



**İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA**
**MÜHENDİSLİK FAKÜLTESİ**
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

# YAPAY SİNİR AĞLARINI KULLANARAK İRİS TANIMA

**HAZIRLAYANLAR:**

**İBRAHİM EYYÜP İNAN-1306150027**

**DANIŞMAN:**

**Dr.Öğr.Üyesi FATİH KELEŞ**

**MAYIS-2020**

# ÖNSÖZ

Gündelik hayatımızda kimliğimizi çeşitli yerlerde, binalara, hava ve tren seyahatlerine güvenli erişim, oy kullanma vb. için kanıtlamamız gerekmektedir. Farklı yaşam alanlarında insanların kimliklerinin kontrol edilmesi sıklıkla karşılaşılan bir güvenlik alma yöntemidir.

Geleneksel olarak, kimliğimizi kanıtlamak için fotoğraf tabanlı kimlik kartları kullanıyoruz ve buna alışığız. Bugüne kadar kimlik kartlarına birçok durumda şifreler eklenmiş ve zamanla gelişen sistemlere göre değiştirilmiştir. Ancak daha sonrasında, bu geleneksel kimlik kanıtları, bu yöntemlerin eksikliklerinin çoğunun üstesinden gelmek için biyometrik kimliklerle değiştirilmektedir.

En tehlikeli güvenlik tehditlerinden biri, birinin başkası olduğunu iddia ettiği kimliğe bürünmedir. Kimlik belirleme ve doğrulama dünyasındaki eğilim biyometriye yönelmiştir. Parmak izleri, yüz tanıma, İris Tanıma, Dna desen tanıma, basılı kimlik kartlarının yerini almaktadır.

Çeşitli biyometrik kimlik doğrulama sistemleri arasında, iris tanıma sistemi, bir bireyi benzersiz şekilde tanımlamak için irislerin görüntülerinde görüntü tanıma tekniklerini kullanan çok önemli bir tekniktir. İris tanıma, iriste bulunan kalıpları analiz ederek bir kişiyi güvenilir bir şekilde tanımlamak için kullanılabilecek belirli bir biyometrik sistem türüdür. Bu çalışma da günümüzde kullanılan biyometrik sistemlerden olan yapay sinir ağları kullanarak iris tanıma sistemleri hakkında bilgi verilmiştir.

TEŞEKKÜRLER

Gelmiş olduğumuz bu noktada ve yapacağımız bu projede bizlerin bugünlerdeki bilgi birikimi seviyesine ulaşmamızda emeği olan tüm İstanbul Cerrahpaşa üniversitesi bilgisayar mühendisliği öğretmenlerine karşılığı olmayan kutsal emeklerinden dolayı en içten ve kalbi duygularla teşekkür ederiz.

**İBRAHİM EYYÜP İNAN-1306150027**

**İÇİNDEKİLER** **SAYFA NO**

## ŞEKİL LİSTESİ

## TABLO LİSTESİ



## SEMBOL LİSTESİ

W :Girdi Boyutu

F :Filtre Boyutu

S :Kaydırma Miktarı

P :Padding Değeri

KISALTMA LİSTESİ

| | |
|---|---|
| PIN | :kişisel tanımlama numarası |
| DNA | :deoksiribonükleik asit |
| YSA | :yapay sinir ağları |
| EXP | :eksponansiyel |
| OCR | :optik karakter tanımlama |
| MAX | :maksimum |
| SVM | :güvenli sanal makine |
| CNN | :Convolutional Neural Network |
| MCNN | : Multi Convolutional Neural Network |
| FCN | : Fully Convolutional Network |
| FC | : Fully Connected Network |

# ÖZET


## YAPAY SİNİR AĞLARINI KULLANARAK İRİS TANIMA

Biyometrik sistemler insanların bazı biyolojik özelliklerini kullanarak elektronik ortamda tanınmalarını sağlamaktadır. Günümüzde varolan güvenlik tedbirleri kişileri birden fazla şifreyi hatırlamaya zorlamaktadır. Bir kişinin kendisini tanıtabilmesi için bir çok araç taşıması gerekmektedir.Bu tarz tedbirler güvenlik ve pratiklik konusunda zayıftır.Günümüzde giderek yaygın hale gelen ve birçok yerde görmeye başladığımız biyometrik sistemler, sağladığı faydalar ve kullanım alanlarıyla dikkat çekmektedir.

İris tanıma,bir örüntü tanıma konusudur.İnsan irisi yüksek oranda ayırt edici özelliklere sahiptir ve bu özellikler biyometrik ayıraçlar olarak kullanılır.Yüz tanıma ve iris tanıma konusunda beynin temel işlevlerini benzetimle gerçekleştiren bilgisayar algoritmaları (Yapay sinir ağları) geliştirilmiştir. Yüz ve iris tanıma, girdi olarak verilen bir yüz görüntüsü ile, bilinen kişilerin yüz görüntülerini içeren bir veritabanı kullanarak, bir bireyi doğrulamak olarak tanımlanabilir. Teknolojinin gelişmesi ve güvenlik konusundaki ihtiyaçların artması ile dünya yeni yöntemlere yönelmeye başlamıştır.

Bu tezde yapay sinir ağlarıyla iris tanıma konusu incelenmiştir.Yaptığımız araştırmalar ve deneyler bu yöntemin iris tanıma için başarılı olduğunu göstermiştir.

# SUMMARY


## IRIS RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

It enables the recognition of biometric systems in electronic environment by using some features of individuals. Today, increasing security measures force people to remember more than one password. In addition, the person becomes obliged to have more than one type of card to introduce himself. Such measures are becoming increasingly distant from reliability and practice, increasing the interest in biometric systems.

Iris recognition is a pattern recognition subject that is rapidly increasing its importance today. Human iris has quite distinctive features and uses these features as biometric identifiers. The problem of face and iris recognition can be defined as identifying or verifying an individual using a database of face images of known persons with a face image provided as input. Due to technological advances and the need for iris recognition applications, research in this area has increased rapidly in recent years. In this thesis, iris recognition is developed by using artificial neural networks. In the iris recognition process, gradient image of the original image is used. A certain number of attribute values were generated from the obtained graph.

Experiments have shown that the developed method is successful for iris recognition.

# GİRİŞ

Günümüz dünyasında teknolojinin sürekli gelişmesine paralel olarak güvenlik tehditleri de aynı oranda artış göstermektedir.İnsanların kendilerini tanıtabilmeleri için kullanabildikleri birden çok sistem vardır. Bunlardan en çok kullanılanı eski bir yöntem olan kimlik kartı, özel şifreler gibi yöntemlerdir. Fakat bunların kaybolması veya unutulması durumları meydana gelebilir. Artan güvenlik problemleri son yıllarda çeşitli güvenlik sistemlerini ortaya çıkarmıştır. Biyometrik sistemlerde, kişilerin fiziksel ve biyolojik farklılıklarını baz alarak elektronik ortamda kimlik tespiti yapılmaktadır. Kişi tanıma işlemi için göz irisinin kullanılabileceği fikrini ilk olarak ortaya atan kişi fransız göz doktoru doktoru Alphonse Bertillon"dur. Bu fikrin ardından 1981"de iki göz doktoru, Aran Safir ve Leonard Flom, göz irisinin kişi tanıma işlemi için kullanılabileceğini savunmuşlardır.. 1989"da Cambridge üniversitesinden Safir ve Flom Dr John Daugman'ın yardımı ile iris tanıma sistemini gerçekleştirmiş ve patentini almışlardır. Ortam şartlarından kaynaklı her zaman istenildiği gibi sonuç üretilemeyebilmektedir. Bundan dolayı systemin ürettiği sonuç belli bir yüzde tutuncaya kadar karşılaştırılır. Belirtilen şartlar sağlanmışsa sistem çalışır ve kişinin sisteme erişmesine izin verilir.

İnternet ve bilgisayar teknolojilerinin performans ve erişilebilirliğinin artması ile birlikte, bazı kişisel bilgilere veya kurum ve kuruluşlara ait gizli verilere, erişme yetkisi olma kişilerin erişmesinin engellenmesi zorunluluk haline gelmiştir. Aynı zamanda bu kişi onaylama işleminin doğruluğu ve hatasız çalışması önemli bir etkendir. Bu sistemler günümüzde daha çok pin gibi bilgiler ile kişiyi onaylar. Giriş bilgilerinin güvenliğini sağlamak ve izinsiz girişleri engellemek oldukça zordur. Bunun yerine biyometrik teknolojiler kişileri doğrudan tanıdıkları için yüksek güvenlik sağlar. Biyometrik sistemlere parmakizi tanıma yüz tanıma iris tanıma gibi örnekler verilebilir, Amaç bakımından çalışmamızı ele alacak olursak insan gözünde hiçbir zaman değişmeyen iris bölgesinin yapay sinir ağlarıyla belirlenmesi ve kişinin tanınmasının en güvenlikli yöntem olduğu gözlemlenmektedir.

Bu çalışmada incelenen makaleler ve tezler ışığında YSA tabanlı iris tanıma sistemleri hakkında genelden özele doğru bilgi veren bir çalışma yapılmıştır.Birinci bölümde biyometrik sistemler genel olarak ve çeşitleriyle birlikte ele alınıp incelenmiş ve yine tezin ana arterlerinden olan yapay sinir ağları ve irisle ilgili olarak açıklama ve bilgi altyapısı titizlikle hazırlanmıştır.Çalışmamızın ilerleyen bölümlerinde ise kullanılan platformlar ve materyallerle ilgili bilgi verilmiş ve son olarak systemin gerçeklenmesi,tartışma ve sonuç bölümlerinde sistemin uygulanmasına dair yorumlar ve bulgular sunulmuştur.

# 2.GENEL KISIMLAR

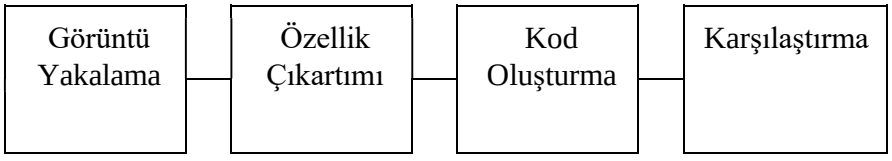
## 2.1.BİYOMETRİK TANIMA SİSTEMLERİ TEKNOLOJİLERİ

Biyometrik tanıma bir insanın fiziksel bazı özelliklerinin kullanılarak tanıma işleminin yapılmasıdır.. Parmak izi, retina ve iris, avuç içi izi,yüz yapısı nsane tanıma günümüzde kullanılan biyometrik tanıma yöntemlerinden bazılarıdır. Bu özellikler her nsane özel olduğu için biyometrik yöntemler hırsızlık ve dolandırıcılığa engel olmaktadır.Bu teknolojinin özelliği parola veya PIN kodu yerine çalınamayan kaybolmayan veya yeniden oluşturulamayan biyometrik özelliğin kullanılmasıdır. Günümüzde de kullanılan bu yöntemlerin dışında bazı yeni keşfedilmeye başlayan biyometrik özelliklerde bulunmaktadır. Bunlar arasında en dikkat çekici olanlardan birisi kulak yapısı tanıma sistemidir. Bu bir insanın keşfedilmemiş bir çok kendisine has özelliğinin bulunabileceğinin göstergesidir. Bu da biyometrik sistemlerin ana amacını oluşturmaktadır.

Biyometrik tanıma sistemlerini kısaca özetlersek, insanın biyolojik özelliklerinden tanıma sistemleridir. Tanıma esnasında kişiye ait özellik belirlenir ve daha sonra onaylama işlemi için saklanır. Doğrulama anında kişiye ait saklanmış olan bu özellikler anlık olarak karşılaştırılır.

### 2.1.1.Biyometrik tanıma nedir?

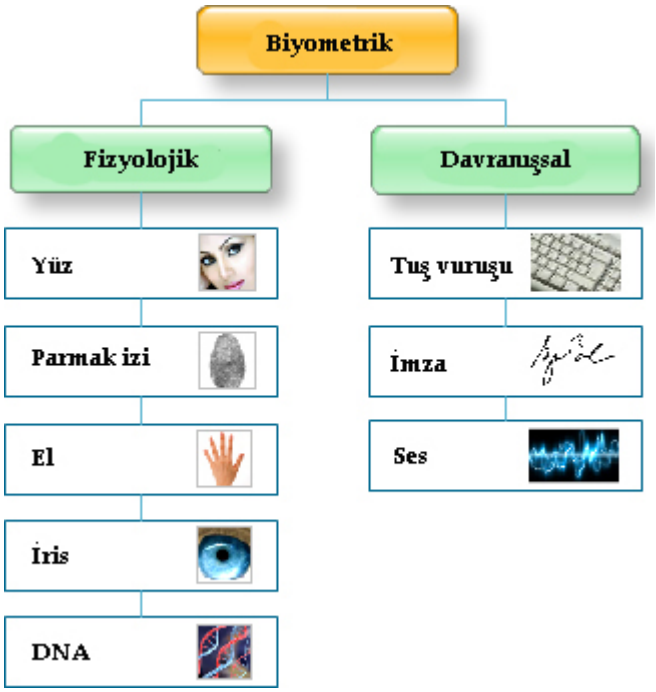
Biyometri bir bireyin fiziksel ya da biyolojik farklılıklarını gözlemleyen ve farklılıklara göre tanıma işlemi yapan bir sistemdir. Başka bir deyişle kimlik kartı pin manyetik kartlar kullanımı yerine kişinin biyolojik özelliklerinden tanıma işlemidir.



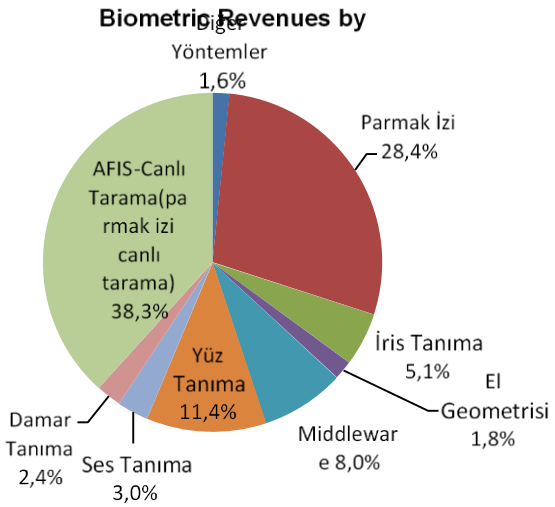

şekil 2.1.biyometrik sistemlerin çalışma prensibi

## 2.1.2.İnsanların Ortak Biyometrik Özellikleri :

Biyometrik özellikler olarak aşağıdaki resimde temsil edilen iki ana gruba ayrılabilir:




şekil 2.1.2.biyometrik sistemlerin siniflandirilmasi



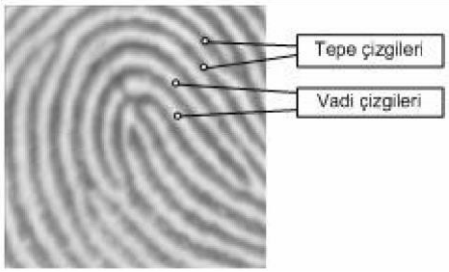
şekil 2.1.2.b.biyometrik sistemlerin kullanim oranlari

### 2.1.3.Yüz tanıma teknolojisi:

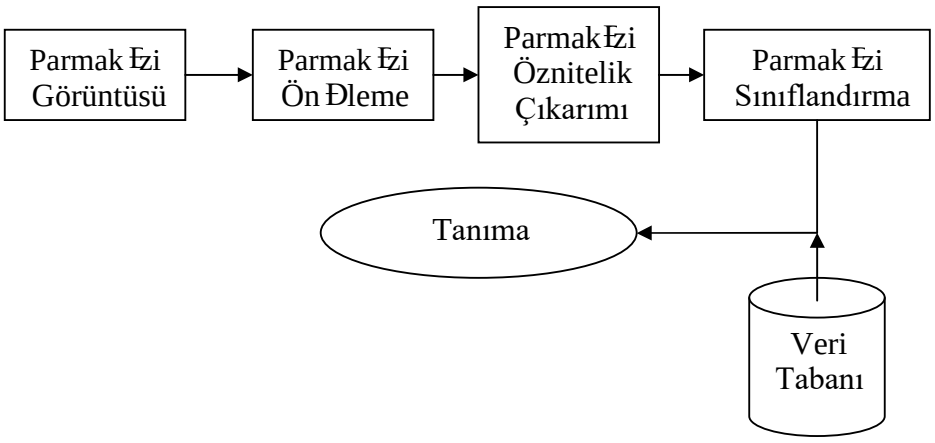
Yüz tanıma biyometri teknolojisi günümüzde en yaygın kullanılan tanıma yöntemlerinden biridir. Gündelik hayatımıza dahil olmuş bu yöntemin akıllı telefonlarda sıkça kullanıldığı görülmektedir. Bu tanıma yönteminin en büyük problemi zamanla kişinin yüz özelliklerinin değişiklik gösteriyor olmasıdır. Saç sakal uzaması bu systemin yanılmasına sebep olan başlıca unsurlardır.

## 2.1.4.Parmak izi tanıma teknolojisi:

Parmak izi parmağın yüzeyindeki çizgilerden oluşur. Bu çizgiler parmağın üzerinde tepe ve vadi noktalarından oluşur.. Bir parmak izinin benzersizliği bu çizgilerin şekillerinin karşılaştırılması ile anlaşılabilir. Parmak izleri genellikle aynı özellikleri taşımayacağı için benzersiz kabul edilirler.[13]




şekil 2.1.4.a.Parmak izinin mikroskobik görüntüsü




şekil 2.1.4.b.parmak izi tanıma algoritması

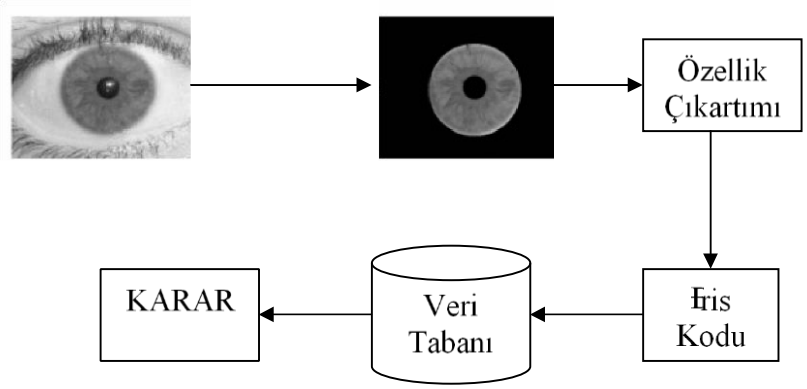
### 2.1.5.El Geometrisi tanıma teknolojisi:

El geometrisi kişileri ellerinin şekillerinden tanıyan biyometrik tanıma sistemidir.

