T.C.
KONYA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

LİSANS BİTİRME PROJESİ

DERİN ÖĞRENME İLE YANGIN TESPİTİ

Hazırlayanlar

Hilal KELEŞ **181222020**

Tuğba GÖNCÜ **171222102**

Danışman

Dr. Öğr. Üyesi Umut ÖZKAYA

2022

Haziran

Konya

T.C.
KONYA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

LİSANS BİTİRME PROJESİ

Derin Öğrenme ile Yangın Tespiti

Hazırlayanlar

Hilal KELEŞ **Tuğba GÖNCÜ**

181222020 **171222102**

Bu çalışma …../…../2022 tarihinde Komisyonumuz tarafından Elektrik Elektronik Mühendisliği Bölümü LİSANS BİTİRME PROJESİ olarak kabul edilmiştir.

SINAV KOMİSYONU

Prof. Dr. Ramazan AKKAYA
Üye

Öğr. Gör. Dr. Yalçın EZGİNCİ
Üye

Dr. Öğr. Üyesi Umut ÖZKAYA
Danışman

…/…/2022
Prof. Dr. A. Afşin KULAKSIZ
Elk. Elktro. Müh. Böl. Bşk.

ÖZET

Bu çalışmada, yangın tespiti için verilen veriler üzerinden YSA (Yapay Sinir Ağları) yapısı önerilmiştir. Önerilen YSA yapısı test ve eğitim verileri olarak sınıflandırılmış %20 eğitim, %80 test olarak ayrılmıştır. Önerilen YSA yapısı en yüksek performans değerlerine yakın değerler vererek bize bu çalışmada yangın tespitinde büyük ölçekte yardımcı olmuştur.

TEŞEKKÜR

Lisans eğitimimiz ve akademik hayatımızdaki desteğini ve yardımlarını esirgemeyen, bilgi ve deneyimlerini bize aktarıp bu çalışmada bize destek olan danışmanımız Dr. Öğr. Üyesi Umut ÖZKAYA'ya en içten teşekkürlerimizi sunarız

İÇİNDEKİLER

ŞEKİL LİSTESİ

KISALTMALAR

YSA:Yapay Sinir Ağı
CNN: Convolutional Neural NetworkRNN
ReLu: Rectified Linear UnitSSD: Single Shot Detector
YOLO: You Only Look Once
R-CNN: Region- Convolutional Neural
ReLU: Rectified Linear Unit
RGB:Red, Green, Blue

1.GİRİŞ

Gezegenimiz oksijen ihtiyacını karşılayabilmek için çok büyük ölçüde ormanlara ihtiyaç duymaktadır. Bununla beraber ekosistemin de büyük bir parçasında rol oynamaktadır. Bu ekosistemde gözle görülen ve gözle görülmeyen türler mevcuttur ve bu türler dünyayı belli bir denge içinde tutmada görev alır. Dünyada insanların bilinçsiz kullanımı, insan kaynaklı olan ya da olmayan yangınlar ve benzeri olaylar sonucu ormanlarımız ve ormanlarımızda yaşayan türler yok olmakta alansal olarak azalmaktadır.

Bir yangının başlaması için yanıcı madde, yakıcı madde ve tutuşma sıcaklığı şarttır. Bunların yanında gerekli bazı etkenler de mevcuttur : havanın durumu, yakıt nemi , ateşin tutuşması ve yayılması, yakıtın türü.

Yangının her türlüsü küçümsenmeyecek derecede can kaybı ve mal kaybına yol açabilmektedir. Ayrıca orman yangınları dolayısı ile atmosfere salınan karbondioksit gazı da atmosferdeki gazların %25'ini oluşturmaktadır. Karbondioksit gazının en bilindik etkisi ise atmosferde ısı emisyonudur. Karbondioksit gazının bu özelliği çağımızın en büyük problemlerinden biri olan küresel ısınmaya sebep olmaktadır.

Küresel ısınmanın dünya üzerinde birçok olumsuz etkileri vardır. Bu etkilere örnek verecek olursak; buzullar erir, deniz seviyesi yükselir, yeryüzünde büyük miktarlarda su kütleleri buharlaşıp atmosfere karışır ve sıcaklık-basınç farkından dolayı şiddetli rüzgarlar meydana gelir. Bu da şiddetli yağmurları, fırtınaları ve tsunamileri beraberinde getirir.

Ülkemiz orman bakımından zengin olduğu için ve yıllın belli dönemleri özellikle Akdeniz bölgesi yüksek sıcaklığa maruz kaldığı için büyük oranda yangın riski bulunmaktadır. Örneğin 2021 yıllından yaşadığımız orman yangınlarında ,28 Temmuz 2021 de Antalya'nın Manavgat ilçesinden başlayıp ve Türkiye'nin birçok şehrine yayıldı. 2021 itibariyle; çoğunluğu Akdeniz, Ege, Marmara, Batı Karadeniz ve Güneydoğu Anadolu Bölgelerindeki ormanların birçoğu tahrip oldu.

Orman yangınlarında can ve mal kayıplarını en aza indirgemek için erken tespit yapmak büyük rol oynar fakat elimizdeki teknolojinin yetersizliği ve maliyet yüksekliği sebebiyle yangın sonrasında bile yanan arazilere ulaşılması zor ve veri toplamak nerdeyse imkansızdır. Bu sebeplerden dolayı erken tespit için uzaktan algılama teknolojilerini kullanırız.

Orman yangınlarının önüne geçilmesi adına uluslararası da çalışmalar yapılmaktadır. Bu ülkeleri örnek olarak ABD, Kanada, Avusturalya ve Akdeniz ülkeleri başta gelmektedir. Yangın aşamalarında risk tahmini, tespiti ve değerlendirmesi için kameralı sistemlerden faydalanarak bilgisayarlı görü ile orman yangınlarında tespit yöntemi uygulanması hedeflenmektedir.

2.MATERYAL VE YÖNTEM

2.1.Derin Öğrenme

Makine öğrenmesi Derin öğrenmeyi bütünü ile kapsar. Yapay zeka da makine öğrenmesini kapsamaktadır. Yapay zekanın ne olduğundan genel anlamda bahsetmek

gerekirse yapay zeka, bilgisayarın insan davranışlarını taklit etmesini sağlayan teknikleri ifade eden genel bir terimdir. Makine Öğrenimi, bu gereksinimleri giderecek eğitilmiş çeşitli dizileri oluşturmaktadır. Makine öğreniminin alt kümesi olan derin öğrenme ise insan beyninin taklit eden bir tür Makine Öğrenimidir. Derin öğrenme çalışmaları, verileri belirli bir mantıksal çerçevede durmadan analiz ederek, tıpkı insanlar gibi benzer sonuçlar çıkarmaya çalışır. Bunu ortaya çıkartabilmek için derin öğrenme, sinir ağları adı verilen çok katmanlı bir algoritma yapısı kullanır.

Derin öğrenmede kullanılacak verilerin çeşitli özellikleri vardır. Bunlardan bazıları yüksek seviyeli özellikler iken diğerleri düşük seviyeli özelliklerdir. Düşük seviyeli özellikler, yüksek seviyeli özelliklerden oluşur. Derin öğrenme, verilerin kendisinden öğrenmeye dayanır. Derin öğrenme yöntemleri göz önüne alındığında, manuel olarak çıkarılan yöntemler yerine verileri en iyi temsil eden hiyerarşik özellikleri çıkarmak için etkili algoritmalar kullanılır. Sinir ağının tasarımı insan beyni düşünme, tepki gibi görevlerde tatbik edilebilir, ağartılabilir bu araçlarda yaşayabilmek için araçlarını uygulayabilir.

Sinir ağlarının tek katmanları, aynı zamanda, aynı zamanda doğru çalışan bir tür olarak da, bu da doğru bir şekilde bir belirleme ve verme ayarlaması için.

İnsan beyniyle aynı şekilde çalışır. Ne zaman yeni bir bilgi öğrensek, beyin onu hakkında bilgileri karşılaştırmaya yarar. Aynı kavram derin sinir ağları tarafından da kullanılmaktadır.[1]

Sinir ağları, kümeleme, sınıflandırma veya regresyon gibi birçok görevi gerçekleştirmemizi sağlar. Sinir ağları ile etiketlenmemiş verileri bu verilerdeki örnekler arasındaki benzerliklere göre gruplayabilir veya sıralayabiliriz. Veya sınıflandırma durumunda, bu veri kümesindeki örnekleri farklı kategorilerde sınıflandırmak için ağı etiketli bir veri kümesi üzerinde eğitebiliriz.

Eski algoritmalar insanlara bağlı olarak çalışırken, derin öğrenme algoritmaları ayırt edici özellikleri kendi kendilerine öğrenebilirler. Derin öğrenme algoritmalarının makine öğrenmesi algoritmalarından farkı; çok büyük veriler ve bu verileri işleyebilecek yüksek performanslı donanımlar gerektirir. Yüksek performanslı donanım, derin öğrenme geliştirme için verimli bir yapıya sahiptir ve eğitim süresinin kısaltılması esastır.

Derin öğrenme, sistemleri büyük ve karmaşık olasılıksal modellerle eğitme ve öğretme yeteneği sağlayan çok etkili bir yöntemdir, en önemli detay bir veri gösteriminde farklı katmanlar kullanmasıdır. Bu farklı katmanların tümü ayrı ayrı önceden eğitilmiştir. Geleneksel yaklaşımlardan bile daha başarılı sonuçlar verirler.[2]

