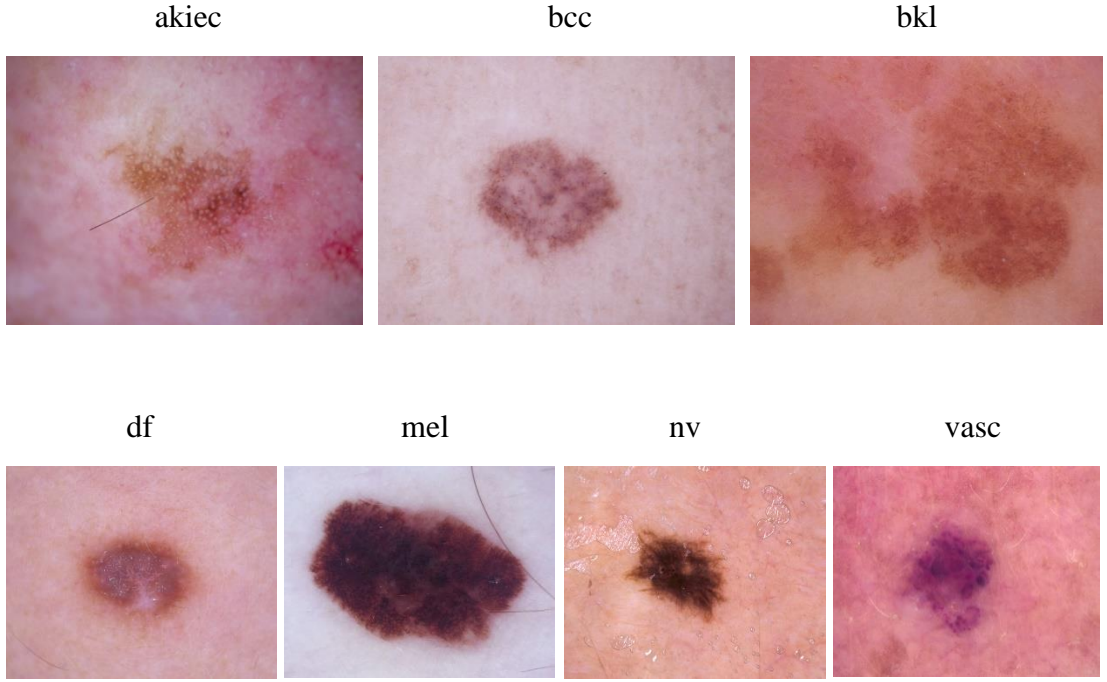


1. GİRİŞ VE TANITIM

Dünya çapında en yaygın kanser türü cilt kanseridir. Cilt kanser türleri arasında en ölümcül şekli olan melanom türüdür. Dermoskopi ile görsel incelemeye kıyasla cilt kanseri lezyonlarının teşhisinde iyileşme gösteren bir cilt görüntüleme yöntemi kullanılmaktadır. Günümüzde cilt kanseri, UV ışınlarına maruz kalma nedeniyle giderek artmaktadır. Cilt kanseri, ölümlerin başlıca nedenlerinden biri haline geldiğinden, cilt kanser türünün tespitini bir saat içinde bulmak için bir ihtiyaç vardır. Bu ihtiyacı karşılamak için cilt lezyonu görüntülerini analiz eden ve sınıflandıran birkaç derin sinir ağı modeli kullanılarak çalışma yapılmıştır. Çalışmada yedi sınıflı veri seti kullanılmıştır. Bu veri seti üzerinde modelinin eğitimi sürecinde kullanılan hold-out (dışarıda tutma) yöntemi ile k-fold cross validation yöntemi kullanılarak sonuçlar elde edilmiştir.

Sınıflandırma için transfer öğrenme kavramını kullanılmıştır. Transfer öğrenme ile öğrenme sürecini sıfırdan başlatmak yerine model, farklı bir problem çözülürken öğrenilen kalıplardan yola çıkar. Bu şekilde model, önceki öğrenmeden yararlanır ve sıfırdan başlamaktan kaçınır. Görüntü sınıflandırmasında, transfer öğrenmesi genellikle önceden eğitilmiş modellerin kullanımıyla ifade edilir. Önceden eğitilmiş bir model, bizim çözmek istediğimize benzer bir sorunu çözmek için büyük bir kıyaslama veri kümesi üzerinde eğitilmiş bir modeldir [1]. Bu çalışmada önceden eğitilmiş ağırlıklar olarak ResNet50, DenseNet201, Inceptionv3 ve VGG16 olmak üzere dört önceden eğitilmiş model kullanılmıştır.



Şekil 1.1. Veri setindeki sınıflara ait örnek görüntüler

1.1. Verisetinde bulunan cilt lezyon türlerini açıklama

- 1 - **Akiec** : Ameliyat olmaya gerek kalmadan lokal tedavi yapılabilir.
- 2 - **Bcc** : Bazal hücreli karsinom nadiren metastaz yapan ancak tedavi edilmediğinde yıkıcı olarak büyüyen epitel cilt kanserinin yaygın bir çeşididir.
- 3 - **Bkl** :İyi huylu keratoz olarak bilinir. Görsel olarak ayırt edilmesi güçtür.
- 4 - **Df** : Dermafibroma olarak bilinen iyi huylu bir cilt lezyonudur.
- 5 - **Mel** :Melenom ile kötü huylu bir neoplazmadır.
- 6 - **Nv** :İyi huylu neoplazmalardır ve melanomun aksine renk ve yapı dağılımı bakımından genel olarak simetriktir.
- 7 - **Vasc** :Vasküler cilt lezyonlarıdır. Kırmızı ve mor renkli lekeler olarak bilinirler. İyi huylu yapılar olarak not edilmiştir [2].

2. LİTERATÜR

Chaturvedi, Grupta ve Prasad (2019), MobileNet modeli üzerinde HAM10000 veri setinin 10015 dermoskopi görüntüsü üzerinde ince ayar yaparak %83,1'lik bir genel doğruluk elde etmişlerdir [3]. Çevik ve Zengin (2019) VGGNET-16 mimarisi ile tasarlanan derin ağ modeli eğitilerek sonuçlar gözlemlemiştir. Veri setini eğitim ve test alanı olarak ayrıştırılmasında K-Fold Cross Validation tekniğinden faydalanılmıştır. Eğitilmiş modelin test aşamasında sınıfların onaylama doğruluğu %85,62 olarak elde etmişlerdir [4]. Ali (2019) Mobilenet CNN, cilt hastalıkları tiplerini girdilerine göre sınıflandırarak kanser hastalıkları alanındaki Derin öğrenmenin uygulanmasının, cilt kanseri görüntülerini sınıflandırmanın ve tanımanın en uygun yolu olabileceğini göstermiş ve mevcut çalışmada %90 doğruluk almıştır [5]. Fırıldak, Çelik ve Talu (2020) çalışmasında başarı oranını %93 alan AlexNet mimarisi üzerinde gerçek lezyon imgelerinden imge üretilmesiyle veri kümesini genişleterek klasik üretme ve Çekişmeli Üretici Ağ (ÇÜA) tekniği kullanarak en yüksek lezyon sınıflandırma doğruluğunu elde etmiştir [2]. Rahman ve Ami (2021) çalışmasında son teknoloji derin öğrenme önceden eğitilmiş modelleri ResNet, Xception ve DenseNet üzerinde sınıflandırarak üç model için sırasıyla %78.15, %81.8 ve %81.9 doğruluk oranlarını almıştır. Daha sonra ortalama topluluğu kullanarak üç modeli birleştirmiş ve teknik herhangi bir ek eğitim almadan %85.3 doğruluk değerini almıştır [6].

3. YÖNTEM

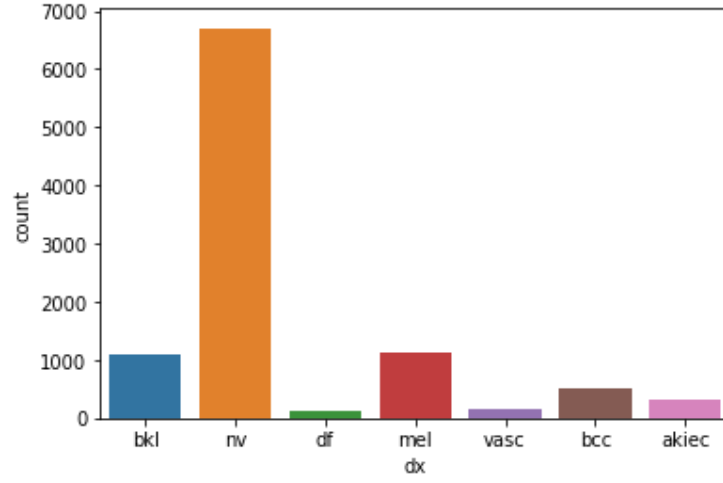
3.1. Veri Seti

Cilt kanseri sınıflandırması için Kaggle web sitesinden halka açık HAM10000 bir veri seti kullanılmıştır. HAM10000 veri seti içerisinde toplamda 10015 adet 650x400x3 boyutunda RGB yedi sınıf cilt lezyon türünü içeren görüntüler bulunmaktadır.

Veri setleri genellikle .csv formatında bulunur. CSV uzantılı dosyalar, veritabanı kullanıcıları için verileri virgüller ile ayırarak belli bir düzende yazıp kaydedilmiş olan dosyalardır. Dosyanın her satırı bir veri kaydını içermektedir. Bir CSV dosyasını dataframe olarak okumak için pandas kütüphanesinin read_csv metodu kullanılmaktadır [7].

