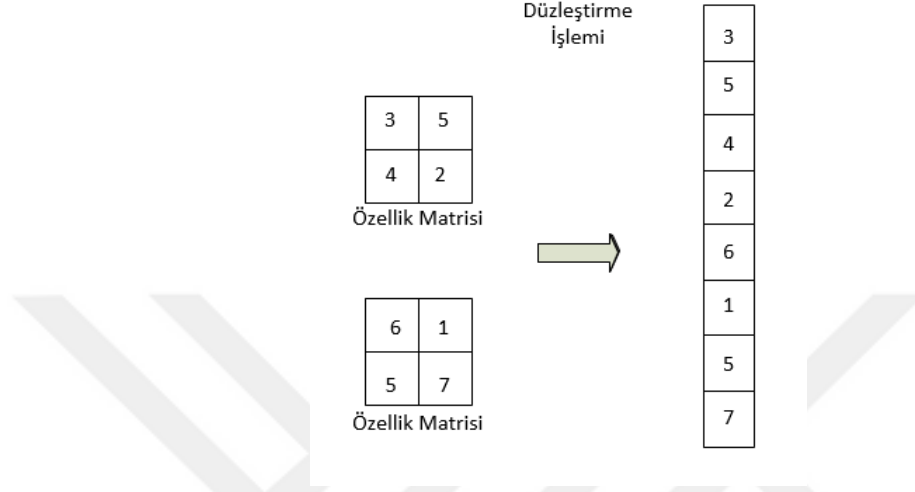
ESA'da evrişim katmanına gelen görüntüye filtreler uygulanır. Örneğin iki boyutlu bir görüntü için evrişim işleminde uygulanacak filtrenin x ve y eksenine göre simetriği alınır. Elde edilen filtre değerleri ile görüntü içindeki filtreye karşılık gelen değerler çarpılır ve çıktı, ilgili matrisin elemanı olarak yazılır [34]. Filtre, tüm resmi belirlenen kaydırma adımına göre tarar. Bu işlem her filtre için yapılır. Elde edilen matris görüntünün farklı filtrelerden geçirilmiş formlarını oluşturur.

Evrişim katmanından sonra havuzlama katmanı gelir. Bu aşamada amaç görüntünün boyutlarını azaltarak belirgin olan öznitelikleri ortaya çıkarmaktır. Havuzlama işleminin maksimum havuzlama, minimum havuzlama, ortalama havuzlama gibi çeşitleri bulunmaktadır. Bunlardan en çok tercih edileni maksimum havuzlama yöntemidir. Havuzlama işlemi için genel olarak 2x2 boyutu seçilir [35]. Maksimum havuzlama işleminin bir örneği Şekil 3.4'de gösterildiği gibidir.



Şekil 3.4. Maksimum havuzlama işlemi

Havuzlama katmanından sonra tam bağlantılı katman gelmektedir. Bu katman basit anlamda bir yapay sinir ağıdır. Evrişim ve havuzlama katmanlarında işlemlerden geçmiş olan görüntü matris formundadır. Söz konusu matrise düzleştirme işlemi uygulanır ve bir boyutlu dizi halini alan görüntü bir özellik vektörünü oluşturur. Tam bağlantılı katmandaki düzleştirme işlemi Şekil 3.5’de gösterildiği gibidir.



Şekil 3.5. Tam bağlantılı katman düzleştirme işlemi

Aktivasyon fonksiyonu

ESA’da gerek evrişim katmanları çıkışında gerekse de tam bağlantılı katmanların çıkışında bazı aktivasyon fonksiyonları kullanılır. Evrişim katmanlarının çıkışında genel olarak ReLU aktivasyon fonksiyonu tercih edilir. Bu fonksiyon negatif girdiler için sıfır değerini almaktadır. Tam bağlantılı katmanın çıkışında ise tekli sınıflandırma işlemi yapılıyor ise sigmoid aktivasyon fonksiyonu, çoklu sınıflandırma işlemi yapılıyor ise softmax aktivasyon fonksiyonu gibi fonksiyonlar genel olarak tercih edilir.

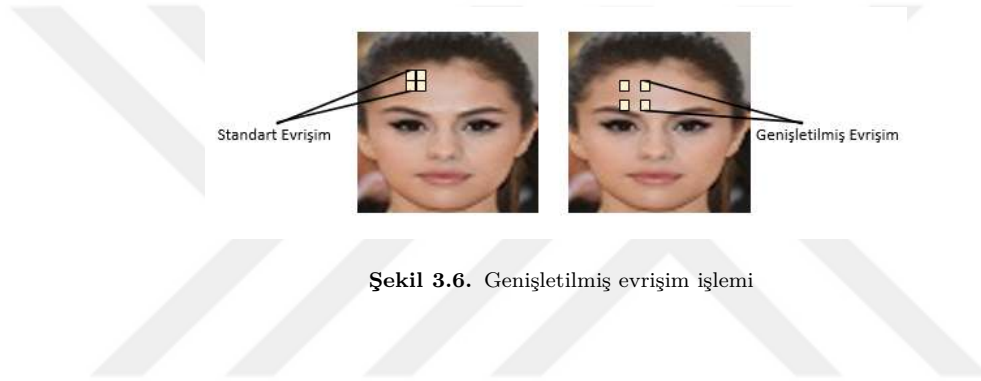
3.3. Hafif Siklet Evrişimli Sinir Ağları

Son dönemlerde görsel verileri sınıflandırma üzerine çok sayıda çalışma yapılmaktadır. Artan veriler ve sınıflandırma ihtiyacı yeni sınıflandırma algoritmalarının doğmasına zemin hazırlamıştır. Görsel verileri sınıflandırma problemlerinde ESA en çok tercih edilen derin öğrenme algoritmalarındandır. ESA’da yüksek performanslı modeller geliştirilmeye devam etmektedir. Ağın performansının iyileştirilmesi istenirken parametre sayısı da artmakta ve daha derin ağlar ortaya çıkmaktadır. ESA’ların yüksek sayıda parametre barındırmaları sınıflandırma işleminin maliyetini arttırmaktadır. Bu durum sınıflandırma maliyetini arttırdığı için bir problem olarak ortaya çıkmıştır. Sınıflandırma işlemi, düşük işlem gücüne sahip cihazlarda da gerçekleştirilmek istenmektedir. Bu soruna bir çözüm olarak parametrelerin azaltıldığı, daha az maliyetle gerçekleştirilebilen çeşitli hafif siklet evrişimli sinir ağı modelleri geliştirilmiştir [36].

ESA'nın sınıflandırma işlemlerini yüksek maliyetli olarak gerçekleştirmesinin en önemli sebeplerinden biri evrişim katmanlarının derinliğidir. Bu sorun hem özellik çıkarımını başarılı bir şekilde gerçekleştirecek hem de derin evrişimler kadar maliyetli sınıflandırma yapmayacak bazı evrişim yöntemlerinin geliştirilmesine yol açmıştır. Farklı stratejiler ile geliştirilen evrişim işlemlerinden bazıları aşağıda verilmiştir.

Genişletilmiş evrişim işlemi

Standart evrişim işlemi sırasında bir alıcı alan aracılığıyla görüntü verisinden bilgiler alınır. Alıcı alan pikseller üzerinde gezerek veriden bilgileri almaktadır. Genişletilmiş evrişimde ise bu alıcı alan daha farklıdır. Alıcı alan normal karesel bir alan olmaktan öte parçalanmış bir alandır. Veriden bilgiler alınırken bu parçalanmış alan bilgileri alır. Bu da alıcı alanı standart alana göre daha geniş kılmaktadır [37]. Genişletilmiş evrişim işlemine bir örnek Şekil 3.6'da verildiği gibidir.

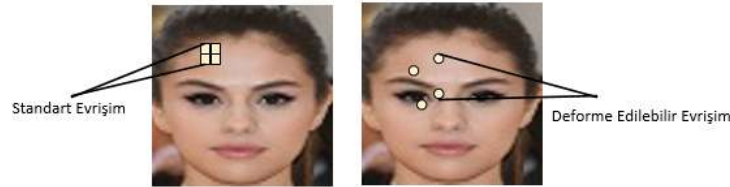


Şekil 3.6. Genişletilmiş evrişim işlemi

Genişletilmiş evrişim işleminde parametre ile genişletme miktarı ayarlanır. Bu işlem ile evrişim işleminin maliyeti azaltılabilmektedir.

Deforme edilebilir evrişim işlemi

Evrişim işlemlerinde genel olarak kare, dikdörtgen gibi düzgün şekillerle veriden bilgi alınır. Deforme edilebilir evrişimde ise şekiller düzenli halde değildir. Şekil 3.7' deki gibi verinin üzerinde dağınık bir şekilde özellik çıkarımı yapılır.

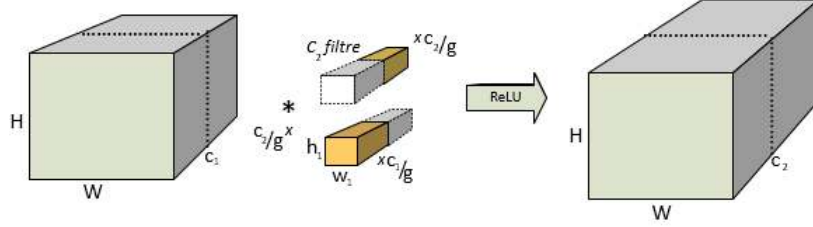


Şekil 3.7. Deforme edilebilir evrişim işlemi

Bu evrişim işleminde şekil, ağırlık geri yayılım hatasına göre ayarlanabilir [36]. Bu özelliği sayesinde deforme edilebilir evrişim, veri kaybı olmadan, daha az maliyet ile veriden daha anlamlı bilgiler alınabilmesini mümkün kılar.

Grup evriřim iřlemi

Grup evriřimi ilk olarak 2012’de AlexNet ile tanıtılmıřtır [13]. Sınırlı donanım kaynaklarını birleřtirerek sınıflandırma iřlemi yapma ihtiyaçı, grup evriřimini ortaya çıkmıřtır. Grup evriřiminde, evriřim parametreleri gruplandırılır [38]. Buna baęlı olarak aynı veri üzerinde farklı evriřim grupları uygulanır.

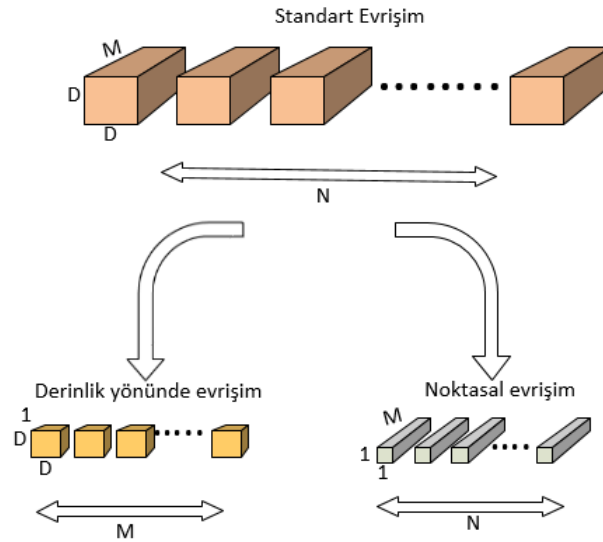


řekil 3.8. Grup evriřimi

řekil 3.8’de evriřim iki gruba ayrılmıřtır. Filtre grupları çeřitli yöntemlerle ayrılabilir. Bu evriřim yönteminde amaç derinlięi arttırırken maliyeti dūřük tutmaktır.