2.1.1. Derin Öğrenme Kullanım Alanları

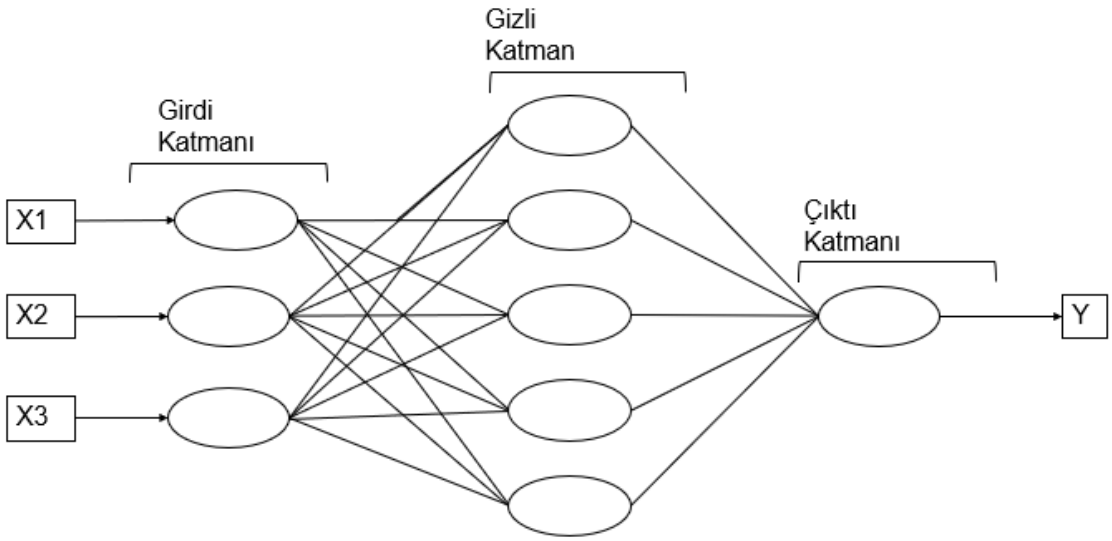
Teknolojinin ilerlemesi ile ortaya çıkan bu yöntem, özellikle; derin öğrenme birçok alanda ihtiyaçlarımızı karşılayacak bir yöntem olarak bilinmektedir. Teknolojinin geldiği son nokta göz önüne alındığında bu yöntem analiz, tespit, otonom araç, savunma sanayi,

sağlık ve güvenlik gibi çeşitli sektörlerde kendine yer bulmuştur. Bunun en büyük nedeni karmaşık problemlerde dahi yüksek doğruluk verebilmesidir.
Algılama sistemleri, özellikle akıllı telefon, bilgisayar veya kapı açma gibi algılamanın gerekli olduğu alanlarda özellikle yüksek güvenlik gerektiren alanlarda tercih edilmektedir. Bu alandaki en bilinen örnek, Google Fotoğraflarım uygulamasında yedeklenen resimlerdeki kişilerin otomatik olarak etiketlenmesi ve sınıflandırılmasıdır.
Telefon bankacılığında ses tanıma sistemi ile banka müşterilerinin daha rahat ve pratik hizmet vermelerine yardımcı olan Derin Öğrenme yöntemi, kendi kendine park eden araçlarda ve otopilot uygulamalarında büyük rahatlık sunuyor. Ayrıca savunma ve güvenlik alanlarında video tanıma özelliği ile Derin Öğrenme yöntemi bu alanda hizmet veren firmaların işini kolaylaştırmaktadır. Örneğin; Bu yöntem ile herhangi bir zamanda kamera kayıtlarının kontrol edilmesi gerekliliği ortadan kalkmakta ve olağandışı hareketlerde alarm sistemi doğrudan devreye girmektedir. Sağlık alanında ise Derin Öğrenme sayesinde teşhis yapılmaktadır. Temel kullanım alanı kanser araştırmaları olan bu yöntemle teşhis süreci çok daha etkili ve hızlı hale gelmektedir.[3]

2.2.Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beynini oluşturan biyolojik sinir ağlarından ilham alan bilgi işleme sistemleridir.
Bir YSA, biyolojik bir beyindeki nöronları gevşek bir şekilde modelleyen, yapay nöronlar adı verilen bağlantılı birimler veya düğümler koleksiyonuna dayanır. Yapay sinir ağlarının öğrenme aşaması verilerle gerçekleştirilir. Öğrenmeye başlayan yapay sinir ağına giriş ve çıkış bilgileri verilir ve belirli parametreler belirlenir.
Sinir ağları, insan beynindeki nöronların birbirine nasıl sinyal gönderdiğinden ilham alan bir tür makine öğrenimi yaklaşımıdır. Sinir ağları, özellikle doğrusal olmayan ilişkileri modellemek için vazgeçilmez bir yöntemdir. Nesne sınıflandırma, ses tanıma, analiz, algılama işlemlerinde kullanılırlar. YSA'lar, şimdiye kadar kullanılan algoritmalardan farklı hesaplama teknikleri kullanır. Ayrıca YSA'lar donanıma bağlıdır.
Sinir ağları, insan beynindeki nöronların birbirine nasıl sinyal gönderdiğinden ilham alan bir tür makine öğrenimi yaklaşımıdır. Sinir ağları, özellikle doğrusal olmayan ilişkileri modellemek için vazgeçilmez bir yöntemdir. Nesne sınıflandırma, ses tanıma, analiz, algılama işlemlerinde kullanılırlar. YSA'lar, şimdiye kadar kullanılan algoritmalardan farklı hesaplama teknikleri kullanır. Ayrıca YSA'lar donanıma bağlıdır.
Yapay sinir ağlarının görüntü ve video üzerinden tespit, nesneleri ayırma , iyi huylu veya kötü huylu olarak sınıflandırmak için patologlara rehberlik ederek kanseri tespit etmek, yüz tanıma, metin çevirisi ve ses tanıma gibi karmaşık tanımlama uygulamalarında makine öğrenimi tekniğinden faydalanmaktadır.

YSA'lar hücrelerinin bir araya gelmesinden oluşmaktadır. YSA'lar temel olarak üç katman içermektedir. İlk katman giriş katmanı , ikinci katman ara (gizli) katman son katman ise çıkış katmanıdır. Şekil 2.1' de basit bit yapay sinir ağının görüntüsü verilmiştir.



Şekil 2.1. Basit YSA Örneği

2.2.1 Yapay Sinir Ağı Katmanları

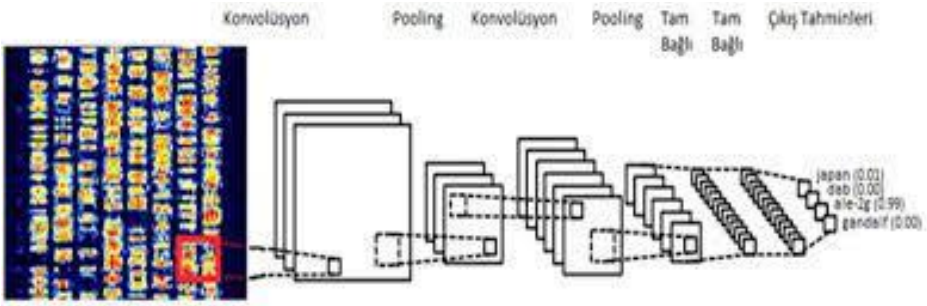
2.2.1.1.Girdi Katmanı (Input Layer):. Bu katmanda öğrenilecek olan veri girdi olarak verilir. Bu veriye ait özellik miktarı kadar da sinir ağı bulunmak durumundadır Genellikle alanın giriş verisi herhangi bir işleme uğramaz ve doğrudan alt katmanlara ulaştırılır. Genel olarak girdi katmanına girdi olarak verilmiş bir veriye ciddi bir işlem yapılmaz doğrudan diğer katmanlara verilmektedir.

2.2.1.2.Çıktı Katmanı (Output Layer): Bu katmanda öğrenilmesi beklenen verilerin sınıflandırılması ve etiketlenmesinin çıkış şeklinde hesaplandığı katmandır. Yapay sinir ağ yapılarına göre bazı ara katman sayıları değişkenlik gösterebilmektedir . Ara katman sayısının artması sınıflandırma ve etiketlenme hesaplarının daha da komplike hale gelmesi demektir. Çıkış katmanı, gelen bilgiyi işleyerek çıktı üretir ve bunu dış dünyaya iletmektedir

2.2.1.3.Gizli Katmanlar (Hidden Layers): Bu katman giriş katmanı ile çıkış katmanı arasında bulunan katmandır.Bu katmanlarda ileri yönlü hesaplamalar ve geri yönlü hata yayılımı yapılır. Katman sayısının çok olması hesaplama karmaşıklığına ve hesaplama süresinin artmasına sebep olur. Daha komplike problemler için veri sayısı, katman sayısı ve nöron sayıları yüksek tutulur. [4]

2.3.Evrişimsel Sinir Ağları

Derin öğrenmede kullanılan ve özellikle bilgisayarlı görü uygulamalarında büyük yol katetmiş olan evrişimsel sinir ağları (CNN) , sıklıkla görüntü tespiti ve analizinde kullanılan çok katmanlı olabilen bir yapay sinir ağıdır. Bu katmanlar kısaca; Giriş (Input) Katmanı, ReLu Katmanı , Havuzlama (Pooling) Katmanı, Konvolüsyon(Convolution) Katmanı, Tam Bağlantılı (Full-Connected) Katman , Sınıflandırma (Classification) Katmanı ve Dropout Katmanlarıdır. Evrişimsel sinir ağlarının katmanlarının en az bir tanesinde olmak üzere geleneksel matris işlemleri yerine evrişim adı verilen işlem kullanılan bir sinir ağı çeşitidir.
Bu sinir ağları obje tespiti, kişi tespiti , otonom sürüşte gerekli olan şerit tespiti gibi çeşit çeşit görüntü işleme temelli problemlerde kullanılan özel bir sinir ağı yapısıdır. Şekilde evrişimsel sinir ağları yapısı görülmektedir.



Şekil 2.2. Evrişimsel Sinir Ağı Yapısı

2.3.1. Evrişimsel Sinir Ağı Katmanları

2.3.1.1.Giriş (Input) Katmanı

Bir sinir ağının giriş katmanı yapay giriş nöronlarından oluşur ve sonraki yapay nöron katmanları tarafından daha fazla işlem için ilk verileri sisteme getirir. Giriş katmanı, yapay sinir ağı için iş akışının en başındadır. 3 Boyutlu matris halindeki görüntüler CNN giriş katmanına verilir. Yani giriş katmanı görüntü halindeki verilerden oluşmalıdır. Örnek olarak 28 x 28 = 784 boyutlarında bir görüntü verisi var, girdi olarak verilmeden önce 784 x 1 boyutuna dönüştürülmesi gerekmektedir. Girdi olarak verilecek olan verinin boyutu büyük olursa eğitim ve test süresi daha uzun sürecek fakat doğruluk ve performans artacaktır. Boyutun küçük tutulması halinde eğitim ve test süresi kısalacak, doğruluk ve performans ise daha düşük olacaktır.