Çizelge 3.1. Veri seti hakkında bilgi

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10015 entries, 0 to 10014
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   lesion_id       10015 non-null  object
1   image_id        10015 non-null  object
2   dx              10015 non-null  object
3   dx_type         10015 non-null  object
4   age             9958 non-null   float64
5   sex             10015 non-null  object
6   localization    10015 non-null  object
dtypes: float64(1), object(6)
memory usage: 547.8+ KB
```



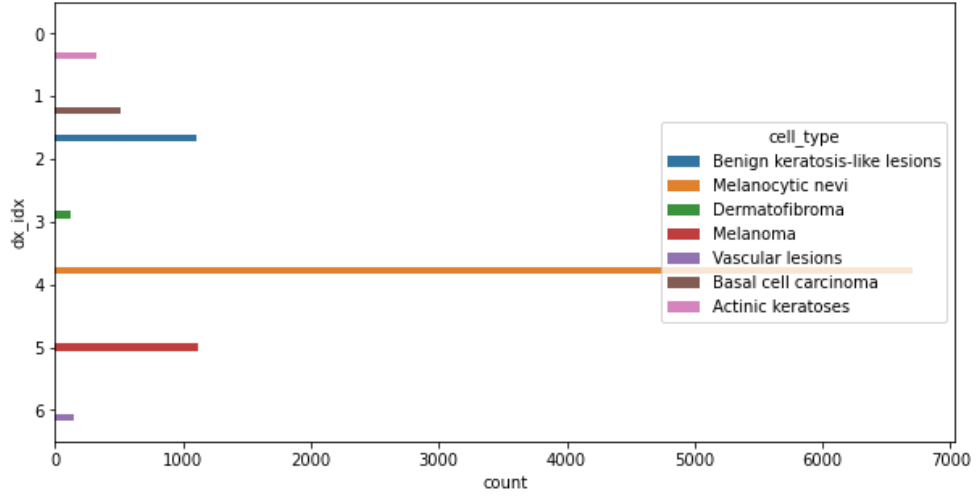
Şekil 3.1. HAM10000 veri setindeki sınıfların dağılımı

Çizelge 3.2. Veri setinde bulunan cilt kanser türlerinin açılımı

dx: kanser hücresi türü	
nv	Melanocytic nevi
mel	Melanoma
bkl	Benign keratosis-like lesions
bcc	Basal cell carcinoma
akiec	Actinic keratoses
vasc	Vascular lesions
df	Dermatofibroma

Çizelge 3.3. Veri setindeki sınıflara ait görüntülerin sayısı

Kanser Türü	Görüntü Sayısı
Melanocytic nevi	6705
Melanoma	1113
Benign keratosis-like lesions	1099
Basal cell carcinoma	514
Actinic keratoses	327
Vascular lesions	142
Dermatofibroma	115
Toplam	10015



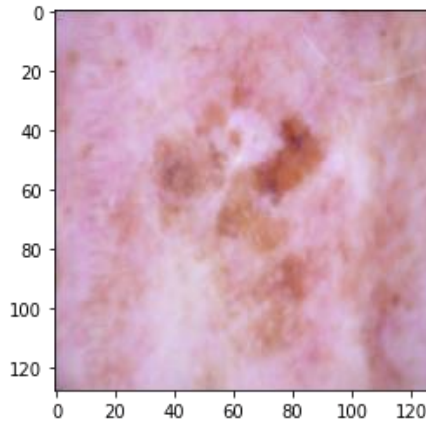
Şekil 3.2. HAM10000 veri setindeki her bir sınıfların etiketlenmesi

Şekil 3.2. 'de görüldüğü gibi Melanocytic nevi diğer sınıflara göre çok daha sık olduğu için dengesiz bir veri setine sahiptir.

3.2. Görüntü Ön İşleme (Image Pre-Processing)

Farklı derin sinir ağı öğrenme modelleri için görüntü girişi farklı olduğundan HAM10000 görüntülerini yeniden boyutlandırılıp $128 \times 128 \times 3$ piksele ayarlanarak pickle dosyasına kaydedilmiştir.

pickle modülü saklanabilen ve başka programlar tarafından ya da aynı programın farklı çalışma zamanlarında kullanılabilecek Python nesneleri yapmanın standart yoludur [8].



Şekil 3.3. Boyutlandırılmış örnek görüntü

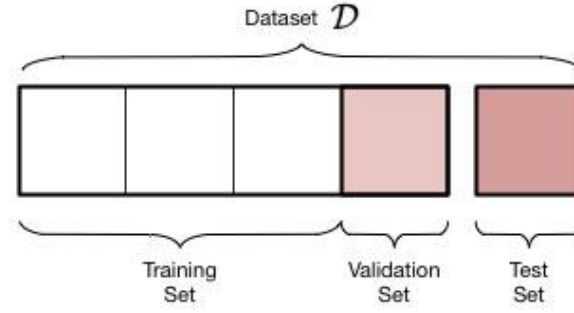
3.3. Veri Setinin Eğitim ve Test Verisi Olarak Ayrılması

Modelinin eğitimi sürecinde kullanılan model performans değerlendirme, hangi modelin daha iyi olduğunu bulmak ve gelecekte öğrenme modelinin ne kadar iyi çalışacağını anlamak için yardımcı olur. Sadece eğitim (training) datayı kullanmak kabul edilebilir durum değildir çünkü metot overfitting (aşırı uyum) durumunda olabilir. Bu durumdan kurtulmak için iki farklı yaklaşım önerilir: hold-out ve cross-validation. Overfitting'ten kaçınmak ve model performansını ölçmek için her iki yaklaşım bir test kümesi (metot eğitimde hiç kullanılmamış) kullanır.

3.3.1. Hold-out (Dışarıda tutma)

Elimizde büyük veri seti varsa biri eğitim, biri doğrulama ve biri de test olmak üzerine üç farklı veri seti elde edebiliriz.

- **Eğitim Verisi (Eğitim Kümesi – Örneklem Kümesi) (Training Set):** girdi özelliklerini karşı ve cevap (çıktı) özellik/lerini içerir.
- **Doğrulama veri seti (Validation dataset):** eğitim aşamasında elde edilen modelin performansını değerlendirmek için kullanılan alt bir veri setidir. Ayrıca, bu veri seti hangi modelin iyi olduğunu belirlemek ve modeller için en uygun parametreleri ayarlamak için bir test platformu sağlar. Tüm modeller doğrulama veri setine gerek duymaz.
- **Test veri seti (Testing Dataset):** (veri setinin alt kümelerinde olmayan hiç görülmemiş durumlar) modelin gelecekteki performansını değerlendirmek için kullanılır. Eğer test verisinde sonuçlar eğitim aşamasına göre kötü çıkıyorsa, overfitting ile karşı karşıya olduğumuz durumu çıkar [9].

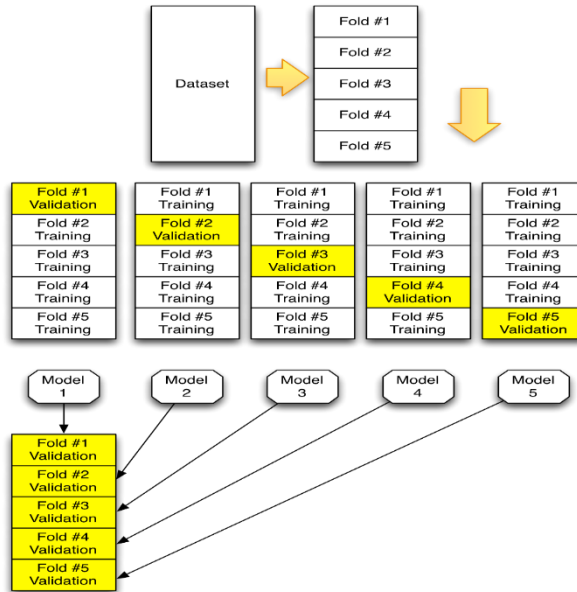


Şekil 3.4. Hold out yöntemi

3.3.2. Cross Validation (Çapraz Doğrulama)

Eğitim verisi alt kümelerle ayrılır. Tek alt kümeyi eğitim için kullanıp diğer kalan kümeleri doğrulama işlemi için kullanılır. Bu işlem çapraz bir şekilde tüm alt kümeler için tekrarlanır. Bu işleme çapraz doğrulama denir. Bu işlem daha önceden belirlenen belli bir k sayısında yapılır. Veri eşit boydaki k parçaya ayrılır ve k kez değerlendirilir. Model değerlendirmeyi iki alt bölüme ayırabiliriz.

- Sınıflandırma Değerlendirme
- Regresyon Değerlendirme



Şekil 3.5. K-Fold Cross Validation Yöntemi

Regresyon ve sınıflandırma, çapraz doğrulama açısından biraz farklı şekilde ele alınır. Regresyon, her bir ögenin nereye düştüğüne çok az dikkat ederek, veri kümesini K katlarına bölebileceğiniz daha basit bir durumdur. Regresyon için, veri ögelerinin mümkün olduğunca rastgele olarak katlara düşmesi en iyisidir. Ayrıca, her katın mutlaka tam olarak aynı sayıda veri ögesine sahip olmayacağını hatırlamak da önemlidir. Veri kümesinin eşit olarak K katlara bölünmesi her zaman mümkün değildir. Regresyon çapraz doğrulaması için Scikit-Learn sınıf KFold'u kullanır.

Sınıflandırma için çapraz doğrulama da KFold nesnesini kullanabilir; ancak bu teknik, sınıf dengesinin her kattaki orijinalinde olduğu gibi aynı kalmasını sağlamayacaktır. Bir modelin eğitildiği sınıfların dengesinin eğitim seti ile aynı (veya benzer) kalması çok önemlidir. Eğitilmiş bir model fiili kullanıma yerleştirildikten sonra bu dağılımdaki bir sapma, izlenmesi gereken en önemli şeylerden biridir. Bu nedenle, çapraz doğrulamanın kendisinin istenmeyen bir kaymaya yol açmadığından emin olmak istiyoruz. Bu, StratifiedKFold (tabakalı örnekleme) olarak adlandırılır ve sınıflandırmayı kullandığınızda KFold yerine Scikit-Learn nesnesi StratifiedKFold kullanılarak gerçekleştirilir. Özetle;

KFold bir regresyon problemiyle uğraşırken, StratifiedKFold bir sınıflandırma problemiyle uğraşır [10].

3.4. Veri Arttırma (Data Augmentation)

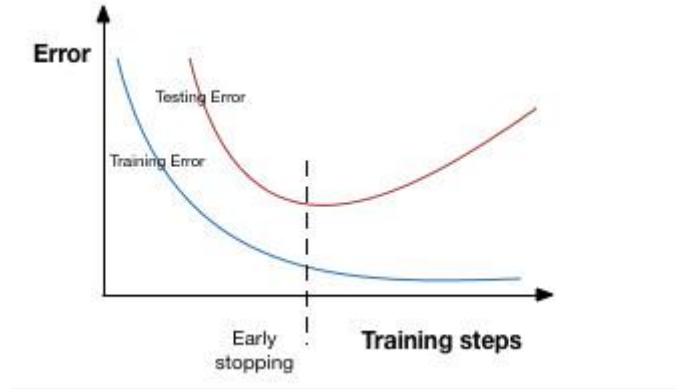
Elimizde ne kadar fazla sayıda veri olursa model o kadar iyi öğrenecektir; bu bilinen bir gerçektir. İyi öğrenme beraberinde yüksek başarıyı da getirmektedir. Verinin fazla olması model başarıyı artımının yanı sıra ezberlemeyi (overfitting) de önlemektedir.

Veri arttırma uygularken amacımız modelin genellenebilirliğini arttırmaktır. Ağımızın sürekli olarak giriş verilerinin yeni, biraz değiştirilmiş sürümlerini gördüğü göz önüne alındığında, ağ daha sağlam özellikler öğrenebilir.

Test sırasında veri artırma uygulamıyoruz ve yalnızca değiştirilmemiş test verileri üzerinde eğitilmiş ağıımızı değerlendiriyoruz- çoğu durumda, belki eğitim doğruluğunda hafif bir düşüş pahasına test doğruluğunda bir artış görülmektedir.

Bu veri setinde en yaygın kullanılan yerinde/anında veri arttırma kullanılmıştır. Bu tür veri arttırma, Keras'ın ImageDataGenerator sınıfı uygulamaktadır. Bu tür bir veri arttırmayı kullanarak, ağıımızın eğitildiğinde her çağda verilerimizin yeni varyasyonlarını görmesini sağlamak istiyoruz. ImageDataGenerator olduğu değil, orijinal veri ve dönüştürülmüş veri hem de geri dönen- sınıfı, sadece rastgele dönüştürülmüş veriler döndürür. Bu büyütme eğitim zamanında yapılır (yani, bu örnekleri önceden/eğitimden önce üretmiyoruz) [11].