Derinlemesine ayrılabilir evriřim iřlemi

Derinlemesine ayrılabilir evriřim iřlemi, řekil 3.9’daki gibi standart evriřim iřlemini derinlik yönünde ve noktasal olarak iki paręaya ayırır.



řekil 3.9. Derinlemesine ayrılabilir evriřim

Standart evriřimin parametreleri $D \times D \times M \times N$ ’dir. Derinlik yönünde evriřimin parametreleri: $D \times D \times 1 \times M$ ve noktasal evriřimin parametreleri: $1 \times 1 \times M \times N$ ’dir. Bu parameteler oranlanırsa:

$$\frac{DxDxM + MxN}{DxDxMxN} = \frac{1}{N} + \frac{1}{D^2} \quad (3.6)$$

Denklem 3.6'daki hesaplamadan da görülmektedir ki derinlemesine ayrılabilir evrişim, standart evrişime göre çok daha az maliyetlidir [39]. Çok fazla parametrenin olması işlem maliyetini arttırdığı gibi aynı zamanda her zaman verimli sınıflandırma işlemi yapamayabilir. Derinlemesine ayrılabilir evrişim bu açıdan tercih edilebilir evrişim yöntemlerindendir [36].

3.4. Kapsül Ağlar

ESA, örüntü tanıma problemlerinde sıkça tercih edilen ve başarılı sonuçlar elde eden bir yöntemdir. Fakat eğitilmiş bir ESA görüntüdeki nesnelerin birbirine olan durumunu, açısını, derinliğini tanımlamak için yeterince elverişli bir yöntem değildir. Bu sorun ESA mimarisinde kullanılan havuzlama katmanlarından kaynaklanmaktadır. Havuzlama katmanında görüntü boyutları düşürülürken görüntüde bulunan bazı önemli öznitelikleri kaybeder. Nesnelerin birbirine göre açı, derinlik gibi olgularının o nesneyi ayırd etmek için önemli olduğu durumlarda özelleştirilmiş bir evrişimli sinir ağı mimarisi olarak bilinen kapsül sinir ağları ve dinamik yönlendirme algoritması tercih edilen algoritmalarındandır.

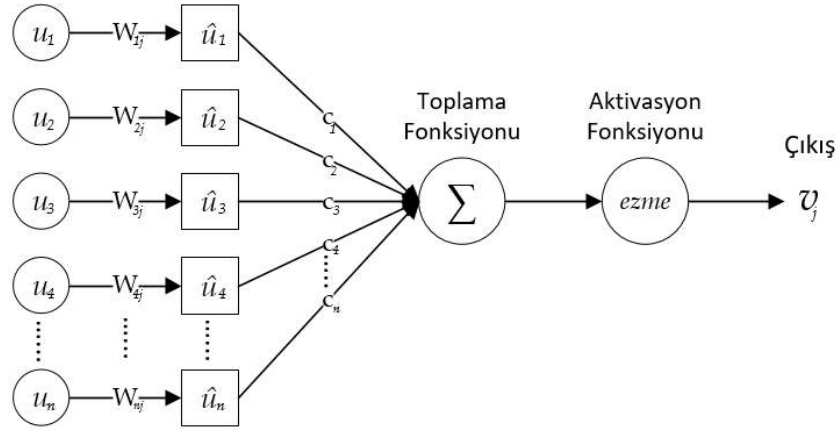
Kapsül ağlar, nöronlardan ziyade kapsüllerden oluşmaktadır. Bir kapsül ise, görüntünün belirli bir bölgesinde bulunan belirli bir nesneyi (örneğin göz) algılamayı öğrenen bir grup nörondan oluşur. Kapsül ağlarının mimarisi ilk olarak alınan verilere bir evrişim katmanı uygulanması ile başlamaktadır. Burada kullanılan ilk evrişim katmanı ESA'larda kullanılan evrişim katmanlarının aynısıdır. Daha sonra öncül kapsül katmanı adı verilen katmana geçilir. Bu katmanda da yine bir evrişim işlemi uygulanır. Uygulanan evrişim işleminin, önceki evrişim işleminden farkı bu katmana gelen görüntüleri vektörel hale getirmek amacıyla bir yeniden boyutlandırma işlemi uygulanması ve ardından aktivasyon fonksiyonu olarak ezme (squash) aktivasyon fonksiyonunun kullanılmasıdır. Bu işlemlerin sonunda dinamik yönlendirme algoritmasının bulunduğu kapsül katmanı yer almaktadır. Bu katmanın sonucunda DigitCaps adı verilen vektörler elde edilmektedir. DigitCaps vektörlerinin sayısı sınıflandırma işlemi için belirlenen sınıf sayısı kadardır.

Dinamik yönlendirme algoritması

Dinamik yönlendirme algoritması, girişleri benzer alt seviyede bulunan kapsülleri, girişleri benzer üst seviyede bulunan kapsüller ile eşleştiren bir algoritmadır.

ESA'da evrişim katmanlarının çıkışlarında genel olarak doğrusal olmayan ReLU aktivasyon fonksiyonu kullanılmakta ve çıkışlar skaler olarak ifade edilmektedir. Kapsül ağlarda ise ReLU aktivasyon fonksiyonuna karşılık ezme (squash) fonksiyonu denilen bir aktivasyon fonksiyonu kullanılır ve çıkışlar skaler değil vektörel olarak ifade edilir. Buradaki vektörler alınan veri bileşeninin, hedef sınıfta bulunma olasılığını ve ayrıca durumunu başka bir deyişle parametrelerini ifade etmektedir. Görüntüde istenilen nesnenin olduğu alanlarda vektör uzunluğu büyük olmadığı alanlarda ise küçüktür [40].

Ezme fonksiyonu vektörlerin uzunluğuna göre vektör uzunluklarının sıfır ile bir arasında bir değer almalarını sağlar. Söz konusu vektörün yönü ise bileşenin durumunu (açı, yönelim, kalınlık, renk vb.) ifade etmektedir. Dinamik yönlendirme algoritmasının uygulanması, havuzlama yöntemlerinin uygulanmasına göre öznetelik çıkarımında daha etkili bir yöntemdir [41].



Şekil 3.10. Kapsül giriş ve çıkışları

Bir kapsülün giriş ve çıkışları Şekil 3.10'da gösterildiği gibidir. Kapsülün girişleri u_i vektörleri olmak üzere bu vektörlere bir dönüşüm matrisi W_{ij} uygulanır. Böylece Denklem 3.7'deki gibi tahmin vektörü olarak bilinen $\hat{u}_{i|j}$ elde edilir. Daha sonra Denklem 3.8'de gösterildiği gibi tahmin vektörünün, kapsül i 'nin kapsül j 'yi ne kadar aktive edebileceğinin ölçümü olan c_{ij} , ile çarpımlarının toplamı alınır, bu bize ağırlıklı bir toplam olan s_j 'yi verir.

$$\hat{u}_{i|j} = W_{ij}u_i \quad (3.7)$$

$$s_j = \sum_i c_{ij}\hat{u}_{i|j} \quad (3.8)$$

Hesaplanan bu değerlere ezme fonksiyonunun uygulanması ve kapsül çıkışı olan v_j 'nin elde edilmesi Denklem 3.9'da gösterilmiştir.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (3.9)$$

Tahmin vektörü $\hat{u}_{i|j}$ ile kapsül çıkışının v_j skaler çarpımlarının alınmasıyla benzerlik b_{ij} , Denklem 3.10'daki gibi hesaplanır.

$$b_{i|j} \leftarrow \hat{u}_{i|j} \cdot v_j \quad (3.10)$$

Daha sonra b_{ij} 'ye softmax uygulanması ile c_{ij} 'nin yeni değeri Denklem 3.11'deki gibi hesaplanır.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (3.11)$$

b_{ij} daha doğru bir hale getirilmek amacıyla Denklem 3.12'de gösterildiği gibi, yinelemeli olarak (genel olarak 3 adımda) güncellenir.

$$b_{i|j} \leftarrow b_{i|j} + \hat{u}_{i|j} \cdot v_{ij} \quad (3.12)$$

Genel olarak dinamik yönlendirme algoritması Tablo 3.1'de gösterildiği gibi kaba kod yardımıyla özetlenebilir.

Tablo 3.1. Dinamik yönlendirme algoritması [2]

Algoritma 1: Dinamik Yönlendirme

1. **algoritma** Yönlendirme ($\hat{u}_{i|j}$, r , l)
 2. tüm kapsüller i , katmanlar l ve bulunulan kapsül katman j olmak üzere
($l+1$): $b_{ij} \leftarrow 0$
 3. for r iterasyon **do**
 4. tüm kapsüller için i . katman l : $c_i \leftarrow \text{softmax}(b_i)$ (denklem(3.11))
 5. tüm kapsüller için j . katman ($l+1$): $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$
 6. tüm kapsüller için j . katman ($l+1$): $v_j \leftarrow \text{ezme}(s_j)$ (denklem(3.9))
 7. tüm kapsüller için i . katman ve l kapsül j . katman ($l+1$): $b_{ij} \leftarrow b_{ij} + \hat{u}_{i|j} \cdot v_j$
 - return** v_j
-

İlk satır, r sayıda dinamik yönlendirme algoritmasının yenilenerek daha düşük bir l seviyesindeki tüm kapsülleri ve bunların çıktılarını ($\hat{u}_{i|j}$) aldığını ifade etmektedir. En son satır ise algoritmanın sonucunda daha yüksek seviyeli bir kapsül çıktısının (v_j) üretileceğini ifade etmektedir.

b_{ij} katsayısı, yinelemeli olarak güncellenen geçici bir değerdir ve prosedür bittikten sonra değeri c_{ij} 'de saklanacaktır. Eğitimin başlangıcında b_{ij} değeri, sıfırdan başlatılır. 3.satır, 4 ile 7. satırlar arasındaki adımların r defa tekrarlanacağını ifade etmektedir.

4. satırdaki adım, daha düşük seviyeli bir kapsül i için tüm yönlendirme ağırlıkları olan, c_i vektörünün değerini hesaplar. Bunu yaparken ise b_{ij} değerlerini softmax fonksiyonundan geçirir. Tüm düşük seviyeli kapsüller için bu işlem yapılır. İlk yinelemede, tüm c_{ij} katsayılarının değeri eşit olacaktır, çünkü ikinci satırda tüm b_{ij} değerleri sıfıra

ayarlanmıştır. Süreç tekrarlandıkça bu tek tip dağılımlar değişecektir.