El geometrisi okuyucuları eli farklı boyutlarda görüntüler ve daha önce kaydedilmiş özelliklerle kıyaslar. El geometrisi araçları 1980lerden itibaren üretilmeye başlamıştır.

**2.1.6.İris Tanıma:**

İris tanıma bireyin gözünün net bir görüntüsünün spesifik iris özelliklerinin tespit edilmesi ile gerçeklenen bir biyometrik tanıma yöntemidir. Genellikle ortam şartlarından etkilenmez. Bu nedenle parmak izine kıyasla daha yüksek bir başarıya sahip olabilir ve daha hijyeniktir. İrisin diğer biyometrik tanıma sistemlerinden en büyük farkı ise hayat boyunca kişide sabit kalıyor olmasıdır. İris sadece doğumdan sonra 16. aya kadar değişiklik gösterir ve ardından ömür boyunca aynı kalır.




Şekil 2.2 İris tanıma sistemi

şekil 2.1.6.iris tanıma sistemi

**2.1.7.DNA Kimlik Teknolojisi:**

DNA biyometrisi herhangi bir bireyi tanımlamanın en garanti ve kesin yoludur. İkiz olmadıkça, DNA bizim fiziksel ve zihinsel kimliğimizi tanımlayan bir yapı olduğu için başka bir insanla aynı genlere sahip olmak mümkün değildir.DNA pek çok örnekten elde edilebilir : kan, saç, tırnak, ağız bezleri, kan lekesi, tükürük… DNA eşleştirme özellikle tecavüz durumlarını kanıtlamak için ceza davalarında popular bir şekilde kullanılmaktadır. DNA biyometrisinin önemli sorunlarından birisi, DNA ile birisini tanımanın yavaş olmasıdır ve oldukça pahalıdır.

## 2.1.8.BİYOMETRİK SİSTEMLERİN KARŞILAŞTIRILMASI

| | Evrensellik | Eşsizlik | Süreklilik | Elde Edilebilirlik | Performans | Kabul Edilebilirlik | Yaygınlık |
|---|---|---|---|---|---|---|---|
| DNA | Y | Y | Y | D | Y | D | D |
| Kulak | O | O | Y | O | O | Y | O |
| Yüz | Y | D | O | Y | D | Y | Y |
| Yüz Termogramı | Y | Y | D | Y | O | Y | D |
| Parmak izi | O | Y | Y | O | Y | O | O |
| El Geometrisi | O | O | O | Y | O | O | O |
| Iris | Y | Y | Y | O | Y | D | D |
| Retina | Y | Y | O | D | Y | D | D |
| İmza | D | D | D | Y | D | Y | Y |
| Ses | O | D | D | O | D | Y | Y |

Y: Yüksek, O: Orta, D: Düşük

Tablo 2.1.8.a.biyometrik sistemlerin özelliklerinin karşılaştırılması

Tabloya bakıldığında biyometrik tanıma sistemleri içerisinde açık ara en zayıf olanın imza sistemi olduğu görülmektedir. İmza tanıma sisteminin zayıf olmasındaki en büyük etken taklit edilebilmesidir. İris,DNA, parmak izi gibi genetik özellikler kişiye has oldukları için böyle bir problemle karşılaşılmaz.
Eşsiz olarak nitelendirilen DNA, parmak izi , iris tanıma ikiz kardeşler arasında dahi farklılıklar taşıdığı için eşsizlik değerleri yüksek olarak belirtilmiştir.
Süreklilik bakımından en güçlü özelliğe sahip olan iris ve DNA dır. Bunun nedeni irisin ve kişinin DNA sının hayat boyunca neredeyse hiç değişikliğe uğramamasıdır.
Elde edilebilirlik özelliği bakımından en düşük olarak sınıflandırılan iris DNA dır. Bu verilerin elde edilmesi diğer tanıma sistemlerine kıyasla daha zordur.

Gözün hassas yapısı kullanılan cihazın temasının ya da kullanılan ışınların verdiği rahatsızlık gibi nedenlerden dolayı iris görüntüsünün elde edilebilirliği diğer biyometrik karakteristiklere göre zordur. En kolay elde edilebilir biyometrik özellikler yüz, el geometrisi ve imzadır.

Doğruluk, hız ve kullanılan teknolojinin sağlamlığı açısından DNA, parmak izi, İris, retina biyometrikleri performansı yüksek olarak sınıflandırılmaktadır.

Kabul edilebilirlik açısından yani biyometrik verilerin ölçüm ve toplanması açısından, yüz, imza ve ses biyometrikleri yüksek olarak sınıflandırılmaktadır. Yaygınlık açısından da yüz, imza ve ses biyometrikleri yüksek olarak sınıflandırılmaktadır. İrisin kabul edilebilirliğinin biraz daha düşük olmasının sebebi sistemin kişiye ait iris görüntüsünü tam olarak inceleyememe ihtimalidir. Günümüz teknolojisinde yüksek çözünürlüklü görüntülerin işlenmesi oldukça zordur.

Aşağıdaki tabloda ise biyometrik sistemlerin doğruluk, kullanım kolaylığı ve kullanıcı kabul oranlarına göre yüksek, orta ve düşük olarak sınıflandırılan bir başka tablo görülmektedir. Tabloda; Y=Yüksek, O=Orta, D=Düşük olarak ifade edilmektedir.

| Faktörler | Doğruluk | Kullanım Kolaylığı | Kullanıcı Kabul Oranı |
|---|---|---|---|
| Parmak izi | Y | O | D |
| El Geometrisi | O | Y | O |
| Yüz | D | Y | Y |
| İris | O | O | O |
| Retina | Y | D | D |
| Ses | O | Y | Y |
| İmza | O | O | Y |

Tablo 2.1.8.b.Biyometrik sistemlerin karşılaştırılması(2)

Tabloya baktığımızda doğruluk faktörünün en yüksek parmak izi ve retinada olduğu görülmektedir. Kullanım açısından en iyi biyometrikler el geometrisi, yüz ve ses olarak belirlenmiştir. Kullanıcı kabul oranının en yüksek olduğu biyometrikler ise yüz, ses ve imzadır.

## 2.2.YAPAY SİNİR AĞLARI(YSA)

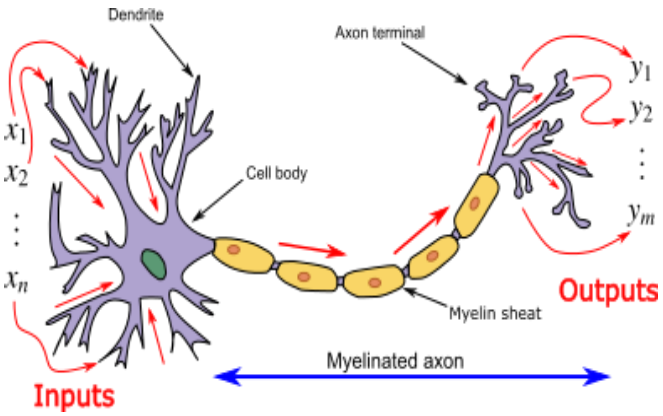
### 2.2.1.YSA'NIN TANIMI VE TARİHÇESİ:

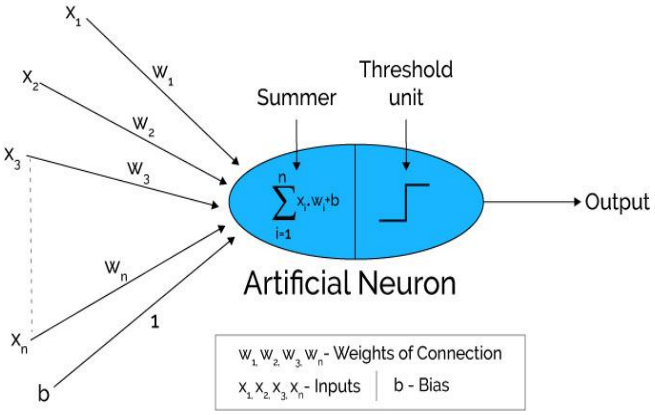
YSA, beynin bir işlemini gerçekleştirmek için yaptığı şeylerden esinlenilerek ortaya atılmıştır. YSA, yapay sinir hücrelerinin birbiri ile bağlanmaları ile oluşur ve birden fazla katmana sahip olabilir. Bu tanıma yakın bir tanımda yapay sinir ağı yazınında çok tanınan Teuvo KOHONEN'e ait bir tanımdır :" Yapay sinir ağları parallel olarak bağlantılı ve çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran hiyerarsik bir organizasyonudur."[13].Beynin bilgi isleme yöntemine uygun olarak YSA, bir öğrenme sürecinin ardından karar verme yeteneğine sahip bir modeldir.

Yapay sinir ağlarının temelleri 1940'ların basında arastırmalarına baslayan W.S. McCulloch ve W.A. Pitts'in, 1943 yılında yayınladıkları bir makaleyle atılmıstır. Mc Culloch ve Pitts, bir biyolojik nöronun basit bir eşik fonksiyonu görevi gördüğü fikrini ortaya atmışlardır.1949'da Donald Hebb ise "The Organization of Behaviour" adlı kitabında hücresel seviyede beyinin öğrenme mekanizmasından bahsetmistir. Bu öğrenme kuralına göre; başka bir nörondan dentrit yoluyla gelen ve bir akson yoluyla alınan giris verisi bu nöronun kendisinden sonraki nörona darbe üretmesine sebep olur. Sonraki aksonal girislerin darbe üretmesi olasılığı artar. Böylelikle yapılan davranısın sonucu ortaya çıkar.

## 2.2.2.YAPAY SİNİR AĞLARININ ÖZELLİKLERİ:

Sinir ağları insan beyninden esinlenilerek oluşturulmuş nöron ağlarıdır. Bu ağlar karar verebilme yeteneğine sahiptirler. Bir nöron tek başına bir anlam ifade etmemektedir. Ancak birçok nöronun birbiri ile iletişim kurması sonucu mantıklı bir sonuçlar doğurabilir.

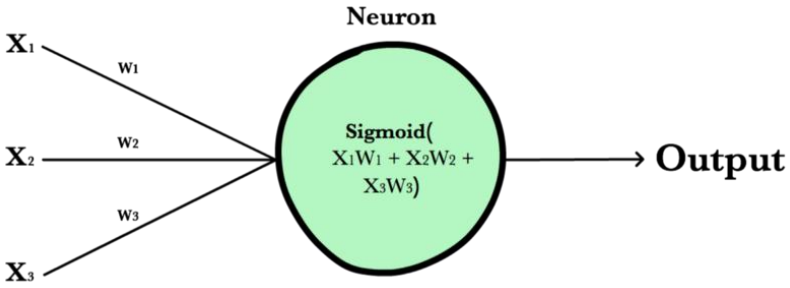



Şekil 2.2.2.a.bir nöronun yapısı şekil 2.2.2.b.yapay bir nöronun gösterimi

Bir nöron bir çok input ve bir output'dan oluşur. Gerçek bir insan nöronuna kıyaslanırsa yapay nöronun input değeri dendrit'e output değeri axon'a benzetilebilir. Genellikle bir nöronun output değeri başka bir nörunun input değeridir.

Nöronların her bir girişinin nöron için farklı bir değeri vardır. Her bir girdi gönderilmiş olduğu input hattı üzerindeki ağırlık değerine göre değerlendirilir. Her bir input değerinin üzerinde bulunduğu hattın ağırlık değeri ile çarpılır. Bu çarpma işleminin sonuçları toplanır ve sonuca bir bias değeri eklenir. Bu işlemin sonucu theshold unit adı verilen bölgeye gider. Bu bölge nöronun aktivasyon fonksiyonunu içerir.

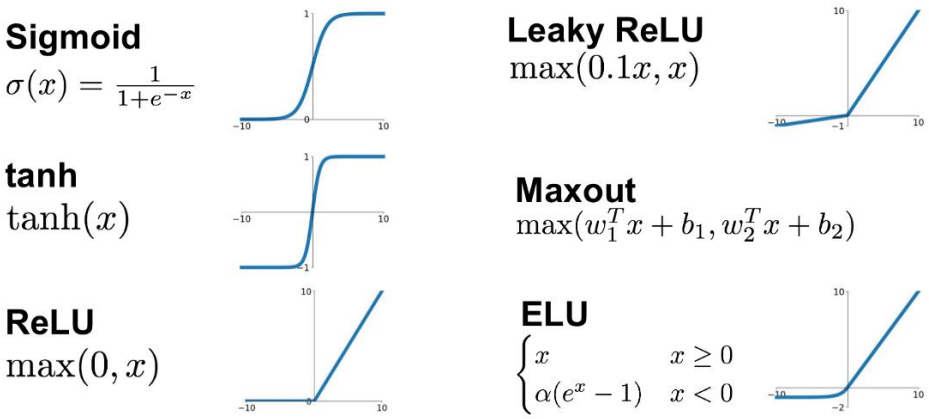



Şekil 2.2.2.c.yapay bir nöronun aktivasyon fonksiyonu

Aktivasyon fonksiyonu gelen değere göre bir output değeri oluşturmakla görevli fonksiyondur. Fonksiyon parametre olarak toplama işleminin sonucunu kullanır.

Birden fazla aktivasyon fonksiyonu bulunmaktadır. Bu fonksiyonların tercihi oluşturulmak istenilen sistemin sahip olduğu özelliklere göre belirlenir. Aktivasyon fonksiyonunun ürettiği sonuç nöronun output değeridir.

### 2.2.3.YSA'da Aktivasyon fonksiyonları:




#### 2.2.3.1.Sigmoid fonksiyonu:

Sigmoid fonksiyonu sıfır ve bir arasında bir output değeri veren bir fonksiyondur. Çıkış değeri mutlak pozitiftir. Geçmişte tercih edilmiş olmasına rağmen günümüzde çok sık kullanılmamaktadır. Bunun sebebi sadece belirli bir aralıkta mantıklı sonuçlar vermesi ve belirli bir değerin altında veya üstünde iken fonksiyonun eğiminin sıfıra yakınsamasıdır. Bir diğer sorunu ise bilgisayar için exp işleminin farklı fonksiyonlara göre daha yavaş çalışmasına sebep olmasıdır. Sigmoid türü aktivasyon fonksiyonu olarakta kabul edilen **softmax** fonksiyonu multi class sistemler için tercih edilen bir fonksiyondur.

$$\mathrm{softmax}(x)_i = \frac{e^{y_i}}{\sum_j^N e^{y_j}}$$

#### 2.2.3.2. tanh fonksiyonu:

Tanh fonksiyonu [-1,1] aralığında değer alır. Sıfır merkezli bir fonksiyondur. Ancak tanh fonksiyonu da sigmoid gibi belirli değerlerden sonra eğim değerinin sıfır olmasına sebep olur.

#### 2.2.3.3.ReLU fonksiyonu:

Relu fonksiyonu parametrenin sıfırdan büyük olması durumunda parametre değeri output değerine eşittir. Bu fonksiyon tanh ve sigmoid fonksiyonlarına göre daha hızlı çalışmaktadır.En büyük eksiği sıfır merkezli olmaması ve sigmoid ve tanh fonksiyonları gibi bazı değerlerde eğim değerini sıfır vermesidir.

### 2.2.3.4.Leaky ReLU fonksiyonu:

Bu fonksiyon Relu fonksiyonu ile benzer özellikler taşımaktadır. Tek farkı fonksiyon parametresinin sıfırdan küçük olduğu durumlarda da eğim değerinin sıfıra eşit olmamasıdır.

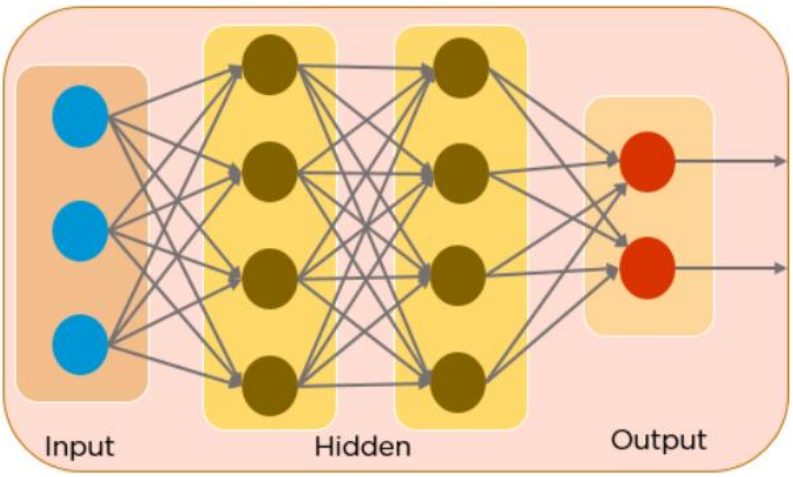
### 2.2.3.5. ELU fonksiyonu:

Bu fonksiyon da ReLU fonksiyonu ile yüksek benzerlik taşımaktadır. Parametrenin sıfırdan büyük olması durumunda parametre output değerine eşittir. Paramtrenin sıfırdan küçük olması durumunda exp işlemi gerçekleştirir.

Bu fonksiyon leaky ReLU'ya göre daha güçlü bir fonksiyondur. Ancak exp işlemi uygulanmak istenilmeyen bir işlemdir.

### 2.2.4.Bir yapay sinir ağının yapısı:

Bir sinir ağı bir giriş değerine göre bir sonuç oluşturduğu için mutlak bir şekilde giriş katmanı ve çıkış katmanı içermek zorundadır. Ancak sinir ağları genellikle iki katmandan oluşacak kadar basit olmaz. Bu sebeple en temel sinir ağı katmanlarından birisi de gizli katmandır.