3.5. Early Stopping



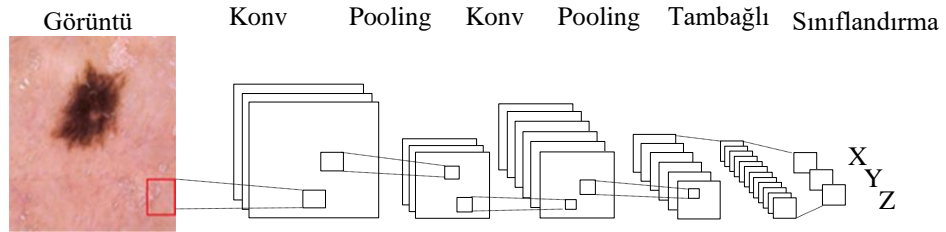
Şekil 3.6. Early Stopping

Öğrenme sırasında validation loss ile test loss arasındaki fark açılmaya başladığı durumda model ezberliyor (overfitting) ya da gürültüyü öğreniyor (noisy) demektir. Bu durumda validation error artmaya başladığında eğitim durdurulur ve bir önceki adıma geri dönülür. Bir önceki adıma geri dönebilmek için eğitim esnasında her bir öğrenme adımında (epoch) bir önceki adımın verileri saklanmış olmalıdır. Yukarıdaki şekilde görüleceği üzere öğrenmenin başlarında training error ve test error her ikisi birlikte azalmaktadır. Bununla birlikte örnekte kesikli çizgiyle belirtilen adımdan sonra test error arttığını görmekteyiz bu noktada ezberleme olmuştur ve bu adımın

öncesinde öğrenme durdurulmalıdır. Overfitting işleminden sonra ağırlıklar üzerindeki değişiklikler modeli olumsuz etkileyecektir. Erken durdurmanın amacı aşırı uydurmayı önlemektir [12].

3.6. Konvolüsyonel Sinir Ağları (Convulational Neural Network)

Çok katmanlı ileri beslemeli bir yapay sinir ağı olan konvolüsyonel sinir ağı (CNN) özellikle görüntü analizlerinin yapılması için kullanılmaktadır. Hayvan görü sistemine dayanan bir yaklaşımla ortaya atılmıştır (Hubel ve Wiesel, 1968). Filtrelemeye dayalı bir yapıdadır. Kullanılacak olan filtre ile görüntünün özelliğini belirtecek öznitelikleri belirgin hale getirir. Özellikle sınıflandırıcı işlemlerinde başarılı sonuçlar üretmektedir. Filtreler farklı boyut ve değerlerde kullanılarak baskınlık düzeyi az olan özniteliklerin ortaya çıkmasını sağlar (Fukushima, 1982; Simard, 2003). Şekil 3.7. 'de konvolüsyonel sinir ağına ait örnek bir mimari görülmektedir.



Şekil 3.7. Cilt görüntülerini sınıflandıran konvolüsyonel sinir ağı modeli

İlk olarak LeCun ve arkadaşları tarafından gradyan temelli bir yaklaşım sunularak ortaya çıkan ağ yapısına konvolüsyonel sinir ağı adı verilmiştir. Oluşturulan bu yapay sinir ağına ise LeNet adı verilmiştir (LeCun vd., 1998).

Çok katmanlı bu sinir ağı içerisinde birden fazla konvolüsyon katmanı, tam bağı katman, aktivasyon katmanı, sınıflandırıcı katman, havuzlama katmanı ve bunlara ek katmanlar yer almaktadır. Her katman kendi işlevini yürüterek sınıflandırıcı katmanda sonuç üretilmektedir. Derin öğrenme yapıları içerisinde en çok kullanılan sinir ağı

konvolüsyonel sinir ağıdır. Daha çok sınıflandırma ve tespit işlemleri için kullanılmaktadır. Sinir ağı içerisindeki katmanlarla sınıflandırılacak öğelere ait öznitelikler belirlenerek sınıflandırıcı katmanı ile öğeler sınıflandırılır.

Imagenet tarafından 2012 ve 2014 yılında yapılan gerçekleştirilen en büyük nesne tanıma yarışması olan Büyük Ölçekli Görsel Tanıma yarışmasında en başarılı sonuçlar Konvolüsyonel sinir ağı ile elde edilmiştir (Girshick vd., 2014).

3.6.1. Giriş (Input) Katmanı

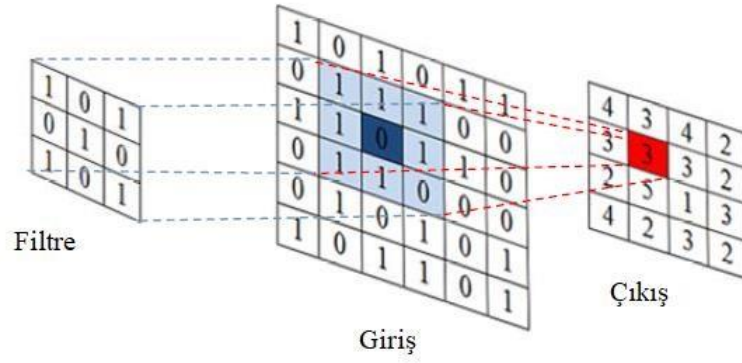
Veri giriş katmanı olarak bilinmektedir. Sinir ağlarında oluşturulacak olan veri seti ağın mimarisine göre belirlenmesi gerekmektedir.

Sinir ağına girecek olan her bir örneklem giriş katmanına ağın eğitimi için girdi verisi olarak kullanılır. Her bir öğeye ait veri arka arkaya gelerek bir veriseti oluşturmaktadır. Bu verisetinin boyutu ağın hızını, test süresini ve bellek ihtiyacını artırmaktadır (Inik ve Ulker, 2017; Tamura vd., 1997).

3.6.2. Konvolüsyon(Convolution) Katmanı

Konvolüsyonel sinir ağlarının temelini oluşturan katmandır. Bu katmanda girdi verisi üzerinde daha önceden belirlenmiş bir filtrenin girdi verisi üzerinde gezdirilerek girdinin belirgin özelliklerinin ortaya çıkmasını amaçlamaktadır. Filtreleme sonucunda giriş verisinden daha küçük bir matris elde edilmesini sağlar (Ciresan vd., 2011).

Derin öğrenme algoritmalarında farklı boyutlarda filtrelemeler kullanılarak konvolüsyon yapılmıştır. Filtreler 1x1, 2x2, 3x3, 5x5, 7x7 gibi boyutlarda kullanılarak konvolüsyon işlemi gerçekleştirilmekte (Lingqiao Liu et al., 2015) ve bu aşamada özellik haritası oluşturulmaktadır. Filtrelemenin eğitim aşamasının her adımında uygulanması ve yinelenmesi önemli bölgelerin tespiti için gereklidir. Şekil 3.8.'de konvolüsyon işlemi görülmektedir [13].



Şekil 3.8. Konvolüsyon işlemi (Liu hui-bin et al., 2016)

Konvolüsyonel sinir ağlarında yapılan hesaplamalar sonucunda, giriş ve çıkış boyutu arasında farklılıklar meydana gelmektedir. Konvolüsyon işleminden sonra oluşacak yeni görüntünün(matris) boyutu aşağıdaki denklem ile hesaplanmaktadır (Altan, 2019).

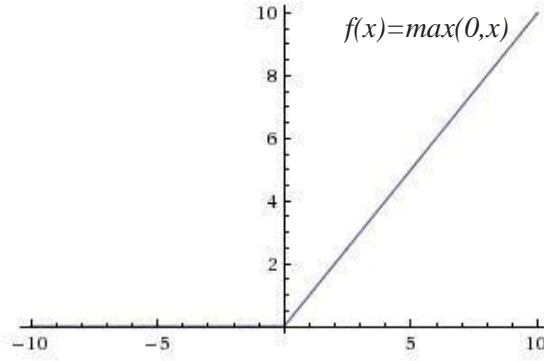
$$n_{out} = \left(\frac{n_{in} + 2p - f}{s} + 1 \right) \quad (1)$$

Bu denkleme göre çıkış matrisinin boyutu n_{out} , giriş resminin boyutu n_{in} , piksel ekleme padding(p), filtre boyutu f, adım kaydırma stride(s) bağlı olarak değişmektedir [13].

Konvolüsyon işlemi derin konvolüsyon sinir ağlarının önemli bir parçasıdır. Bu katmanda yapılan filtreleme işlemi ile öğeye ait özellikler ağa daha iyi şekilde yansımış olacaktır. Tercih edilecek olan filtre ağın eğitim sürecini ve başarısını doğrudan etkileyecektir (Pang vd., 2017).

3.6.3. Aktivasyon (Relu) Katmanı

Genellikle konvolüsyon katmanından sonra aktivasyon katmanı gelir. Aktivasyon fonksiyonu olarak sigmoid, hiperbolik tanjant, sinüs, step, eşik değer fonksiyonları kullanılmaktadır. Çok katmanlı yapay sinir ağları içerisinde doğrusal olmayan dönüşümler kullanılmaktadır. Pek çok aktivasyon fonksiyonu olmasına karşın derin öğrenme mimarilerinde Relu (Rectified Linear Unit) (Nair ve Hinton 2010) $f(x) = \max\{f_0\}(0, x)$ fonksiyonu kullanılmaktadır (Krizhevsky vd., 2012).



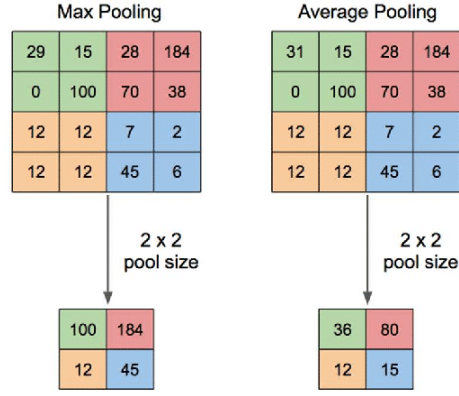
Şekil 3.9. Relu Aktivasyon fonksiyonu

Aktivasyon işlemi sonucunda elde edilen değer negatif ise 0 pozitif ise 1 değerini alır. Bu durum Şekil 3.9.'te gösterilmiştir (Jarrett vd., 2009).

3.6.4. Havuzlama (Pooling) Katmanı

Daha çok relu katmanı sonrasında yer alan havuzlama katmanı verilerin indirgenerek ağda sonraki katman için giriş boyutunu azaltır. Havuzlama işlemi ile veri üzerinde kayıplar oluşmaktadır. Ancak ağda dolaşacak olan veri miktarındaki azalma ağın daha hızlı olmasını sağlamaktadır. Ağdaki veri miktarının azalması ağdaki hesaplama miktarlarının ve kullanılacak olan bellek miktarının da azalması anlamına gelmektedir (Hinton vd., 2012).

Havuzlama katmanında uygun bir filtre kullanılarak işlem yapılır. Kullanılacak olan $N \times N$ boyutlu bir filtre veri havuzunda gezdirilerek matris içerisinde kalan uygun değerlerin bulunmasını sağlar. Değerin elde edilmesi için en çok kullanılan ortalama (average pooling) ve en büyük değer (max pooling) işlemleridir. En büyük değer havuzlamasında $N \times N$ boyutlu matris içerisinde kalan en büyük değer oluşturulacak olan yeni veri matrisinin 1×1 boyutundaki yeni değeri olur. Bu durum şekil 3.10. 'de gösterilmektedir (Zeiler ve Fergus, 2013). Ortalama havuzlamada ise $N \times N$ boyutundaki matris içerisinde kalan tüm değerlerin ortalaması alınarak elde edilecek yeni veri matrisinin 1×1 boyutundaki değerini içerir.