2.3.1.2.Konvolüsyon (Convolution) Katmanı

Görüntünün öznitelikleri konvolüsyon katmanında çıkartılır. Öncelikle görüntünün bir kısmı konvolüsyon katmanına bağlanır (giriş görüntüsünün filtre ile aynı boyuttaki yerel bölgesidir) ve konvolüsyon işlemini gerçekleştirmek ve nokta çarpımını hesaplamak için filtre uygulanır. Ardından, filtre aynı giriş görüntüsünün bir sonraki alıcı alanı üzerinde bir adım ile kaydırılır ve aynı işlem tekrar yapılır. Aynı işlem, görüntünün tamamını gözden geçirene kadar tekrar tekrar yapılır. Çıktı bir sonraki katmanın girdisi olacaktır. Konvolüsyonel sinir ağlarının yeniliği, görüntü sınıflandırması gibi belirli bir tahmine dayalı modelleme sorununun kısıtlamaları altında bir eğitim veri kümesine özgü çok sayıda filtreyi otomatik olarak öğrenme yeteneğidir [5]

2.3.1.3.Havuzlama (Pooling) Katmanı

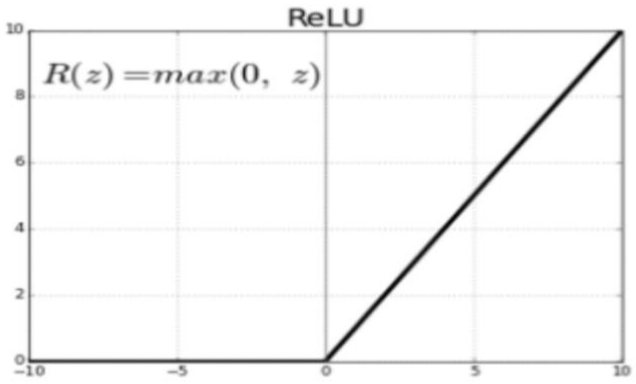
Evrişim katmanından sonra havuzlama katanı eklenir. Bu katmanın amacı evrişim katmanında çıkarılmış öznitelikler daha basit forma indirgenip sıkıştırılmış hale getirtilir. Havuzlama katmanı, aynı sayıda havuzlanmış özellik haritasından oluşan yeni bir set oluşturmak için her bir özellik haritası üzerinde ayrı ayrı çalışır.
Havuzlama, özellik haritalarına uygulanacak bir filtre gibi, bir havuzlama işlemi seçmeyi içerir. Havuzlama işleminin veya filtrenin boyutu, özellik haritasının boyutundan daha küçüktür; özellikle, neredeyse her zaman 2 piksellik adımlarla uygulanan 2×2 pikseldir.
Bu, havuzlama katmanının her bir özellik haritasının boyutunu her zaman 2 kat azaltacağı anlamına gelir, örneğin her boyut yarıya indirilir ve her bir özellik haritasındaki piksel veya değer sayısı boyutun dörtte birine düşer. Örneğin, 6×6 (36 piksel) bir özellik haritasına uygulanan bir havuzlama katmanı, 3×3 (9 piksel) bir çıktı havuzlanmış özellik

haritasıyla sonuçlanacaktır. İki çeşittir:Ortalama Havuzlama ,Maximum Pooling (veya Max Pooling).

2.3.1.4. ReLu Katmanı

Aktifleştirme katmanı olarak da isimlendirilen ReLu Katmanı konvolüsyon katmanlarından sonra gelir. Evrişimsel sinir ağı nöronlarının çıktıları için en yaygın şekilde devreye sokulan doğrultucu birim olarak bilinir. İşlem denklem ile ifade edilecek olursa eşitlikte de olduğu gibi tanımlanmaktadır ve grafiği şekil 2.3' te gösterilmektedir. İnput olarak alınan verilerin negatif değerini sıfıra çekmektedir. Bu katmana geçmeden önceki adımda çeşitli matematiksel işlemler gerçekleştiğinden dolayı doğrusal ağ yapısı vardır. Bu derin ağı doğrusal olmayan bir yapıya sokmak için bu katman uygulanır. Kullanılan bu katman sayesinde ağ daha çabuk öğrenme işlemini gerçekleştirmektedir.[6]

$$f(x) = \{\ 0\ eğer\ x < 0\ ,\ x\ eğer\ x \geq 0\ \}$$



Şekil 2.3. ReLu Katmanının Çıkışa Etkisi

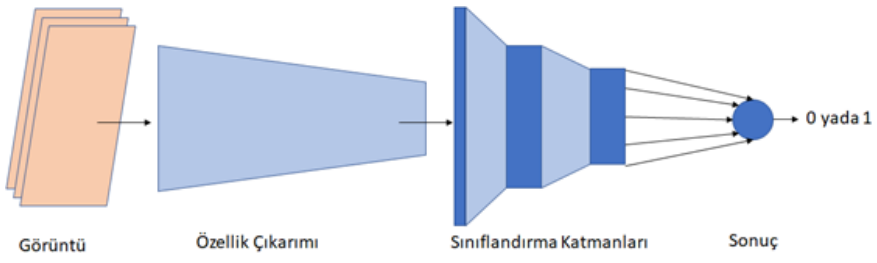
2.3.1.5. Tam Bağlı (Full-Connected) Katman

Tam bağlı katman temel gizli katman birimidir. Geleneksel CNN mimarileri için de, girdi özelliklerinin doğrusal olmayan ilişkilerini daha fazla modellemek için genellikle sondan

bir önceki katman ile çıktı katmanı arasına tam bağlantılı bir katman eklenmektedir.[7] Tam bağlı katman çoğunlukla evrişimsel sinir ağı modellerinde sonlara devreye girmektedir. Bu katmanın asıl amacı sınıflandırmanın doğru bir şekilde tamamlanmasıdır. Burada bulunan tüm nöronlar bir sıra halinde görülmektedir. Bu katman bir önceki katman ile bağlı durumdadır.
Katmandaki nöronların tamamı bu katmana bağlı önceki katmandaki aktivasyonların hepsine tam bağlı durumdadır.

2.3.1.6. Sınıflandırma (Classification) Katmanı

Tam bağlı katmandan hemen sonra devreye giren bu katman sınıflandırma yapmaktadır. Bu katmanda girdi olarak verilen öge miktarınca çıktı alınmaktadır. Örnek olarak 20 farklı türdeki obje sınıflandırması yapılıyor ise bu katmanın çıkış değeri 20 olmaktadır. Tam bağlı katmanda başarısı dolayısıyla softmax sınıflandırıcısı tercih edilmektedir**.** Sınıflandırmada 20 farklı nesne 0-1 aralığında belli bir değerde çıkış üretir. 1'e yakın sonucu üreten çıkış, ağın tahmin ettiği nesne olduğu anlaşılır.

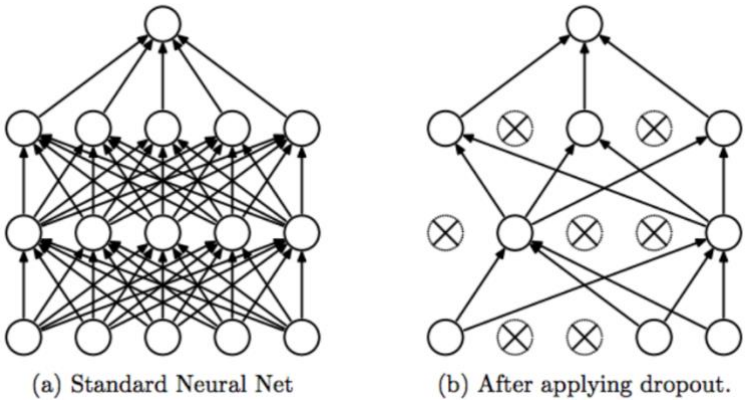


Şekil 2.4. Sınıflandırma Katmanı

2.3.1.7. DropOut Katmanı

Çok katmanlı yapay sinir ağlarında büyük veriler ile eğitim gerçekleştirilirken bazı durumlarda istenmeyen bir durum olarak; ağ ezberleme yapar. Bu noktada ezberlemenin ortadan kaldırılabilmesi için dropout katmanı kullanılmaktadır. Dropout katmanı tam bağlı katmanlar için bir düzenleme katmanı olarak Hinton ve arkadaşları tarafından önerilmiştir Bu katmanda kullanılan esas mantık ağda bulunan bazı düğümlerin ortadan

kaldırılmasıdır. Şekil 2.5 te (a) ağın ilk durumu gösterilmektedir , (b)'de DropOut katmanı uygulandıktan sonraki durumu gösterilmektedir.



Şekil 2.5. (a)Yapay sinir ağı (b)Dropout uygulanmış sinir ağı (Çarpı atılmış nöronlar ağdan çıkarılmıştır)

2.3.2. Evrişimsel Sinir Ağlarında Hiper-Parametreler

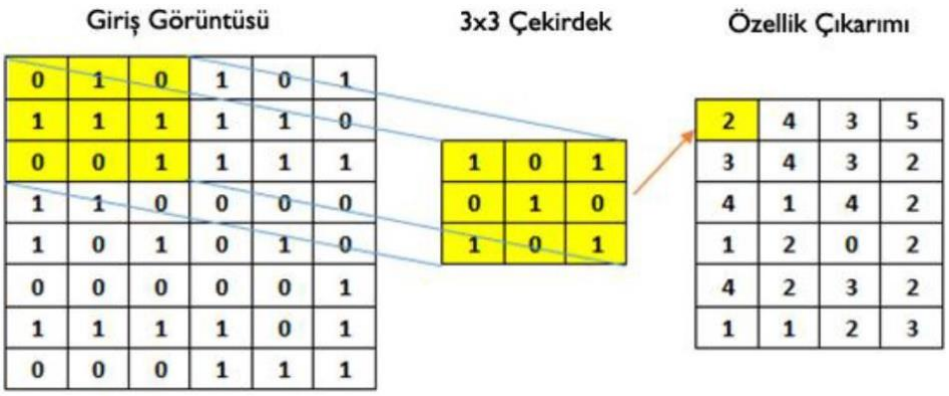
Evrişimsel sinir ağlarında söz edilen hiper parametre; modeli geliştiren kişinin tercihi doğrultusunda, söz konusu amaca ve kullanılacak olan veri setine göre değişkenlik gösteren parametreler hiper parametre (hyperparameters) olarak adlandırılır.
Hiper-parametrelerin uygun seçimi, geliştirilen mimarinin ağırlık, yanlılık gibi model parametreleri üzerinde daha başarılı bir kestirim yapabilmesine olanak tanımakta, bu da sinir ağının eğitimi sonucunda daha başarılı çıktılar elde edilmesini sağlamaktadır. Evrişimsel sinir ağlarında en sık karşılaşılan hiper-parametreler veri seti boyutu, adım aralığı, loss fonksiyonu , gizli katman sayısı, yığın sayısı, dönem sayısı, öğrenme oranı ve aktivasyon fonksiyonudur. Ne olması gerektiği, modeli tasarlayan kişiye bırakılmış, probleme, veri setine göre değişiklik gösteren parametreler hiper-parametre (hyperparameters) olarak adlandırılmaktadır. Mesela hiper-parametre KNN sınıflandırmasında k'nın ne olacağına mimariyi geliştiren karar vermektedir. Aynı şekilde SVM çalışmasında kullanılacak olan kernel fonksiyonu tercihini modeli geliştiren karar vermektedir. Bu örneklere benzer olarak derin öğrenme modellerinde veri seti boyutu , gizli katman sayısı gibi değerlere mimariyi geliştiren karar vermektedir. Bu değerlerin doğru şekilde belirlenebilmesi için modeli geliştiren kişi modeli, amacı, veri seti boyutu gibi etkenleri göz önüne alarak tercih yapmalıdır.[8]

2.3.2.1. Veri Seti Boyutu

Veri seti boyutu diğer algoritmalar gibi derin öğrenme algoritmalarında da büyük önem arz etmektedir. Bir mimarinin başarılı olabilmesi için veri setinin de yeterli bir büyüklükte olması gerekmektedir. Veri seti boyutu ne kadar fazla olursa eğitimde geçen süre de aynı oranda uzayacaktır. Veri seti amaca uygun olarak çeşitli biçimlerde olabilir. Örneğin ses tanıma ile ilgili çalışma yapılmakta ise veri seti ses kayıtlarından oluşur. Obje tanıma ile ilgili çalışma yapılmakta ise veri seti görüntülerden oluşmaktadır. Veri seti boyutu büyüklüğünün yanında başka önemli noktalar da vardır. Örneğin veri seti görüntü içeriyor ise bu görüntüler farklı çeşitlerde görüntüler olmalıdır.