Tüm düşük seviyeli kapsüller için tüm ağırlıklar (c_{ij}) hesaplandıktan sonra , daha yüksek seviyeli kapsüllere bakılması için 5. satıra geçilir. Bu adımda, önceki adımda belirlenen c_{ij} yönlendirme katsayıları ile ağırlıklandırılan girdi vektörlerinin doğrusal bir kombinasyonunu hesaplar. Bu da s_j çıktı vektörünü üretir. Tüm yüksek seviyeli kapsüller için bu işlem yapılır.

Son adımdaki vektörler, vektörün yönünün korunmasını sağlar ve uzunluğunun 1'den fazla olmamasını sağlayan, doğrusal olmayan ezme (squash) aktivasyon fonksiyonundan geçirilir. Bu adım, tüm yüksek seviyeli kapsüller için v_j çıktı vektörünü üretir. Bu adımlar, her bir yüksek seviyeli kapsül j'ye bakar ve ardından her girişi inceler ve formüle göre karşılık gelen b_{ij} ağırlık değerini günceller. Böylece düşük seviyeli kapsül, çıktısını benzer olan daha yüksek seviyeli kapsüle gönderecektir. Bu benzerlik, iç çarpım tarafından yakalanmaktadır.

Bu algoritma, sonraki yinelemelerde 3. adımdan başlar ve işlemi r kez tekrar eder. Böylece ESA'dan farklı olarak nesnelerin birbirine göre olan durumları da hesaba katılmış olur.

Kayıp hesaplama:

Eğitim sırasında kayıp, Denklem 3.13'deki eşitliğe göre sınıf sayısı kadar olan vektörlerin her biri için hesaplanacaktır. Ardından son kaybı hesaplamak için bu hesaplanan sınıf sayısı kadar olan değerler toplanacaktır. Kayıp fonksiyonu formülünde sınıf tahmini, T_c 'nin değerini belirler. Sınıf doğru tahmin edilmiş ise T_c 'nin değeri 1 aksi halde 0 olur.

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 + \lambda(1 - T_c) \max(0, \|v_c\| - m^-)^2 \quad (3.13)$$

Formüle göre, bir örneğin sınıfı doğru tahmin edildiğinde bir doğru sınıf ve sınıf sayısının bir eksiği kadar da yanlış sınıf oluşur. Doğru sınıfın T_c değeri 1 ve geri kalan sınıfların T_c değeri 0 olacaktır. Bu durumda doğru sınıfın kayıp değeri hesaplanmak istendiğinde eşitliğin birinci kısmı kullanılacaktır (Denklem 3.14).

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 \quad (3.14)$$

Doğru tahmin durumunda digit kapsülün kaybını hesaplamak için, digit kapsülün çıkış vektörü m^+ 'dan (0.9 sabit değer) çıkarılır. Ardından, elde edilen değerın karesi alınır. Bu işlem ile kapsül çıkışı 0.9 değerinden fazla olduğu durumlarda hata sıfır, diğer durumlarda sıfırdan büyük bir sayı elde edilir.

Doğru etiketle eşleşmeyen digit kapsüller için T_c değeri sıfır olacak ve bu nedenle eşitliğin ikinci kısmı kullanılacaktır (Denklem 3.15).

$$L_c = \lambda(1 - T_c) \max(0, \|v_c\| - m^-)^2 \quad (3.15)$$

Yanlış tahmin durumunda digit kapsülün kaybını hesaplamak için, digit kapsülün çıkış vektöründen m^- (0.1 sabit değer) değeri çıkarılır. Ardından, elde edilen değerın karesi alınır. Bu işlem ile kapsül çıkışı 0.1 değerinden az olduğu durumlarda hata sıfır, diğer durumlarda sıfırdan büyük bir sayı elde edilir.

Son olarak, formülde, eğitim sırasında sayısal kararlılık için lambda katsayısı değeri (0.5 sabit değer) dahil edilmiştir.

Kapsül ağırları ESA'nın havuzlama katmanından kaynaklı olarak oluşturduğu sorunlara, dinamik yönlendirme algoritmasıyla yeni bir yöntem öneren bir modeldir. Diğer bir deyişle ESA'da bulunan havuzlama katmanının yerini kapsül ağırlarda dinamik yönlendirme algoritması almaktadır. Dinamik yönlendirme algoritması sayesinde verilerden ve söz konusu verileri oluşturan bileşenlerinden daha fazla bilgi öğrenilir. Bu özelliklerinden ötürü kapsül ağırları öğrenilecek verilerde; derinlik, örtüşme, kalabalık gibi eğitim işlemini zorlaştıran özelliklerin var olduğu durumlarda tercih edilen yöntemlerdendir.

4. MATERYAL VE METOT

Bu bölümde bu tez kapsamında yapılan çalışmalara yer verilmiştir. Çalışmalarda kullanılmak üzere bir kapsül ağılar modeli hazırlanmıştır. Yüz verilerini sınıflandırma işleminde kullanılan kapsül ağılar modeli temel olarak altı katmandan oluşturulmuştur.

Giriş katmanı

Giriş katmanı modelin ilk katmanıdır ve 64x64 boyutunda olan 3 kanallı (rgb) eğitim görüntü verilerinden oluşur. Giriş olarak alınan bu veriler evrişim katmanına gönderilir.

Evrişim katmanı

Girişler evrişim katmanına iletilir. Bu katman ESA'nın mimarisinde bulunan evrişim katmanlarının aynısıdır. Evrişim katmanının parametreleri Tablo 4.1'de gösterildiği gibidir. Bu katmandan sonra özellik kapsül katmanı olarak adlandırdığımız, öncül kapsül katmanı gelmektedir.

Tablo 4.1. Evrişim katmanı parametreleri

Evrişim Katmanı Parametreleri	
Filtre Sayısı	256
Çekirdek Boyutu	9
Kaydırma Adımı	1
Doldurma	Valid
Aktivasyon Fonksiyonu	ReLU

Görüntü verilerinin boyutu, evrişim katmanından sonra (çekirdek boyutu - 1 = 8) 8 piksel kaybeder. Uygulanan evrişim katmanından sonra görüntü boyutları 56x56 piksele düşmüştür. Daha sonra elde edilen bu görüntülere özellik kapsülü katmanı uygulanmıştır.

Özellik kapsülü katmanı

Bu katman evrişim katmanı çıkışlarını, giriş olarak almaktadır. Evrişim katmanından farklı olarak bu katmanda yeniden boyutlandırma işlemi ve ezme aktivasyon fonksiyonu uygulanmaktadır. Özellik kapsülünün parametreleri Tablo 4.2'de verildiği gibidir.

Tablo 4.2. Özellik kapsülü katmanı parametreleri

Özellik Kapsülü Katmanı Parametreleri	
Kapsül Boyutu	16
Çekirdek Boyutu	5
Kaydırma Adımı	2
Kanal Sayısı	16
Doldurma	Valid

Evrişim katmanıya 56x56 boyutlarına indirgenen görüntünün, özellik kapsülü katmanının kaydırma adımı 2 olan parametresi ile boyutları 28x28 (2'ye bölümü) boyutlarına düşürülmüştür. Bununla birlikte çekirdek boyutu 5 alınarak (çekirdek boyutu - 1 = 4 / 2 = 2) görüntü 2 piksel daha kaybedip bu katman çıkışında 26x26 boyutuna

gelmiştir. Ayrıca bu katmanda test edilerek model için en optimum değer olarak kabul edilen 0.3 değerinde Dropout işlemi uygulanmıştır.

Daha sonra bu görüntüleri 16 boyutlu vektörler haline getirmek için bir yeniden boyutlandırma işlemi uygulanarak, görüntüler vektör boyutuna indirgenmiştir (26x26x16x16).

Özellik kapsül katmanı, digit kapsül katmanına tamamen bağlı olduğundan, çıkış vektörlerinin ezme (squash) aktivasyon fonksiyonundan geçirilmesi gerekir. Bu aşamada ezme fonksiyonuna, görüntüyü tekrar oluşturma kaybı olan küçük bir epsilon değeri eklenmelidir. Model için bu değer 0.005 olarak seçilmiştir. Ezme fonksiyonunun çıktısı digit kapsül katmanına iletilir.

Digit kapsül katmanı

Bu katmandaki kapsül sayısı, tahmin edilmesi gereken sınıfların sayısı (5) ile belirlenir. Ayrıca bu katmanda bulunan kapsüller, özellik yönelimlerine dayalı olarak sınıflar hakkında tahminlerde bulunacaktır. Ancak bu tahminlerin öncül kapsüller tarafından yapılan tahminlere uyması gerekir ve bu yöntem, dinamik yönlendirme algoritması ile sağlanır. Bu katmanda giriş, özellik kapsül katmanının çıktısı olan (26x26x16x16) 16 boyutlu vektörlerdir. Kapsülün iç işleyişine göre, giriş vektörlerinin her biri, 16 boyutlu girdi uzayını 32 boyutlu kapsül çıktı uzayına eşleyen kendi 16x32 boyutlu ağırlık matrisini alır. Dolayısıyla, bu katmanda (26x26x16x16x32x5) 27.688.960 parametre bulunmaktadır. Modelin katmanları ve parametreleri Tablo 4.3’de gösterildiği gibidir.

Tablo 4.3. Model katmanları ve parametreleri

Katman	Giriş	Parametre	Çıkış
Giriş Katmanı	64x64 resim	0	64x64x3
Evrişim Katmanı	64x64 resim	62464	56x56x256
Özellik Kapsül Katmanı Evrişim	56x56x256 tensor	1638656	26x26x256
Özellik Kapsül Katmanı Yeniden Boyutlandırma	26x26x 256 tensor	0	10816x16
Digit Kapsül Katmanı	26x26x16x16 tensor	27688960	32x5
Kod Çözücü	32x5 matris	66879	64x64

Kod çözücü

Kod çözücü, 32 boyutlu bir vektörü doğru digit kapsülden alır ve onu etiketli bir görüntüye dönüştürmeyi öğrenir. Kod çözücü bir düzenleyici olarak kullanılır. Doğru digit kapsülün çıktısını girdi olarak alır ve 64’e 64 piksellik bir görüntüyü yeniden oluşturmaya amaçlar. Kayıp işlevi, yeniden yapılandırılmış görüntü ile giriş görüntüsü arasındaki öklid mesafesidir. Kod çözücü; kapsülleri, orijinal görüntüyü yeniden oluşturmak için yararlı olan özellikleri öğrenmeye zorlar. Yeniden yapılandırılmış görüntü, giriş görüntüsüne ne kadar yakınsa o kadar iyidir. Bu katmanda da 66.879 parametre kullanılmaktadır.

Sonuç olarak bu ağda kullanılan toplam parametre sayısı 29.456.959’dır. Toplanan yüz görüntüsü verileri oluşturulan modelde kullanılarak eğitim, doğrulama ve test işlemleri gerçekleştirilmiştir.