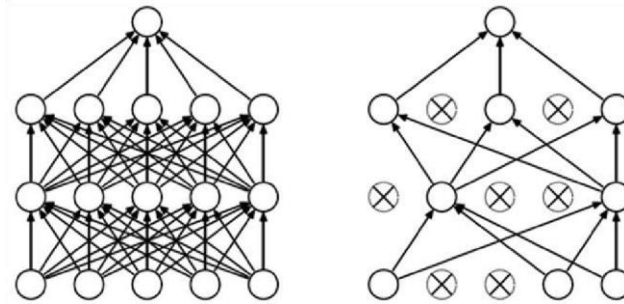


Şekil 3.10. Havuzlama işlemi [14].

Bu katmanda genellikle en büyük değerleri havuzlama yöntemi daha başarılı sonuçlar vermektedir.

3.6.5. Dropout Katmanı

Çok katmanlı yapay sinir ağlarında sinir ağı eğilirken aşırı öğrenme adı verilen ağın ezberlenmesi olayı gerçekleşir. Bu istenmeyen bir durumdur. Ağın ezberlenmesinin önüne geçmek için ağda ezber yapan bazı düğümlerin ortadan kaldırılmasıdır. Böylelikle ağın ezberlenmesi ortadan kaldırılmaya çalışılır (Srivastava vd., 2014). Dropout katmanı tam bağlı katmanlar için bir düzenleme katmanı olarak Hinton ve arkadaşları tarafından önerilmiştir. Dropout işlemi, sinir ağının düzenlenmesi yeteneğini artırdığı yapılan test performanslarıyla ortaya konmuştur (Hinton vd., 2012). Şekil 3.11. 'da çok katmanlı sinir ağı ve dropout yapılmış sinir ağı yapısı gösterilmektedir. Dropout işlemi yapılmış olan sağdaki şekilde bazı düğümler ortadan kaldırılmış olarak görülmektedir.



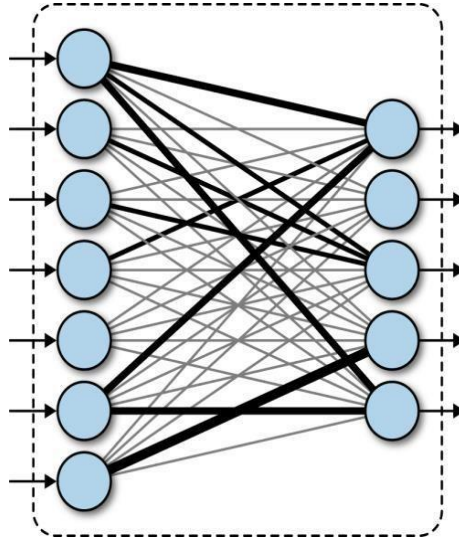
Şekil 3.11. Yapay sinir ağı a)

Dropout uygulanmış sinir ağı (Çarpı atılmış nöronlar ağdan çıkarılmıştır) b)

3.6.6. Tam Bağlı (Full-Connected) Katman

Bu katmandaki tüm nöronlar bir dizi şeklinde görünür. Katmandaki nöronların tamamı bu katmana bağlı önceki katmandaki aktivasyonların hepsine tam bağlı durumdadır. Tam bağlı katmanın özelliği önceki katmana bağlı olarak ortaya çıkar. Nesneyi belirleyecek olan özelliklerin hangi sınıfla ilişkili olduğu belirlenir. Bir örüntü ele alınacak olursa; örüntüdeki bir insanın tespit edilmesini sağlayan özellikler aktivasyon haritalarında yüksek değere sahip nöronda yer alacaktır. Bir tam bağlı katman bir sınıfla ilişki derecesi yüksek olan yüksek seviyeli özelliklere bakar. Bu özellikleri belirten ağırlıkların olduğu nöronlara bakılarak hangi sınıfa ait olduğu ortaya çıkmış olur (Lin vd., 2013; LeCun ve Bengio, 1995).

Örneğin en son katmanda elde edilen veri boyutu $3 \times 5 \times 64$ değerlerinde olduğu düşünülürse, bu katman sayesinde verilerin yeni boyutu $3 \times 5 \times 64 = 960$ olacaktır. Şekil 3.12. 'de tam bağlı katman yapısı görülmektedir. [13].

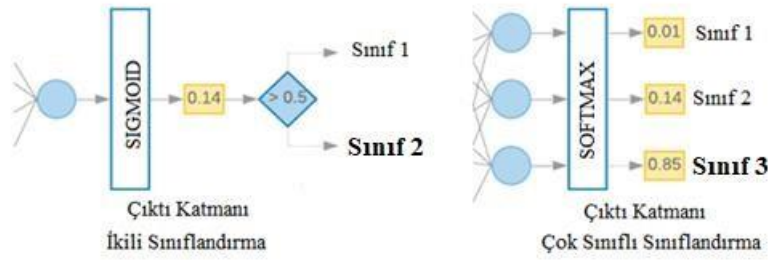


Şekil 3.12. Tam bağlı katman yapısı [13].

3.6.7. Sınıflandırma (Classification) Katmanı

Sınıflandırma katmanının görevi, tam bağlı katmandan elde edilen veriyi belirlenen sınıf sayısı kadar sınıflandırma yaparak çıkış değeri üretmektedir. Sınıflandırma

katmanında genellikle sigmoid ve softmax aktivasyon fonksiyonları kullanılmaktadır (Cireşan et al., 2011). Örneğin 10 farklı nesnenin sınıflandırılması yapılacaksa, sınıflandırma katmanı değeri 10 olmalıdır. Tam bağlı katman sonucunda çıkış değeri 960 seçilirse, sınıflandırma katmanı için 960×10 ağırlık matrisi elde edilmektedir. Sınıflandırma sonucunda 10 farklı nesne 0-1 aralığında bir değer olarak, değeri en yüksek olan nesne sınıflandırma tahmin değerini oluşturmaktadır.



Şekil 3.13. Sınıflandırma işlemi [13].

3.6.8. Yumuşatma (Softmax) Katmanı

Sınıflandırma katmanı olarak yer alan softmax katmanı kendisinden önce gelen tam bağlı katmandan girdi verisini alır ve sınıflandırmak için kullanır. Olasılıksal bir girdi verisinin belirli bir sınıfa ait olma durumunu ortaya koyar. Hangi sınıfa daha yakın olduğuna dair değer üretir. Derin öğrenme ağı içerisinde üretilen olasılıksal hesaplama gerçekleştirilerek her bir sınıf için olasılık değerini çıkarır. Bu işlemler için çapraz entropi kullanılır (Tang, 2013).

3.6.9. Normalizasyon (Normalization) Katmanı

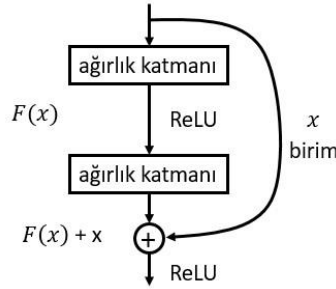
Derin konvolüsyonel sinir ağlarının eğitilmesi hesapsal olarak ciddi bir süreç almaktadır. Eğitim süresinin azaltılmasının bir yolu nöronların aktivasyonlarını normalize etmektir. Normalizasyon katmanı geri beslemeli ağlarda gizli katmanlardaki durumları stabil hale getirmede oldukça etkilidir. Genellikle Relu katmanı sonrasında normalleştirme gerçekleştirilir (Li vd., 2015).

Normalizasyon ağı performansını etkilemektedir. Bu sebeple katmanlardan gelen veriler belirli bir düzen içerisinde olmalıdır. Girdi verileri çok büyük ya da çok küçük içerebilir. Bu değerlerin normalize edilerek belirli aralıkta kullanılması eğitim ve süreç açısından önemlidir. Girdi verilerinin normalize edilerek belirli aralıkta temsil edilmesi gereklidir (Ioffe ve Szegedy, 2015) [15].

3.7. Derin Öğrenme Modelleri

3.7.1. ResNet

ResNet diğer mimarilere göre daha derin olarak tasarlanmıştır ve toplam 152 katmandan meydana gelmektedir. Bu mimarinin en önemli özelliği kalıntı bloklarından oluşmasıdır. Kalıntı bloкта girdi olarak alınan değer sırasıyla evrişim-aktivasyon-evrişim işlemlerinden geçirilerek, $F(x)$ fonksiyonu adında bir çıktı elde edilmektedir. Daha sonra bu fonksiyona orijinal girdi değeri eklenerek yeni çıktı değeri elde edilmektedir. Bu çıktı $H(x)$ fonksiyonu olarak adlandırılmakta ve $H(x) = F(x) + x$ olarak ifade edilmektedir (He et al., 2016). Ağı oluşturan bu yapı Şekil 3.14.'da verilmektedir.

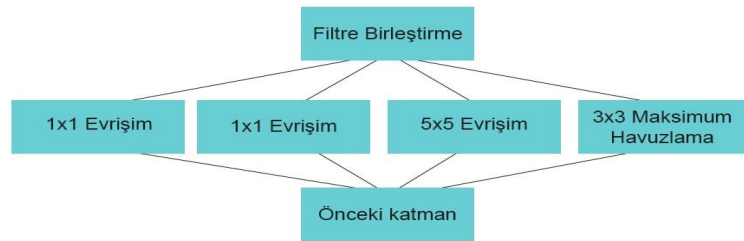


Şekil 3.14. Kalıntı öğrenme yapısı

Klasik evrişim işleminde $H(x)$, $F(x)$ fonksiyonuna eşit iken bu yapıda evrişim işleminden sonra girdiye orijinal veri de eklenmektedir. Şekil 3.15. 'de klasik model ve ResNet model yapısı görülmektedir.

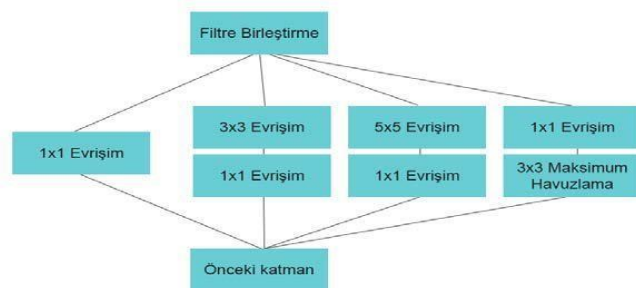
3.7.3. GoogLeNet (Inception)

Inception modülü adı verilen evrişimsel sinir ağı modeli 144 katmandan oluşmaktadır. Bu modelde paralel ağ parçaları kullanılarak daha önceki modellerden farklı bir sıralı yaklaşım kullanılmaktadır. Inception modülü evrişim katmanlarında 1x1, 3x3, 5x5 filtre boyutu ve 3x3 boyutunda maksimum havuzlama işlemlerinin paralel gerçekleştirilmesiyle oluşmaktadır (Kızrak & Bolat, 2018). Şekil 3.17. 'de temel inception modülü görülmektedir.