2.3.2.2. Adım Aralığı (Stride)

Sinir ağlarındaki filtreler görüntü pikselleri üzerinden teker teker ilerlemektedir. Bu ilerleme sayısına adım aralığı (stride) denmektedir. Adım aralığı değeri eğer fazla seçilirse filtre de büyük olacağından, filtre görüntü üzerinde daha az gezmiş olup daha küçük bir görüntü matrisi ortaya çıkarmış olur. Eğer küçük seçilirse daha fazla alanda gezeceği için daha büyük bir görüntü matrisi ortaya çıkar. Öznitelikler de buna göre değişken olmaktadır. Şekil 2.6 da adım aralığı (stride) örneği görülmektedir.



Şekil 2.6. 3x3 Filtre görüntü üzerinde (1,1) adım aralığı ile ilerlemektedir

2.3.2.3. Loss Fonksiyonu

Derin öğrenme sinir ağlarındaki en son katmanda loss fonksiyonu görev almaktadır. Bu fonksiyon mimarinin hata miktarını belirlerlemekte ve bunun yanında mimariye ait başarı

oranını da hesaplamaktadır. Loss fonksiyonunun asıl amacı mimarinin eğitim sonrası tahmini ile gerçek problem arasındaki farkın miktarını belirlemektir. Özetlemek gerekirse başarılı bir mimarinin tahmini ile problemin gerçek değeri arasında loss fonksiyonunun değeri de oldukça düşük olacaktır. Loss fonksiyonunun büyük bir değerde olması mimarinin başarısız olduğunu gösterir. Loss fonksiyonun 0'a yakın olması beklenmektedir.[9]

2.3.2.4. Gizli Katman Sayısı

Derin öğrenme ve Yapay Sinir Ağlarındaki en belirgin farklardan biri katman sayılarıdır. Mimaride bulunan gizli katmanları arttırmak başarı sağlamaktadır.Fakat bunun yanında hesaplama zorluğunu da beraberinde getirir.

2.3.2.5. Yığın Sayısı (Batch Size)

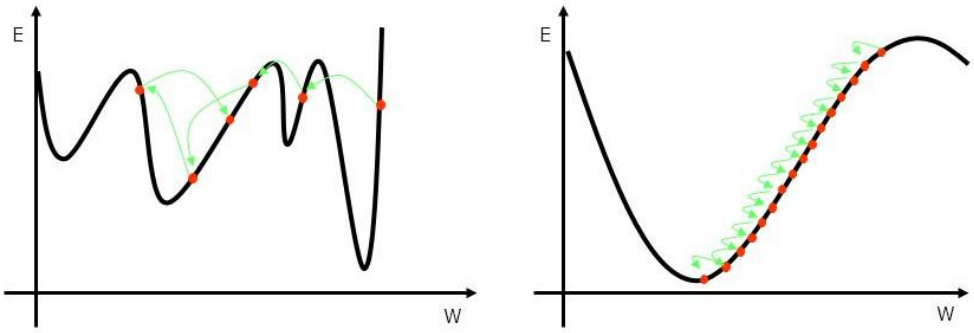
Eğitim yapılırken mimarideki sinir ağlarına gruplar halinde veriler yollanır. Bu grupların ne kadarlık yığınlar halinde olacağına mimariyi geliştiren karar verir. Bu grupların miktarı yığın sayısı (batch size) olarak adlandırılmaktadır. 16-128 arasında seçilebilen bu sayılar 2 nin üstleri olacak biçimde tercih edilmelidir.

2.3.2.6. Dönem Sayısı (Number Of Epoch)

Dönem sayısı, tüm veri setinin sinir ağından geçme sayısı olarak tanımlanmaktadır. Modelin eğitim sonucu ile test sonucu arasındaki hata en aza inene kadar dönem sayısı arttırılmalıdır. [9]

2.3.2.7. Öğrenme Oranı (Learning Rate)

Derin öğrenme çalışmalarındaki hiperparametreler arasında bulunan learning rate yani öğrenme oranı, hata katsayısı olarak da adlandırılmaktadır. Öğrenme oranının büyük olması halinde minimum olan noktaya ulaşılamaz, düşük olması halinde ise model daha geç öğrenmektedir. Deneme yanılma yolu ile kullanılacak en iyi öğrenme katsayısı bulunup kullanılmalıdır. Şekil 2.7 de öğrenme katsayısı farkı görülmektedir

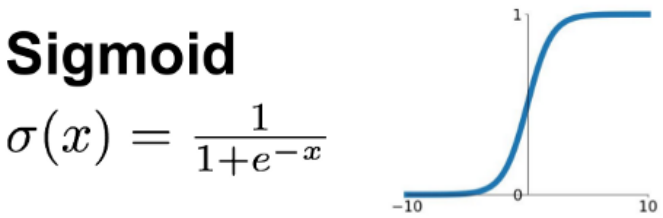


Şekil 2.7. Öğrenme katsayısının büyük olması durumunda grafik (sol), öğrenme katsayısının küçük olması durumunda grafik (sağ)

2.3.2.8. Aktivasyon Fonksiyonu

Aktivasyon fonksiyonunun amacı (doğrusal olmayan) girdini çıktıya eşlenmesini sağlamaktır. . Girdi ve eğer mevcut ise yanlılığı ile beraber ağırlıklı toplamı hesaplamaktadır. Bu işlemler bütünün özü, hesaplanmış y değerinde o nöronun yapısına göre nöronun aktif olup olmayacağına hüküm vermesidir. Bu fonksiyonların esas hedefi yapay sinir ağlarını non linear yani doğrusal olmayan hale getirmektir. Yani aktivasyon fonksiyonları çok katmanlı sinir ağlarında non linear hesaplamalarında kullanım olarak öne çıkmaktadır. Hesaplamalar için seçilen aktivasyon fonksiyonuna çıktı olarak verilecek sonuçta büyük görev düşmektedir. Alınacak sonuçtan iyi bir performans elde edilebilmesi için ise aktivasyon fonksiyonu amaca en uygun şekilde tercih edilmelidir.[10] Çeşitli aktivasyon çeşitleri mevcuttur. Bunlara örnek verecek olursak;

Sigmoid: Girdi olarak gerçek sayılar verilir. Çıkış olarak 0 ve 1 arasında sonuç alınır. Fonksiyonun matematiksel denklem ve grafiği şekil 2.8'de verilmiştir.



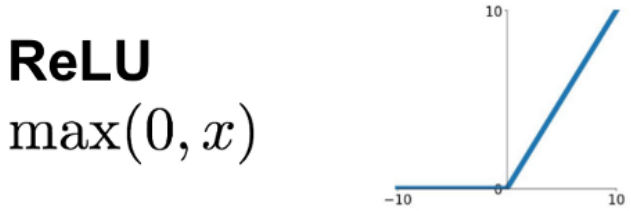
Şekil 2.8. Sigmoid fonksiyonun grafiği ve denklemi

- **Tanh:**Girdi olarak reel sayılar verilmesi sebebi ile sigmoid fonksiyonuna benzer fakat çıktılar -1 ile 1 arasında sonuç alır. Fonksiyonun grafiği şekil x te verilmiştir.

Şekil 2.9. Tanh fonksiyonunun grafiği

- **ReLu**: CNN' de sıklıkla tercih edilmektedir. Girdi olarak hangi sayı verilirse verilsin pozitife çevirmektedir. Dasit bir yaklaşıma sahiptir ve daha kolay eğitilebilmektedir.

.



Şekil 2.10. ReLu fonksiyonunun grafiği

2.3.3. Evrişimsel Sinir Ağlarında Transfer Öğrenme

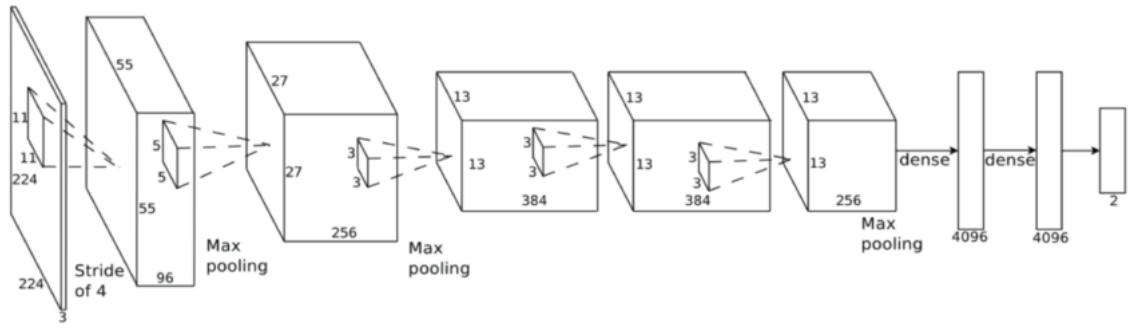
Transfer öğrenme kısaca başka bir problem çözümü için eğitilmiş modelin yeni bir problemi çözebilmek için tekrardan kullanılmasıdır. Basit bir örnek verilecek olursa; hayvanat bahçesi görüntüsünde köpek olup olmadığını tahmininin yapılabilmesi için sınıflandırıcıyı eğitirken hayvanları ayırt edebilmek için de bir önceki eğitim esnasında kazanılmış bilgiler kullanılabilir . Transfer öğrenmesi özellikle derin öğrenmede büyük boyutlu verisetine ihtiyaç duyulmadığı için oldukça tercih edilen bir yaklaşımdır.