Şekil 2.2.4.Bir yapay sinir ağının yapısı

Bu katman bir giriş katmanından veya başka bir gizli katmandan aldığı değerler ile bir sonuç oluşturur ve bu sonucu çıkış katmanı veya başka bir gizli katmana aktarır.

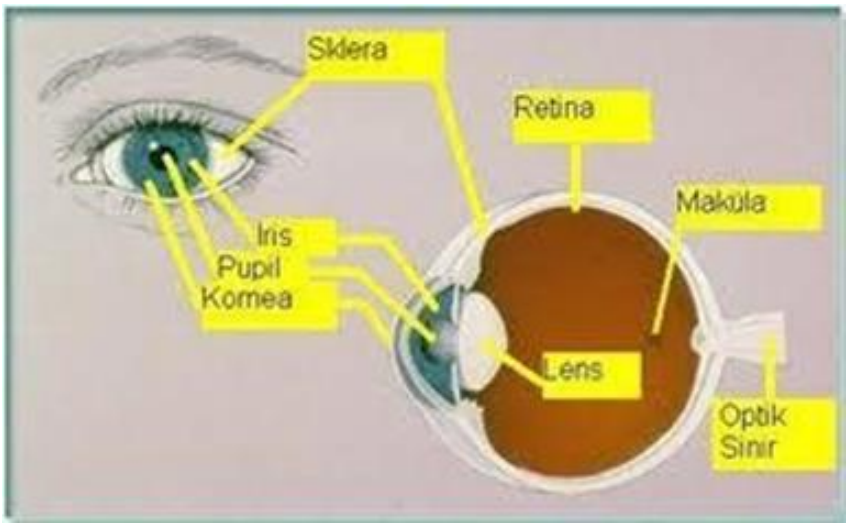
### 2.2.5.YSA'da Ağırlık fonksiyonu:

Bir sinir ağır birden fazla katmandan oluşabilmektedir. Bu katmanlardaki nöronların ağırlık değerleri sınıflandırma aşamasında en önemli rolü oynar. Bu ağırlık değerleri sinir ağına öğretilmiş olan sınıfların özelliklerini taşır. Bir ağırlık fonksiyonunun bu özellikleri öğrenmesi için gerçekleştirilen aşama ilerleyen sayfalarda anlatılacaktır.

## 2.3.İRİS

### 2.3.1. İRİSİN BİYOLOJİK YAPISI VE ÖZELLİKLERİ

İris, gözün ön kısmında bulunan ve lifli dokudan oluşan renkli tabakadır. İriste 250'den fazla görsel karakteristik bulunmaktadır. Bunlar daireler, benekler, çizgiler gibi belirleyici şekillerdir İris, bebek embriyo olarak anne karnındayken oluşur ve insanın ölümüne kadar değişmez. İris tarama biyometrik taramalar içerisinde en basit olanlarından biridir. Sıradan bir CCD kamera kullanılarak yaklaşık 15–20 cm uzaklıktan tarama yapılır . Genellikle retina ile karıştırılmaktadır. Fakat aşağıdaki şekilde görüldüğü gibi retina ile iris birbirinden çok farklıdır.
İris gözün ön kısmında bulunan lifli dokudan oluşan renkli tabakdır. Bu tabaka göz bebeğinin çevresinde bulunur. Göz irisi bir çok karakeristik özellik barındırmaktadır. İrisin en önemli özelliklerinden birisi ise hayat boyunca kişide neredeyse hiç değişiklik göstermemesidir.




**şekil 2.3.1. irisin biyolojik yapisi**

İrisin,bazı biyometrik özellikleri ;

•Dünyada bir kişiyi ait bir irisin başka bir kişide de olma ihtimali neredeyse yoktur.
•Tek yumurta ikizleri aynı DNA yapısına sahip olmasına ragmen farklı iris yapısına sahiptirler.
•Göz irisi kalıtımsal hastalıklardan etkilenmez.
•Farklı cinsiyette veya ırkta bulunmak irisin yapısını etkilemez.
•Gözle görülebilen ve hassasiyet ile ölçülebilen bir organdır.
•Ömür boyu değişmeyen tek organdır.
•İnsanın doğumun 16. ayından itibaren ölüme kadar değişmez.
•Göz insanın hayatını kaybetmesinin ardından en çabuk özelliğini kaybeden organdır.
•İris tanıma sistemi kişiye zarar vermez. Lazer ya da benzeri görüntü alma tekniklerine ihtiyaç duymadan basit bir CCD kamera ile görüntü alınabilmektedir.

## 2.3.2.İRİS TANIMA SISTEMLERININ ÇALIŞMA PRENSİBİ

İrisi tanıma sistemlerini çalışma şekli irisin farklı göz bebeği boyutlarındaki örneklerinin alınarak öğrenilmesi şeklinde gerçekleştirilmektedir. Bu işlem için tanıtılacak olan kişinin gözünün yakın mesafeden bir veya birden çok resminin çekilmesi ile elde edilir. Kullanılacak system bu görüntüden kişiyi ayırt etmeye çalışır.

**Şekil 2.3.2.İris Tanıma Sistemlerinin Çalışma prensibi**

## 2.3.3.İris Tanıma sistemlerinin bazı avantajları :

•Şifre, pin numarası gibi unutulma ve paylaşılma riski olan bilgilere gerek yoktur.
•Sistemin hatalı kabul olasılığı 1/1042'dir.
•Diğer biyometrik sistemler ile karşılaştırıldığında doğrudan temas olmadığı için hijyeniktir.
•Şifre pin numarası gibi unutulabilir bilgilere gerek yoktur.
•Sistemin hata olasılığı oldukça düşüktür.
•Göz rengi belirleyici bir faktör değildir.
•İris insan bedeninin ölümünün ardından ilk olarak ortadan kaybolan özelliktir.

# 3. KULLANILAN ARAÇ VE YÖNTEM

## 3.1. KULLANILAN PROGRAMLAMA DİLİ

### 3.1.1.PHYTON PROGRAMLAMA DİLİ

Python, ilk sürümü Guido van Rossum tarafından 1991'de ortaya konulmuş genel amaçlı bir programlama dilidir. Python programlama dili nesneye yönelik bir programlama dilir. C programalama dili ile geliştirilmiş olan bu dilin en büyük özelliği matematiksel işlemler için sağladığı kolaylıktır.

#### 3.1.1.1.Python Nerelerde Kullanılır?

Python programlama dili mühendislikten finansa bir çok alanda kullanılmakta olan bir programlama dilir. 2000'li yıların başından itibaren popülerleşmeye başlayan dil günümüzde en sık kullanılan diller arasında yerini almıştır.

Donanımsal açıdan, işlem gücü ve bellek (depolama) kapasitesinin herkesin kolayca ulaşabileceği şekilde artıp yaygınlaşması ile, Python programlama dili gibi bazı durumlarda daha düşük seviyeli dillere göre (C gibi) daha yavaş çalıştığı ve daha fazla bellek tükettiği gözlemlense de bu dezavantajlar gün geçtikçe ortadan kaybolmaktadır.

#### 3.1.1.2.Python Bilimsel Hesaplama Kütüphaneleri

Python ile temel bilimsel hesaplama için üç ana kütüphane kullanılmaktadır:
Hızlı dizi yapıları ve matris işlemleri gibi bazı temel işlevler için Numpy; Numpy veri yapılarının üzerinde sayısal entegrasyon, diferansiyel denklem çözümü, optimizasyon ve istatistik gibi herkesçe ihtiyaç duyulabilecek işlevler için SciPy ve iki boyutlu ve belli bir düzeyde üç boyutlu görselleştirme için Matplotlib kütüphanesi kullanılmaktadır. Bu projede de bu kütüphanelerden sıkça faydalanılmıştır.

Ayrıca, Python'un bilimsel kütüphaneleri de açık kaynak kodlu olduklarından, derste ele alınan konular için kullandıkları modüllerin kodlarını inceleyebilecek ve gereği halinde değiştirerek farklı amaçlara uyarlayabileceklerdir.

Tensorflow kütüphanesi Numpy kütüphanesinin daha komplike bir şekilde çalışan halidir. Bir çok barındırdığı fonksiyon Numpy modülü içerisinde benzeride de bulunmaktadır. Bu kütüphanenin kullanılmasındaki en büyük sebep yapay zeka için bazı işlemleri kolaylaştırması ve ekran kartı kullanımı sağlamasıdır.

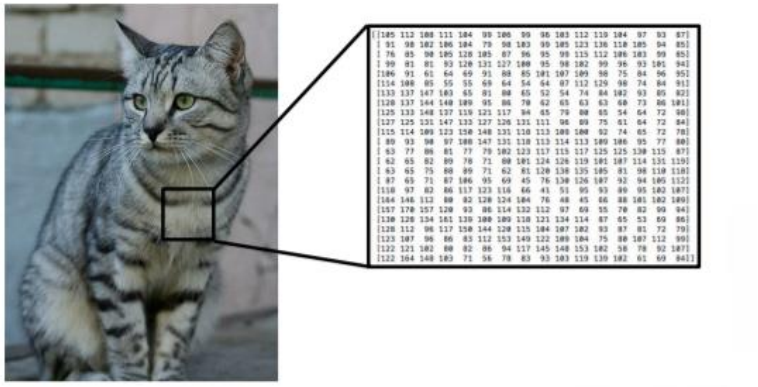
# 3.2.KULLANILAN YÖNTEM

## 3.2.1.CNN(Convolutional neural network)

### 3.2.1.1.Convolutional Neural Networks nedir ve nasıl işler ?

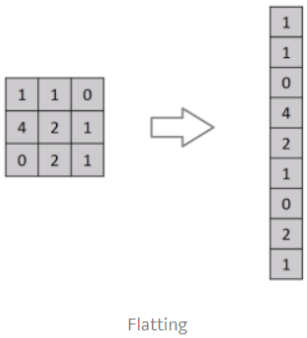
Evrişimsel sinir ağları bir görüntünün kategorize edilebilmesi için kullanılan bir yöntemdir. Bu yöntem içerisinde görüntünün kategorize edilebilmesi için gerçekleştirilen birden fazla katman bulunmaktadır.

Bir görüntünün kategorize edilebilmesi için birçok yöntemler bulunmaktadır. Ancak bu bu sistemlerden bazıları yüksek başarı sağlayamamakta ve bazıları ise aşırı derecede fazla işlem yapılmasına sebep olmaktadır. Görüntünün kategorize edilmesinde diğer yöntemlere göre çok daha yüksek başarı elde eden bir diğer yöntem sinir ağlarıdır.



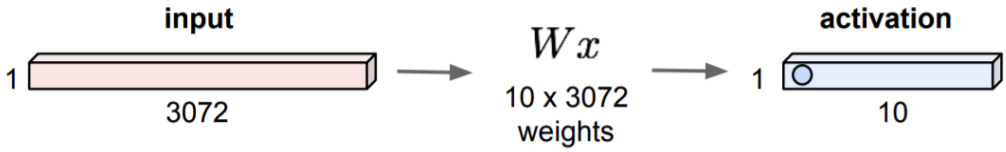
Bir resim bilgisayar ortamında bir matirisi ifade etmektedir. Her bir pixelin renk değeri matris olarak tutulmaktadır. Bu görüntü renkli bir görüntü olması durumunda kırmızı, yeşil ve mavi tonlarını ifade eden üç farklı matristen oluşmaktadır. Bu matrisler genellikle her bir pixel için 0 ile 255 arasında renk tonlaması bulundururlar. Bir sinir ağının resim hakkında öğrenebileceği özellikler resmin matris haline getirilmiş olan verilerdir.

Bir sinir ağı içerisinde bulunan nöronların girdi noktaları bulunmaktadır. Bu girdi noktaları nöronun kendi içerisinde yapacağı işlem için gerekli bilgileri barındıran verilerin teslim alındığı noktalardır. Bir görüntünün sinir ağına iletilebilmesi için ilk olarak bir sinir ağının input layer'i haline getirilmesi gerekmektedir. Bu sayede görüntü sinir ağının içerisine aktarılmaktadır. Örneğin 32x32 boyutlarındaki bir görüntünün bilgisayar ortamındaki matrisleri 3 adettir ve her biri 32x32 boyutlarındadır. Bu verinin bir sinir ağına aktarılabilir hale getirilmesi için tek boyutlu bir dizi haline getirilmesi gerekmektedir.



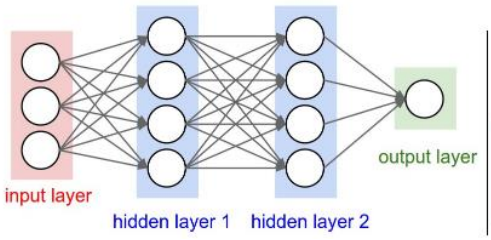

3.2.1.1.a.Cnn'de flatting(tek boyutlu dizi haline getirme) işlemi [2]

Bu işlemin gerçekleştirilmesi ile birlikte 32x32x3 boyutlarındaki bir görüntü 3072x1 boyutunda bir diziye dönüştürülmüş olur. Bu veri sinir ağının input layer'ına aktarılması ile birlikte nöronların sahip oldukları ağırlık değerleri nokta çarpımı işlemi yapılır ve aktivasyon fonksiyonuna iletilmek üzere toplanır. Temel olarak standart sinir ağları ile görüntünün analiz edilmesi işlemi bu şekilde gerçekleşir. Ancak bu yöntem birçok yönden gereksiz işlem yoğunluğuna sebep olmaktadır.

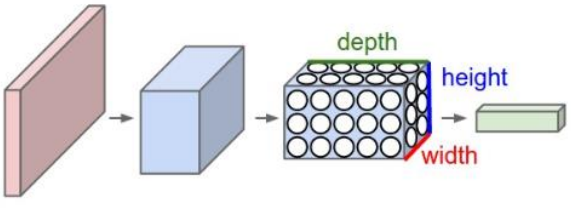



Örneğin 3072x1 input değerine sahip bir fully connected layer içerisinde 10 adet ağırlık fonksiyonu ile çalışılmak istenmesi durumunda 10x3072 boyutunda bir veri oluşturmaktadır. Bu verinin içerisindeki her bir 3072 adet ağırlık değeri bir nöronun ağırlık değerleridir. Her bir 3072 boyutundaki dizinin input değerleri ile nokta çarpımı bir nöronun aktivasyon değeridir.[1]

Evrişimsel sinir ağlarında standart sinir ağlarından farklı olarak bir görüntünün uzaysal yapısı korunmaktadır. Evrişimsel sinir ağları standart sinir ağlarına yüksek oranda benzerlik taşımaktadır. Standart sinir ağlarında yapmış olduğumuz tek boyutlu bir dizi haline getirmek yerine 3 boyutlu hali korunmaktadır.




Şekil 3.2.1.1.b.Standart bir sinir ağı [2]




Şekil 3.2.1.1.c.ConvNet [2]

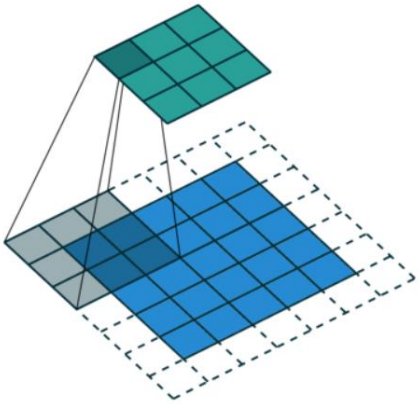
### 3.2.1.2.KATMANLAR:

Bir evrişimsel sinir ağı, bir girdi ve bir çıktı katmanından ve ayrıca birçok gizli katmandan oluşur. En temel olarak kullanılan katmanlar convolutional layer , pooling layer ve fully connected layer'dır.

#### 3.2.1.2.1.Convolutional Layer:

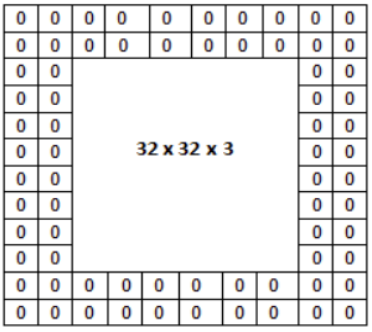
Bu katman bir evrişimli sinir ağları sisteminin mutlak olarak ilk katmanıdır. bu katmanının parametreleri bir dizi öğrenilebilir filtre içerir. Görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bu fitreler uygulanır. Her filtre boyut olarak küçüktür ancak bütün bir görüntünün üzerinde gezdirilir.

Bir filtre bir nöronun ağırlık fonksiyonudur. Filtrenin içerisindeki değerler bir nöronunun girişlerinin ağırlık değerleridir. Bu katmanda bir ağırlık fonksiyonu ile birden fazla nöron üretilmektedir. Sinir ağlarında bir görüntü matrisi bir ağırlık fonksiyonu ile nokta çarpımı yapılıp bir aktivasyon değeri üretilmekteydi. Ancak evrişimsel sinir ağlarında standart sinir ağlarındaki ağırlık fonksiyonuna kıyasla çok daha küçük bir fonksiyon ile bir nöron topluluğu olan aktivasyon map'i üretilmektedir.



Bir filtre görüntünün sol üst köşesinden başlayarak sıra ile sağa ve aşağıya kaydırılarak nokta çarpımı yapılır. Bu işlem ile aktivasyon map'i oluştururur. Aktivasyon map'inin boyutları filtrenin görüntü üzerinde gezdirilme şekline göre değişiklik gösterir.

Görüntü üzerinde bir filtrenin gezdirilmesi esnasında görüntünün kenar ve köşelerininde filtrelenebilmesi için görüntü matrisinin kenarları 0 ile doldurulur. Bu işleme **zero-padding** denir.