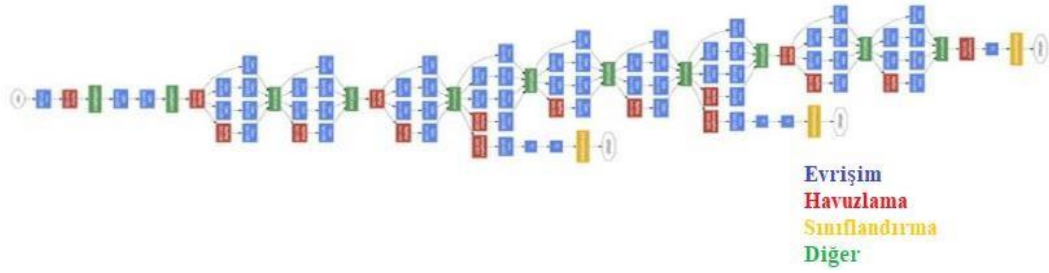
Şekil 3.17. Inception modül yapısı

Inception modülünde 3 farklı filtre boyutlu evrişim işlemi ve maksimum havuzlama işlemi uygulanmaktadır. Klasik evrişim ağıının her katmanında evrişim işlemi, filtre boyutu ya da havuzlama işlemine gerek duyulurken, inception modülünde tüm bu işlemler paralel olarak gerçekleştirilmektedir. Bu modülde filtre boyutunun artması parametre sayısını ve hesaplama maliyetini artırmaktadır (Szegedy et al., 2017).



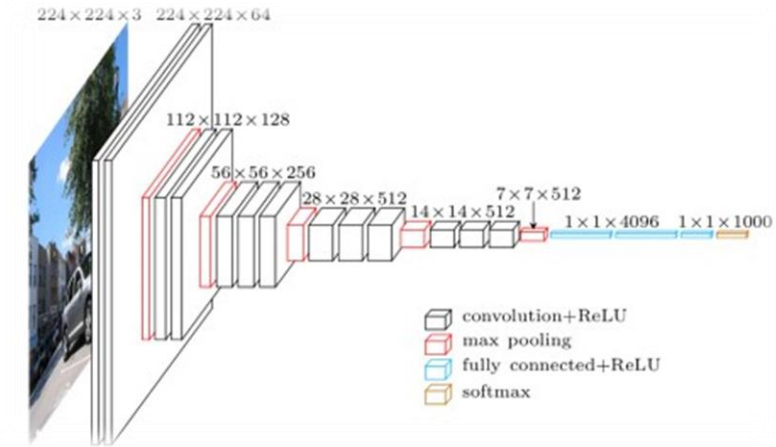
Şekil 3.18. Boyut düşürme işleminden sonra Inception yapısı

Hesaplama maliyetini düşürmek için, Şekil 3.18. 'de gösterildiği gibi Inception modülü içerisine 1x1 boyutunda evrişim katmanını eklenerek, evrişim işlemi gerçekleştirilmektedir. Yapılan bu işlem darboğaz olarak adlandırılmakta ve parametre sayısını 10 kat azaltmaktadır. Bununla birlikte hesap karmaşıklığını azaltarak modelin performansını artırmaktadır. Şekil 3.19. 'de GoogLeNet model yapısı görülmektedir.



Şekil 3.19. GoogLeNet Ağ model yapısı (Szegedy et al., 2015)

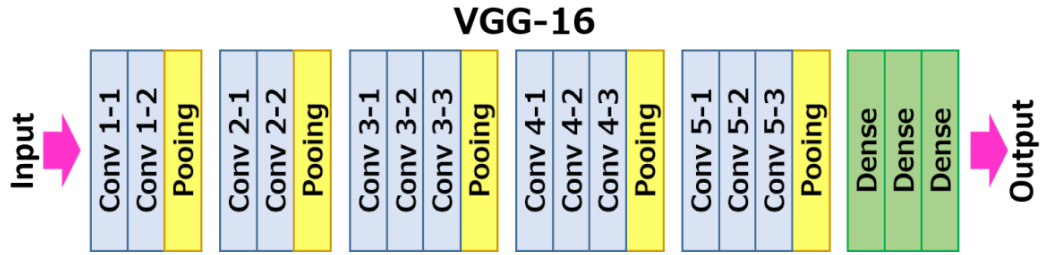
3.7.4. VGGNet



Şekil 3.20. VGGNet model yapısı [13].

VGGNet'i diğer modellerden ayıran en önemli farkı, ikili veya üçlü evrişim katmanlarını havuzlama katmanlarının izlemesidir. Şekil 3.20. 'de VGGNet modeli görülmektedir. Bu modelde yaklaşık 138 milyon parametrenin hesabı yapılarak, (Zeiler & Fergus, 2014) 1000 sınıflı nesne tanımlanmaktadır. VGGNet, VGG16 ve VGG19 adlı iki farklı model yapısı kullanılmaktadır.

VGG16 modelinde 13 evrişim katmanı, 3 tam bağlı katman, havuzlama, aktivasyon fonksiyonu, dropout ve sınıflandırma katmanlarıyla birlikte toplam 41 katman bulunmaktadır. VGG19 modelinde ise VGG16 modeline ek olarak 3 evrişim katmanın eklenmesiyle oluşmaktadır. Bu modelde de diğer modellerde olduğu gibi girişten çıkışa doğru veri boyutu azalırken kanal sayısı artmaktadır (Simonyan & Zisserman, 2015). Ayrıca her evrişim işleminde giriş ve çıkış boyutları aynı olup 3x3 filtre boyutu, havuzlama katmanlarında 2 adım sayısı ve 2x2 filtre boyutu kullanılmaktadır (Krizhevsky et al., 2012).



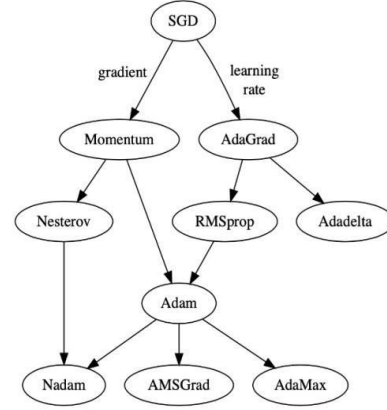
Şekil 3.21. VGG16 modelin daha sezgisel bir düzeni [18].

3.8. Optimizasyon Algoritmaları

Konvolüsyonel sinir ağlarında, ağırlıkların doğru bir şekilde güncellenmesi öğrenme işlemi için çok önemlidir. Hata hesaplandıktan sonra elde edilen hataya göre ağırlıklar güncellenir. Ağırlıkların güncellenmesi için Geri Yayılım Algoritması (Back-Propagation) kullanılır. Geri yayılım algoritması ile sinir ağındaki her bir ağırlığın, hesaplanan hataya olan etkisini hesaplamak için gradyan tabanlı Stokastik Gradyan Azaltma (SGD), RMS-Prop, Adam ve Adadelta metotları sıklıkla kullanılmaktadır [19].

Çizelge 3.4. Gradient Descent yöntemlerinin karşılaştırılması [20].

Yöntem	Tarih	Öğrenme Katsayısı	Gradient
SGD	1951	✓	✓
Momentum	1964		✓
Adam	2014	✓	✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	



Şekil 3.22. Gradient Descent yöntemlerin evrimsel haritası [20].

- Gradyan Azaltma (Gradient Descent): Gradyan azaltma algoritması (Batch Gradient Descent), tüm veri kümesinde gradyan hesabını gerçekleştirir. Bu durum bir iterasyonun çok uzun sürmesine neden olmakta ve erken yakınsama ile yerel optimumlara yakalanma riskini arttırmaktadır. Bu gibi problemlerden dolayı mini-batch gradient descent yöntemi önerilmiştir. Bu yöntemde eğitim veri seti belirlenen sabit değerde yığınlara (batch) bölünmüştür ve bölünen her bir yığın için toplam hata hesaplanır ve ağırlıklar güncellenir. Eğer mini-batch değeri “1” olacak şekilde seçilirse stokastik gradyan azaltma (SGD) algoritması uygulanmış olur; bu durumda her defasında bir örnek için hata hesaplanır ve ağırlıklar güncellenir, dolayısıyla vektörizasyonun getirmiş olduğu hız kaybedilir.

- Momentumlu Gradyan Azaltma (Gradient Descent with Momentum): Gradyan azaltma yöntemlerinde öğrenme adımlarının en iyi sonuca doğru daha hızlı hareket etmesi ve daha az sapmaların olması istenir. Atılan öğrenme adımları bazen çok fazla olabilir ve atılan bu yanlış adımlar en iyi sonuca ulaşma süresini doğrudan etkileyebilir. Bu problemleri iyileştirmek amacıyla momentumlu gradyan azaltma yöntemi önerilmiştir. Geçmişteki öğrenme adımlarının ortalama hızı bir sonraki öğrenme adımının hızını sınırlamak için kullanılabilir. Bu yöntem ile üstel olarak azalan geçmişteki gradyanların dinamik ortalaması tutulur ve bu dinamik ortalamalar hesaba katılarak o yönde bir ilerleme sağlanır. Bu sayede öğrenme adımları en iyi sonuca doğru daha hızlı hareket eder ve daha az sapma olur.

- RMS-Prop (Root Mean Square Propagation): RMS-Prop yöntemi, Geoffrey Hinton'ın Coursera'da verdiği derste önerdiği bir optimizasyon yöntemidir. Bu yöntem uyarlanabilir öğrenme oranı kullanan algoritmalar alt kümesine girmektedir. Yani öğrenme adımları boyunca her bir parametre için ayrı bir öğrenme oranı kullanılır ve bu öğrenme oranları elde edilen sonuçlara göre uyarlanabilir, güncellenebilir. Bu yöntem üstel olarak azalış gösteren ortalama yaklaşımını kullanarak, uzak geçmişteki noktaları dikkate almaz. Bu sayede yakınsamayı hızlandırabilir.

Adadelata optimizasyon yönteminde, parametrelerin kendi öğrenme hızları vardır. Bu öğrenme hızları giderek azalmaktadır ve sistem belirli bir noktadan sonra öğrenme işlemini gerçekleştiremez. RMS-Prop yöntemi bu sorunu çözmek amacıyla önerilmiştir.

- Adam (Adaptive Moment Estimation): Adam, klasik stokastik gradyan azaltma (SGD) yöntemi yerine kullanılabilecek daha verimli, adaptif bir optimizasyon algoritmasıdır. Yani her bir parametre için dinamik bir şekilde öğrenme oranını (learning rate) günceller. Hesaplama yükü olarak verimlidir ve düşük bellek gereksinimlerine ihtiyaç duyar. Adam, RMS-Prop ve Momentum ile birlikte gradyan düşürme yöntemlerine benzer bir parametre güncelleme metodu kullanmaktadır [19].

3.9. Model Performansının Değerlendirilmesi

Derin öğrenmede, sistem performansının belirlenmesi için model performansının değerlendirilmesi önemlidir.

3.9.1. Karışıklık Matrisi (Confusion Matrix)

Makine öğrenmesindeki sınıflandırma probleminde, bir karışıklık matrisi, verilerin değerlendirilmesini sağlayan özel bir tablo düzenidir. Bir algoritmanın performansı, matrisin her satırı veya sütunu tahmin edilen değerleri temsil ederken, her satır veya sütun gerçek bir değerdeki değerler ile temsil eder.