Evrişimsel sinir ağları için kullanılan çeşitli transfer öğrenme yaklaşımları mevcuttur. Bunlara örnek olarak ; özellik çıkarıcı metot , kısmı özellik çıkarıcı metot.

2.3.4. Evrişimsel Sinir Ağı Modelleri

2.3.4.1.AlexNet

Alex Krizhevsky Ilya Sutskever ve Krizhevsky tarafından 2012 yılında geliştirilen bu çalışma sayesinde derin öğrenme çalışmaları tekrardan ün kazanmıştır. AlexNet mimarisi 8 katmandan oluşmaktadır. Bunların 5'i evrişim katmanı 3'ü havuzlama katmanlarını içermektedir . AlexNet mimarisinde aktivasyon fonksiyonu görevi için relu fonksiyonu tercih edilmiştir. Normal şartlarda mimarilerde kullanılan tanh ve sigmoid'e göre daha hızlı bir performans sergilemesi tercih edilme sebebi olmuştur . Yaklaşık 60 milyon parametrenin kullanıldığı AlexNet, paralel çift GPU üzerinde çalışan ilk model olma özelliğini de taşımaktadır. ImageNet ILSRVC yarışmasında sınıflandırmada sergilemiş olduğu doğruluk başarısı %74,3'ten %83,6'ya doğru bir artış göstermiştir. Ayrıca geliştirilen mimari sayesinde overfitting denilen aşırı öğrenme probleminin önüne geçilmiştir.[11]
Şekil 2.11'de AlexNet mimarisi gösterilmektedir.

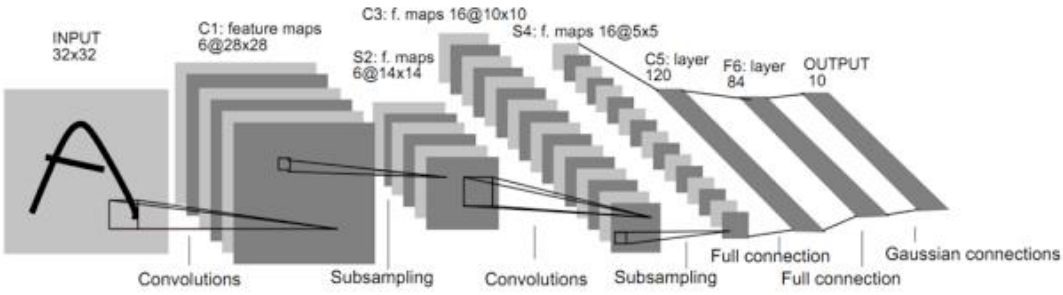


Şekil 2.11. AlexNet Mimari Yapısı

2.3.4.2. Lenet-5

Lenet-5, Yann LeCun ve diğerleri tarafından 1998 yılında Gradient-Based Learning Applied to Document Recognition adlı araştırma makalesinde önerilen en eski önceden eğitilmiş modellerden biridir. El yazısı ve makine baskılı karakterleri tanımak için bu mimariyi kullandılar. Lenet-5 mimarisinin ün kazanmasının asıl nedeni, sade ve anlaşılır

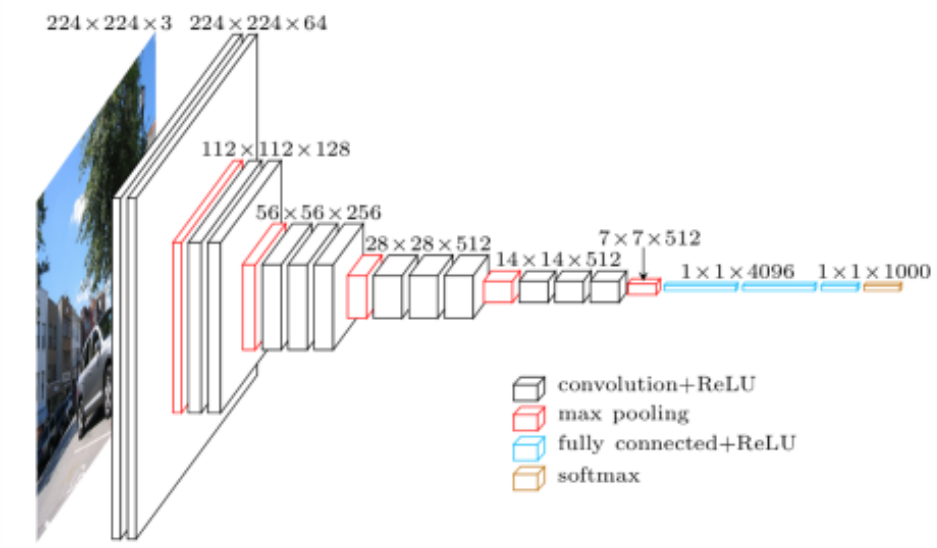
bir mimariye sahip olmasıdır. Görüntü sınıflandırmasında terch edilen için çok katmanlı bir evrişim sinir ağıdır. Bu modelin öteki modellerden farkı boyut küçültme adımında ortalama ortaklama hesabının tercih edilmesidir. Bunlara ek olarak aktivasyon fonksiyonu olarak sigmoid ve hiperbolik tanjant kullanılmaktadır.[13]
Şekil 2.12'de Lenet-5 mimari yapısı gösterilmektedir.



Şekil 2.12. Lenet-5 Mimari Yapısı

2.3.4.3. VGG

VGG, klasik bir evrişimli sinir ağı mimarisidir. VGG, bu tip ağların derinliğinin nasıl arttırılmasına gerektiğine dair bir incelemeye dayanmaktadır. Ağ, küçük 3 x 3 filtreler kullanmaktadır. Aksi takdirde, ağ basitliği ile karakterize edilir: diğer bileşenler yalnızca havuz katmanları ve tamamen bağlı katmanlardır.VGG mimarisi ile takribi olarak 138 milyon parametrenin hesabı yapılmaktadır.[13]
Şekil 2.13'te VGG mimarisi gösterilmektedir.



Şekil 2.13. VGG Mimari Yapısı

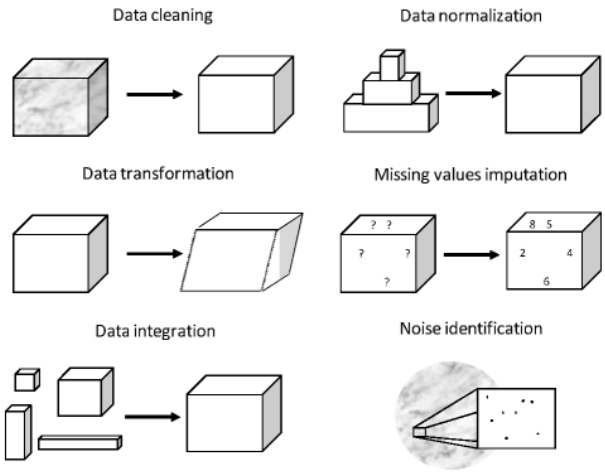
2.3.5. Nesne Tespit ve Takip Algoritmaları

Gerçek zamanlı tespit, bir dizi görüntü üzerinden nesneleri tespit etmeye, takip etmeye çalışan ve aynı zamanda eskiyen geleneksel yaklaşımı değiştirerek nesne davranışını anlamaya ve tanımlamaya çalışan bilgisayarlı görü çalışmalarında aktif bir araştırma konusudur. Nesne algılama ve izleme, gözetim, araç navigasyonu ve otonom robot navigasyonu gibi birçok bilgisayarlı görü çalışmalarında önemli ve zorlu görevlerdir. Nesne tespiti bir video dizisi etrafında nesnelerin konumlarının tespit edilmesini içermektedir. Her izleme yöntemi, görüntüde ya da videoda nesne ilk kez göründüğünde tespit edilebilmesi için bir nesne tespit sistemi gerektirmektedir.[14]
Görüntü üzerinden sonuçlar elde edebilmek için çeşitli sayısal görüntü işleme uygulamaları tercih edilmektedir. Fakat gerçek zamanlı tespit ve takip analizlerinde sonuçlar elde edilebilmesi görüntüye göre daha zordur. Bu sebeplerden dolayı derin öğrenme algoritmaları kullanmak işi kolaylaştıracaktır. Görüntüler üzerinde nesnelerin hızlı bir şekilde tespitini gerçekleştiren ilk algoritma Viola Jones (Viola ve Jones, 2001) olmuştur. Derin öğrenme kullanılarak obje analizi ve takibi hakkındaki uygulamalar incelendiğinde çoğunlukla dört adım uygulanmaktadır;

- **Görüntünün Ön İşlemesi**

Projede kullanılacak olan veri setinin girdi olarak verilmeden önce yapılan düzeltmelere ön işleme denir. Ön işleme ;veri setinde kendini tekrar eden veriyi temizlemek, eksik halde olan verileri tamamlamak ve bunun dışında veri boyutu, optimizasyon, dönüştürme

gibi işlemler bütünü veri ön işlemesini içermektedir. Her zaman ön işleme yapılması şart değildir.



Şekil 2.14. Veri ön işlemesi aşamaları örnek çizimi

- **Nesnenin Tespiti**

Gerçek zamanlı nesne tespiti iki aşama ile gerçekleştirilmektedir. Birinci adımda nesne belirgin hale getirilir. İkinci adımda ise nesne arka plandan ayrılır. Nesne tespitinin doğru bir şekilde yapılabilmesi için probleme uygun bir algoritma tercih edilmesi gerekmektedir.

Uzun bir süredir nesne analizi ve takibi çalışmalarında son derece önemli gelişmeler elde edilmiştir. Nesne algılama gerçekleştirilirken elimizde bulunan veri üzerinde nesne taraması yapılır. Ufak bir örnekle açıklanacak olursa; elimizde yangın var, yangın yok olmak üzere iki etiket var, bu etiketler ile model eğitilir. Eğitim sonrası test aşamasında görüntüde yangın olan ve olmayan noktalar belirlenip kare içine alınır. Bu örnek 2 sınıflı olan bir çalışma örneği içindir. Daha fazla sınıf içeren çalışmalarda ise bir görüntü üzerinde birden fazla nesne taraması yapılır. Nesne algılamayı basit sınıflandırmadan ayıran en belirgin özellik görüntü üzerinde birçok nesneyi tanıyor olmasıdır.

- **Nesnenin Sınıflandırılması**

Veri setindeki bir görüntünün sınıflandırma aşaması gerçekleştirilirken görüntüde bulunan objenin niteliklerine uygun olarak farklı gruplara bölünmesine nesne sınıflandırılması denilmektedir. Görüntünün niteliklerine uygun olarak

sınıflandırılmasından kasıt, görüntüdeki eşit değeri, kontrastı, piksel değeri ve benzeri olarak açıklanabilir.