4.1. Kapsül Ağlar ve Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması

Bu çalışmada; küçük bir veri kümesi oluşturularak, kapsül ağlar ve bazı evrişimli sinir ağları ile sınıflandırma uygulamaları gerçekleştirilmiştir. Çalışmanın amacı bu modellerin başarımlarını kıyaslamaktır.

Çalışmada, yüz verilerini sınıflandırma işlemi evrişimli sinir ağı algoritmalarından olan AlexNet, Vgg16, Vgg19, ResNet50 modelleri ve kapsül ağlar modeli kullanılarak Google Colaboratory çalışma ortamında Tesla K80 donanımı ile gerçekleştirilmiştir. Vgg16, Vgg19 ve ResNet50 modelleri Keras kütüphanesinin uygulamalarından alınarak Python programlama dilinde oluşturulmuştur. AlexNet ve Kapsül ağlar modeli ise Tensorflow ve Keras kütüphaneleri kullanılarak Python programlama dilinde oluşturulmuştur.

Bu çalışma için 5 kişinin her biri için 200 tane olmak üzere kişilerin yüz verilerini içeren 64x64 boyutunda toplam 1000 renkli görüntüden oluşan bir veri seti, internet ortamından ünlü kişilerin yüz görüntüleri toplanarak hazırlanmıştır. Bu çalışmada kullanılan veri kümesi örneği Şekil 4.1’de verildiği gibidir.

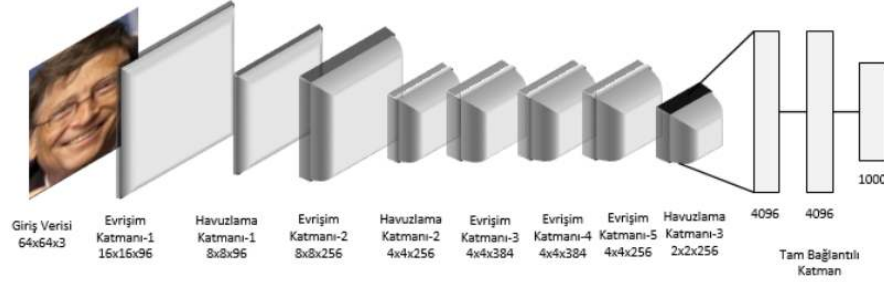


Şekil 4.1. Kapsül ağlar ve evrişimli sinir ağları ile yüz verilerinin sınıflandırılması çalışması veri kümesi örneği

Hazırlanan veri kümesinin %60’ı eğitim işlemi için, % 20’si doğrulama işlemi için ve kalan %20’si ise test işlemi için kullanılmak üzere ayrılmıştır.

ESA’nın kabul görmüş birçok modeli bulunmaktadır. İlk olarak 2012’de düzenlenen ImageNet yarışmasında önerilen AlexNet modeli sınıflandırma başarımı ile ESA’nın literatüründe önemli bir yer almaktadır [13]. AlexNet modelinin sınıflandırma üzerinde gerçekleştirmiş olduğu başarımlar, sınıflandırma işleminde ESA’ya olan ilgiyi büyük ölçüde

arttırarak, bu alanda yeni modeller geliştirilmesine ilham olmuştur. Söz konusu AlexNet modelinin yapısı Şekil 4.2’de verilmiştir. Düzenlenen ImageNet yarışmaları ile 2014 yılında birinci olan VGGNet [42] ve 2016 yılında birinci olan ResNet [43] modelleri de AlexNet modeli gibi literatürde yerini alan diğer yüksek başarımlı modellerdendir.



Şekil 4.2. AlexNet modeli yapısı

Bu bölümde gerçekleştirilen sınıflandırma uygulamalarında, kapsül ağlar modeli ile karşılaştırılan evrişimli sinir ağları modelleri yer almaktadır.

AlexNet

AlexNet modeli başarımlı kanıtlamış başlıca evrişimli sinir ağı modellerinden birisidir. Bu modelde 5 adet evrişim katmanı, 3 adet havuzlama katmanı ve 3 adet tam bağlantılı katman yer almaktadır.

VGG16

Vgg16 modeli toplam 16 katmandan oluşan bir evrişimli sinir ağı modelidir. Modelin yapısı AlexNet modelinin yapısına oldukça benzemektedir. Bu model, yapısında 13 adet evrişim katmanı ve 3 adet tam bağlantılı katman bulundurmaktadır.

VGG19

Vgg19 modeli toplam 19 katmandan oluşan bir evrişimli sinir ağı modelidir. Modelin yapısı Vgg16 modelinin yapısının daha da derinleştirilmiş bir hali olarak düşünülebilir. Modelin yapısında 16 adet evrişim katmanı ve 3 adet tam bağlantılı katman yer almaktadır.

ResNET50

ResNet50 modeli diğer ESA modellerinden farklı olarak ardışık ağ mimarisi yerine bünyesinde artık blok yapısı bulundurmaktadır [44]. Bu yapının kullanılmasının nedeni, ağın yapısı derinleştikçe başarımın düşmesine engel olmak, başarımın düşeceği durumlarda ise başarımın aynı kalmasını sağlamaktır.

4.2. Hafif Siklet Evrişimli Sinir Ağları ile Kapsül Ağların Yüz Verilerini Sınıflandırma Performanslarının Karşılaştırılması

Görsel verileri sınıflandırma üzerinde tercih edilen ESA’ların çok derin ağlar olmaları, eğitim maliyetlerini arttırmaktadır. ESA’lar kadar derin ve maliyetli olmayan buna karşın

yüksek sınıflandırma başarımları elde edebilecek bir çok hafif siklet evrişimli sinir ağı modeli geliştirilmiştir [36].

Bu çalışmada hazırlanan küçük bir veri kümesi üzerinde, hafif siklet evrişimli sinir ağları ile kapsül ağların sınıflandırma başarımları ve verimlilikleri karşılaştırılmak istenmiştir.

Çalışmada, yüz verilerini sınıflandırma işlemi hafif siklet evrişimli sinir ağları modellerinden olan MobleNetV2, SqueezeNet, ShuffleNet, EfficientNetB0 modelleri ve kapsül ağları modeli kullanılarak Google Colaboratory çalışma ortamında Tesla K80 donanımı ile gerçekleştirilmiştir. MobileNetV2 ve EfcientNetB0 modelleri Keras kütüphanesinin uygulamalarından alınarak Python programlama dilinde oluşturulmuştur. ShuffleNet, SqueezeNet ve kapsül ağlar modeli ise Tensorflow ve Keras kütüphaneleri kullanılarak Python programlama dilinde oluşturulmuştur.

Bu çalışmada, 5 ünlü kişinin her biri için 120 tane olmak üzere, kişilerin yüz verilerini içeren 64x64 boyutunda toplam 600 renkli görüntüden oluşan bir veri seti, internet ortamından, yüz görüntüleri toplanarak hazırlanmıştır. Bu çalışmada kullanılan veri kümesi örneği Şekil 4.3’de verildiği gibidir.



Şekil 4.3. Hafif siklet evrişimli sinir ağları ve kapsül ağlar ile yapılan çalışmada kullanılan veri kümesi örneği

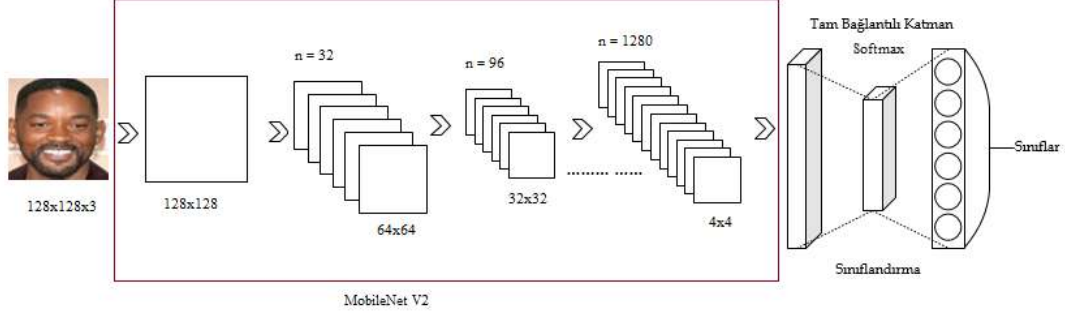
Hazırlanan veri kümesinin %60’ı eğitim işlemi için, % 20’si doğrulama işlemi için ve kalan %20’si ise test işlemi için kullanılmak üzere ayrılmıştır.

Bu bölümde ise gerçekleştirilen sınıflandırma uygulamasında kullandığımız hafif siklet evrişimli sinir ağı modelleri hakkında bilgiler verilmiştir.

MobileNet

MobileNet’in ilk versiyonu, bilinen standart evrişim katmanları yerine derin evrişim katmanlarını içeren bir yapı ile tanıtıldı [20]. Derinlemesine ayrılabilir evrişim katmanlarının kullanılmasının amacı, ağı parametre sayısını azaltmaktır [36]. Önerilen ikinci versiyonda

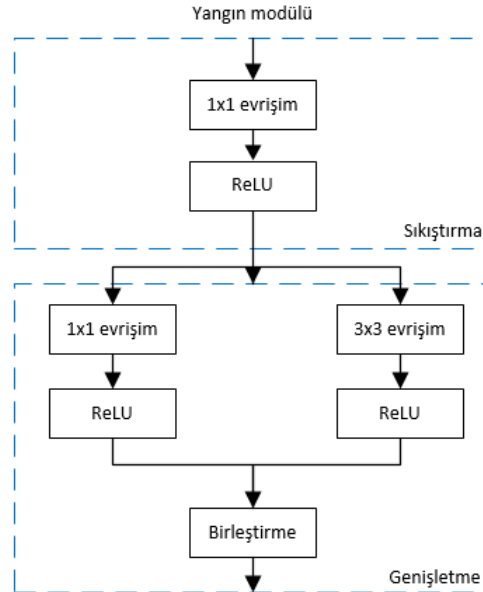
[21], bu algoritma ters kalıntı yapıları ve 1x1 noktasal evrişimlerle desteklenmiştir [45]. Bu sayede verilerden bilgi kaybının azaltılması ve daha doğru sınıflandırma işlemlerinin gerçekleştirilmesi amaçlanmaktadır. MobileNetV2 modelinin yapısı Şekil 4.4’de gösterildiği gibidir.



Şekil 4.4. MobileNetV2 modelinin yapısı

SqueezeNet

SqueezeNet, AlexNet’ten 50 kat daha az parametreyle sunulan hafif bir evrişimsel sinir ağı olmasıyla bilinmektedir [19]. SqueezeNet, biri başlangıçta ve biri sonda olmak üzere 2 evrişim katmanı ve bunların arasında 8 yangın modülü içerir. Bu yangın modülleri temel olarak 1x1 filtre sıkıştırma ve 3x3 filtre genişletme katmanlarından oluşur [46]. Bu katmanların çıkışında ReLU aktivasyon fonksiyonu kullanılmaktadır. SqueezeNet modelinde, evrişim katmanlarının arasında yer alan yangın modülünün yapısı Şekil 4.5’de verildiği gibidir.