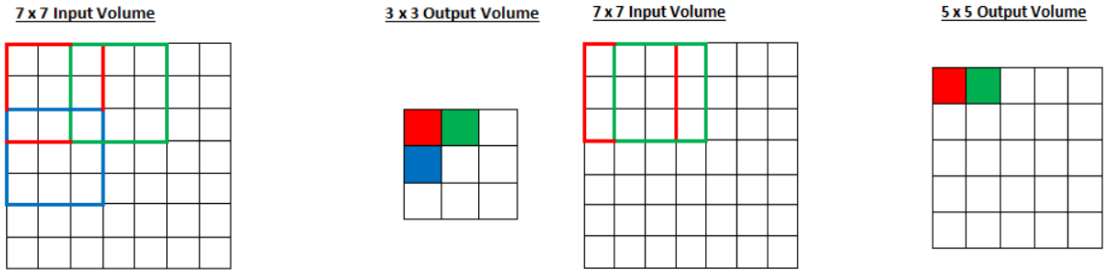



Padding

Şekil 3.2.1.2.1.a.zero padding işlemi [1]

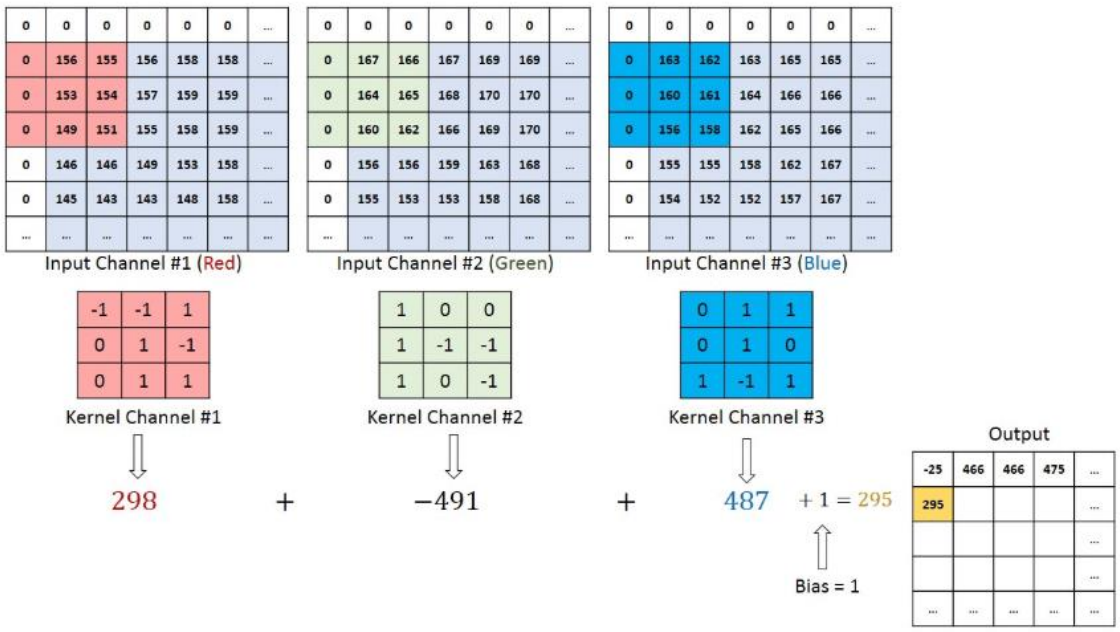
Bu işlem için görüntü matrisine eklenmesi gereken 0 kenarı sayısı filtrenin boyutlarına göre belirlenir. 3x3 boyutlarındaki bir filtrenin görüntünün sol üst köşesindeki pixele etki edebilmesi için eklenmesi gereken kenar sayısı birdir. Eklenmesi gereken kenar sayısı NXN boyutundaki bir filtre için **(N-1)/2** formülü ile hesaplanır. Ancak bu bir zorunluluk değildir. Geliştirici kişi tarafından tercihen kullanılır.

Oluşacak aktivasyon map'inin boyutunu etkileyebilecek bir diğer özellikse filtrenin görüntü üzerinde kaydırılması esnasında gerçekleştirilen atlama miktarıdır.




**şekil 3.2.1.2.1.b.** 2 basamak kaydırma [1] şekil **3.2.1.2.1.**c. 1 basamak kaydırma [1]

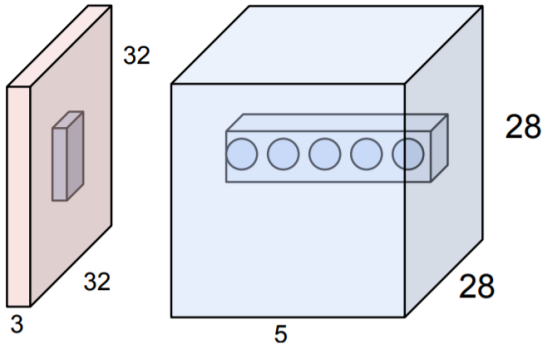
Bir aktivasyon map'inin sahip olacağı boyutları W değerinin girdi boyutu, F değerinin aktivasyon map'indeki her bir nöronun girdi sayısı (filtre boyutu), S değerinin kaydırma miktarı (stride) ve P değerinin uygulanmış padding değeri olması durumunda **(W−F+2P)/S+1** formülü ile hesaplanır.[2]




**Şekil 3.2.1.2.1.d.**MXNX3 boyutundaki bir görüntüye 3x3x3 boyutunda filtreleme[2]

Üç kanallı bir görüntünün (RGB) filtrelenebilmesi için kullanılan filtrenin de 3 kanal içermesi gerekmektedir. Aktivasyon map'in oluşturulması esnasında görüntünün her bir kanalına uygulanan filtrelerin oluşturduğu sonuçların toplanır.

Bir convolutional layer içerisinde genellikle birden fazla filtre uygulanır. Bu filtrelerin her biri kendi aktivasyon map'ini oluşturur. Örneğin 32x32x3 boyutundaki bir görüntünün 5 adet filtre ile filtrelenmesi sonucunda oluşacak hacim MxNx5 boyutlarında olacaktır. Bu hacmin her bir kanalına **depth slice** denilmektedir. Depth slice bir filtrenin oluşturduğu aktivasyon map'idir. Her bir slice üzerindeki aynı konumda bulunan nöronlar görüntü üzerinde aynı noktaya bakmaktadır.




Evrişimsel sinir ağlarının standart sinir ağlarından en büyük farkı nöronların belirli bir bölgeye bakıyor olmasıdır. Standart sinir ağlarında tek bir nöron bütün bir görüntüye bakıyorken evrişimsel sinir ağlarında sadece filtre boyutunda bir alana bakmaktadır.

32x32x3 boyutlarındaki bir görüntünün standart sinir ağlarında 32*32*10=10240'dır. Bu rakam filtre sayısının ve görüntü boyutlarının daha da büyütülmesi ile çok daha yüksek rakamlara ulaşabilmektedir. Bu durum istenilmeyecek düzeyde işlem yoğunluğuna sebep olur. Ancak aynı aktivasyon map'in üzerindeki nöronların parametreleri paylaşması ile elde edilen convolutional layer için gerekli parametre sayısı 5x5x3 boyutlarında 10 filtre ile filtrelenmesi durumunda 5*5*3*10=750'dir. Bu açık bir şekilde standart sinir ağlarından daha iyi bir rakamdır.[6]

### 3.2.1.2.2.Activation Layer:

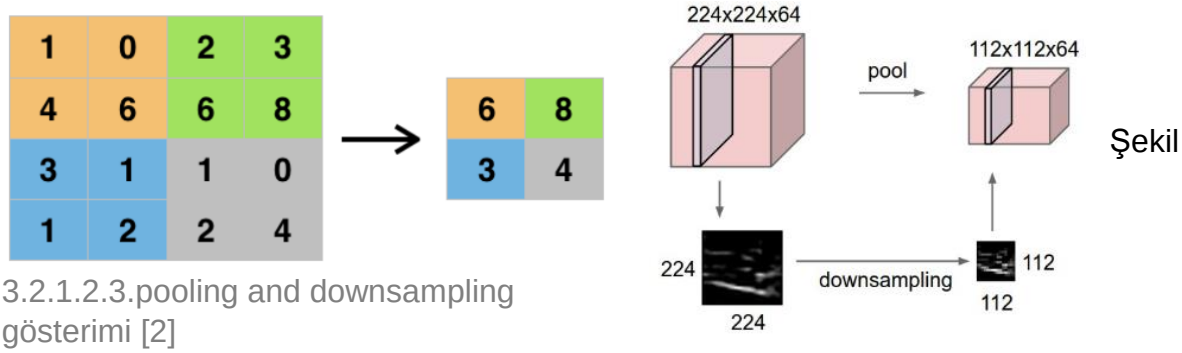
Aktivasyon katmanı genellikle convolutional layer'dan sonra kullanılan katmandır. Bu katman convolutional layer'da oluşturulmuş olan nöronların aktivasyon fonksiyonlarına göre sonraki katmana iletecekleri değerlerin belirlendiği katmandır. Bir nöron girdi değerleri ile gerçekleştirdiği işlem (genellikle toplama) sonucu nöronun aktivasyon fonksiyonunda bir çıktı oluşmasını sağlar. Bu çıktı bir sonraki katmanın girdilerini oluşturacak değerdir.

Bu katmanda daha önce anlatılmış olan aktivasyon fonksiyonları kullanılmakla birlikte çoğunlukla reLU fonksiyonu tercih edilmektedir. ReLU fonksiyonuna göre oluşmuş olan aktivasyon map'in içerisinde bulunan değerler (bu değerler nöronlara ulaşmış olan değerleri temsil etmektedir.) sıfır değerinden yüksek olması durumunda sonraki katmana değerin kendisi gönderilmektedir. ReLU Fonksiyonu

**f (x) = max (0, x)** ile ifade edilmektedir. Aktivasyon map'teki siyah değerler negatiftir. ReLU fonksiyonu uygulandıktan sonra siyah değerler kaldırılır onun yerine 0 konur. Bu katman non-linearity olarak da adlandırılmaktadır.

### 3.2.1.2.3.Pooling layer:

Görüntünün boyutlarının fazla olması durumunda işlem yoğunluğundan kurtulabilmek için kullanılan katmandır.



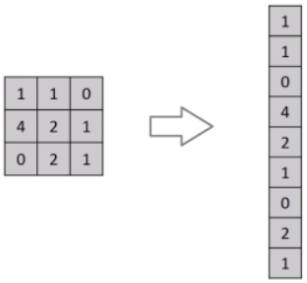

Şekil 3.2.1.2.3.pooling and downsampling gösterimi [2]

Farklı pooling yöntemlerinin bulunması ile birlikte en çok tercih edilen pooling yöntemi max pooling'dir. Bu yönteme göre görüntü matrisinin içerisinden 2x2 boyutunda matrisler belirlenir bu matrisin içerisindeki en büyük değer yeni görüntü matrisine eklenir. Bu şekilde görüntü veya aktivasyon map'lerinin boyutları kayıp en aza indirgenerek küçültülmüş olur.

Pooling işlemi uygulanan bir hacmin derinliğinde değişiklik olmaz.4x4x10 boyutundaki bir görüntü matrisi pooling işleminden sonra 2x2x10 boyutlarına sahip olur.

### 3.2.1.2.4.Flattening layer:

Flattening layer bir geliştiricinin tercihine göre sıralanmış olan katmanların ardından sonuç olarak oluşan aktivasyon map'inin fully connected layer'a iletilebilmesi için gerçekleştirilen katmandır.




Flatting

Bu katmanın sonucunda en son üretilmiş olan aktivasyon map'i tek boyutlu bir dizi haline getirilmektedir. Bu dizinin verileri sınıflandırma işleminde kullanılacak olan sinir ağlarının input değerileri olarak kullanılacaktır.

#### 3.2.1.2.5.Fully connected layer:

Sonuç olarak elimize ulaşan aktivasyon map'i görüntünün birçok özelliği kontrol edilerek elde edilen değerlerden oluşmaktadır. Bu değerlerin bir sinir ağları aşamasından geçirilmesi ile sınıflandırılması yapılmaktadır. Bunun sonucunda görüntü hakkında bir karar verilmektedir.

### 3.2.1.3.Loss function:

Bir sinir ağı içerisindeki nöronların ağırlık fonksiyonlarının mantıklı sonuçlar üretebilmesi için eğitilmiş olması gerekmektedir. Bu eğitme işlemi eğitim için verilmiş olan veri üzerinden gerçekleştirilir. Standart bir sinir ağı içerisinde bir nöronun girdilerinin ağırlık değerlerinin doğru ve yanlış olan girdi seçeneklerini seçebiliyor olması gerekir ancak başlangıçta input değerlerinin hangi özelliklere sahip olduğu bilinmediği için sinir ağı tamamı ile mantıksız bir sonuç üretir. Bu hataya sebep olan ağırlık fonksiyonunun daha mantıklı bir sonuç üretebilmesi için ilk olarak bu fonksiyonun ne boyutlarda bir hata yaptığının tespit edilmesi gerekmektedir. Bu işlemi gerçekleştiren fonksiyona **loss function** denir.

Bir loss function data loss ve regularization loss değerlerinin toplamıdır. Data loss birçok farklı fonksiyon ile hesaplanabilmektedir. Bunlara örnek MAE,MSE,Cross Entropy, multiclass SVM fonksiyonları verilebilir.

Kayıp fonksiyonunun seçimi, sinir ağının çıkış katmanında kullanılan aktivasyon fonksiyonuyla doğrudan ilgilidir. Sıklıkla kullanılan loss function örnekleri:

$$S = \sum_{i=1}^{n} |y_i - f(x_i)|. \qquad S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

MAE(L1 loss) [20] MSE(L2 loss) [20]

Bu fonksiyonların birbirlerine göre avantaj ve dezavantajları bulunmaktadır. Data set olarak kullanılan veriye göre elde edilen başarı değişiklik gösterebilmektedir.

| | L1 loss | | L2 loss | |
|---|---|---|---|---|
| data set | training time | test accuracy | training time | test accuracy |
| news20 | 0.69 | 85.50% | 0.80 | 85.00% |
| MNIST | 2.98 | 92.93% | 11.64 | 92.54% |
| sector | 2.98 | 94.36% | 3.59 | 94.14% |
| rcv1 | 1.66 | 88.64% | 1.53 | 88.50% |

Tablo 3.2.1.3. L1 ve L2 loss fonksiyonları

Data loss değerinin hesaplaması için multiclass SVM fonksiyonunu kullanacak olursak data loss değeri:

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$ [20]

formülü ile hesaplanır. Bu işlem tek bir eğitim örneğinin ağırlık fonksiyonları ile oluşturduğu sonuçlarda yanlış sınıfın değerinin doğru sınıfın değerinde çıkarılıp bir eklenmesi ile hesaplanır. Bu işlem esnasında bir değerinin eklenmesinin amacı ağırlık fonksiyonunun doğru sınıf için verdiği sonucun yanlış sınıf için verdiği sonuç arasında birden fazla fark olmasını sağlamaktır.

Bu formülün herbir eğitim örneği için uygulanması gerekmektedir.

$$L(W) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W), y_i)$$ [20]

Ortaya çıkan her bir sonucun ortalaması eğitim setimiz için uygulanan ağırlık fonksiyonlarının loss function değerini vermektedir.[9]

Cross entropy loss fonksiyonu genellikle softmax veya sigmoid aktivasyon fonksiyonu ile kullanılmakta olan bir kayıp fonksiyonudur. Bu iki farklı türü bulunmaktadır. Sadece iki farklı class için kullanılan cross entropy fonksiyonu :

$-(y\log(p)+(1-y)\log(1-p))$ [20]

(p: olasılık değeri y : gerçek sonuç)

Şeklinde gösterilir. Eğer sistem iki adet'ten fazla sınıfa sahip ise :

$-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c})$ [20]

(c: her bir sınıf , o: her bir sınıf için gözlemlenen olasıık değeri )

Şeklinde ifade edilir.

Bunun yanında data loss değerine ek olarak regularization loss adı verilen bir kayıp fonksiyonu daha genel kayıp fonksiyonuna eklenmektedir. Regularization teknikleri genelde tasarlanan modelde overfitting önleyerek başarımı artırmak için kullanılmaktadır. Regularization loss fonksiyonlarına birçok farklı versiyonu bulunmaktadır. En sıklıkla kullanılan regularization loss fonksiyonları :

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

L2 regularization loss [9]

$$R(W) = \sum_k \sum_l |W_{k,l}|$$

L1 regularization loss [9]

Bu fonksiyonlardan hangisinin kullanılacağının seçimi genellikle filtrenin sahip olduğu değerlerin dağılımına göre belirlenmektedir.

Regularization loss değeri genellikle bir hiperparametre ile çarpılır.

$$\lambda R(W)$$

λ parametresi düzenlileştirme boyutunu belirleyen parametredir. Bu parametreye geliştirici tarafından duruma göre en iyi sonucu veren değer atanır.

Gerçekçi bir kayıp değerine ulaşılabilmesi için gerekli olan data loss ve regularization loss değerlerinin toplamı bize sahip olduğumuz filtre değerlerinin kayıp değerini verir.

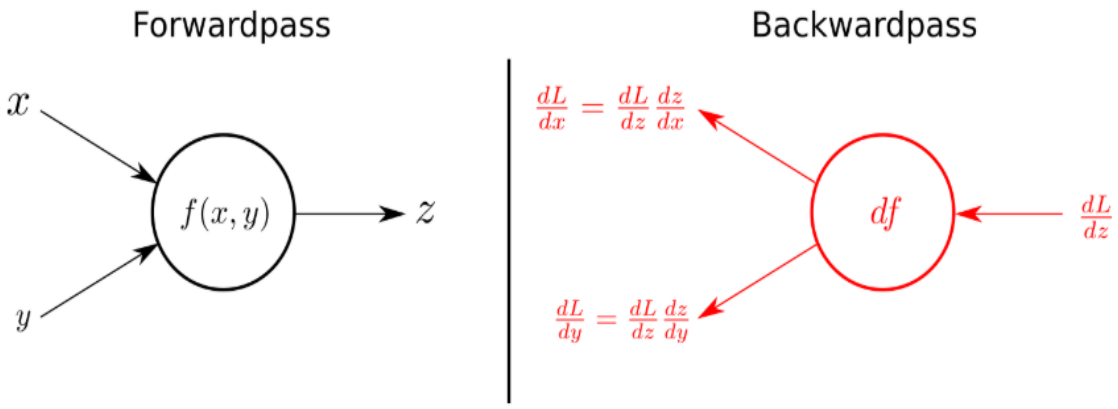
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + R(W)$$ Full loss

Full loss değeri sahip olduğumuz ağırlık fonksiyonunun kayıp değerini bize vermektedir.[5]

### 3.2.1.4.Backpropagation:

Sinir ağlarında eğitim aşaması iki aşamada gerçekleşir. İlk aşama eğitilmek istenilen filtre ile standart katmanların gerçekleştirilmesidir. Bu işleme **forward propagation** denir. İkinci aşama ise geriye doğru gerçekleştirilen aşamadır. Bu aşamada ilk aşamada hesaplanan değerlere göre filtre içerisindeki ağırlık değerleri düzenlenir. Bu aşamaya **backward propagation** denir.