- **Nesnenin Takibi**

Nesne takibinin gerçekleşebilmesi için takipten önce elimizde bulunan materyaldeki takibi yapılması hedeflenen nesnenin tespitinin yapılması birincil şarttır. Tespiti yapılanan nesne etiket altına alındıktan sonra takibi de bu etiket üzerinden gerçekleşir.
Nesne takibinin yapılmasının en önemli basamağı doğru nesne tespiti yapmaktır**.**
Nesne tespiti ve takibini gerçeğe en yakın ve hızlı bir şekilde gerçekleştirebileceğimiz çeşitli alternatifler mevcuttur. Bunlara örnek olarak SSD, YOLO, R-CNN , Fast R-CNN, Faster R-CNN verilebilir. Rcnn işlemini 3 adımda gerçekleştirirken faster r CNN tek adımda gerçekleştirmektedir. Ağa verilen görüntü input olarak alınır ve hemen ardından sınıf bulunma ihtimali ve sınırlayıcı kutular oluşturulur. R- CNN de her bölge önerisi ayrı yapılır. Örnek verilecek olursa bir bölgenin tespiti 2 saniye sürerse 5 bölgenin tespiti 10 saniye sürer. Faster R-CNN tüm bölge önerisi ve hesaplamaları tek seferde yaptığı için daha hızlıdır. SSD ise boyutu küçük objeler ele alındığında Faster R-CNN e göre daha düşük performans sergilemektedir. SSD algoritması boyutu düşük objeleri yalnızca çözünürlüğü fazla katmanlarda analiz edebilir. Tüm bunlar göz önüne alındığında R-CNN algoritmasına göre daha fazla hata yapmaktadır.

3.DENEYSEL KURULUM

3.1.Problemin Belirlenmesi

Dünyada her yıl binlerce bölgede büyük ya da küçük çaplı yangınlar çıkmaktadır. Bu yangınlar kimi zaman erken müdahale ile söndürülüp can ve mal kaybına yol açmazken kimi yangınlar büyük kayıplara sebebiyet vermiştir. Orman yangınlarının en yaygın sebepleri arasında insan kaynaklı olan ihmal ve tedbirsizliktir. Fakat insan kaynaklı yangınlar dışında özellikle ülkemizde Ege, Akdeniz ve Marmara bölgelerinde özellikle de yaz mevsiminde sıcaklık, yağış, rüzgar gibi iklimsel sebeplerden ötürü yangına elverişli ortamlar oluşmaktadır. Bu tez çalışmasında ise sinir ağları yardımı ile bir sistem geliştirilmesi hedeflenip bu olumsuz durumun önüne geçilmesi ya da erken tespiti amaçlanmıştır.

3.2.Problem Analizi

Yangınlara erken müdahale için erken tespitin yapılması şarttır. Erken tespit için ise bir sistem tasarımı yapılmalıdır. Bu sistem yangın çıkan herhangi bir ortamı kamera ile okuyup derin öğrenme algoritmaları ile tespitini yapmalıdır.

4.GELİŞTİRİLEN YÖNTEM

4.1.Problem İçin En Uygun Sinir Ağı Modelinin Tensorflow İle Oluşturulması

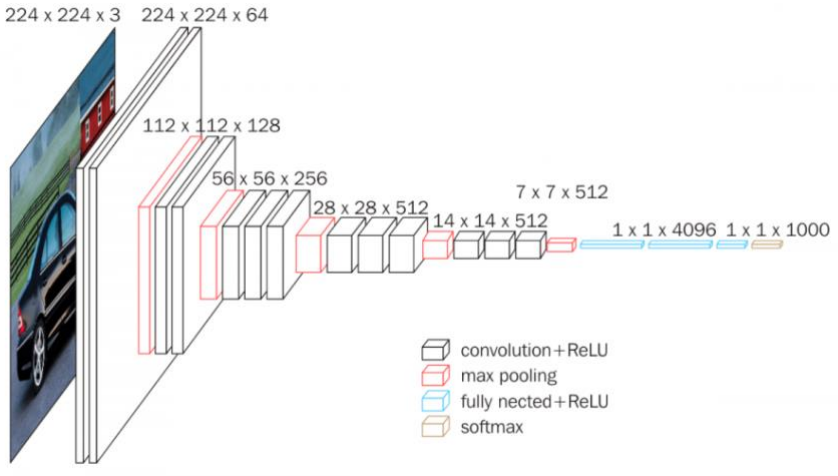
TensorFlow, Google tarafından geliştirilmiş derin ve makine öğrenimi için popüler bir çerçevedir. Ücretsiz ve açık kaynaklı bir yazılım kütüphanesidir. Birden çok veri kümesine sahip yapay sinir ağlarıyla çalışma alanı sağlar. TensorFlow'un üstünde tflearn, tf-slim, skflow vb. kullanabilirsiniz. En popüler TensorFlow uygulamaları arasında nesne tanımlama, nesne tespit etme, konuşma tanıma, sınıflandırma ve daha fazlası bulunur.Bu bilgisayarlı görü uygulamaları geliştirmek için tensorFlow API'lar yararlanılır .TensorFlow API'lar transfer öğrenme yöntemini kullanarak önceden hazır eğitilmiş nesne algılama modellerinin kullanılmasını yada yeni modeller hazırlanmasını ve eğitilmesini sağlar . Temelinde Python programlama dili kullanımı ile geliştirilmiş olan TensorFlow API'lar Python dışında C++, Java, C#, JavaScript gibi birçok farklı programlama dilini de desteklediği görülmektedir.
Kullanılacak olan bir tensorflow API'sinin başlıca yararlandığı kütüphane keras kütüphanesidir. Buradaki çalışmada da modelleri oluşturmak için Python yazılım dili ve Keras kütüphanesi kullanılmıştır.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Data augmentation ve generator (klasörlerden)
train_augmentation = ImageDataGenerator(
                                        rescale = 1. / 255)
train_generator = train_augmentation.flow_from_directory(
                                        directory = training_path,
                                        target_size = (224, 224),
                                        batch_size = 16)
val_augmenatation = ImageDataGenerator(
                                        rescale = 1. / 255)
val_generator = val_augmenatation.flow_from_directory(
                                        directory = val_path,
                                        target_size = (224, 224),
                                        batch_size = 1)
test_augmenatation = ImageDataGenerator(
                                        rescale = 1. / 255)
test_generator = test_augmenatation.flow_from_directory(
                                        directory = test_path,
                                        target_size = (224, 224),
                                        batch_size = 1)