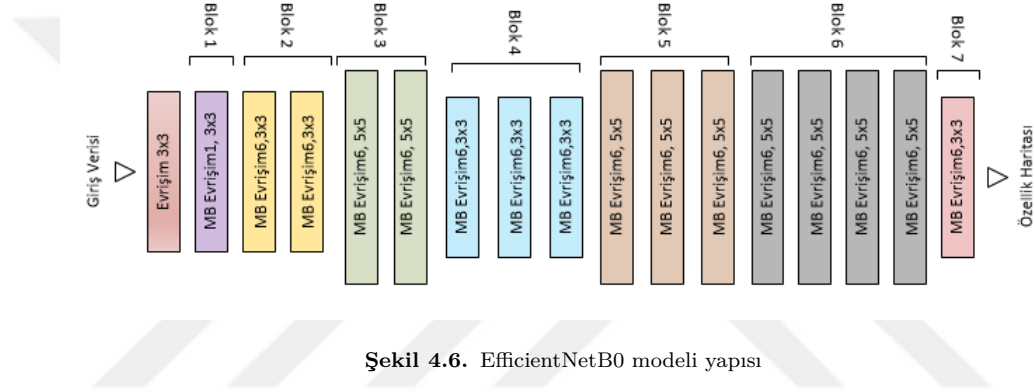
Şekil 4.5. SqueezeNet yangın modülü yapısı

ShuffleNet

ShuffleNet, grup evrişimi ve kanal karıştırma işlemi ile geliştirilmiş hafif siklet evrişimli bir sinir ağıdır [47]. Grup evrişiminde, verilerden elde edilen bilgilerin gruplarla sınırlandırılabilmesi sorunu vardır. Bu noktada sorunu çözmek için ShuffleNet, kanal shuffle katmanı sayesinde tensörün kanallarının rastgele shuffle işlemine tabi tutma yaklaşımını kullanır [48].

EfficientNet

Hafif siklet evrişimli sinir ağlarından bir tanesi de EfficientNet'tir. EfficientNet, bir dizi sabit kullanarak ağı derinliğini, çözünürlüğünü ve genişliğini ölçeklendirir [49]. EfficientNet, B0'dan B7'ye kadar 8 modele sahiptir ve modelin numarası arttıkça performansı da artmaktadır. EfficientNetB0 modelinin yapısı Şekil 4.6'da verildiği gibidir.



Ayrıca EfficientNet, ReLU aktivasyon fonksiyonunu kullanmak yerine Swish adlı bir aktivasyon fonksiyonunu kullanmayı tercih eder [22].

$$f(x) = x.\text{sigmoid}(\beta x) \quad (4.1)$$

$$f(x) = x.\text{sigmoid}(x) \quad (4.2)$$

Swish aktivasyon fonksiyonu Denklem 4.1'de verilmiştir. β parametresi burada öğrenilebilir bir parametredir fakat bu parametre neredeyse hiçbir versiyonda kullanılmamıştır [50]. Genel olarak tercih edilen hali Denklem 4.2'de verildiği gibidir.

4.3. Kapsül Ağların Farklı Boyutlardaki Veri Kümelerinde Sınıflandırma Durumlarının İncelenmesi

Bu çalışmada üç farklı boyutta veri kümesi hazırlanarak, sınıflandırma çalışması kapsül ağlar modeli ile gerçekleştirilip karşılaştırılmıştır. Uygulamalar Google

Colaboratory çalışma ortamında gerçekleştirilmiştir. Gerçekleştirilen uygulamalar için Tesla P100 donanımı kullanılmıştır.

Çalışmada kullanılan veri kümesi internet ortamından ünlü kişilerin yüz görüntüleri toplanarak hazırlanmıştır. Veri kümesi 5 kişinin, kişi başı 300 görüntü olacak şekilde toplam 1500 görüntüden oluşan 1.veri seti, kişi başı 500 görüntü olacak şekilde toplam 2500 görüntüden oluşan 2.veri seti ve kişi başı 700 görüntü olacak şekilde toplam 3500 görüntüden oluşan 3.veri seti şeklinde ayrılmıştır. Veriler 64x64 boyutunda renkli yüz görüntülerinden meydana gelmektedir.

Hazırlanan veri kümesinin bir örneği Şekil 4.7’de gösterildiği gibidir.



Şekil 4.7. Kapsül ağların farklı boyutlarda başarımının incelenmesi için hazırlanan veri kümesi örneği

Veri kümeleri toplanırken kişilerin yüz görüntülerinin, farklı açılardan, farklı yaşlardan, farklı aksesuarlardan gibi bazı özelliklere sahip örneklerden toplanılmasına özen gösterilmiştir.

Hazırlanan her üç veri kümesinin de %60'ı eğitim işlemi için, % 20'si doğrulama işlemi için ve kalan %20'si ise test işlemi için kullanılmak üzere ayrılmıştır.

5. SONUÇLAR

Bu bölümde tez kapsamında yapılan çalışmalardan elde edilen sonuçlara yer verilmiştir.

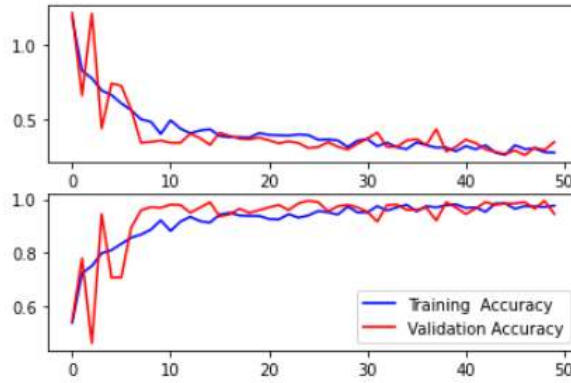
5.1. Kapsül Ağlar ve Evrişimli Sinir Ağları ile Yüz Verilerinin Sınıflandırılması Çalışması Sonuçları

Hazırlanan yüz görüntüleri veri kümesi, 4 farklı evrişimli sinir ağı modeli ile eğitim işleminden geçirilerek, her biri için test işlemi başarımları alınmıştır. Modellerin her biri için eğitim işlemi 50 adımda (epochs) gerçekleştirilirken yalnızca ResNet50 modeli 100 adımda eğitim işleminden geçirilmiştir. Yapılan uygulamadan elde edilen sonuçlar Tablo 5.1'de gösterildiği gibidir. Modeller arasında en iyi sonucu veren AlexNet modelinin eğitim

Tablo 5.1. Evrişimli sinir ağı modelleri sonuçları

Model	Test Başarımları
AlexNet	%94
VGG16	%91
VGG19	%93.5
ResNet50	%75

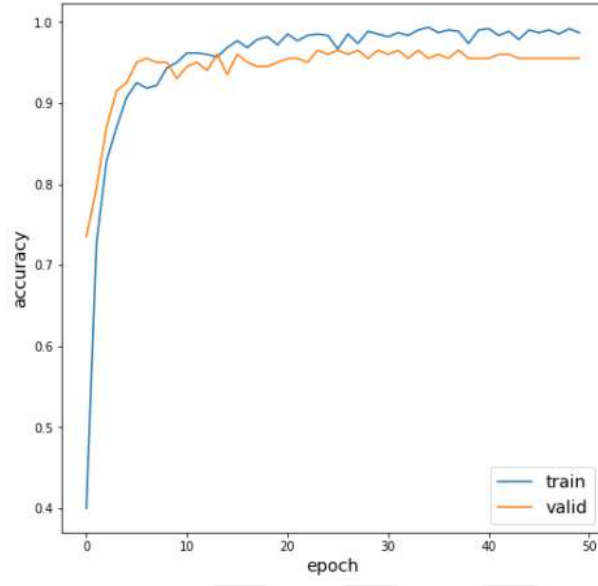
ve doğrulama işlemleri üzerindeki kayıp ve başarımlar grafiği Şekil 5.1'de gösterildiği gibidir.



Şekil 5.1. AlexNet modeli eğitim ve doğrulama grafiği

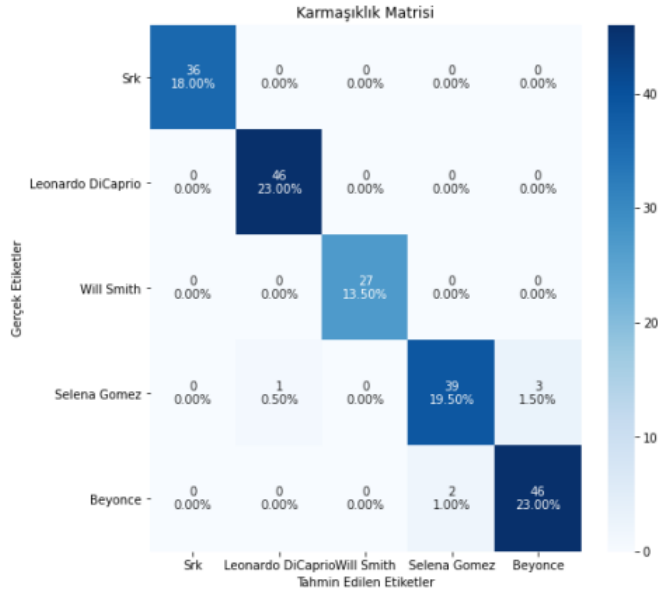
Diğer modellere görece AlexNet modelinin hazırlanan veri kümesi üzerindeki yüksek sınıflandırma başarımları elde etmiş olması, ağı derinliği arttıkça başarımların her zaman artmadığını göstermektedir [51].

Kapsül ağlar ile hazırlanıp eğitilen modelde ise test işlemi %97 başarımlarıyla sonuçlanmıştır. Modelin eğitim ve doğrulama süreci grafiği Şekil 5.2'de gösterildiği gibidir.