Şekil 3.2.1.4.Forward and backward

Forward pass işleminde x ve y input değerlerine göre f fonksiyonu ile z değeri hesaplanmaktadır. Bu standart bir nöron içerisindeki aktivasyon fonksiyonu ile benzeştirilebilir. Backward pass işleminde ise kayıp fonksiyonunun (L fonksiyonu) z değerine göre kısmi türevi alınmaktadır. Ağ üzerindeki her bir input değerinin local eğim değerinin hesaplanabilmesi için zincir kuralı kullanılmaktadır.[8]

Backward pass işlemindeki ∂L/dout değerindeki out terimi convolutional layer içerisindeki output değerini temsil etmektedir. F değerinin filtreyi temsil etmesi durumunda backward pass işlemi ile dL/dF denklemi bulunur. Bu değer zincir kuralına göre **∂L/∂F= ∂L/∂out * ∂out/∂F** eğitliği ile bulunur. ∂L/∂F değeri filtrenin loss  gradient değerini bulur. Bu değer filtre içerisindeki değerin ne kadar değiştirilmesi gerektiğini belirlemektedir. Bu F değeri convolutional layer içerisinde birden fazla output değerini etkilediği için her bir output değerine göre eğim hesaplanmalıdır.

$$\frac{\partial L}{\partial \text{filter}(x,y)} = \sum_i \sum_j \frac{\partial L}{\partial \text{out}(i,j)} * \frac{\partial \text{out}(i,j)}{\partial \text{filter}(x,y)}$$

Bu formül filtre içerisindeki her bir ağırlık değerinin etkilediği output değerlerine göre ağırlık değerinin loss gradient değerini hesaplar. Filtre değerlerinin güncellenmesi için kullanılan formül :

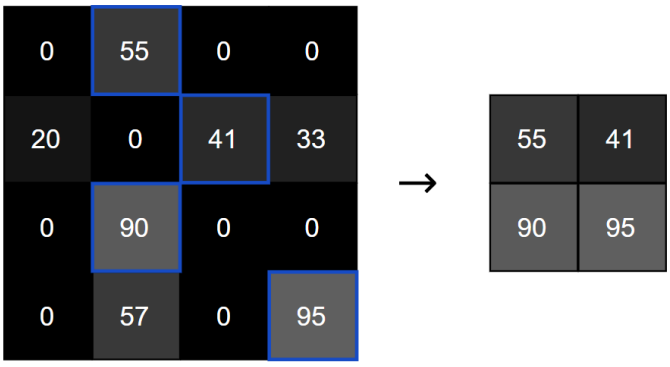
$$W_{x,y} = W_{x,y} - \gamma * ( \partial L / \partial W_{x,y} )$$

Formül içerisindeki $\gamma$ değeri **learn rate** değeridir. Duruma göre geliştirici tarafından en iyi değer verilmektedir.[8]

Bu işlem sinir ağı içerisindeki her bir katman için gerçekleştirilmektedir. Her bir katman içerisindeki loss gradient değeri zincir kuralı ile üst katmanlardan alt katmanlara iletilmektedir

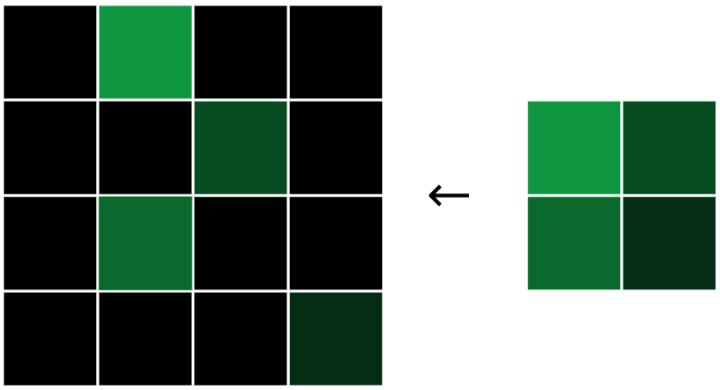
**Max-pooling için Backpropagation:**

Max pooling layer bir öğrenme aşaması içermez çünkü ağırlık değerleri yoktur ancak bir backpropagation metoduna ihtiyaç duyulmaktadır. Bunun sebebi üst katmanlara loss gradient değerinin iletilmesi gerekmesidir.




An example forward phase that transforms a 4x4 input to a 2x2 output

Max pooling yönteminde 2x2 boyutundaki alan içerisinde en yüksek değer yeni matrisi oluşturmaktadır.



An example backward phase that transforms a 2x2 gradient to a 4x4 gradient

Backward aşamasında 2x2 boyutundaki bir output gradient matrisinin boyutları iki katına çıkar. 4x4 boyutundaki gradient matrisi içerisinde maximum değere bakmayan input noktalarının gradient değeri 0 olur. Eğer bir input pixeli maximum değere bakıyor ise output gradient matrisi içerisindeki değer input gradient matrisi içerisindeki değer ile eşit olur. Bu matris içerisinde maximum değerin hangi konumdan geldiğinin bir ön bellekte tutulması gerekmektedir.

### 3.2.1.5.Weight initialization:

Sinir ağı içerisinde kullanılacak olan ağırlık fonksiyonlarının eğitimi kayıp fonksiyonunun eğimi aracılığı bulunmaktadır. Ancak ağırlık fonksiyonlarının başlangıç değerlerinin doğru seçilmesi de eğitim aşamasında büyük bir önem taşımaktadır. Seçilen değerler genellikle rastgele değerlerdir. Rastgele değer atamasının yapıldığı aralık ağırlık fonksiyonu için büyük önem taşımaktadır.

Seçilecek olan ağırlık değerlerinin varyans değerinin temkinli bir şekilde seçilmesi gerekmektedir. Varyans değerinin aşırı yüksek bir değer seçilmesi durumunda fazla sayıda saklı katman içeren sinir ağlarında aktivasyon değerlerinin aşırı yüksek çıkmasına ve sigmod, tanh gibi aktivasyon fonksiyonların doymasına sebep olur. Böyle bir durumda eğim aktivasyon map'in bütün eğim değerlerinin sıfır olmasına sebep olur. Bu sebeple aşırı yüksek bir değer tercih edilmemelidir. Varyans değerinin gereğinden küçük olması eğim değerlerinin aşırı küçülmesine ve öğrenme aşamasının yavaşlamasına ve durmasına sebep olur.

```
input layer had mean 0.001776 and std 1.001663
hidden layer 1 had mean 0.000196 and std 0.213644
hidden layer 2 had mean 0.000030 and std 0.047696
hidden layer 3 had mean -0.000009 and std 0.010655
hidden layer 4 had mean -0.000001 and std 0.002383
hidden layer 5 had mean 0.000000 and std 0.000531
hidden layer 6 had mean 0.000000 and std 0.000119
hidden layer 7 had mean 0.000000 and std 0.000027
hidden layer 8 had mean -0.000000 and std 0.000006
hidden layer 9 had mean 0.000000 and std 0.000001
hidden layer 10 had mean -0.000000 and std 0.000000
>>>
```

0.01 standart sapma ile

```
input layer had mean 0.000886 and std 1.000348
hidden layer 1 had mean 0.000863 and std 0.982128
hidden layer 2 had mean -0.000688 and std 0.981584
hidden layer 3 had mean 0.000211 and std 0.981664
hidden layer 4 had mean 0.001785 and std 0.981564
hidden layer 5 had mean -0.001634 and std 0.981635
hidden layer 6 had mean 0.004385 and std 0.981795
hidden layer 7 had mean -0.000859 and std 0.981712
hidden layer 8 had mean -0.001264 and std 0.981655
hidden layer 9 had mean -0.000412 and std 0.981658
hidden layer 10 had mean 0.000005 and std 0.981395
>>>
```

1.0 standart sapma ile

Bu güne kadar en mantıklı ağırlık fonksiyonu tanımlama yöntemlerinde biri Glorot algoritmasında kullanılan **xavier initialization'**dur . Bu yöntem bir katman içerisindeki nöron sayısını kullanarak bir ağırlık fonksiyonu oluşturur. Nöron sayısının N olması durumunda rastgele oluşturulmuş ağırlık fonksiyonunun $1/\sqrt{n}$ değeri ölçeklenmesi ile oluşturulur. Ancak reLU fonksiyonu için başarılı değildir.

Bir diğer ağırlık fonksiyonu tanımlama yöntemi ise **kaiming initialization**'dur. Bu yöntemin xavier initialization'a göre reLU fonksiyonu içinde yüksek başarı ile çalışmaktadır. Rastgele oluşturulmuş olan ağırlık fonksiyonunun her bir değerinin $\sqrt{(2 / n)}$ değeri ile çarpılması ile elde edilir.[7]

## 3.2.2 FCN(Fully convolutional neural network)

### 3.2.2.1 Semantic Segmentation:

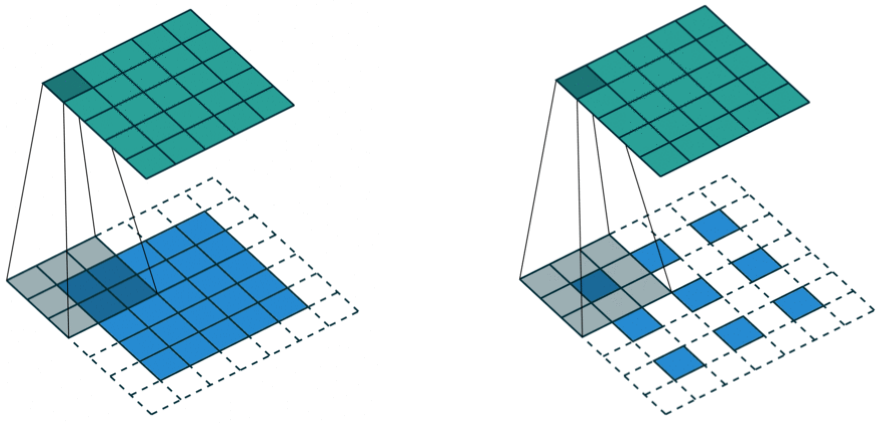
Semantic segmentation işlemi görüntü içerisindeki cismin pozisyonunun tespit edilmesi için kullanılan bir işlemdir. Aynı özellikteki objelerin birbirinden ayırt edilebilmesi için kullanılmaktadır. Bu işlem CNN işleminin aksine görüntüyü bir bütün olarak değil pixel bazında inceler. Bunun sebebi görüntüdeki her bir pixelin bir objeye ait olmasıdır.

**3.2.2.2. FCN'nin aşamaları:**

FCN işlemi de CNN işleminin özellik yakalama kısmı ile aynı şekilde gerçekleştirilir. Defalarca ve istenilen sonuca en yakın sonuç üretilene kadar arka arkaya convolution ve max pooling katmanları dizilir. Oluşan sonuç girdi olarak verilen görüntüden çok daha küçük olması muhtemeldir. Oluşan görüntünün mantıklı bir görüntü üretmesi için upsampling işlemi gerçekleştirilir. Bu işlem ile convolution katmanlarının ürettiği matrisin boyutlarının gözle ayırt edilebilir bir sonuç oluşturması için gerçekleştirilir.

**3.2.2.2.1 Deconvolutinal layer(upsampling):**

Convolution katmanının ürettiği sonuç sistem özelliklerine göre değişiklik gösterebilmektedir. Ancak bir convolution katmanının sonucu bir matristir ve bu matris "[yığın boyutu, yükseklik, genişlik, derinlik]" formundadır. Yığın boyutu değeri sisteme kaç adet farklı örneğin eğitildiğini ifade etmektedir. Yüksekli ve genişlik ve derinlik değerleri ise matrisin boyutlarını ifade etmektedir. Genellikle birden fazla convolution katmanı ve max pooling uygulandığı ve uygulanan katmanların stride değerleri birden büyük olduğu için çıktı matrisinin yükseklik ve genişlik değerleri girdi olarak verilen görüntünün boyutlarından küçüktür. Girdi boyutlarında bir çıktının elde edilebilmesi için convolutional katmanının tersinin gerçekleştirilmesi gerekmektedir. Bu katmana deconvolutional layer adı verilir.

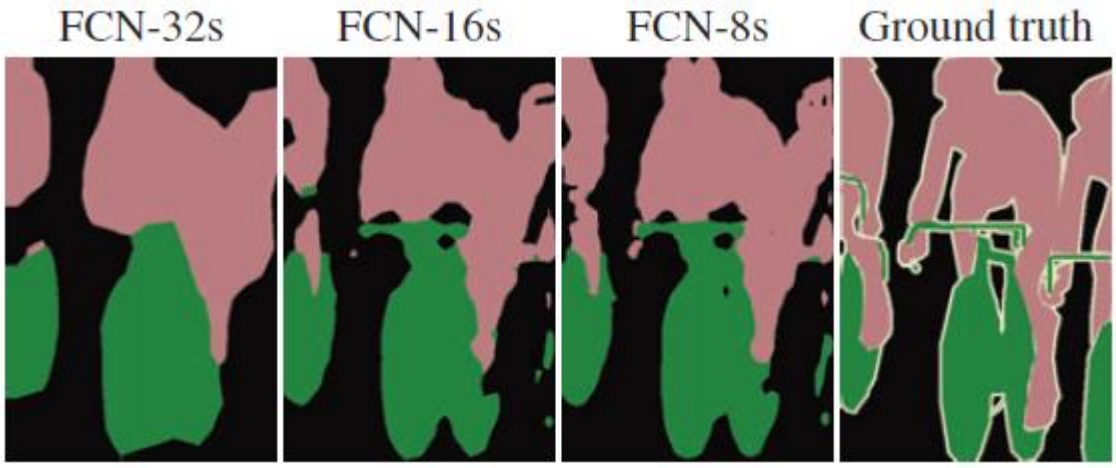


**3.2.2.2.1** Sol resim Convolution ; sağ resim Deconvolution

Bu işlem tıpkı convolution katmanı gibi filtreler ile girdi olarak alınan verinin birden fazla filtre ile matris çarpımının gerçekleştirilmesi ile elde edilir. Ancak standart bir convolution katmanının aksine stride(adım atma) işlemi gerçekleştirilmez. Bunun yerine girdi matrisinin yükseklik ve genişlik değerlerini elde edilmek istenilen yükseklik ve genişlik değerlerine arada boşluklar bırakarak elde eder ve filtreler ile matris çarpımını gerçekleştirir. Bu işlemin sonucu sisteme verilen girdi görüntüleri aynı yükseklik ve genişlik değerlerine sahip olur.

Deconvolution katmanının sonucunun derinliği kullanılacak filtre sayısı ile belirlenir. Sistemin kaç farklı objeyi ayırt etmesi isteniyor ise filtre sayısı ona göre belirlenmelidir. Upsampling işleminin sonucunda oluşan matrisin derinliğinin her bir indisi farklı bir objenin segmentlenmiş görüntüsünü içerisinde barındırır. Bu görüntünün elde edilebilmesi için sonuç matrisi transpose edilerek "[yığın, yükseklik, genişlik, derinlik]" formundan "[yığın, derinlik , yükseklik , genişlik]" formuna dönüştürülür. Bu formun ikinci eksenindeki matrislerin her biri bir objenin bulunduğu pozisyonun görüntüsüdür.

#### 3.2.2.2.2 Skipping:




3.2.2.2.2 Farlı katmanlarda skip işlemi gerçekleştirilerek üretilmiş semantic segmentation[18]

Sadece son convolutional katmana upsampleing işleminin gerçekleştirilmesi her zaman mükemmel sonuç üretmez. En son katman genellikle objenin kabataslak bir görüntüsünü ortaya çıkarır. Daha ayrıntılı bir görüntünün elde edilebilmesi için sistemin farklı seviyelerindeki convolution katmanlarının sonuçları için deconvolution işlemi uygulanmalı ve her bir deconvolution katmanının sonucu basit bir toplama işlemi ile birleştirilmelidir. Bu işleme "skip" adı verilmektedir.

### 3.2.3 Preprocessing:

Daugman's rubber işlemi ile düzenlenmiş olan data bir Numpy array ile tutulmaktadır. Bu datanın bir CNN modelinde işlenebilmesi için ilk olarak datanın uygun hale getirilmesi gerekmektedir. Bir CNN modelinde convolution veya gizli katmanla için tanımlanmış olan ağırlık değerleri girdi değerleri ile matris çarpımına tabii tutulur.

Bu matris çarpımının sonucunun varyans değerinin aşırı derecede küçük veya aşırı derecede büyük olmaması gerekir. Aksi takdirde sistem gereksiz büyüyebilir veya küçülebilir ve eğitilemez hale gelebilir. Bu nedenle katmana girdi datasının varyans ve ortalama değerlerinin ön görülebilir değerler olması gerekir. İki farklı tipte değişkenin matris çarpımının varyans değeri eğer değişkenlerin beklenen değerleri sıfıra eşit ise her bir değişkenin varyans değerinin çarpımına eşittir. Bu bu sayede doğru bir ağırlık fonksiyonu varyans değeri ile katmanın giriş ve çıkış verilerinin varyans değeri sabitlenmiş olur.

Datanın eğitiminin daha kolay olması için kullanılan yöntemlerden biri standadization'dur. D'nin datanın kendisini , M nin datanın ortalama değerini, S'nin ise datanın standart sapma değerini ifade etmesi halinde:

$D_S = ( D - M ) / ( S + epsilon)$ (epsilon : 1e-8)

formülü ile hesaplanır.