# Model ve eğitim
from tensorflow.keras.applications.vgg16 import VGG16
model = VGG16(weights = None, classes=2)
model.summary()
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import categorical_crossentropy
opt = Adam(lr=0.001)
model.compile(optimizer = opt, loss = categorical_crossentropy, metrics=['accuracy'])
model.fit_generator(train_generator, epochs = 2000, steps_per_epoch = int(3044 / 16),
                    validation_data = val_generator, validation_steps = 389,
                    verbose = 1)
```

Şekil 4.1. VGG16 ile oluşturulan modelin yapısı

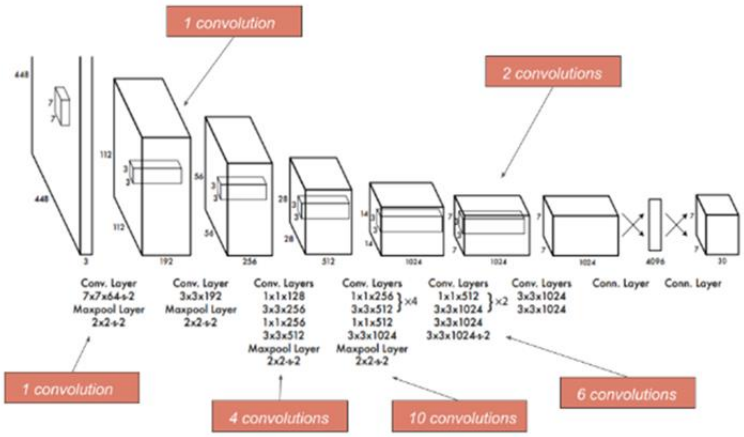


Şekil 4.2. VGG16 modellinin ağ yapısı

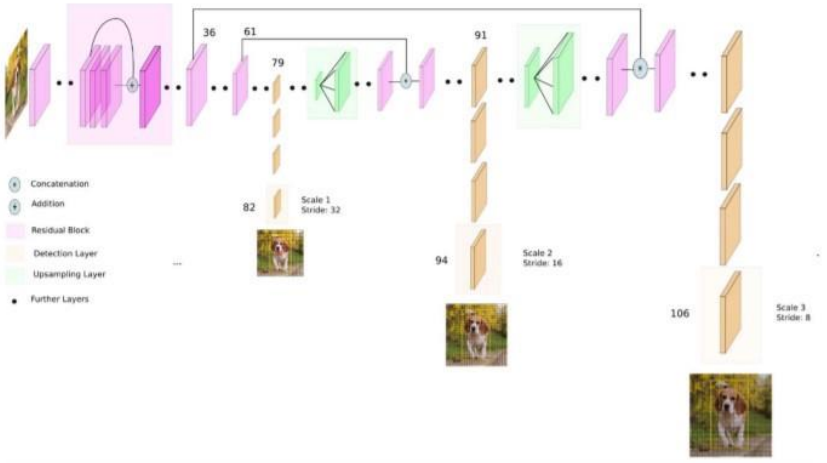
4.2.Problem İçin En Uygun Sinir Ağının Seçilmesi

Yolo, bir nesne algılama odaklı ve yalnızca bir sinir ağından oluşur. Yolo, görüntüleri veya video karelerini girdi olarak sayar ve sınırlayıcı kutular ve bir sınıfın bu kutular içinde olma olasılığını üretir. Gerçek zamanlıdır.
Yolo'nun Fast Yolo veya Tiny Yolo olarak çeşitleri vardır, bu alternatifler hızlıdır ancak daha az doğruluk vermektedir.
Yangın tespitini doğruluğu yapmasının yanında, çalışma alanında yangın olan bir alanda hızlı tespit etmek de oldukça önemli olmaktadır. Bundan dolayı bu problem için hem yüksek hızlı hem de yüksek doğruluk da çalışan YoloV3 algoritması ile tercih edilmiştir ve framework olarak Darknet'ten yararlanılmıştır.
YoloV3 algoritması girdi resmini NxN'lik parçalara bölmektedir. Her parça, alan içerisinde nesne olup olmadığını bulmaktadır. Bu parçaların her biri sınırlayıcı kutuların adeti kadar tahmin vektörü oluşturmaktadır. Sınırlayıcı kutu metodunun kullanılmasının amacı ise herhangi bir parça da birden fazla nesne bulunma durumunda algoritmanın bu parça da bulunan tüm nesneleri tespit etmektir. Tahmin vektörü içerisinde ise güven skoru, nesnenin orta noktasının enlem(x) ve boylam(y) koordinatı, bağlı sınıf olasılığını içermektedir. Güven skoru, parçanın içinde nesne olup olmadığından ne kadar emin olduğunu göstermektedir. Bağlı sınıf olasılığı ise model de bulunan sınıf sayısı ile doğru orantıdadır. Yangın tespiti problemi için model de 2 sınıf mevcut olduğu için 2 tane bağlı sınıf olasılığı olacaktır. Nesne tespitinin ardından temel amaç tespit edilen nesnenin etrafına bir dikdörtgen çizmektir.
YoloV3'de genel kaybı etkileyen birçok neden bulunmaktadır. Bunlar; sınıflandırma kaybı, konum kaybı ve güven kaybıdır.
Sınıflandırma kaybı, tahmini yapılan objenin hata miktarıdır. Konum kaybı, tahmini yapılmış olan objeye ait kutunun hata miktarıdır. Bu üç fonksiyonun toplanmış hâli genel hata toplamına

eşittir. Yolov3 ye yüksek doğruluğa ulaşabilmek için hata toplamı düşük tutulmalıdır. [16].
YoloV3'de genel hata toplamı ne kadar düşük olursa o kadar yüksek doğruluğa ulaşılmaktadır.



Şekil 4.3. YOLO mimarisi

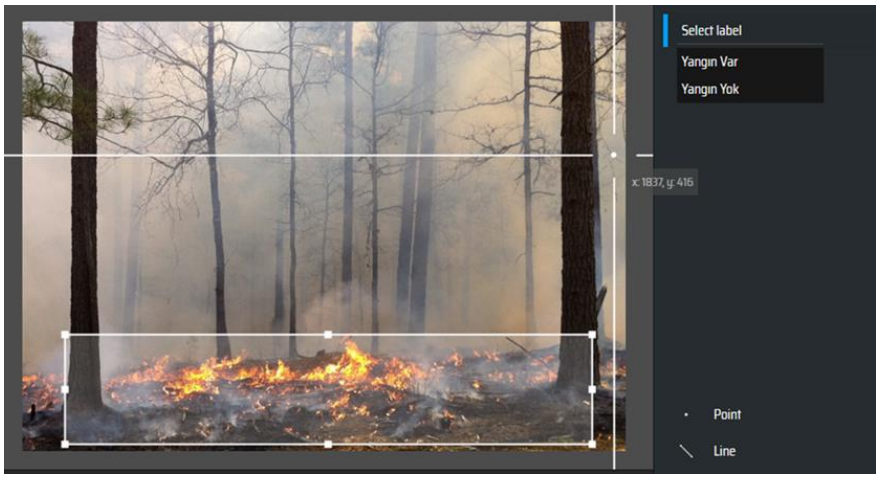


Şekil 4.4. Yolov3 mimarisi

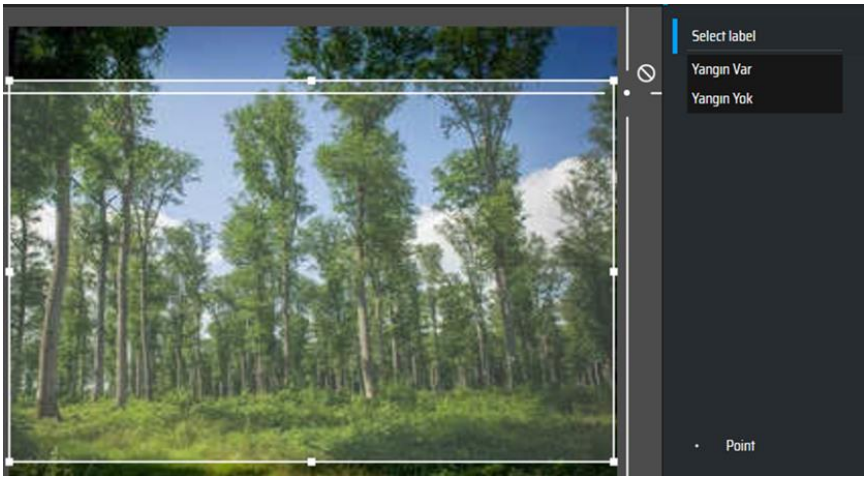
4.3.Veri Setinin Oluşturulması Ve Etiketleme İşlemi

Problem için kullanılacak veri seti büyük ve çeşitli olması öğrenme ile doğru orantılıdır. Ayrıca veri setinin büyüklüğü eğitime harcanan zamanı ve eğitim sonucunda oluşacak dosyanın büyüklüklüyü ile doğru orantılıdır. Ama başarılı bir model için sadece veri setinin büyüklüğü yeterli değildir, aynı zamanda veri çeşitliliği de modelin başarısında önemli bir etkendir. Bunun için aynı görüntüden farklı açılarda, farklı renklerde, farklı ölçeklerde ve farklı gürültülerde verilerin olması önemlidir.

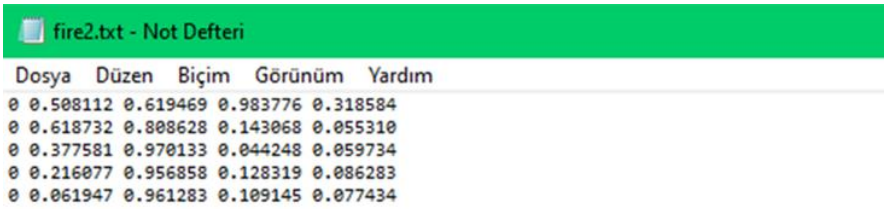
Bu tez kapsamında kullanılacak veri seti için önceden çekilmiş yangın fotoğraflarından oluşan bir veri seti elde edilmiştir. Kullanılan veri seti 999 fotoğraftan oluşmaktadır. Veri setinde bulunan tüm fotoğraflar sınıflandırmaya göre etiketleme işlemi yapılmıştır. Etiketleme işleminin amacı bir fotoğraf da istenilen nesnenin tespit etmektir. Etiketleme işleminde nesne fotoğrafın hangi koordinatlarda bulunduğunu bir text dosyasına kaydetmektir. Bu proje kapsamında veri setinin içindeki görüntüleri etiketlemek için makesense.ai den yararlanılmıştır. Proje için iki etiket oluşturuldu. Bu etiketler yangın var ve yangın yok şeklindedir.
Fotoğraf etiketlemesinin ardından, veri seti train ve test olarak parçalanmıştır. Bu parçalanma %20 eğitim, %80 test olarak ayrılmıştır.



Şekil 4.5. Yangın var etiketi



Şekil 4.6. Yangın yok etiketi



Şekil 4.7. Görüntünün koordinat bilgileri

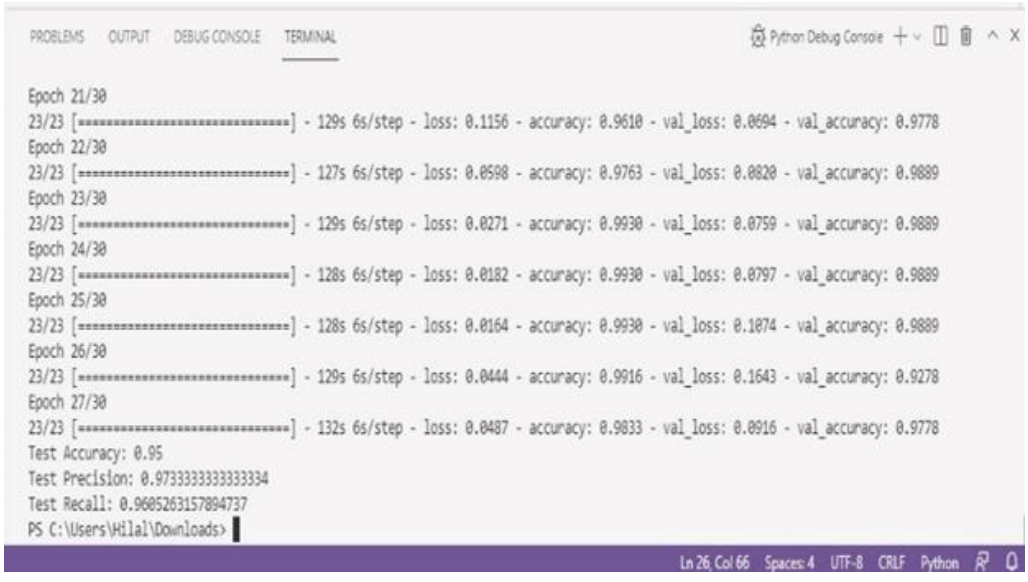
fire751.jpg	30.744	29.215	JPG Dosyası	12.04.2022 21:38	2F68FA71
fire751.txt	37	31	Metin Belgesi	12.04.2022 21:39	ACFDB6B6
fire752.jpg	248.574	243.893	JPG Dosyası	12.04.2022 21:38	747ED703
fire752.txt	76	57	Metin Belgesi	12.04.2022 21:39	B7707693
fire753.jpg	379.445	363.654	JPG Dosyası	12.04.2022 21:38	F0B3760D
fire753.txt	37	32	Metin Belgesi	12.04.2022 21:39	88B33B03
fire754.jpg	41.662	40.452	JPG Dosyası	12.04.2022 21:38	D89F6ECC
fire754.txt	37	34	Metin Belgesi	12.04.2022 21:39	AF7958EB
fire755.jpg	176.532	167.915	JPG Dosyası	12.04.2022 21:38	F155163D
fire755.txt	115	73	Metin Belgesi	12.04.2022 21:39	8BC9B724
non_fire1.jpg	56.623	51.201	JPG Dosyası	12.04.2022 21:38	28613485
non_fire1.txt	38	35	Metin Belgesi	12.04.2022 21:39	ABA9AED8
non_fire2.jpg	222.422	218.241	JPG Dosyası	12.04.2022 21:38	E05EEEDA
non_fire2.txt	37	32	Metin Belgesi	12.04.2022 21:39	41DA4A0F
non_fire3.jpg	159.496	151.580	JPG Dosyası	12.04.2022 21:38	2999103E
non_fire3.txt	76	57	Metin Belgesi	12.04.2022 21:39	E3ED344E
non_fire4.jpg	146.685	143.424	JPG Dosyası	12.04.2022 21:39	678FB3A5
non_fire4.txt	76	55	Metin Belgesi	12.04.2022 21:39	D536E9F3
non_fire5.jpg	59.904	58.463	JPG Dosyası	12.04.2022 21:39	3AFC28ED
non_fire5.txt	115	71	Metin Belgesi	12.04.2022 21:39	5540C245
non_fire6.jpg	237.620	237.432	JPG Dosyası	12.04.2022 21:39	D650320F
non_fire6.txt	76	56	Metin Belgesi	12.04.2022 21:39	05224B57
non_fire7.jpg	228.126	220.639	JPG Dosyası	12.04.2022 21:39	9EF789D5
non_fire7.txt	115	73	Metin Belgesi	12.04.2022 21:39	EE5F1181

Şekil 4.8. Görüntülerin text dosyaları

4.4.Sinir Ağı Parametrelerin Ayarlanması Ve Eğitim İşlemi

Veri setinin niteliklerine göre sinir ağının parametrelerinin doğru belirlenmesi oldukça önemlidir. Parametre değerleri sinir ağlarında farklılık gösterebilmektedir ayrıca aynı sinir ağlarında yapılan bir ayar apayrı iki veri setini de aynı şekilde etkileyemeyebilir.
Veri seti büyüdükçe ve çeşitliliği artıkça öğrenme de doğru oranda etkilenmektedir fakat bu eğitim süresini artırmaktadır.
Veri setinin niteliklerine göre sinir ağının parametreleri ayarlandıktan sonra eğitim işlemine başlanılabilir. Sinir ağı parametrelerinden olan batch ve learning rate eğitim işlemi için çok önemlidir. Batch değeri bir iterasyonda okuduğu adet fotoğrafıdır. Eğitim