Karışıklık matrisi, verilerin mevcut durumu ile sınıflandırma modelinin doğru ve yanlış tahminlerinin miktarlarını gösterir. Bir karışıklık matrisi Şekil 3.23. 'de görülebilir. Matris elemanı, tahmin edilen değerlerin sayısına göre değişir. Gerçek değerler veri setinde bulunan çıkış değerleridir. Pozitif bir durum için pozitif bir değer ve negatif bir durum için negatif bir değer alır. Derin öğrenme yöntemi ile oluşturulan modelin amacı, hem pozitif hem de negatif değerleri doğru bir şekilde tahmin etmektir.

Confusion Matrix		Actual Values (Gerçek değerler)	
		Positive	Negative
Predicted Values (Tahmin edilen değerler)	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Şekil 3.23. Karışıklık matrisi

- Karışıklık matrisinde bulunan değerler

Gerçek pozitif (TP) = Doğru olarak pozitif olarak işaretlenen örneklerin sayısı.

Yanlış pozitif (FP) = Yanlış olarak pozitif olarak işaretlenen örneklerin sayısı.

Gerçek negatif (TN) = Doğru olarak negatif olarak işaretlenen örnek sayısı.

Yanlış negatif (FN) = Yanlış olarak negatif olarak işaretlenen numune sayısı [21].

3.9.2. Sınıflandırma Ölçütleri

Sınıflandırma aşamasında; en uygunun ayırt edilmesi, sınıflandırma yönteminin seçilmesi, sınıflar arasında önemliliğin belirlenmesi ve tahminleme gibi kriterlerin bulunabilmesi için ölçütler kullanılmaktadır. Çizelge 3.5'te sınıflandırma performanslarını karşılaştırmak için çalışmada kullanılan ölçütler verilmiştir [22].

Çizelge 3.5. Sınıflandırma ölçütleri [23].

Ölçüt	Formül	Açıklama
Doğruluk (Accuracy)	$\frac{(TP + TN)}{(TP + FP + TN + FN)}$	Bir sınıflandırıcının başarısını göstermek için kullanılan en yaygın ölçüttür. Doğru tahminlerin toplam örnekleme oranıdır. Toplam örnek sayısı üzerindeki doğru tahminler
Geri Çağırma (Recall)	$TP / (TP + FN)$	Doğru pozitif oranı. Sınıflandırıcı tarafından bulunan doğru pozitif örneklerin, tüm pozitif örneklere oranıdır. Doğru sınıflandırılmış pozitiflerin değerini ölçme
Kesinlik (Precision)	$TP / (TP + FP)$	Sınıflandırıcı tarafından bulunan doğru pozitif sayısının, pozitif olarak sınıflandırılan örneklerin toplamına oranıdır. Pozitif olarak etiketlenen örneklerin pozitif olarak sınıflandırılmış toplam örnek sayısı
F-Skor (F-Score)	$\frac{2*TP}{(2*TP + FP + FN)}$	Gerçek pozitif oranının (Recall) ve Kesinliğin (Precision) harmonik ortalamasıdır. Sistemin doğru optimize edilmesi, dengesiz sınıflar için yararlı hibrit metrik
Özgüllük (Specificity)	$TN / (TN + FP)$	Doğru negatif oranı. Sınıflandırıcı tarafından bulunan doğru negatif örneklerin, tüm negatif örneklere oranıdır. Gerçek negatif örneklerin oranı

3.9.3. İşlem Karakteristik Eğrisi (ROC)

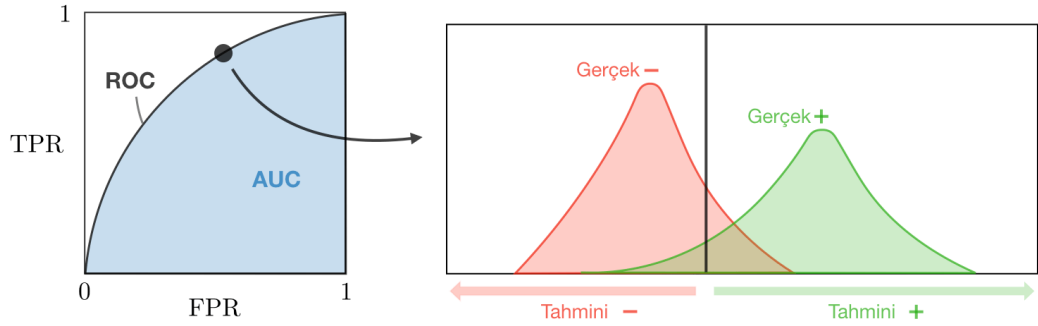
Herhangi bir sınıflandırma modelinin performansını kontrol etmek için en önemli değerlendirme ölçütlerinden biridir. Özellikle dengesiz veri setlerinin bulunduğu durumlarda, makine öğrenmesi algoritmalarının performansını değerlendirmek için en yaygın kullanılan ölçümlerden biridir. Ve modelin tahmininde ne kadar iyi olduğunu açıklar.

ROC farklı sınıflar için bir olasılık eğrisidir. Tipik bir ROC eğrisinde X ekseninde Yanlış Pozitif Oran (FPR) ve Y ekseninde Gerçek Pozitif Orana (TPR) vardır. [24]. İşlem Karakteristik Eğrisi (receiver operating curve), eşik değeri değiştirilerek Doğru Pozitif Oranı-Yanlış Pozitif Oranı grafiğidir. Bu metrikler aşağıdaki tabloda özetlenmiştir:

Çizelge 3.6. ROC eğrisinde yer alan metrikler

Metrik	Formül	Eşdeğer
True Positive Rate TPR	$TP / (TP + FN)$	Geri çağırma (Recall)
False Positive Rate FPR	$FP / (TN + FP)$	1-specificity

İşlem Karakteristik Eğrisi (ROC)'nin altındaki alan arttıkça kapsanan alan ne kadar büyükse, makine öğrenme modelleri o kadar iyi verilen sınıfları ayırt etmede daha iyidir. AUC için ideal değer 1'dir. [25].



Şekil 3.24. ROC Eğrisi

4. SONUÇLAR VE TARTIŞMA

Çalışmada kullanılan görüntüler ağa girdi olarak vermeden önce aynı boyuta getirilmelidir. Bu çalışma da tek bir veri seti kullanıldığı için bu durum bir sorun olmamıştır; ancak birden fazla veri setinden görüntüler alındığı durumda ya da aynı veri setinde ama farklı boyutlarda görüntüler kullanıldığında bunların mutlaka aynı boyuta getirilmesi gerekmektedir. Bu çalışmada model eğitiminin çok uzun süreler almaması için Google Colaboratory kullanılmıştır. Önerilen model eğitilirken kullanılan deneysel çalışmalar Python programlama dili ve Keras kütüphanesi ortamı kullanılarak gerçekleştirilmiştir.

Çalışmada Bölüm 3.3’de yer alan veri setlerinin her biri için Bölüm 3.7’de yer alan sınıflandırma algoritmaları uygulanmıştır. Her bir veri seti için iki farklı değerlendirme yöntemi kullanılarak sınıflandırıcı modeller oluşturulmuştur. Kullanılan yöntemler Hold-out ve 5 katlı çapraz doğrulama (5-CV) yöntemidir.

Çapraz doğrulama yöntemi makine öğrenmesi modellerinde genellikle tercih edilen bir yöntem olmaktadır. Bunun en önemli nedeni modeli eğitirken ve test ederken verinin tamamını hem eğitim hem de test amaçlı kullanmasıdır. Hold-out yöntemi ise verinin sadece ilk aşamada yani model eğitme ve test aşamasında bir kez bölünmesi ile gerçekleştiğinden model oluşturulurken test bölümünde kalan bir verinin modelin yanlış öğrenmesine sebep olma durumu olabilmektedir. Hold-out yöntemi verinin çok fazla olduğu durumlarda zaman kaybını gidermek için tercih edilebilmektedir [26].

4.1. Hold-Out Yöntemi ile Değerlendirme

Hold-out yönteminde, veri setini modeli oluşturmak için eğitim ve test parçalarına bölünme işlemi yapılmaktadır. Eğitim verisi ile model oluşturulurken, test verisi ile de bu modelin doğruluğunun testi yapılmaktadır. Holdout yönteminde %80 train, %20 test, %10 train veri seti içinden validation set olacak şekilde kullanılmıştır.

Çizelge 4.1. Hold-Out Yöntemi ile ayrılmış veri setinin görüntü sayıları

Training set	7210
Testing set	2003
Validation set	802
Toplam	10015

Çizelge 4.2. Hold-out yönteminde kullanılan modellerin hiper parametre değerleri

Model Adı	optimizer	learning_rate (lr)	loss – function	epoch	batch_size
ResNet50	ADAM	0.0001	categorical_crossentropy	30	32
DenseNet201	ADAM	0.0001	categorical_crossentropy	40	64
InceptionV3	ADAM	0.0001	categorical_crossentropy	40	64
VGG16	ADAM	0.0001	categorical_crossentropy	25	64

Yukarıdaki çizelgeye bakılarak farklı farklı optimizasyonlar deneyerek (SGD-adadelta), öğrenme oranı üzerinde değişiklikler yaparak genellikle aynı olsa da sonuç olarak en iyi bu halin olduğu gözlemlenmiştir. Çalışmada kullanılan Adam algoritmasının uygulanması kolaydır, hızlı öğrenme süresine sahiptir, hesaplama açısından verimlidir, az bellek gerektirir ve büyük veri veya parametrelerle ilgili problemler için çok uygundur.

Kayıp fonksiyonu (loss function), tasarlanan bir modelin hem hata oranını hem de performansını ölçen bir fonksiyon türüdür. Derin öğrenmede, Bir sinir ağının son katmanı, fonksiyon kaybının tanımlandığı katmandır. Derin öğrenme uygulamalarında fonksiyon, tasarlanan modelin tahmini ile istenen gerçek değer arasındaki fark iyi tahmine sahip bir model olması durumunda kapasite tasarlanırsa, gerçek değer ile tahmin edilen değer arasındaki fark daha düşük olacaktır. Başka bir deyişle, kayıp değeri daha düşüktür. Daha yüksek kayıp değerine sahip bir çıktı, tasarlanan modelin

kusurlar içerdiğini gösterir. Ancak, iyi tasarlanmış bir modelde, bir kayıp sıfıra yakın bir değer beklenmelidir [21]. Bu çalışmada, kategorik çapraz entropi kaybı fonksiyonu kullanılmıştır. Kategorik çapraz entropi etiketler ve tahminler arasındaki çapraz entropi kaybını hesaplar. Sonuç olarak çok sınıflı sınıflandırma problemlerinde bu fonksiyon kullanabilmektedir.

Batch_size sayısı artırmanın ve epoch sayısı artırmanın doğruluk yüzdesini artırırken süreyi uzattığı gözlemlenmiştir. Ayrıca daha fazla epoch sayısı vermenin de aşırı öğrenmeye uğradığı görülmüştür.