### 3.2.4 Dataset:

Bu projenin gerçeklenmesi sonucunda eğitilmesi ve test edilmesi hedeflenen veri Casia-iris-thousand dataset'idir. Bu dataset bin farklı kişiye ait yirmi bin görüntü içermektedir. Her bir kişiye ait sağ ve sol göz ayrı olmak üzere 40 görüntü bulunmaktadır.[21]

# 4. SİSTEMİN GERÇEKLENMESİ:

CNN (convolutional neural networks) bir görüntünün özelliklerinin yakalanması ve özellikleri tespit edilmiş bir görüntünün sınıflandırılması ile gerçekleşmektedir. Bu nedenle görüntü içerisinde spesifik noktalara hedef olmak bazen mantıksız olabileceği gibi bazı durumlarda gerekli olabilmektedir. Resmin tamamı sistemin özelliklerini öğrenmesi için gerekli olması durumunda resmin tamamının işlenmesi mantıklıdır. Ancak resmin sadece küçük bir kısımındaki özellikler sistem için önemli ise görüntünün geriye kalan kısımları sistemin gereksiz bilgiler edinmesine sebep olabilmektedir. Örneğin bir kedi resminde arka plandaki bir insan görüntüsü sistemin insana ait özellikleri yakalar. Buda sistemin öğrendiği bazı özelliklerin gereksiz olmasına sebep olur. Bu sorunun üstesinden gelmek için sisteme öğretilen öğretilen dataset'in daha fazla örnek veri bulundurması veya data augmentation işlemi ile sahip olunan dataset'in genişletilmesi ile çözülebilmektedir. Ancak dataset'in sınırlı olduğu durumlarda bu sorunun üstesinden gelmenin yolu segmentation işlemidir.

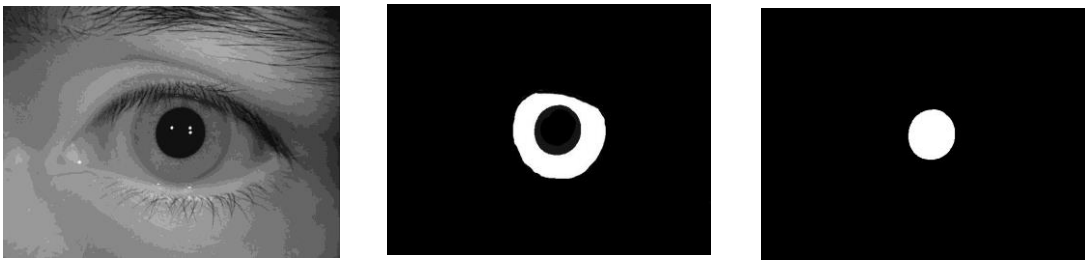
## 4.1 Kullanılan araçlar:

Sinir ağı oluşturmak için günümüzde sci-learn, keras, tensorflow, pytorch gibi kütüphaneler kullanılmaktadır. Bu projenin gerçekleştirilmesi aşamasında sadece sinir ağı oluşturulmayacak, birçok matris işlemi gerçekleştirilecektir. Bu nedenle daha çok matematiksel işlemler üzerine yoğunlaşmış ve ekran kartı kullanımını destekleyen tensorflow tercih edilmiştir. Tensorflow kütüphanesi gerçekleştirdiği bir çok işlemin sonucunu bir Numpy dizisi olarak döndürür. Bu nedenle Numpy kütüphanesi kullanılması kaçınılmazdır.

## 4.2 Fcn’nin (fully convolutional network) geçekleştirilmesi:

### 4.2.1 Ground truth dataset oluşturulması

İlk olarak sistemin doğru olanı ayırt edebilmesi için bir ground truth ( temel gerçek) dataseti’nin oluşturulması gerekmektedir. Bu aşamada Fcn işleminde eğitim için kullanılacak olan iris görüntülerinin iris ve göz bebeği noktaları çizilmesi gerekmektedir.

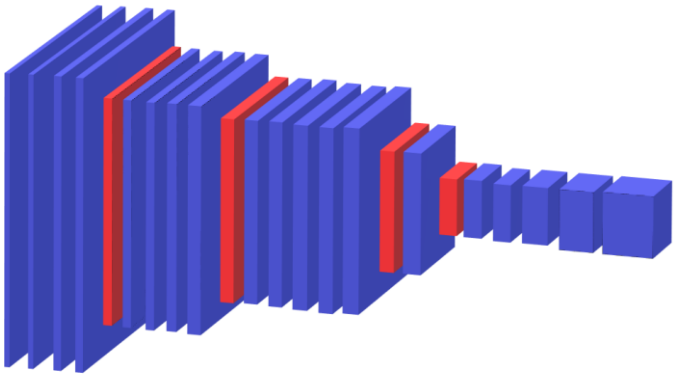


4.2.1. Sol görüntü resim, orta görüntü İris, sağ görüntü Göz bebeği

Program bu iki çizimin haricinde bir görüntü daha oluşturur. Bu görüntü hem iris hemde göz bebeği olmayan her bir pixel için beyaza boyanır geriye kalan pixeller siyaha boyanır. Oluşan bu üç çizim FCN işleminin kayıp fonksiyonuna verilecek olan doğruluk değerleridir.
Bu işlem sistemin genel bir başarı elde edebilmesi için birçok görüntü için Görüntü için uygulanması gerekmektedir.

### 4.2.2. FCN convolution katmanlarının oluşturulması:

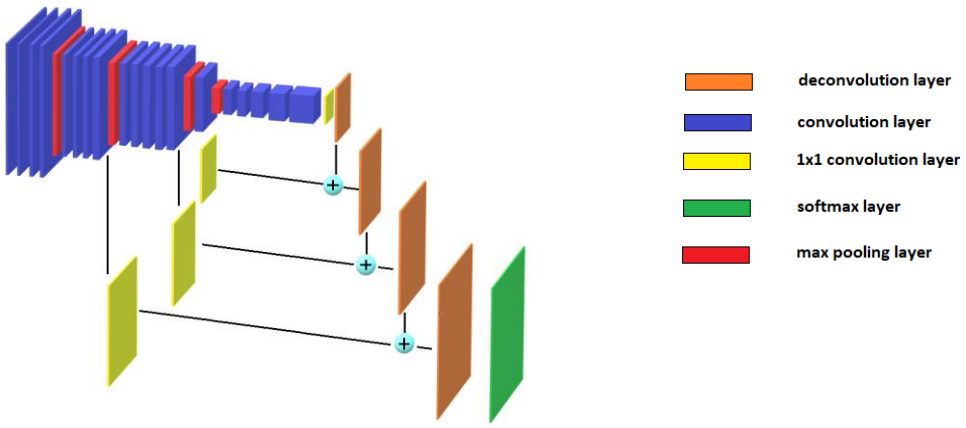
Bu işlem tanıtılacak olan datasetin zorluk seviyesine göre değişiklik gösterebilen tamamı ile deneme yanılma yolu ile en yüksek başarının elde edildiği modelin bulunması aşamasıdır. Aşırı derecede derin bir model sistemin gereğinden fazla bilgi edinmesine ve aşırı sığ bir model ise sistemin yeterince bilgi edinememesine yol açabilmektedir. Bu projede gerçekleştirilen model 19 adet convolution katmanı ve 4 adet max pooling katmanı kullanılmıştır.



4.2.2..Fcn modeli Kırmızı: max pooling mavi: convolution

Bu katmanların arasından farklı boyutlarda olan 4 adet convolution katman seçilmiş ve herbirinin sonuna 1x1 boyutlarında filtreler ile işlem yapan convonlution katmanı eklenmiş ve bunun ardından upsampling işlemi gerçekleştirilmiştir. Upsampling işleminin ardından oluşan 4 farklı matris matris toplama işlemi ile toplanmış ve sonuç olarak 3 derinliğe ve girdi görüntünün yükseklik ve genişlik değerlerine eşit boyutlarda veri elde edilmiştir.

### 4.2.3 Upsampling ve skipping aşaması:




4.2.3.Upsampling ve skip modeli

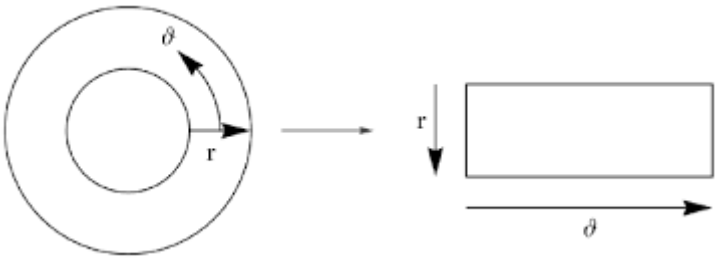
Bu katmanların gerçeklenmesinin ardından systemin kayıp değerinin hesaplanabilmesi için bir kayıp fonksiyonu tanımlanmıştır. Bu sistemde kayıp fonksiyonu olarak softmax fonksiyonu ile birlikte sıkça çoğunlukla tercih edilen cross entropy fonksiyonudur. Bu fonksiyon sistemin vermis olduğu cevaba göre bir kayıp değeri oluşturur. Bu kayıp fonksiyonu optimizere parameter olarak verilir ve optimizer bu değerin olabildiğince düşük hale getmeye çalışır.

**Adam Optimizer:** Yapılan testlerde Adam optimizerin diğer optimizerlerden daha hızlı bir şekilde zirveye ulaştığı ve maksimum noktasının diğer optimizer'lerin maksimum noktasından daha yüksek olduğu gözlemlenmiştir. Bu nedenle modelin optimizasyon işlemi adam optimizer ile gerçekleştirilmiştir.

FCN işlemi aşamasında kayıp değerinin düşük olması hayati önem taşımaktadır. Bunun sebebi kayıp değeri çok küçük dahi olsa bazı durumlarda spesifik özelliklerin öğrenilememesine sebep olabilmesidir (CNN gibi modellerde 0.1 gibi bir kayıp değeri göz ardı edilebilir ancak Fcn modelinde bu sistemin hata yapmasına sebep olur). Bu proje için geliştirilen Fcn modelinin eğitimi esnasında karşılaşılan sorunlardan bir tanesi göz bebeği içerisine yansıyan ışığın öğrenilmesi olmuştur. Kayıp değerinin 0.1 gibi değerlerde eğitimin durdurulması durumunda system göz bebeği içerisindeki ışık yansımasını genellikle tanıyamamıştır. Dataset'in genişletilmesi ve kayıp değerinin 0.001 gibi değerlere ulaşana kadar eğitimin devam ettirilmesi bu sorunun üstesinden gelmiştir.

## 4.3 Daugman’s Rubber:

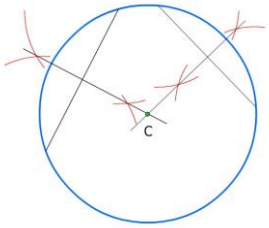
Bir çok kişiye ait iris görüntüsü içerisindeki irisin boyutları ve pozisyonu değişiklik gösterebilmektedir. Bu nedenle irisin pozisyonunun belirlenmesi ve CNN ile eğitilebilir hale getirilmesi gerekmektedir. CNN gibi modeler sabit büyüklükteki verileri işleyebilmelerinden kaynaklı her bir irisin sahip olduğu boyutların aynı olması gerekir. Görüntünün sabit bir boyuta getirilmesi numpy modülünün resize fonksiyonu gibi yöntemlerle de yapılabilmesine karşın bu her zaman mantıklı olmayabilir. Bu nedenle irisin sabit bir boyuta indirgenmesi işlemi için daugman’s rubber yöntemi tercih edilmiştir. Daugman’s rubber yuvarklak bir yapıya sahip bir görüntünün düz bir yapıya dönüştürülmesini sağlamaktadır.




4.3.Daugman’s rubber model

### 4.3.1 Merkez noktası ve yarıçapların bulunması

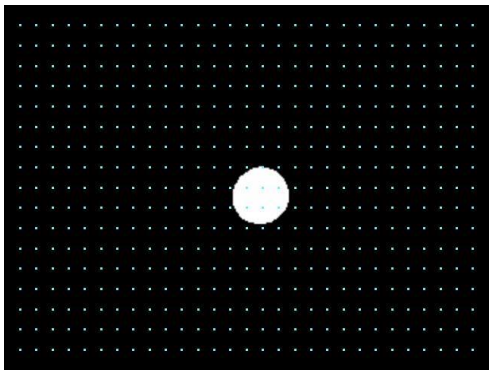
İlk olarak göz bebeğinin merkez noktasının tespit edilmesi gerekmektedir. Bu merkez noktası daha sonra iris üzerinde yeni görüntü içinde kullanılacak olan pixel noktalarını hesaplarken gereklidir.




4.3.1.a Çemberin merkezi

Çemberin (göz bebeğinin) merkez noktası çember üzerindeki rastgele noktalar arasındaki doğruların dikey doğrularının kesişim noktasıdır. Bu noktanın tespit edilebilmesi için ilk olarak çemberin üzerinde rastgele noktalar bulunur.

Rastgele noktaların bulunması için FCN işlemi ile tespit edilmiş olan göz bebeği üzerinde hangi pixel bazında hangi yükseklik ve genişlik değerleri arasında olduğu tespit edilebilmesi için görüntünün yükseklik ve genişlik değerleri belirli sayılarda aralıklara bölünür. Bu işlemin amacı sistemin performansının düşmesinin engellenmesidir. Görüntünün tamamının incelenmesi büyük bir yığının eğitilmesi amaçlandığında performansın aşırı derece düşmesine sebep olur. Bu nedenle belirli noktalara bakılarak tahmini bir pozisyon belirlenir.



4.3.1.b Kontrol edilen noktalar

Yukarıdaki resimde parçalara ayrılmış görüntünün kontrol edilecek olan köşe noktaları gösterilmiştir. Eğer bu noktaların değeri 128'in üzerinde ise (128 üzeri göz bebeği olduğu anlamına gelir. Aksi durumlar için 128'den küçüktür) bu nokta işaretlenir. Bütün köşe noktaları kontrol edildikten sonra göz bebeğinin bulunduğu yükseklik ve genişlik değerleri tahmini olarak bulunmuş olur.

Pozisyonu tespit edilmiş çemberin üzerindeki tespit edilir. Bu noktaların arasındaki doğrular ve bu doğruların dikey doğrularının kesişim noktası tespit edilir. Bu kesişim noktası çember üzerindeki dört nokta aracılığı ile bulunan merkez noktasıdır.

Bazı durumlarda elde edilen göz bebeği görüntüsü tam olarak yuvarlak olmayabilir. Bu duruma genellikle gözü incelenen kişinin gözünün bir kısmının kapalı olması neden olur. Bu nedenle daha gerçekçi bir sonuç için birçok nokta ile merkez noktası hesaplanmalı ve ortalama değeri merkez noktası olarak alınmalıdır. Bu projede 16 adet çember üzerindeki noktadan 8 adet doğru elde edilmiş ve bu doğrular ile 8 adet merkez noktası hesaplanmıştır.

Bu merkez noktalarının standart sapma değeri belirli bir değerin (üç pixel) üzerinde olması durumunda standart sapmanın büyük olmasına sebep olan noktalar elenir. Standart sapma değeri istenilen değerin altına düşmesi halinde geride kalan noktaların ortalama değeri genel merkez noktasıdır.

Merkez noktasının tespit edilmesinin ardından irisin yarı çapının hesaplanması gerekir. FCN ile tespit edilen iris görüntüsü üzerinde merkez noktasından başlanarak irisin ve göz bebeğinin yarıçap değeri hesaplanır. Bazı durumlarda irisin bir kısmının gözükmeme ihtimalinden dolayı farklı açı değerlerine göre birden fazla yarı çap değeri hesaplanır ve ortalama değeri yarıçap olarak belirlenir.

### 4.3.2 Görüntünün normalize edilmesi:

İrisin ve göz bebeğinin yarı çapının belirlenmesinin ardından oluşması istenilen düzleştirilmiş görüntünün yükseklik ve genişlik değerlerine atılacak adımların büyüklük değerleri hesaplanır. İrisi yarı çapının $R_i$ , göz bebeği yarı çapının $R_g$ , düzleştirilmiş görüntünün yükseklik değerinin $Y_d$ olması durumunda adım genişliği :

$$\text{Adım} = (R_i - R_g) / Y_d$$

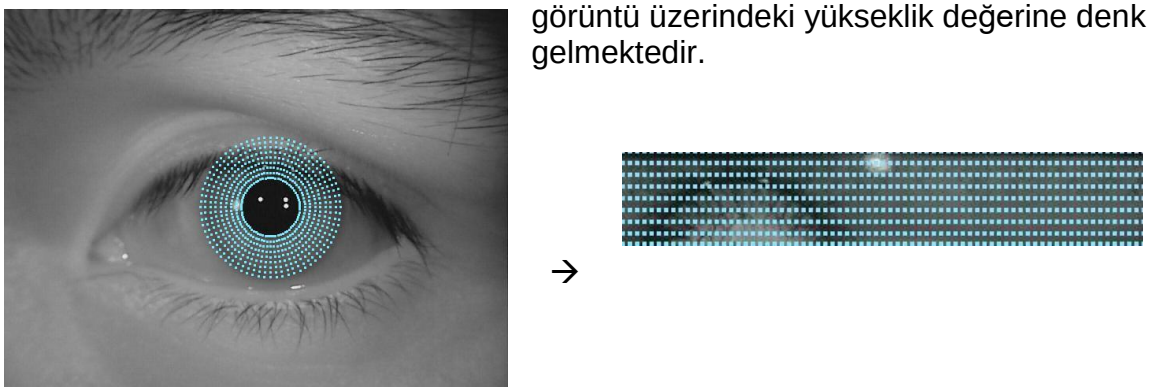
formülü ile belirlenir .

Benzer şekilde açı değerinin değişimi düzleştirilmiş görüntünün genişlik değerinin $X_d$ olması durumunda adım genişliği:

$$\text{Açı değişimi} = ( 360 / X_d )$$

formülü ile belirlenir.