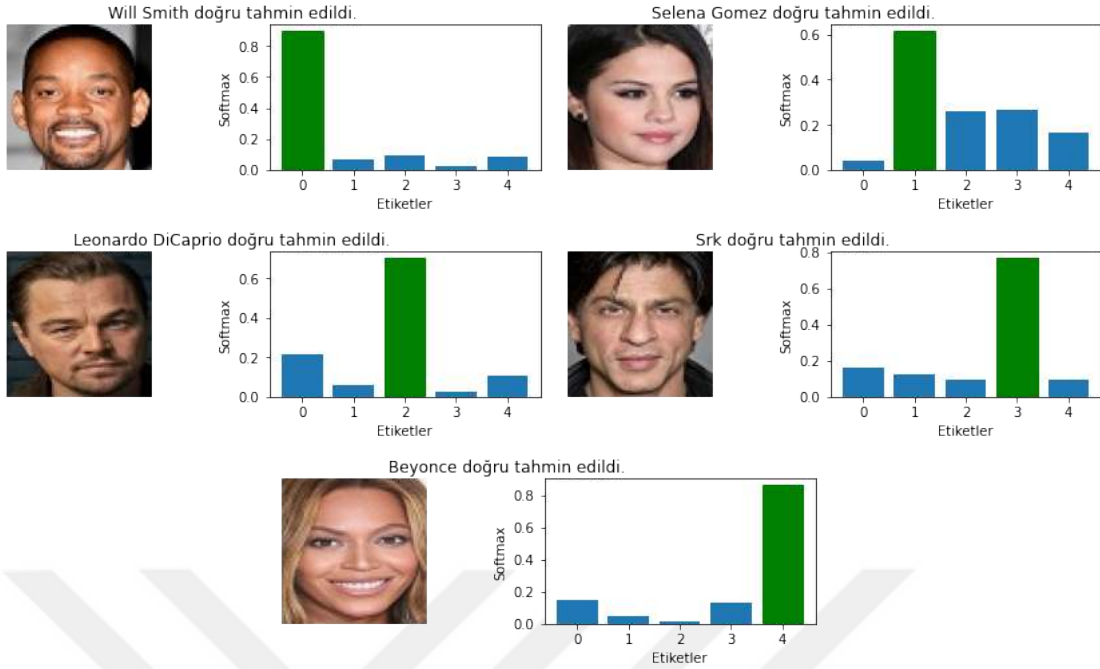
Şekil 5.2. Kapsül ağılar modeli eğitim ve doğrulama grafiği

Bu başarımların değerleri modelin çok kez çalıştırılmasıyla elde edilen 5 en iyi sonucun ortalaması olan değerlerdir. Modelin sonucunda ortaya çıkan karmaşıklık matrisi Şekil 5.3'de gösterildiği gibidir.



Şekil 5.3. Kapsül ağılar modeli karmaşıklık matrisi

Modelin, kullanılan yüz verilerinden bazılarının sınıflarını tahmin etmesine dair bir örnek Şekil 5.4'deki gibidir.



Şekil 5.4. Kapsül ağılar modeli tahminleme örnekleri

Yapılan uygulamalar sonucunda kapsül ağılar ile hazırlanan modelin test doğruluğu evrişimli sinir ağıları modellerine kıyasla daha yüksek sonuçlanmıştır. Bunun nedeninin evrişimli sinir ağlarında bulunan havuzlama katmanının verilerde öznelilik kaybına neden olması olduğu tahmin edilmektedir. Buna bağlı olarak kapsül ağların yüz verilerini sınıflandırma problemi üzerinde başarılı sonuçlar verebileceği gözlemlenmiştir.

Tablo 5.2. Modellerin eğitim işlemi tamamlama süreleri

Model	Eğitim Süresi
AlexNet	2 dakika 47 saniye
VGG16	1 dakika 25 saniye
VGG19	1 dakika 29 saniye
ResNet50	6 dakika 50 saniye
Kapsül Ağlar	13 dakika 44 saniye

Buna karşılık Tablo 5.2’de verilen modellerin eğitim işlemi tamamlama sürelerine bakıldığı zaman ise kapsül ağılar modelinin diğer modellere göre daha uzun sürede bu işlemi gerçekleştirdiği görülmektedir. Modellerin eğitim tamamlama sürelerinden yola çıkılarak daha büyük veri kümelerinde kapsül ağılar modeli ile yapılacak eğitim işlemlerinin daha maliyetli olacağı göz önünde bulundurulabilir.

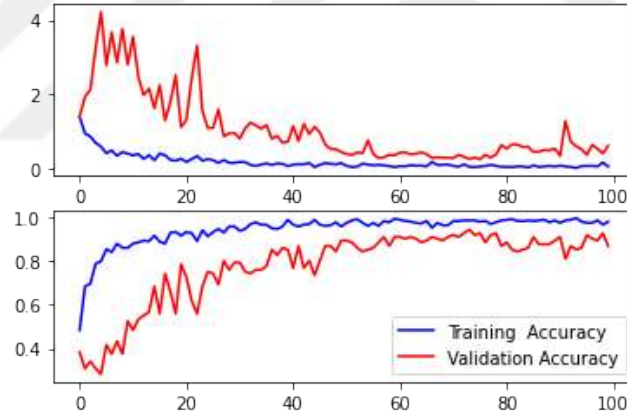
5.2. Hafif Siklet Evrişimli Sinir Ağları ile Kapsül Ağların Yüz Verilerini Sınıflandırma Performanslarının Karşılaştırılması Çalışması Sonuçları

Hazırlanan yüz görüntüleri veri kümesi, 4 farklı hafif siklet evrişimli sinir ağı modeli ile eğitim işleminden geçirilerek, her biri için test işlemi başarımları alınmıştır. Modellerin her biri için eğitim süreci 100 adımda (epoch) gerçekleştirilirken, sadece kapsül ağlar modeli 50 (epoch) adımda eğitilmiştir. Bunun nedeni ise kapsül ağlar modelinin eğitim süresinin uzun sürmesidir. Yapılan uygulamadan elde edilen sonuçlar Tablo 5.3'de gösterildiği gibidir. Modeller arasında en iyi sonucu veren MobileNetV2 modelinin eğitim

Tablo 5.3. Hafif siklet evrişimli sinir ağı modelleri sonuçları

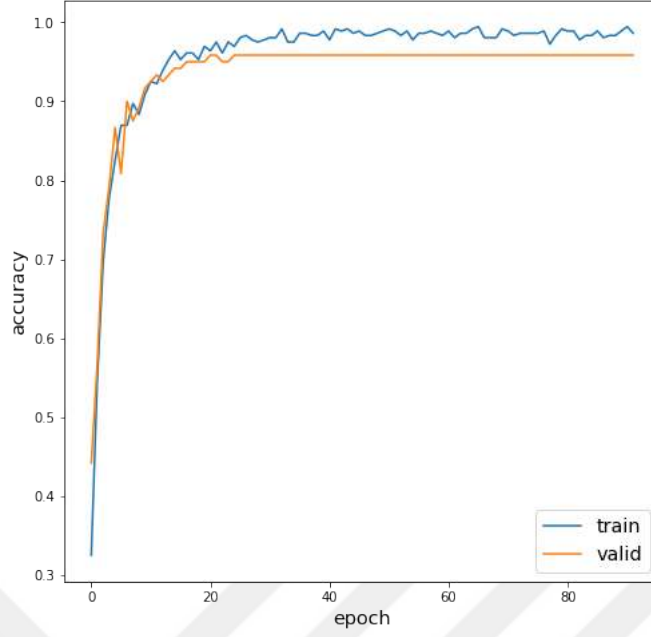
ESA Modelleri	Model Test Başarımları
MobileNetV2	%90.8
SqueezeNet	%74.1
ShuffleNet	%81.6
EfficientNet	%80.8

ve doğrulama işlemleri üzerindeki kayıp ve başarımlar grafiği Şekil 5.5'de gösterildiği gibidir.



Şekil 5.5. MobileNet modeli eğitim ve doğrulama grafiği

Kapsül ağlar ile hazırlanıp eğitilen modelde ise test işlemi %92.5 başarımlarıyla sonuçlanmıştır. Modelin eğitim ve doğrulama süreci grafiği Şekil 5.6'da gösterildiği gibidir.



Şekil 5.6. Kapsül ağılar modeli eğitim ve doğrulama grafiği

Modellerin eğitim işlemi tamamlama süreleri Tablo 5.4’de verildiği gibidir.

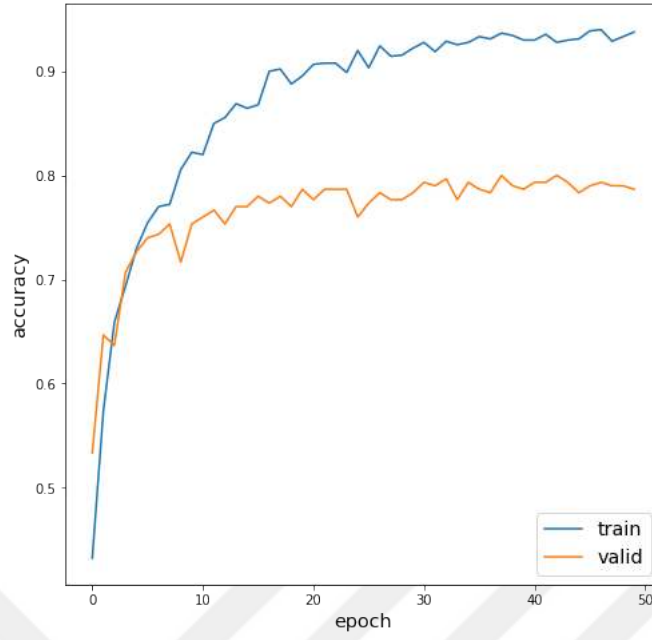
Tablo 5.4. Modellerin eğitim işlemi tamamlama süreleri

Model	Eğitim Süresi
MobileNetV2	2 dakika 32 saniye
SqueezeNet	6 dakika 10 saniye
ShuffleNet	20 dakika 5 saniye
EfficientNet	3 dakika 28 saniye
Kapsül Ağlar	27 dakika 2 saniye

İlk çalışmada olduğu gibi diğer modellerden daha az eğitim adımı uygulanmasına rağmen, zaman açısından en maliyetli olan model kapsül ağlardır. Buna karşılık kapsül ağların, küçük veri kümelerinde olan başarımları diğer modellere göre daha yüksek sonuçlanmıştır.

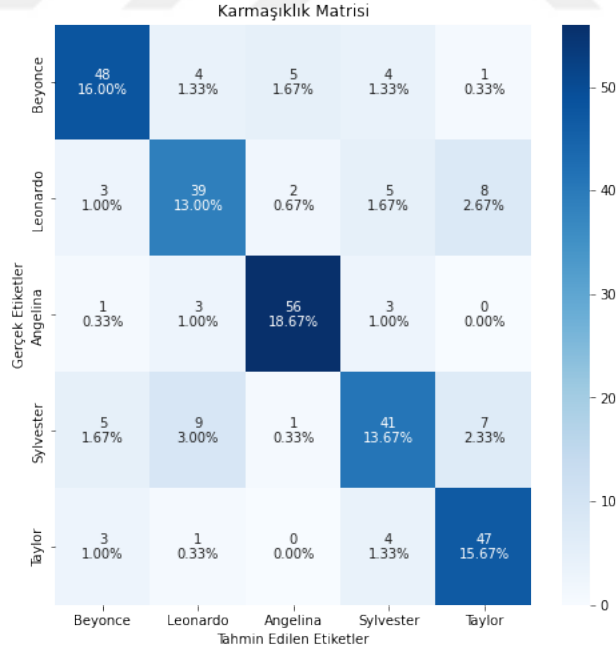
5.3. Kapsül Ağların Farklı Boyutlardaki Veri Kümelerinde Sınıflandırma Durumlarının İncelenmesi Çalışması Sonuçları

Bu çalışmada hazırlanan üç farklı veri kümesi kapsül ağlar modeli ile eğitilmiştir. Veri kümeleri önceki çalışmalarda kullanılan veri kümelerine göre daha fazla kişilerin farklı durumlarından alınan görüntülerle oluşturulmuştur. Hazırlanan ilk veri kümesi üzerinde kapsül ağların test başarımları %77 olarak sonuçlanmıştır. Modelin ilk veri kümesi ile elde edilen eğitim ve doğrulama grafiği Şekil 5.7’deki gibidir.