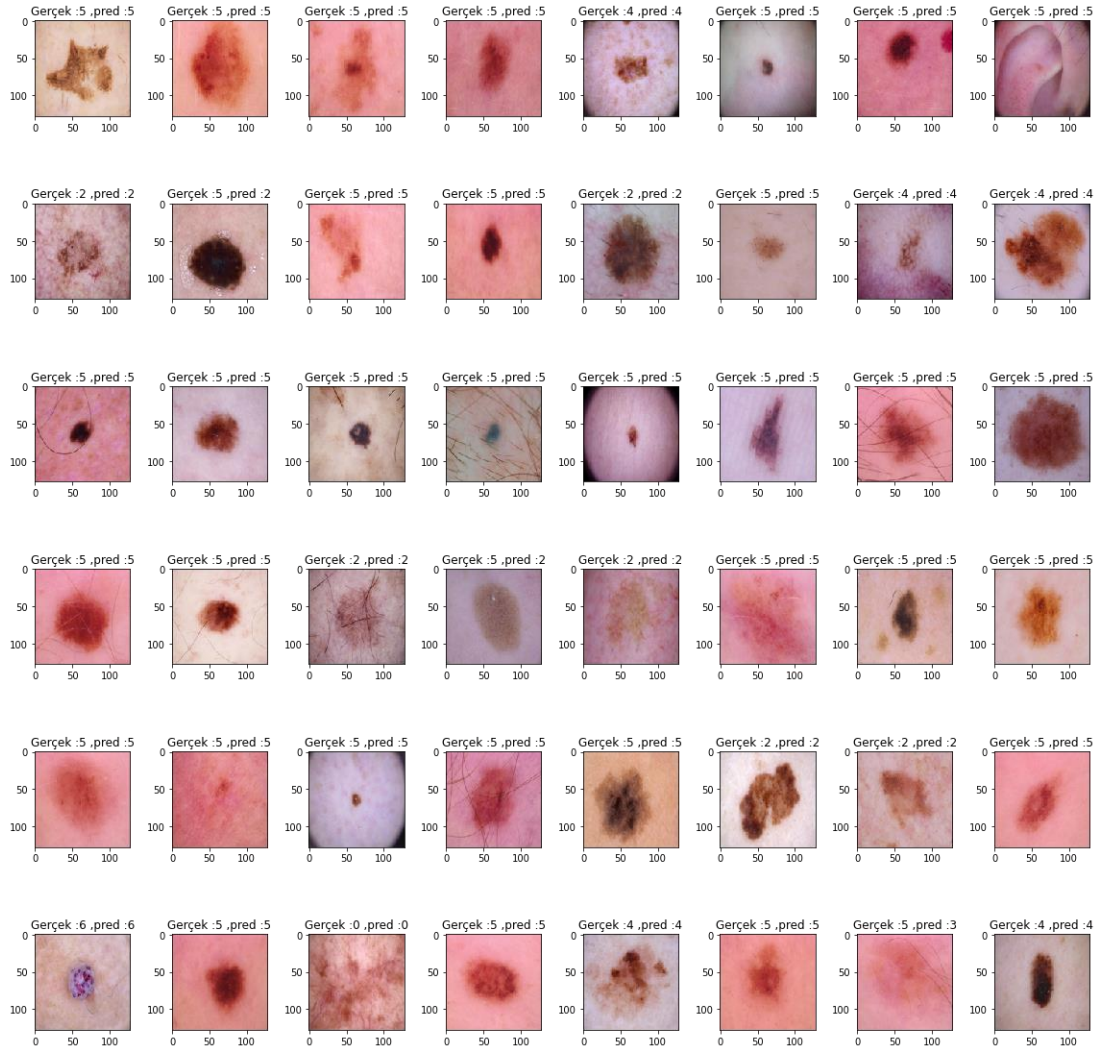
Çizelge 4.3. Hold-out yönteminde kullanılan modellerin başarı oranları

Model Adı	Test Başarı Oranı	Validation Başarı Oranı
ResNet50	%86.37	%86.53
DenseNet201	%88.82	%88.90
InceptionV3	%85.97	%83.79
VGG16	%82.08	%83.92

Sonuçlardan görüldüğü üzere ResNet50, DenseNet201, InceptionV3 ve VGG16 olmak üzere dört farklı önceden eğitilmiş ağ kullanılmış ve gerçekleştirilen deneysel çalışmalar ile her birisi ile elde edilen sonuçlar kendi arasında karşılaştırılmıştır. Ulaşılan sonuçlar dermoskopi görüntüler üzerinde cilt kanserinin tespiti ve sınıflandırılmasında DenseNet201 ağının daha başarılı olduğunu göstermektedir.

Çizelge 4.4. Hold-out yönteminde kullanılan modellerin test verisi üzerinde sonuçları

Model Adı	Toplam Test Verisi	Doğru Tahmin Edilmiş Veri	Yanlış Tahmin Edilmiş Veri
ResNet50	2003	1730	273
DenseNet201	2003	1779	224
InceptionV3	2003	1722	281
Vgg16	2003	1644	359



Şekil 4.1. En iyi doğruluk yüzdesini alan DenseNet201 modelin random test görüntüleri üzerinde sonuçları

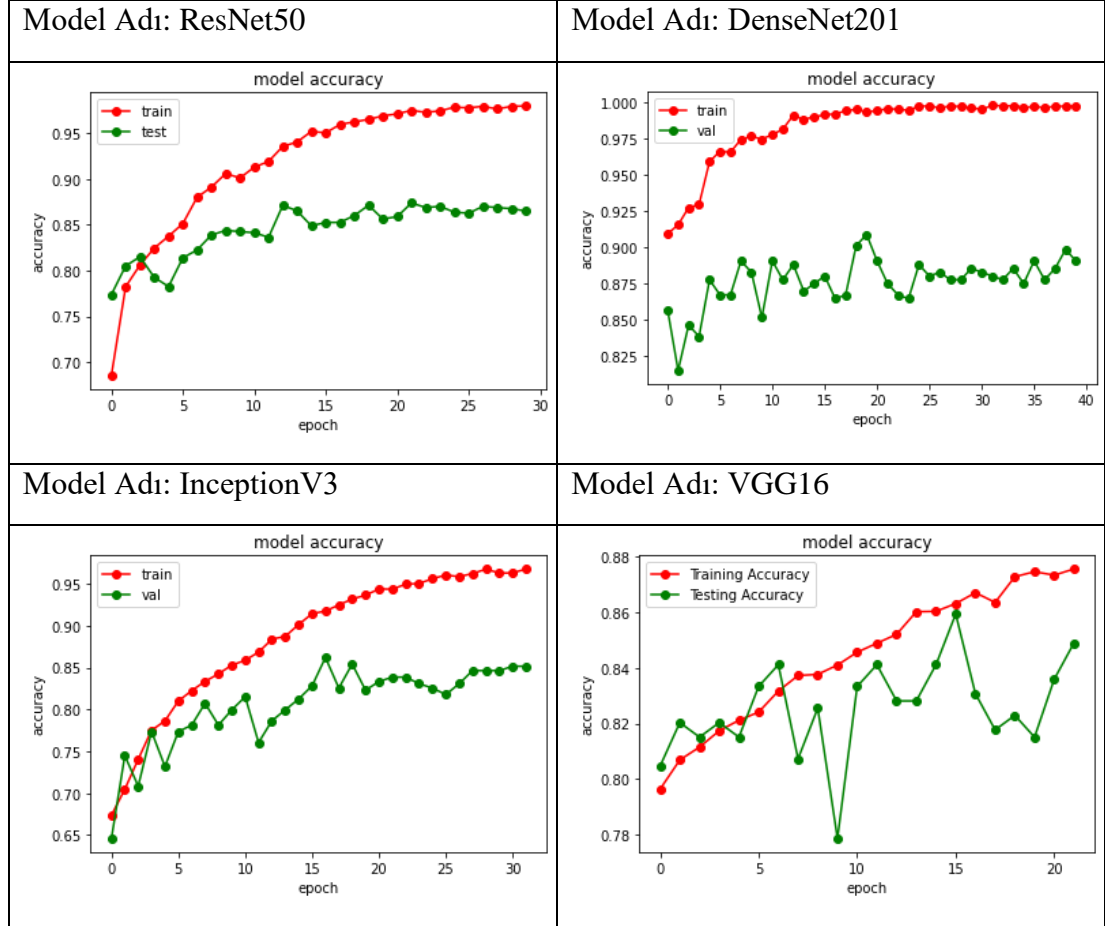
Şekil 4.1 'de görülen en iyi başarı elde eden DenseNet201 modelin random görüntülerinden alınan test verisi üzerinde sonuçlandırılmasında gerçek ve tahmin edilen değerleri göstermektedir. Burada yer alan değerler (0-akiec, 1-bcc, 2-bkl, 3-df, 4-mel, 5-nv, 6-vasc) her bir sınıfa ait değerlerinin doğru tahmin edip edilmediğini sonuçlandırabilmektedir.