Merkez noktası, yarıçap ve adım değerlerinin hesaplanmasının ardından sıfır açı değerinden başlanarak göz bebeği yarıçapının sonundan iris yarıçapının sonuna kadar belirlenen adımlarla pixel'ler kişinin resminden alınır ve düzleştirilmiş görüntünün içerisine yerleştirilir. Hesaplaması yapılan her bir açı değeri düzleştirilmiş görüntü üzerindeki yükseklik değerine denk gelmektedir.



4.3.2 Görüntü üzerinde belirlenen noktalar ve düzleştirilmiş noktalar

Her bir açı değeri için pixel konumlarının hesaplanması ve görüntünün üzerine yerleştirilmesi ile daugman's rubber işlemi tamamlanmış olur.

## 4.4 MCNN gerçeklenmesi:

### 4.4.1 Preprocessing:

Datanın varyans değerinin sabitlenebilmesi ve xavier initializerin amacına uygun bir şekilde hareket edebilmesi için ortalama değeri 0'a ve varyans değeri 1'e getirilmelidir. Bu işlem data içerisindeki her bir değerden datanın ortalama değerinin çıkarılması ve standart sapma değerine bölünmesi ile elde edilir. Bu işlemin ardından datanın ortalama değeri 0 ve varyans değeri 1 olur. Bu sayede aşırı derecede küçük veya büyük varyans değerlerinden kaçınılmış ve modelde daha derin katmanlardaki varyans değerlerinin patlaması veya ölmesi engellenmiş olur. Sistemin ağırlık fonksiyonları Relu aktivasyon fonksiyonları için kaiming initializer, softmax aktivasyon fonksiyonu için xavier normal initializer kullanılmıştır.

Eğitim için kullanılacak olan dataset ilk olarak standardization işlemine tabii tutulur. İlk olarak datanın varyans ve ortalama değerleri hesaplanır. Ardından dataset'in içerisindeki her bir değerden ortalama değeri çıkarılır ve varyansın karekköküne bölünür. Bu işlem test dataset'i içinde uygulanması gerekmektedir. Ancak test dataset içinde eğitim için kullanılan dataset'in ortalama ve varyans değeri kullanılır. Sistemde test edilecek olan her dataset'in eğitim datasetinin ortalama ve varyans değeri ile standardize edilmesi gerekmektedir.

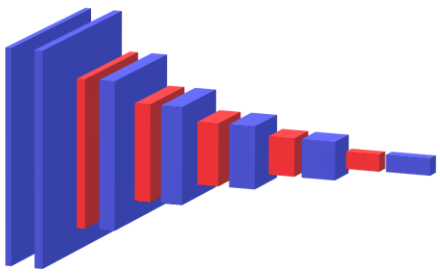
Eğitim ve test dataset'lerinin oluşturulmasının ardından sistemin hangi sınıfın doğru olduğunu anlayabilmesi için bir label oluşturulur. Bu label içerisine her bir örnek için sistemde bulunan sınıf sayısı adedince 0 eklenir ve sadece doğru cevabın adresindeki değer 1 yapılır. Bu label [örnek sayısı , sınıf sayısı ] boyutundadır.

### 4.4.2 MCNN aşaması:

#### 4.4.2.1 CNN modelinin gerçeklenmesi:

MCNN birden fazla sayıda CNN yapısının birleştirilmesinden oluşmaktadır. Farklı boyutlarda farklı özelliklerin yakalayabilmekte daha yüksek bir başarı elde edeceği düşünüldüğünden tercih edilmiştir.

İlk olarak FCN ve daugman's rubber ile casia iris-thousand dataseti CNN için kullanılabilir hale getirildi. Ardından MCNN için kullanılmak üzere bir CNN modeli tasarlanmıştır. Bu işlem esnasında data için en yüksek başarının hangi model için elde edildiği gözlemlendi.



4.4.2.1 CNN özellik yakalama aşaması modeli

Bu modelin düzeni şu şekilde gerçekleştirilmiştir:

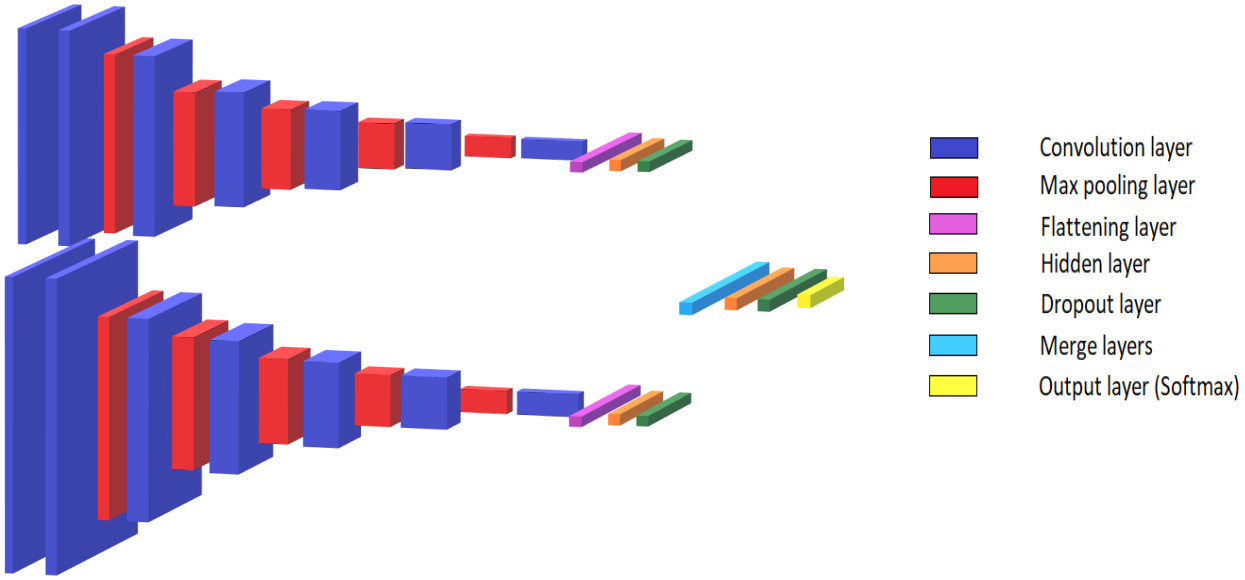
- 360x48 boyutlarında giriş görüntüsü ( Daugman's rubber ile düzleştirilmiştir.)
- 11x11 boyutlarında 16 adet kernel (çıktı boyutu aynı)
- 5x5 boyutlarında 32 adet kernel (çıktı boyutu aynı)
- Max pooling (çıktı boyutu 180x24)
- 3x3 boyutlarında 48 adet kernel (çıktı boyutları aynı)
- Max pooling (çıktı boyutları 90x12)
- 3x3 boyutlarında 64 adet kernel (çıktı boyutları aynı)
- Max pooling (çıktı boyutları 45x6)
- 3x3 boyutlarında 96 adet kernel (çıktı boyutları aynı)
- Max pooling (çıktı boyutları 24x3)
- 3x3 boyutlarında 128 adet kernel (çıktı boyutları aynı)
- Max pooling (çıktı boyutları 12x2)
- 3x3 boyutlarında 196 adet kernel (çıktı boyutu 12x2)

Bu model sistemin özellik yakalama kısmıdır. Modelin çıktısı bir sınıflandırma aşamasından geçirildikten sonra bir sonuç elde edilebilir. Bu nedenle standart bir CNN'nin nasıl bir başarı elde edebildiği gözlemlenebilmesi için model için bir tamamen FC (fully connected layer) geliştirilmiştir. FC ye input olarak modelin son katmanının ürettiği matris verilir. Bu matrisin FC ile işlenebilmesi için ilk olarak düzleştirilmesi gerekir. Bu işleme flattening denir. Boyutları [yığın boyutu, 12, 2] olan model çıktısı [yığın boyutu, 24] formuna dönüştürülür. Ardından iki adet gizli katman üretilir. Her bir gizli katmanda girdi boyutunun üçte ikisi kadar nöron bulunmaktadır. Ardından eğitilen datasetin durumuna göre çıktı sayısı belirlenir.

#### 4.4.2.2 CNN modellerin birleştirilmesi

Geliştirilen CNN modeli 360x48 boyutlarında çalışmaktadır. MCNN modelleri bir dataset'in farklı boyutlarında geliştirilmiş olan bir modeldir. MCNN modelinde birden fazla CNN modeli bir araya getirilerek ortak bir sonuç elde edilmesi ile üretilir.

Proje kapsamında geliştirilen sistemde ilk olarak farklı boyutlar için çalışan CNN modelleri geliştirilir. Bu nedenle önceden geliştirilen 360x48 girdi boyutuna sahip olan CNN ile aynı modelde ancak farklı girdi boyutlarına sahip ekstra bir adet model daha geliştirilmiştir. Bu modelin katmanları ise ilk model ile aynı ancak girdi boyutu 270x32'dir.




4.4.2.2 MCNN modeli

Birleştirilmek üzere tasarlanmış farklı boyutlarda girdi alan bu 3 modelin birleştirilmesi için ilk önce CNN modellerinin çıkış noktalarının düzenlenmesi gerekir. Bu nedenle ilk olarak her bir CNN modelinin son katmanı olan çıkış katmanı kaldırılır. Geriye kalan modelin çıkış noktası FC kısmının ikinci gizli katmanı olur. Her bir CNN modelinin son katmanı bir birleştirme işlemi ile birleştirilir. Birleştirilen çıktı değerleri iki gizli katmana sahip yeni bir FC modeli ile sınıflandırılır.

## 4.5 Olasılık Alt Limitinin belirlenmesi:

Modelin son katmanında softmax kullanıldığı için sistemin oluşturacağı sonuç 0 ile 1 arasında bir olasılık değeridir. Bu durumda dataset için gerçekleştirilen başarı testi birçok farklı sınıf arasından en yüksek olasılığa sahip olan sınıf sistemin kararıdır. Ancak sistem verdiği cevap belirli bir olasılık değerin altında olması durumunda aslında verdiği cevaptan emin olmadığı anlamına gelmektedir. Sistemin kendinden emin olması verdiği cevabın doğruluk değerini arttırır. Ancak eğer sistem yanlış bir cevaptan emin olması kabul edilebilir bir durum değildir.

Modelin eğitilmesi aşamasında genellikle bir başarı elde edilebildiğinin düşünülebilmesi için test için kullanılan data için yüksek bir başarı elde etmesi gerekir. Ancak iris tanıma sisteminde hatalı bir kişinin doğru cevap olarak verilmesi kabul edilemez. Bu nedenle sistemin gösterdiği doğru sınıfın diğer sınıflara göre daha yüksek olması yetmez. Bu sınıfın gerçekten kabul edilebilmesi için olasılık değerinin belirli bir değerin üzerinde olması gerekir. Bu sayede sistem emin olmadığı kişiler için onay verilmemiş olur.

Eğitim işleminin tamamlandığına karar verilebilmesi için bir test başarısı için bir sınır koyulur. Bu sınır eğitimin sonlandırılması için gerekli şarttır. Daha sonra bu test başarısından ne kadar emin olunduğunun test edilmesi gerekir. Bunun için eğitilen kişi sayısına göre değişiklik gösterecek olan bir olasılık değeri sınırı koyulur. Bu sınırı altındaki değerler kabul edilmez ve sistemin kişiyi tanıyamadığına kanaat getirilir. Bu sayede yanlış bir kişiye onay verilmemiş olur.

Olasılık alt sınırı olarak belirlenen değer yüzde 95'tir. Bu sınırın altında hiçbir sonuç doğru olarak kabul edilmeyecektir. Ancak sistemin bu rakamın üzerinde bir değer üretmesi de kesin bir sonuç değildir.

Sistemin üreteceği ihtimal değerlerinin güvenilirliği tartışılırdır. Modelin büyük bir model olması durumunda sistem bir olasılık değeri üretmekten çıkar ve tam anlamı ile emin cevaplar vermeye başlar. Bu durumda doğru cevap verdiği örnek için yüzde yüze yakın olasılık üretirken yanlış bir cevap verdiği örnek içinde yüzde yüze yakın bir sonuç üretebilmektedir. Bu durumun en büyük sebebi modelin büyük olması ve sistemin aşırı uç noktaları öğrenmesidir. Bu duruma **miscalibration** denilir. [22]

Sistemin sadece bir eşik değeri uygulanarak verdiği cevabın kesin olarak algılanması sistemin yanılmasına sebep olabilir. Bu nedenle miscalibration olayının üstesinden gelinmesi gerekir. Bu durumun üstesinden gelmek için birden fazla yöntem uygulanabilir. Bunlardan en mantıklı olanı regularization'dur. Ancak regularization'da her zaman mükemmel bir sonuç oluşturmayabilir ve yüksek bir l2 regularization hiper parametresi sistemin başarı değerlerinin düşmesine sebep olur.

Miscalibration olayının üstesinden gelmenin en zararsız yolu **temperature scaling** işlemidir. [22]

$$\mathrm{softmax}(x)_i = \frac{e^{y_i}}{\sum_j^N e^{y_j}} \quad \mathrm{softmax}(x)_i = \frac{e^{\frac{y_i}{T}}}{\sum_j^N e^{\frac{y_j}{T}}}$$

Sol: Normal softmax Sağ: ölçeklenmiş softmax [22]

Standart olarak uygulanan softmax fonksiyonunun parametrelerinden input değeri bir skaler değer ile bölünür. Bu durum engellenmesi durumunda softmax katmanı daha makul daha yumuşak olasılık değerleri ile sonuç üretmeye başlayacaktır.

Günümüzde popüler olan bu sorun tam anlamı ile çözülebilmesi mümkün değildir.Temperature scaling işlemi ancak softmax katmanının daha makul sonuçlar üretmesini sağlar ancak bunun bir garantisi yoktur.

## 4.6 Sistemin tekrardan eğitilebilirliği:

Bir CNN modelinin tekrardan eğitilebilmesi için birçok farklı yöntem bulunmaktadır. Bu yöntemlerden en sık kullanılanlardan biri “joint training” işlemidir. Hazırda bulunan modelin sahip olduğu sınıflara yeni sınıfların eklenmesi işlemi geçmişte öğrenilmiş olan bazı özelliklerin kaybolmasına sebep olabilmektedir. Bu nedenle yeni dataset direk olarak modele verilemez. Aksi taktirde model eski sınıfları tanıyamaz hale gelir. Bu nedenle yeni sınıfların eski modelin özellikleri ile eğitilmesi gerekir.

İlk olarak eski modelin son katmanının yerine yeni bir çıkış katmanı yerleştirilir. Bu işlemin amacı öncesinde sadece eski modelin sahip olduğu sınıf sayısı kadar çıktı bulunduran katmanın eski ve yeni olan sınıflar için sonuç üretebilecek şekilde düzenlenmesidir. Bu işlemin ardından yeni çıktı katmanı için bir kayıp fonksiyonu ve optimizer tanımlanır. Optimizasyon işleminde dikkat edilmesi gereken en önemli etken hangi katmanların eğitilecek olduğudur. Eğer eski eğitim dataset’i yeni eğitim dataset’ine göre çok daha büyük ise eğitilmesi gereken katmanlar sadece son katmanlardır. Çünkü ilk katmanlar daha genel bilgiler bulundurduğu için yeni dataset içinde iyi bir başarı elde edebilir. Eğer yeni dataset büyük ise daha fazla katmanın eğitilmesi sistemin başarı değerinin artmasını sağlayacaktır.

Bu modelde orijinal bir dataset’in üzerine daha küçük bir dataset’in eğitilmesi aşamasında sadece sınıflandırma işlemi yapan FC modelindeki gizli katmanlar yeniden eğitilmektedir. Eğer yeni dataset orijinal dataset’ boyutlarında ise modelin tamamı eğitilmektedir.

Modelin sadece sınıflandırma aşamasının dondurulması durumunda modelin büyük bir kısmı her epoch değerinde gereksiz yere hesaplanır. Bu nedenle yeniden eğitilmesi planlanan katmanlar için bir besleme noktası oluşturulur. Bu besleme noktası dondurulmuş olan katmanların verilen input değerne göre oluşturacağı sonuç ile beslenir. Bu sayede dondurulmuş katmanlar defalarca gereksiz bir şekilde hesaplanmasından kaçınılmış olur.

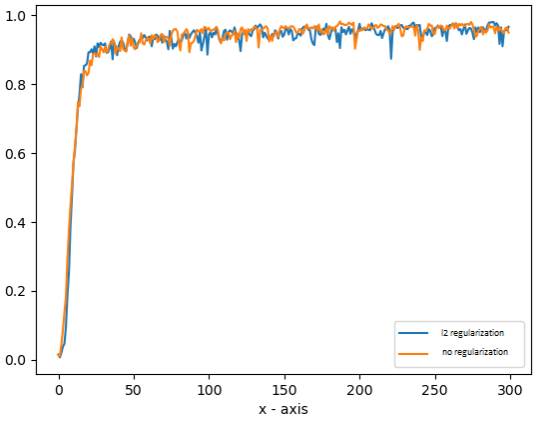
# 5. Bulgular:

Bir FCN modelinde girdi görüntünün segmente edilmiş halinin detaylı olarak istenmesi durumunda upsample ve skip işlemleri birçok farklı seviyede ki katmanın birleştirilmesi gerekmektedir.

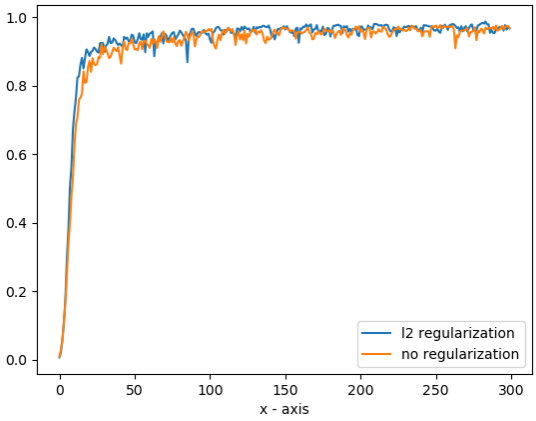
FCN modelinde overfitting olayı standart NN veya CNN modellerine göre daha nadir görülür. CNN için kayıp değerinin aşırı derecede küçük olması her zaman iyi bir sonuç üretmeyebilir ancak FCN modelinin başarılı olabilmesi için kayıp değerinin olabildiğince düşük olması gerekir.