Şekil 5.7. Modelin ilk veri seti ile elde edilen eğitim ve doğrulama süreci grafiği

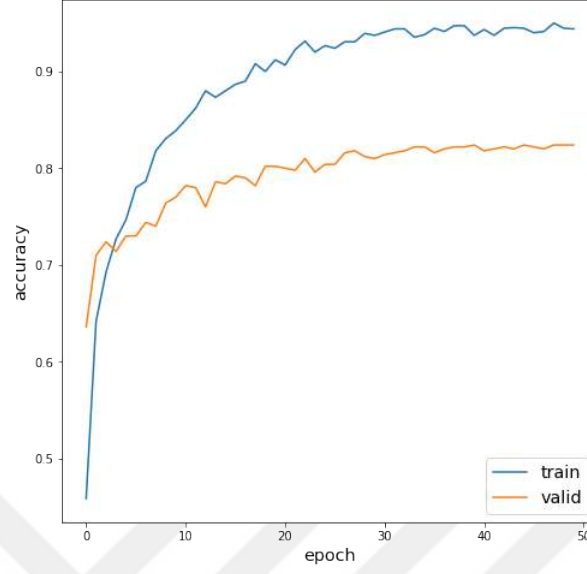
Kapsül ağlar modelinin ilk veri kümesi ile gerçekleştirdiği sınıflandırma uygulamasının sonucunda ortaya çıkan karmaşıklık matrisi Şekil 5.8'de gösterildiği gibidir.



Şekil 5.8. Modelin ilk veri kümesi ile elde edilen karmaşıklık matrisi

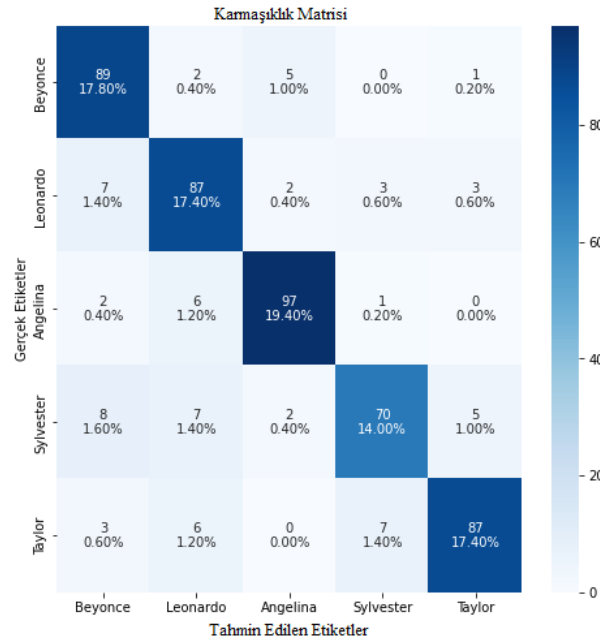
Hazırlanan ikinci yüz görüntüleri veri kümesi, ilk veri kümesinin verilerinin arttırılması ile oluşturulmuştur. Bu veri kümesi üzerinde kapsül ağlar modelinin elde ettiği

test başarımı %86'dır. Modelin ikinci veri kümesi ile elde edilen eğitim ve doğrulama grafiği Şekil 5.9'daki gibidir.



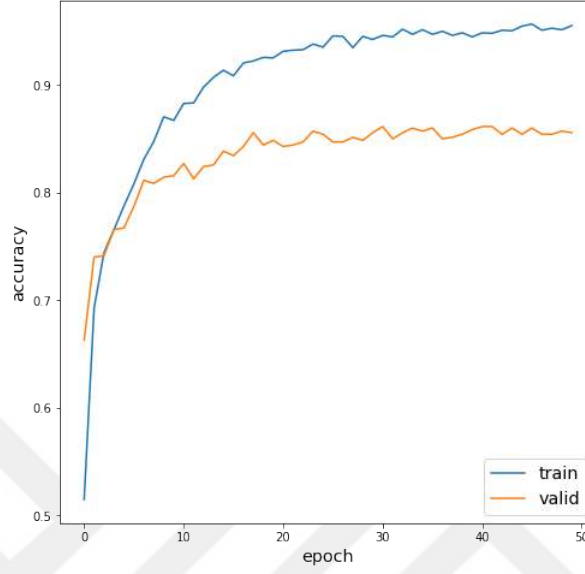
Şekil 5.9. Modelin ikinci veri seti ile elde edilen eğitim ve doğrulama süreci grafiği

Kapsül ağlar modelinin ikinci veri kümesi ile gerçekleştirdiği sınıflandırma uygulamasının sonucunda ortaya çıkan karmaşıklık matrisi Şekil 5.10'da gösterildiği gibidir.



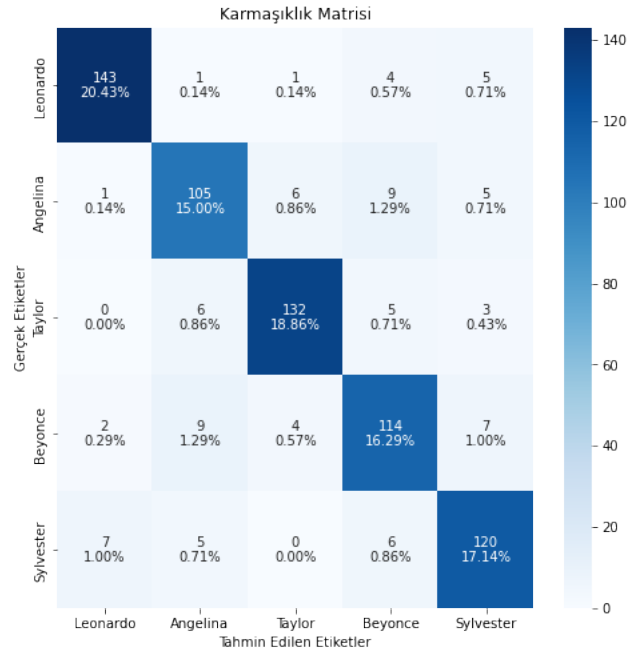
Şekil 5.10. Modelin ikinci veri kümesi ile elde edilen karmaşıklık matrisi

Hazırlanan üçüncü yüz görüntüleri veri kümesi, ikinci veri kümesinin verilerinin arttırılması ile oluşturulmuştur. Bu veri kümesi üzerinde kapsül ağılar modelinin elde ettiği test başarımı %87.7'dir. Modelin üçüncü veri kümesi ile elde edilen eğitim ve doğrulama grafiği Şekil 5.11'deki gibidir.



Şekil 5.11. Modelin üçüncü veri seti ile elde edilen eğitim ve doğrulama süreci grafiği

Kapsül ağılar modelinin üçüncü veri kümesi ile gerçekleştirdiği sınıflandırma işleminin sonucunda ortaya çıkan karmaşıklık matrisi Şekil 5.12'da gösterildiği gibidir.



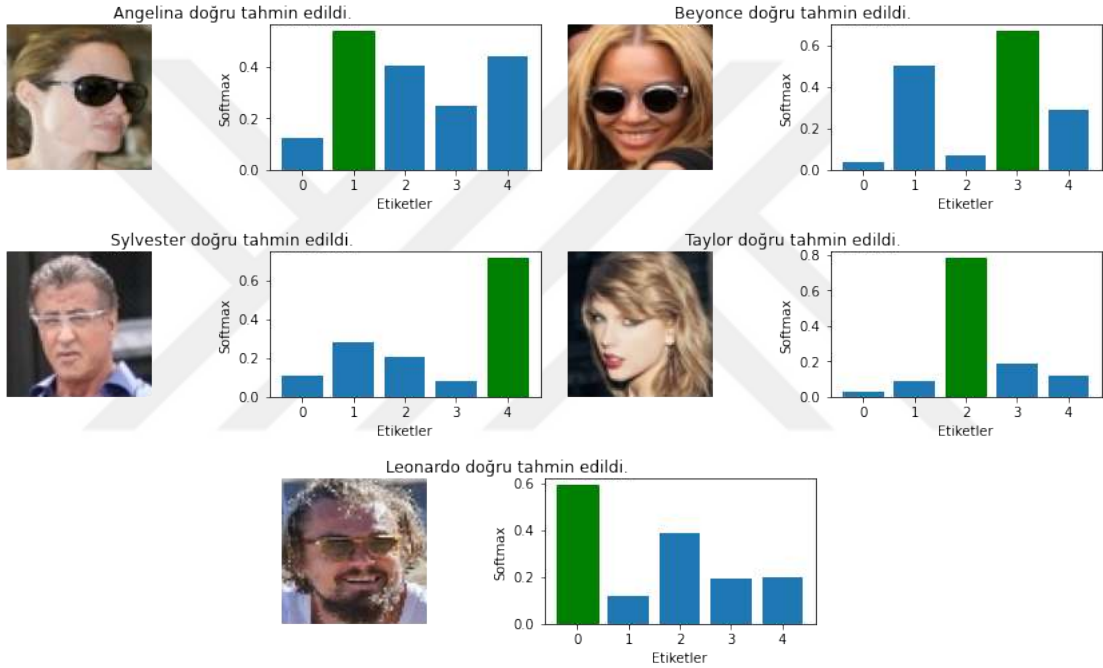
Şekil 5.12. Modelin üçüncü veri kümesi ile elde edilen karmaşıklık matrisi

Üç veri kümesi ile de eğitim işleminden geçirilen modellerin, eğitim süreleri Tablo 5.5’de verildiği gibidir. Kapsül ağılar modelinin eğitim işlemleri için fazla zaman alması

Tablo 5.5. Kapsül ağı modelinin eğitim işlemleri tamamlama süreleri

Veri Kümesi	Eğitim Süresi
1. Veri kümesi	71 dakika 12 saniye
2. Veri kümesi	220 dakika 15 saniye
3. Veri kümesi	491 dakika 24 saniye

problemi, kapsül ağıların dezavantajıdır. Kapsül ağılar modelinin kullanılan yüz verilerinden bazılarının sınıflarını tahmin etmesine dair bir örnek Şekil 5.13’deki gibidir.