işlemi esnasında her iterasyon da ağırlık dosyalarını güncellemektedir. Batch değeri büyük olması iterasyon başına geçen süreyi artırmakta ve eğitim süresi artırmaktadır.
Eğitim esnasında iterasyon başına bir ortalama loss değeri hesaplanmaktadır. Loss değeri tahmin edilen değerlerin gerçeğe ne kadar yakın olduğunu göstermektedir. Öğrenme oranın değerine göre ortalama loss değeri farklılık göstermektedir. Öğrenme oranı için en iyi yollardan biri de öğrenme oranını ilk yüksek tutup gittikçe azaltmaktır. Öğrenme hızının düşük tutulması eğitim süresini artırmaktadır.
Eğitim için kullanılacak olan veri seti yeterince büyük değilse transfer learning ile daha kolay bir öğrenme yapılabilir. Transfer learning, belirli bir problemi çözmek için öğrenilen bilgileri benzer ama farklı bir problem de yararlanılır. Eğitim durdurma işlemi hata verilerinde bir değişim olmuyorsa ve average loss değeri 0.08 gibi oldukça düşük değerlere ulaşmışsa eğitim durdurulmalıdır. Eğitimin düşük olmadığı değerlerde sürekli aynı salınımı yapıyorsa sinir ağı parametrelerini gözden geçirilmeli veya veri setinin büyütülmesi gerekmektedir.
Bu tez kapsamında YoloV3 ve VGG16 parametreleri ve eğitim işlemleri özetlenmiştir**.**
YoloV3' de sisteme alınacak olan görüntülerin boyutları 608x608 olarak ayarlanılır. Eğer istenirse 418x418 olarak da sisteme alınabilir. Fakat bu boyut ne kadar büyük olursa küçük nesnelerin tespiti de o oranda iyi olmaktadır. Bunun yanında eğitim yapılacak ekran kartının özelliklerine göre bu boyutlar 32'nin katları oranında artırılabilmektedir. YoloV3 için, batch ve learning rate parametre değerleri 1 ve 0.001 olarak alınmıştır. Eğitim 3600 adımda durdurmuştur.
VGG16 de sisteme alınacak olan görüntülerin boyutları 224x224 olarak ayarlanılır. VGG16 için, batch ve learning rate parametre değerleri 1 ve 0.001 olarak alınmıştır. Eğitim 2000 adımda durdurulmuştur.



Şekil 4.9. VGG16 eğitim aşaması

Şekil 4.10. Yolov3 eğitim aşaması

4.5.Test İşlemi

Eğitim işleminden sonra test işlemi, son iterasyonda oluşturulan veya daha önceki iterasyonlarda oluşturulan ağırlık dosyaları kullanarak yapılabilmektedir. Tahminde bulunduğu her görüntü için 0-1 arasında bir sınıf değeri döndürür, hangi sınıfın değeri daha yüksek ise eşik değerine göre sınıf veya sınıflar tahmin olarak verilmektedir.
Test aşamasında eğitim işlemi yapılan veri setinin dışında görüntü kullanılarak yapılmaktadır. Test edilen fotoğraflarda doğruluk ve hız çok önemlidir. Eğer test işleminde nesne tespitini hızlı bir şekilde yapılmıyorsa başka bir sinir ağları kullanılabilir, düşük doğruluk olduğu durumda ise farklı sinir ağı parametrelere değerleri kullanılabilir.

4.6.Performans Analizi

Eğitim ve test aşamalarından sonra veri setine dayanan modelin ne kadar etkili performans verdiği hesaplanmalıdır. Performans hesaplanması için birçok metrik yöntemleri vardır. Bunların bazıları F skoru, ROC eğrisi, PR eğrisidir.

5.DENEYSEL SONUÇLAR

Bu tez çalışmasında yangın tespiti üzerine farklı derin öğrenme mimarileri araştırılmış, analiz edilmiş ve veri seti kullanarak eğitimleri yapılmıştır. İki adet derin öğrenme ağ mimarisi (VGG16 ve Yolov3) ve bu mimarilere ait hiper parametrelere bağlı olarak eğitimler transfer öğrenme yöntemiyle gerçekleştirilmiş ve sonuçlar kaydedilmiştir. Bu

uygulamalara bağlı olarak çeşitli sonuçlar elde edilmiştir. Tez çalışmasının sonuçları aşağıda maddeler halinde ifade edilmektedir.

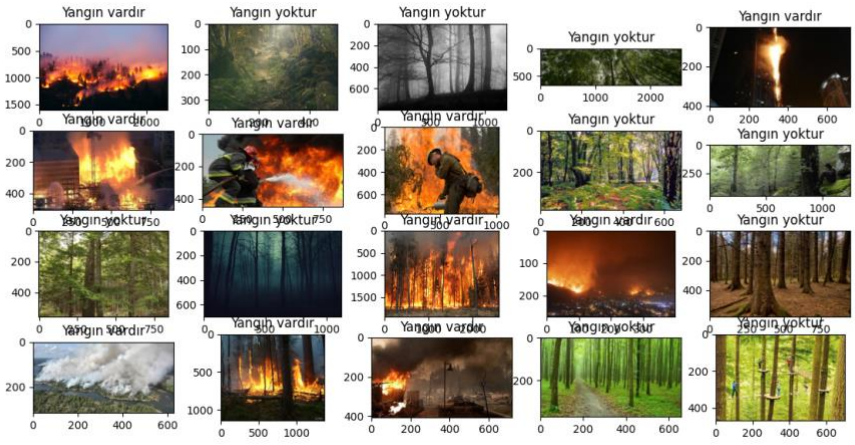
Kullanmış olduğumuz veriseti toplamda 999 görsel içermektedir. Bunların 755 tanesi yangın fotoğrafları 244 tanesi yangın unsuru içermeyen fotoğraflardan oluşmaktadır.

Ve bu çalışmada 2 nesne etiketi kullanılmıştır. Bu etiketler yangın var, yangın yoktur. Verisetinin elde edilmesinin ardından etiketleme adımı makesense.ai platformunda gerçekleştirilmiştir. Tüm verisetinin etiketinin tamamlanması ise yaklaşık 4 saat sürmüştür.

Veri seti ile alakalı yapılması gerekenler tamamlandıktan sonra eğitimin önemli adımı olan hiper parametre ayarları eğitim amacına uygun şekilde seçilmiştir. Eğitimler parçalar halinde yapılmıştır. Öğrenme oranı iki eğitim içinde 0.001 seçildikten sonra yapılan eğitimin son iterasyon ağırlıkları ile test işlemi gerçekleştirilmiştir. Test işleminin çıktıları şekil 5.1. de ve şekil 5.2. de görülmektedir.

Yolov3 ile yapılan eğitimde ilk 1700 iterasyonluk eğitim sonucunda tahmin sonucu yapmadığı görülmüştür. Eğitim ve iterasyon miktarları arttırılarak en iyi sonuca varılması hedeflenmiştir. Eğitim yaklaşık 7 saat sürmüştür. 3600 iterasyon eğitim yapılmıştır.

VGG16 ile yapılan eğitimde ilk 2000 iterasyonluk eğitim sonucunda en iyi tahmin sonuca varıldığı görülmüştür.



Şekil 5.1. VGG16 İle test aşaması

Şekil 5.2. Yolov3 İle test aşaması

6.KAYNAK

[1] Moolayil, J., (2019). Learn Keras for Deep Neural Networks A Fast-Track Approach to Modern Deep Learning with Python, Apress, Berkeley, CA, ABD.

[2] Kızrak, M. ve Bolat, B., (2018),"Derin Öğrenme ile Kalabalık Analizi Üzerine DetaylıBir Araştırma", Bilişim Teknolojileri Dergisi, c.11, no.11, s. 263
-286.

[3] Chollet, F., (2017). Deep Learning with Python, Manning, Shelter Island, NY, ABD.

[4] Lau, S., (2017). A Walkthrough of Convolutional Neural Network — Hyperparameter Tuning,

[5] Ioffe, S., Szegedy, C., (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, 7-9 Temmuz 2015, Lille, Fransa.

[6] Özkan İNİK, Erkan ÜLKER (2017) "Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri"

[7] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi1and Matthias Dehmer (2020) "An Introductory Review of Deep Learning for Prediction Models With Big Data"

[8] [81] Lau, S., (2017). A Walkthrough of Convolutional Neural Network — Hyperparameter Tuning,

[9] S. S. Haykin, « Neural Networks and Learning Machines,» *Pearson Education*, cilt

3, p. 906, 2009.
[10] Ziya Tan (2019) “Derin Öğrenme Yardımı İle Araç Sınıflandırma “
[11] Rupesh Kumar Rout (2013) “ A Survey on Object Detection and Tracking Algorithms “
[12] A. Krizhevsky, I. Sutskever, G. E ve .. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks,» NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, cilt 1, pp. 1097-1105, 2012.
[13] Y. Lecun, L. Bottou, Y. Bengio ve P. Haffner, «Gradient-based learning applied to document recognition,» *proceeding of the IEEE,* cilt 86, no. 11, pp. 2278-2324, 1998.
[14] M. D. Zeiler ve R. Fergus, «Visualizing and Understanding Convolutional Networks,» *European Conference on Computer Vision*, pp. 818-833, 2013.
[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg “SSD: Single Shot MultiBox Detector” , 2016
[16] Ziya Tan (2019) “Derin Öğrenme Yardımı İle Araç Sınıflandırma “