FCN modelinin eğitimi esnasında kullanılan yığın boyutu (batch size) düşük tutulmalıdır. Bunun nedeni sistemin büyük yığın boyutunun kullanıldığı esnada küçük ayrıntıları görememesidir. Bu nedenle kayıp değeri istenilen değerlere ulaşmakta güçlük çeker ve istenilen başarıya ulaşmak için çok fazla denemenin gerekli olmasına sebep olur.

CNN modellerinde overfitting olayını önleyici bir işlem gerçekleştirilmediği sürece derin modellerin kullanılması test başarısının düşmesine sebep olmaktadır. Sadece regularization veya dropout gibi yöntemlerle overfitting engellendiyse test başarısı derin modellerde sığ modellere göre daha yüksek olur.




Sığ model




Derin model

Yukarıdaki resmimde derin modelde sığ modele göre l2 regularization'un daha başarılı olduğu görülmektedir. Derin model l2 regularization ile 80 kişi üzerinden yapılmış testte 98.8 test başarısı elde etmiştir. Ancak  regularization işleminin olmadığı durumda maksimum 97.8 değerine ulaşabilmiştir. Bu durum derin modellerde regularization'un etkisini göstermektedir.

Yapılan testlerde momentum optimizer'in ( SGD+momentum ) adam optimizer'e göre çok daha yavaş çalıştığı tespit edilmiştir.

Learning rate'in her bir iteration'un gerçekleştirilmesinin ardından küçük adımlarla küçültülmesi sistemin başarısını arttırdığı gözlemlenmiştir.

Küçük yığın boyutuna ile yapılan eğitimin büyük yığın boyutu ile yapılan eğitimden daha hızlı zirveye ulaştığı ancak maksimum noktasına ulaşmaya çalışırken başarı değerlerinde daha fazla dalgalanmaya sebep olduğu gözlemlenmiştir. Yığın boyutunun küçük olması ekran kartının kullanım performansını düşürdüğü bundan kaynaklı daha yavaş çalıştığı gözlemlenmiştir.

Sistemin tanınmayan kişilere ait görüntülere verdiği cevabın doğru olma ihtimali tanınmadığı için yoktur. Bu nedenle sistemin bu kişilere verdiği cevabın doğru olarak kabul edilmemesi için bir olasılık eşik değeri uygulanır. Ancak bu olasılık değeri he zaman mükemmel bir sonuç üretmeyebilir. Yanlış bir kişi için yüzde yüze yakın bir sonuç üretmesi durumunda bunun fark edilmesi imkansız hale gelir. Bu durum daha önce miscalibration olarak ifade edilmiştir. Miscalibration olayına sebep olan en önemli etkenler modelin fazla büyük olması, overfitting ve softmax fonksiyonudur. Softmax fonksiyonu genellikle multiclass sistemlerde kategorik cross entropy loss fonksiyonu ile birlikte kullanılır. Bu fonksiyon genellikle ikiden fazla sınıfın bulunduğu durumlarda, ikili cross entropy fonksiyonu sadece iki adet sınıf olduğunda tercih edilir. Softmax fonksiyonu ile kategorik cross entropy loss fonksiyonunun kullanıldığı durumlarda sistem daha yüksek başarı elde ettiği ancak verdiği ve daha kendinden emin cevaplar ürettiği ancak miscalibration olayının ortaya çıkmasına sebep olduğu gözlemlenmiştir.

# 6. Tartışma Ve Sonuç:

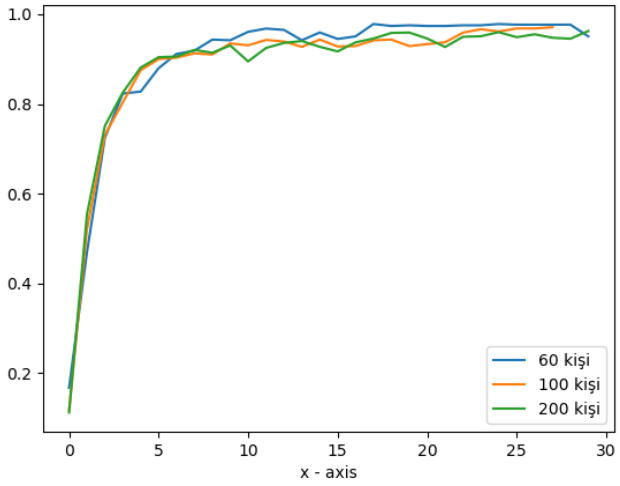
Uzun süredir kullanılmakta olan İris Tanıma Sistemi, Biyometrik sistemler arasında son dönemde belirgin hale gelmiştir. Diğer sistemlere göre artı yanları olduğu gibi, körlerde ve gözü titreyen bireylerde uygulanmaması negatif yönü olarak görünmektedir.

En çok tercih edilen sistem olan parmak izi tarama sisteminden daha ayırt edici olması, kişilerde yaşam boyunca değişmemesi ve genetik faktörlerden etkilenmemesi önümüzdeki zamanlarda iris tanıma sisteminin daha geniş bir kullanım alanı olacağını göstermektedir.

Bu çalışmada; biyometrik sistemler arasında olan İris Tanıma Sisteminin çok kullanılan tanıma sistemlerine göre avantajlı olduğu
görülmüştür. Ama henüz yeteri kadar yaygın kullanılmadığı fark edilmiştir.

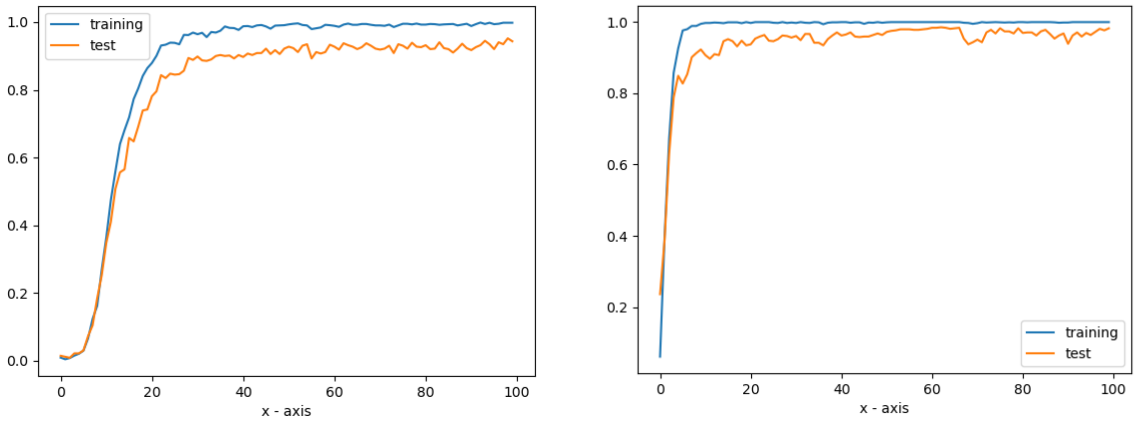
İris tanıma projesi için yapay zeka ile tanıma sisteminin gerçekleştirilmesinin en büyük problemlerinden biri yapay sinir ağlarının öğrenme hızıdır. Öğrenmez hızı genellikle düşük olmasından kaynaklı olarak büyük bir datanın eğitilmesi oldukça uzun bir vakit alabilmektedir. Ancak günümüzde yapay sinir ağlarının öğrenme hızı ekran kartlarının performanslarının artması ile birlikte oldukça yüksek duruma gelmeye başlamıştır. Bu nedenle gelecekte tanıma sistemleri içerisinde iris tanıma sisteminin yerinin vazgeçilmez olacağı aşikardır.

Sistemin genel modelinin oluşturulmasının ardından dataset için en iyi sonuçları üreten model için farklı parametreler ile birlikte birçok deneme yapılmıştır.



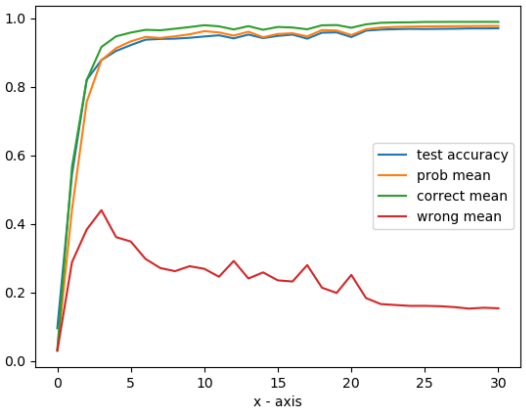

6.1 Başarı grafiği

Farklı sayıda kişi üzerinden sistemin 30 epoch çalıştırılmasının sonucu yukarıdaki grafikte gösterilmektedir. Kişi sayısında bir artış olması sistemin aynı başarı değerine daha geç ulaşmasına sebep olmaktadır. Modelin farklı boyutlardaki başarı değerinin değişiklik gösterdiği görülmektedir. Ancak büyük modeller yeterli miktarda eğitildiklerinde düşük boyutlu modeller kadar başarı elde edebildiği gözlemlenmiştir. Uygun yığın boyutu, öğrenme katsayısı, dropout katmanı katsayısı bulunması ile birlikte sistem performansının yükseldiği ve overfitting veya underfitting durumlarına veya test başarısındaki dalgalanmaya engel olunmuştur.




6.2 soldaki görüntü Direkt dataset , sağdaki görüntü Düzenlenmiş dataset

FCN semantic segmentation ve ardından Daugman’s rubber işlemlerinin uygulanmasının ardından sistemin başarı seviyesi ve sistemin öğrenme hızındaki artış gözlemlenmiştir. Yukarıdaki grafikte sabit boyutlarda görüntülerden oluşan dataset’in üzerinde hiçbir değişiklik yapılmadan CNN modeli ile ürettiği başarı değeri ve iris noktasının tespit edilip görüntü içerisindeki gereksiz bölgelerin ortadan kaldırılarak CNN modelinin sadece irise ait özelliklere odaklanmasının sağlanması ile elde edilmiş CNN modeli başarı değeridir.




6.3 İhtimal değerleri

Sistemin yüzde 97 başarı değeri elde edilene kadar eğitildikten sonra oluşan sonuç ve ortalama ihtimal değerleri yukarıdaki grafikte belirtilmiştir. Sistemin doğru cevap verme oranı mavi, ürettiği ortalama ihtimal değeri sarı, doğru cevaplar için ürettiği ihtimal değeri yeşil ve yanlış cevaplar için ürettiği ihtimal değeri kırmızı ile gösterilmiştir.

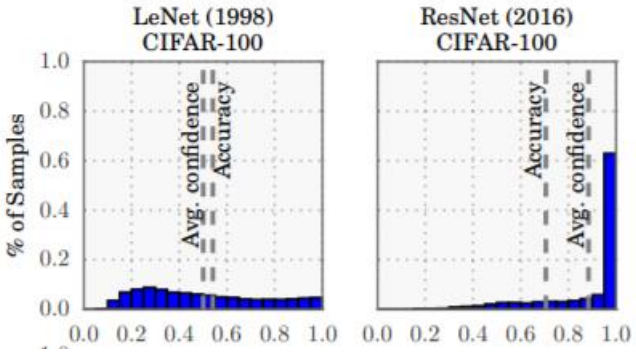
# 7. Benzer Çalışmalar:

Bu zamana kadar birçok farklı yöntemle iris tanıma işlemi gerçekleştirilmiştir. Bu çalışmalar 1980 li yıllara dayanır.İlk deneme olan daugman yöntemi ile gerçeklenen iris tanıma sistemi gelecek iris tanıma sistemleri için örnek olmuştur. Yakın zamanda iris tanıma işlemi için daha sıklıkla sinir ağları kullanılmaya başlanmıştır. İris tanıma işleminde kayda değer bir başarıyı sağlayan ilk model alexnet’tir. Alexnet’in ardından hızlı bir gelişim gösteren CNN modeli zamanla güçlenen teknoloji ile daha derin olarak tasarlanmaya başlayan modeller daha da yüksek sonuçlar elde etmişlerdir. Bu güne kadar iris tanıma işlemi için kullanılmış modellerin en önemlileri Daugman, AlexNet, VGG, Inception(GoogleNet), ResNet, DenseNet’tir. [23]

| Model | Başarı değeri |
|---|---|
| DenseNet | 98.8% |
| ResNet | 98.2% |
| VGG | 93.1% |
| Daugman | 91.0% |

7. Farklı modellerin casia-iris-thousand dataset'i üzerindeki başarı değerleri [23]

Bu modellerin başarı değerleri yukardaki grafikte belirtilmiştir. Yukarıda gösterilmekte olan modellerin yüksek başarı sağladıkları görülmektedir. Ancak DenseNet, ResNet gibi modeller aşırı derin modeller oldukları için calibrasyon konusunda iyi olmayan modellerdir. Örneğin LeNet yüksek bir başarı değerine sahip değildir. Ancak ürettiği ihtimal değerleri daha stabil ve daha gerçekçidir.




LeNet ve ResNet başarı ve eminlik grafikleri [26]

Yukarıdaki grafikte daha küçük bir model olan LeNet'in başarı değerinin daha düşük ancak sonuçtan eminliğinin daha güvenilir olduğu görülmektedir. Modelin büyük olmasından kaynaklanan güvenilmezlik üzerinde çalımakta olduğumuz modelde de gözlemlenmektedir. Sistem eğitilmiş olduğu dataset üzerinde tanımlayamadığı kişilerde yüksek ihtimal üretmezken, tanımadığı bir dataset üzerinde çalıştığı esnada kendinden emin bir şekilde yanlış cevap verebilmektedir.

Bu durum oran olarak azaltılabilse dahi tamamen hata yapmamasının sağlanması neredeyse imkansızdır.

# KAYNAKÇA:


[1] Deshpande , A . A Beginner's Guide To Understanding Convolutional Neural Networks Part 2 . adeshpande3.github . https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

[2]Lee, F . Convolutional neural Networks . cs231n . http://cs231n.github.io/convolutional-networks/

[3]Okumuş ,R . Convolutional Neural Networks (Evrişimsel Sinir Ağları) . Medium .https://medium.com/@rabiaokumus96/convolutional-neural-networks-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-cceb887a2979

[4]Kim, S. A Beginner's Guide to Convolutional Neural Networks (CNNs).towardsdatascience. https://towardsdatascience.com/a-beginners-guide-to-convolutional-neural-networks-cnns-14649dbddce8

[5]Deshpande, M . Convolutional Neural Network-II . towardsdatascience. https://towardsdatascience.com/convolutional-neural-network-ii-a11303f807dc

[6]Bozarık , E. Sinir ağları ve derin öğrenme – gradyan inişi . medium . https://medium.com/deep-learning-turkiye/sinir-a%C4%9Flar%C4%B1-ve-derin-%C3%B6%C4%9Frenme-v-grandyan-d%C3%BC%C5%9F%C3%BC%C5%9F%C3%BC-8c6d15a3d965

[7]dellinger,J. Weight initialization in Neural Networks: A journey from basics to kaiming.Towardsdatascience. https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79

[8]Jefkine . Backpropagation in convolutional neural networks. DeepGrid. https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

[9]Amit Shekar . What are L1 and L2 Loss functions? .AfterAcademy . https://afteracademy.com/blog/what-are-l1-and-l2-loss-functions

[10]Agarval , M. Backpropagation in convolutional neural networks- intution and code . medium. https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199

[11] Allibhai E . Building a convolutional neural Networks in Keras . Towardsdatascience. https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5

[12]https://ab.org.tr/ab13/bildiri/143.pdf

[13]https://acikerisim.deu.edu.tr/xmlui/bitstream/handle/20.500.12397/7922/305570.pdf?sequence=1&isAllowed=y

[14]http://www.artelektronik.com/yuz-tanima-sistemleri-biyometrik-tanima-sistemleri-nedir.html

[15]https://teknolojiprojeleri.com/programlar/python-nedir-ne-ise-yarar-nerelerde-kullanilir

[16]https://medium.com/yapayzekanedir/python-programlama-dili-ile-yapay-zeka-c6e345b49b5f

[17] Li Z, Hoiem D . Learning without Forgetting . https://arxiv.org/abs/1606.09282

[18] Tsang S . Review: FCN — Fully Convolutional Network (Semantic Segmentation) . Towardsdatascience. https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1

[19] Cui Z, Chen W, Chen Y . Multi-Scale Convolutional Neural Network for Time Series Classification (MCNN) . https://www.cse.wustl.edu/~z.cui/projects/mcnn/

[20] Fortuner , B. Loss Functions . github . https://github.com/bfortuner/ml-glossary/blob/master/docs/loss_functions.rst

[21] Tan T, Sun Z . Center for biometric and security research . CSBR . http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp

[22] Douillard A. How to be confident in your neural network.Towardsdatascience.

https://towardsdatascience.com/how-to-be-confident-in-your-neural-network-confidence-10d10dcf8003

[23] Nguyen K, Fookes C, Ross A. Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective. IEEE Xplore .

https://ieeexplore.ieee.org/document/8219390?denied=

[24] Brownlee J. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. Machinelearningmastery . https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[25] Neuman L, Zisserman A, Vedaldi A. Relaxed Softmax: Efficient Confidence Auto-Calibration for Safe Pedestrian Detection. https://www.robots.ox.ac.uk/~vgg/publications/2018/Neumann18c/neumann18c.pdf

[26] Guo C, Pleiss G, Sun Y. On Calibration of Modern Neural Networks.Cornell University. https://arxiv.org/abs/1706.04599

# ÖZGEÇMİŞ

## İBRAHİM EYYÜP İNAN

**1997 yılında Trabzonda doğdu. Cudibey ilköğretim okulunu bitirdi. 2011-2014 yılları arasında akaçaabat anadolu lisesinde okudu.**
**2014 yılında kanuni anadolu lisesine geçti ve 2015 yılında mezun oldu. 2015 yılından itibaren İstanbul Üniversitesi-Cerrahpaşa Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne devam etmektedir.**