Şekil 5.13. Kapsül ağılar modelinin tahminleme örnekleri

Tahminleme örneklerinden de görüldüğü gibi farklı durumlardan alınan örnekler de doğru sınıflanabilmiştir. Kapsül ağılar modelinin ilk veri kümesi üzerinde elde ettiği başarımlar her ne kadar çok iyi olmasa da ikinci ve üçüncü veri setlerinde başarımlar iyileşmiştir. Çalışmada görülmek istenen kapsül ağlarda, farklı boyuttaki veri kümelerinden elde edilen başarımların, birbirine yakın olması ve veri miktarının düşük olduğu durumlarda da başarımın fazla etkilenmemesidir. Bu durum ikinci ve üçüncü veri kümesi için sağlanırken birinci veri kümesi için sağlanamamıştır. Bunun nedeni, birinci veri kümesinin yetersiz örnek barındırması olarak değerlendirilmiştir.

ÖNERİLER

Kapsül ağılar modelinde evrişim katmanlarının sayısı veya hiperparametreleri değiştirilerek daha iyi sonuçlar elde edilebilir. Bununla birlikte modelde özniteliklerin çıkarılması aşamasında farklı teknikler kullanılarak sonuçların incelenmesi, kapsül ağların gelişimi için önemli bir adım olacaktır.

Benzer veri kümeleri ile başka ağlarla da sınıflandırma işlemi yapıp sonuçlar karşılaştırılabilir. Çalışmada yüz verilerini sınıflandırma işlemi küçük veri kümeleri üzerinde incelenmiştir. Gelecekte farklı boyut ve çeşitlikte veri kümeleri üzerinde kapsül ağlar ile oluşturulmuş bir model denenerek daha gelişmiş bir analiz yapılabilir. Farklı derin öğrenme algoritmaları ile karşılaştırma yoluna gidilebilir.



KAYNAKLAR

- [1] İnik, Ö., Ülker, E. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, C.6, Sayı 3, 85-104.
- [2] Sabour, S., Frosst, N. ve Hinton, G. E. (2017). Dynamic Routing Between Capsules, *arXiv preprint*, arXiv:1710.09829
- [3] Dabanoğlu, M. B. (2018). *Makine Öğrenmesi ile Gerçek Zamanlı Videodan Yüz İfadesi Analizi*, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- [4] Bahurupi, S. P., Chaudhari, D. S. (2012). Principal Component Analysis for Face Recognition, *International Journal of Enginnering an Advanced Technology*, C.1, 91-94.
- [5] Bartlett, M. S., Movellan, J. R. and Sejnowski, T. J. (2002). Face Recognition by Independent Component Analysis, *IEEE Transactions on Neural Networks*, C.13, Sayı 6, 1450-1464.
- [6] Lu, J., K., Plataniotis, N. and Venetsanopoulos, A. N. (2003). Face recognition using LDA based algorithms, *IEEE Transactions on Neural Networks*, C.14, Sayı 11, 195-200.
- [7] Huang, L.L., Shimizu, A., Kobatake, H., (2005). Robust face detection using Gabor filter features, *Pattern Recognition Letters*, 26(11), 1641-1649.
- [8] Lowe, D. G. (2004). Distinctive image features from scale invariant keypoints, *International Jorunal of Computer Vision*, C.60, Sayı 2, 91-110.
- [9] Turk, M., Pentland, A. (1991). Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, 3 (1), 71-86. 10.1162/jocn.1991.3.1.71
- [10] Phillips, P. J. (1998). Support Vector Machines Applied to Face Recognition, *Proceedings of the 11th International Conference on Neural Information Processing Systems*, 803-809.
- [11] Guo, G., Li, S. and Chan, K. (2000). Face recognition by support vector machines, *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 196-201.
- [12] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11), 2278-2324.
- [13] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, 25.
- [14] Lawrence, S., Giles, C. L., Tsoi, A. C., Back, A. D. (1997). Face recognition: a convolutional neural-network approach, *IEEE Trans Neural Networks*, C.8, Sayı 1, 98-113. doi:10.1109/72.554195.
- [15] Taigman, Y., Yang, M., Ranzato, M. A. and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701-1708.
- [16] Alimovski, A. (2019). *Derin Öğrenmeye Dayalı Güçlü Yüz Tanıma Sistemi İçin GAN ile Veri Çoğaltma*, Yüksek Lisans Tezi, İstanbul Sabahattin Zaim Üniversitesi, Fen Bilimleri Enstitüsü.
- [17] Yi, D., Lei, Z., Liao, S., Li, S. (2014). Learning Face Representation from Scratch.
- [18] Parkhi, O. M., Vedaldi, A. and Zisserman, A. (2015). Deepface: Closing the gap to humanDeep face recognition, *In BMVC*, C.1, Sayı 3, 6.
- [19] Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W., Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, *arXiv*. doi:10.48550/ARXIV.1602.07360.
- [20] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *arXiv*. doi:10.48550/arXiv.1704.04861.
- [21] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. -C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks, *CVPR*, 4510-4520. doi:10.1109/CVPR.2018.00474.

- [22] Tan, M., Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *Proceedings of the 36th International Conference on Machine Learning*, 6105-6114. doi:10.48550/ARXIV.1905.11946.
- [23] Lee, J. -H., Chan, Y. -M., Chen, T. -Y., Chen, C. -S. (2018). Joint Estimation of Age and Gender from Unconstrained Face Images Using Lightweight Multi-Task CNN for Mobile Applications, *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 162-165. doi:10.1109/MIPR.2018.00036.
- [24] Duong, C. N., Quach, K. G., Jalata, N. L., Luu, K. (2019). MobiFace: A Lightweight Deep Learning Face Recognition on Mobile Devices, *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 1-6. doi:10.1109/BTAS46853.2019.9185981.
- [25] Rouhsedaghat, M., Wang, Y., Ge, X., Hu, S., You, S., Kuo, C.C.J. (2021). FaceHop: A Light-Weight Low-Resolution Face Gender Classification Method, *ICPR International Workshops and Challenges. ICPR 2021*. doi:10.1007/978-3-030-68793-9_12.
- [26] Patrick, M., Adekoya, A., Ayidzoe, M., Baagyere, E. (2019). Capsule Networks – A survey, *Journal of King Saud University - Computer and Information Sciences* doi:10.1016/j.jksuci.2019.09.014.
- [27] Osman, A. A. (2020). *Face Identification Using Capsule Network with Small Data Set*, Master Thesis, Tallin University of Technology.
- [28] Andina, D., Pham, D., Vega-Corona, A., Seijas, J., Torres-García, J. (2007). Neural Networks Historical Review, *Computational Intelligence*, 39-65. doi:10.1007/0-387-37452-3_2.
- [29] Öztemel, Ercan. (2012). Crystal Growth, *Yapay Sinir Ağları*, Papatya Yayıncılık Eğitim, İstanbul.
- [30] Zafari, A., Kianmehr, M.H. and Abdolazadeh, R. (2013). Modeling the effect of extrusion parameters on density of biomass pellet using artificial neural network, *Int J Recycl Org Waste Agricult*, C.2, Sayı 9 doi:10.1186/2251-7715-2-9
- [31] Elmas, Ç. (2003). *Yapay Sinir Ağları*, Seçkin Yayıncılık, Ankara.
- [32] Asilkan, Ö. ve Irmak, S. (2009). İkinci el otomobillerin gelecekteki fiyatlarının yapay sinir ağları ile tahmin edilmesi, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Dergisi*, C.14, Sayı 3, 375-395.
- [33] Kurt, M. S. (2018). *Derin Öğrenme Modelleri ile Web Sayfası Sınıflandırma*, Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü.
- [34] Kızrak, A., Bolat B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma, *Bilişim Teknolojileri Dergisi*. doi:10.17671/gazibtd.419205.
- [35] Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017). Understanding of a convolutional neural network, *International Conference on Engineering and Technology (ICET)*, 1-6. doi:10.1109/ICEngTechnol.2017.8308186.
- [36] Zhou, Y., Chen, S., Wang, Y., Huan, W. (2020). Review of research on lightweight convolutional neural networks, *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 1713-1720. doi:10.1109/ITOEC49072.2020.9141847.
- [37] Yu, F., Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions, *arXiv*. doi:10.48550/ARXIV.1511.07122.
- [38] Zhang, T., Qi, G. -J., Xiao B., Wang, J. (2017). Interleaved Group Convolutions, *2017 IEEE International Conference on Computer Vision (ICCV)*, 4383-4392. doi:10.1109/ICCV.2017.469.
- [39] Sifre, L., Mallat, S. (2014). Rigid-Motion Scattering for Texture Classification, *arXiv*. doi:10.48550/ARXIV.1403.1687.
- [40] Kızrak, M. A., Beser, F., Bolat, B., Yildirim, T. (2018). Kapsül Ağları ile İşaret Dili Tanıma Recognition of Sign Language using Capsule Networks, *26th Signal Processing and Communications Applications Conference (SIU)*, 1-4. doi:10.1109/SIU.2018.8404385.

- [41] Kınlı, F., Kırac, F. (2020). FashionCapsNet: Clothing Classification with Capsule Networks, *Bilişim Teknolojileri Dergisi*, C.13, Sayı 1, 87-96. doi:10.17671/gazibtd.580222.
- [42] Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint*. doi:10.48550/ARXIV.1409.1556.
- [43] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- [44] Doğan, F. ve Türkoğlu, İ. (2018). Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması, *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.
- [45] Saxen, F., Werner, P., Handrich, S., Othman, E., Dinges L., Al-Hamadi, A. (2019). Face Attribute Detection with MobileNetV2 and NasNet-Mobile, *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 176-180. doi:10.1109/ISPA.2019.8868585.
- [46] Qiang, B., Zhai, Y., Zhou, M., Yang, X., Peng, B., Wang, Y., Pang, Y. (2021). SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition, *IEEE Access*, 77661-77674. doi:10.1109/ACCESS.2021.3079337.
- [47] Wang, Y., Yan, J., Sun, Q., Zhao, Y., Liu, T. (2021). ShuffleNet-based comprehensive diagnosis for insulation and mechanical faults of power equipment, *High Volt*, 6(5), 861– 872. doi:10.1049/hve2.12035.
- [48] Gomes, R., Rozario P., Adhikari, N. (2021). Deep Learning optimization in remote sensing image segmentation using dilated convolutions and ShuffleNet, *2021 IEEE International Conference on Electro Information Technology (EIT)*, 244-249. doi:10.1109/EIT51626.2021.9491910.
- [49] Atila, Ü., Uçar, M. , Akyol, K. , Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model, *Ecological Informatics*, 244-249. ISSN:1574-9541.
- [50] Ramachandran, P., Zoph, B., Le, Q. V. (2018). Searching for Activation Functions, *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*. doi:10.48550/arxiv.1710.05941.
- [51] Çoban, A., Özyurt, F. (2022). Kapsül Ağları ile Yüz Verilerinin Sınıflandırılması, *Avrupa Bilim ve Teknoloji Dergisi*, C.33, 176-183. doi:10.31590/ejosat.999055.

ÖZGEÇMİŞ

Ayşe ÇOBAN

[illegible]