Çizelge 4.5. Kullanılan modellerin parametreleri

Model Adı	Parametreleri																																																												
Resnet50	<table><tr><td>conv5_block3_1_bn (BatchNormal1</td><td>(None, 4, 4, 512)</td><td>2848</td><td>conv5_block3_1_conv[0][0]</td></tr><tr><td>conv5_block3_1_relu (Activation</td><td>(None, 4, 4, 512)</td><td>0</td><td>conv5_block3_1_bn[0][0]</td></tr><tr><td>conv5_block3_2_conv (Conv2D)</td><td>(None, 4, 4, 512)</td><td>2359808</td><td>conv5_block3_1_relu[0][0]</td></tr><tr><td>conv5_block3_2_bn (BatchNormal1</td><td>(None, 4, 4, 512)</td><td>2848</td><td>conv5_block3_2_conv[0][0]</td></tr><tr><td>conv5_block3_2_relu (Activation</td><td>(None, 4, 4, 512)</td><td>0</td><td>conv5_block3_2_bn[0][0]</td></tr><tr><td>conv5_block3_3_conv (Conv2D)</td><td>(None, 4, 4, 2048)</td><td>1050624</td><td>conv5_block3_2_relu[0][0]</td></tr><tr><td>conv5_block3_3_bn (BatchNormal1</td><td>(None, 4, 4, 2048)</td><td>8192</td><td>conv5_block3_3_conv[0][0]</td></tr><tr><td>conv5_block3_add (Add)</td><td>(None, 4, 4, 2048)</td><td>0</td><td>conv5_block2_out[0][0] conv5_block3_3_bn[0][0]</td></tr><tr><td>conv5_block3_out (Activation)</td><td>(None, 4, 4, 2048)</td><td>0</td><td>conv5_block3_add[0][0]</td></tr><tr><td>global_average_pooling2d_7 (Glo</td><td>(None, 2048)</td><td>0</td><td>conv5_block3_out[0][0]</td></tr><tr><td>dropout_7 (Dropout)</td><td>(None, 2048)</td><td>0</td><td>global_average_pooling2d_7[0][0]</td></tr><tr><td>dense_7 (Dense)</td><td>(None, 7)</td><td>14343</td><td>dropout_7[0][0]</td></tr><tr><td colspan="4">Total params: 23,602,055</td></tr><tr><td colspan="4">Trainable params: 23,548,935</td></tr><tr><td colspan="4">Non-trainable params: 53,120</td></tr></table>	conv5_block3_1_bn (BatchNormal1	(None, 4, 4, 512)	2848	conv5_block3_1_conv[0][0]	conv5_block3_1_relu (Activation	(None, 4, 4, 512)	0	conv5_block3_1_bn[0][0]	conv5_block3_2_conv (Conv2D)	(None, 4, 4, 512)	2359808	conv5_block3_1_relu[0][0]	conv5_block3_2_bn (BatchNormal1	(None, 4, 4, 512)	2848	conv5_block3_2_conv[0][0]	conv5_block3_2_relu (Activation	(None, 4, 4, 512)	0	conv5_block3_2_bn[0][0]	conv5_block3_3_conv (Conv2D)	(None, 4, 4, 2048)	1050624	conv5_block3_2_relu[0][0]	conv5_block3_3_bn (BatchNormal1	(None, 4, 4, 2048)	8192	conv5_block3_3_conv[0][0]	conv5_block3_add (Add)	(None, 4, 4, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]	conv5_block3_out (Activation)	(None, 4, 4, 2048)	0	conv5_block3_add[0][0]	global_average_pooling2d_7 (Glo	(None, 2048)	0	conv5_block3_out[0][0]	dropout_7 (Dropout)	(None, 2048)	0	global_average_pooling2d_7[0][0]	dense_7 (Dense)	(None, 7)	14343	dropout_7[0][0]	Total params: 23,602,055				Trainable params: 23,548,935				Non-trainable params: 53,120			
conv5_block3_1_bn (BatchNormal1	(None, 4, 4, 512)	2848	conv5_block3_1_conv[0][0]																																																										
conv5_block3_1_relu (Activation	(None, 4, 4, 512)	0	conv5_block3_1_bn[0][0]																																																										
conv5_block3_2_conv (Conv2D)	(None, 4, 4, 512)	2359808	conv5_block3_1_relu[0][0]																																																										
conv5_block3_2_bn (BatchNormal1	(None, 4, 4, 512)	2848	conv5_block3_2_conv[0][0]																																																										
conv5_block3_2_relu (Activation	(None, 4, 4, 512)	0	conv5_block3_2_bn[0][0]																																																										
conv5_block3_3_conv (Conv2D)	(None, 4, 4, 2048)	1050624	conv5_block3_2_relu[0][0]																																																										
conv5_block3_3_bn (BatchNormal1	(None, 4, 4, 2048)	8192	conv5_block3_3_conv[0][0]																																																										
conv5_block3_add (Add)	(None, 4, 4, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]																																																										
conv5_block3_out (Activation)	(None, 4, 4, 2048)	0	conv5_block3_add[0][0]																																																										
global_average_pooling2d_7 (Glo	(None, 2048)	0	conv5_block3_out[0][0]																																																										
dropout_7 (Dropout)	(None, 2048)	0	global_average_pooling2d_7[0][0]																																																										
dense_7 (Dense)	(None, 7)	14343	dropout_7[0][0]																																																										
Total params: 23,602,055																																																													
Trainable params: 23,548,935																																																													
Non-trainable params: 53,120																																																													
DenseNet201	<table><tr><td>conv5_block32_1_conv (Conv2D)</td><td>(None, 4, 4, 128)</td><td>241664</td><td>conv5_block32_0_relu[0][0]</td></tr><tr><td>conv5_block32_1_bn (BatchNormal</td><td>(None, 4, 4, 128)</td><td>512</td><td>conv5_block32_1_conv[0][0]</td></tr><tr><td>conv5_block32_1_relu (Activatio</td><td>(None, 4, 4, 128)</td><td>0</td><td>conv5_block32_1_bn[0][0]</td></tr><tr><td>conv5_block32_2_conv (Conv2D)</td><td>(None, 4, 4, 32)</td><td>36864</td><td>conv5_block32_1_relu[0][0]</td></tr><tr><td>conv5_block32_concat (Concatena</td><td>(None, 4, 4, 1920)</td><td>0</td><td>conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]</td></tr><tr><td>bn (BatchNormalization)</td><td>(None, 4, 4, 1920)</td><td>7680</td><td>conv5_block32_concat[0][0]</td></tr><tr><td>relu (Activation)</td><td>(None, 4, 4, 1920)</td><td>0</td><td>bn[0][0]</td></tr><tr><td>global_max_pooling2d (GlobalMax</td><td>(None, 1920)</td><td>0</td><td>relu[0][0]</td></tr><tr><td>dense (Dense)</td><td>(None, 512)</td><td>983552</td><td>global_max_pooling2d[0][0]</td></tr><tr><td>dropout (Dropout)</td><td>(None, 512)</td><td>0</td><td>dense[0][0]</td></tr><tr><td>dense_1 (Dense)</td><td>(None, 7)</td><td>3591</td><td>dropout[0][0]</td></tr><tr><td colspan="4">Total params: 19,309,127</td></tr><tr><td colspan="4">Trainable params: 19,000,071</td></tr><tr><td colspan="4">Non-trainable params: 229,056</td></tr></table>	conv5_block32_1_conv (Conv2D)	(None, 4, 4, 128)	241664	conv5_block32_0_relu[0][0]	conv5_block32_1_bn (BatchNormal	(None, 4, 4, 128)	512	conv5_block32_1_conv[0][0]	conv5_block32_1_relu (Activatio	(None, 4, 4, 128)	0	conv5_block32_1_bn[0][0]	conv5_block32_2_conv (Conv2D)	(None, 4, 4, 32)	36864	conv5_block32_1_relu[0][0]	conv5_block32_concat (Concatena	(None, 4, 4, 1920)	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]	bn (BatchNormalization)	(None, 4, 4, 1920)	7680	conv5_block32_concat[0][0]	relu (Activation)	(None, 4, 4, 1920)	0	bn[0][0]	global_max_pooling2d (GlobalMax	(None, 1920)	0	relu[0][0]	dense (Dense)	(None, 512)	983552	global_max_pooling2d[0][0]	dropout (Dropout)	(None, 512)	0	dense[0][0]	dense_1 (Dense)	(None, 7)	3591	dropout[0][0]	Total params: 19,309,127				Trainable params: 19,000,071				Non-trainable params: 229,056							
conv5_block32_1_conv (Conv2D)	(None, 4, 4, 128)	241664	conv5_block32_0_relu[0][0]																																																										
conv5_block32_1_bn (BatchNormal	(None, 4, 4, 128)	512	conv5_block32_1_conv[0][0]																																																										
conv5_block32_1_relu (Activatio	(None, 4, 4, 128)	0	conv5_block32_1_bn[0][0]																																																										
conv5_block32_2_conv (Conv2D)	(None, 4, 4, 32)	36864	conv5_block32_1_relu[0][0]																																																										
conv5_block32_concat (Concatena	(None, 4, 4, 1920)	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]																																																										
bn (BatchNormalization)	(None, 4, 4, 1920)	7680	conv5_block32_concat[0][0]																																																										
relu (Activation)	(None, 4, 4, 1920)	0	bn[0][0]																																																										
global_max_pooling2d (GlobalMax	(None, 1920)	0	relu[0][0]																																																										
dense (Dense)	(None, 512)	983552	global_max_pooling2d[0][0]																																																										
dropout (Dropout)	(None, 512)	0	dense[0][0]																																																										
dense_1 (Dense)	(None, 7)	3591	dropout[0][0]																																																										
Total params: 19,309,127																																																													
Trainable params: 19,000,071																																																													
Non-trainable params: 229,056																																																													
InceptionV3	<table><tr><td>batch_normalization_187 (BatchN</td><td>(None, 2, 2, 192)</td><td>576</td><td>conv2d_187[0][0]</td></tr><tr><td>activation_179 (Activation)</td><td>(None, 2, 2, 320)</td><td>0</td><td>batch_normalization_179[0][0]</td></tr><tr><td>mixed9_1 (Concatenate)</td><td>(None, 2, 2, 768)</td><td>0</td><td>activation_181[0][0] activation_182[0][0]</td></tr><tr><td>concatenate_3 (Concatenate)</td><td>(None, 2, 2, 768)</td><td>0</td><td>activation_185[0][0] activation_186[0][0]</td></tr><tr><td>activation_187 (Activation)</td><td>(None, 2, 2, 192)</td><td>0</td><td>batch_normalization_187[0][0]</td></tr><tr><td>mixed10 (Concatenate)</td><td>(None, 2, 2, 2048)</td><td>0</td><td>activation_179[0][0] mixed9_1[0][0] concatenate_3[0][0] activation_187[0][0]</td></tr><tr><td>global_max_pooling2d_1 (GlobalM</td><td>(None, 2048)</td><td>0</td><td>mixed10[0][0]</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 512)</td><td>1049088</td><td>global_max_pooling2d_1[0][0]</td></tr><tr><td>dropout_1 (Dropout)</td><td>(None, 512)</td><td>0</td><td>dense_2[0][0]</td></tr><tr><td>dense_3 (Dense)</td><td>(None, 7)</td><td>3591</td><td>dropout_1[0][0]</td></tr><tr><td colspan="4">Total params: 22,855,463</td></tr><tr><td colspan="4">Trainable params: 22,821,031</td></tr><tr><td colspan="4">Non-trainable params: 34,432</td></tr></table>	batch_normalization_187 (BatchN	(None, 2, 2, 192)	576	conv2d_187[0][0]	activation_179 (Activation)	(None, 2, 2, 320)	0	batch_normalization_179[0][0]	mixed9_1 (Concatenate)	(None, 2, 2, 768)	0	activation_181[0][0] activation_182[0][0]	concatenate_3 (Concatenate)	(None, 2, 2, 768)	0	activation_185[0][0] activation_186[0][0]	activation_187 (Activation)	(None, 2, 2, 192)	0	batch_normalization_187[0][0]	mixed10 (Concatenate)	(None, 2, 2, 2048)	0	activation_179[0][0] mixed9_1[0][0] concatenate_3[0][0] activation_187[0][0]	global_max_pooling2d_1 (GlobalM	(None, 2048)	0	mixed10[0][0]	dense_2 (Dense)	(None, 512)	1049088	global_max_pooling2d_1[0][0]	dropout_1 (Dropout)	(None, 512)	0	dense_2[0][0]	dense_3 (Dense)	(None, 7)	3591	dropout_1[0][0]	Total params: 22,855,463				Trainable params: 22,821,031				Non-trainable params: 34,432											
batch_normalization_187 (BatchN	(None, 2, 2, 192)	576	conv2d_187[0][0]																																																										
activation_179 (Activation)	(None, 2, 2, 320)	0	batch_normalization_179[0][0]																																																										
mixed9_1 (Concatenate)	(None, 2, 2, 768)	0	activation_181[0][0] activation_182[0][0]																																																										
concatenate_3 (Concatenate)	(None, 2, 2, 768)	0	activation_185[0][0] activation_186[0][0]																																																										
activation_187 (Activation)	(None, 2, 2, 192)	0	batch_normalization_187[0][0]																																																										
mixed10 (Concatenate)	(None, 2, 2, 2048)	0	activation_179[0][0] mixed9_1[0][0] concatenate_3[0][0] activation_187[0][0]																																																										
global_max_pooling2d_1 (GlobalM	(None, 2048)	0	mixed10[0][0]																																																										
dense_2 (Dense)	(None, 512)	1049088	global_max_pooling2d_1[0][0]																																																										
dropout_1 (Dropout)	(None, 512)	0	dense_2[0][0]																																																										
dense_3 (Dense)	(None, 7)	3591	dropout_1[0][0]																																																										
Total params: 22,855,463																																																													
Trainable params: 22,821,031																																																													
Non-trainable params: 34,432																																																													
VGG16	<table><tr><td>block4_conv1 (Conv2D)</td><td>(None, 16, 16, 512)</td><td>1180160</td><td></td></tr><tr><td>block4_conv2 (Conv2D)</td><td>(None, 16, 16, 512)</td><td>2359808</td><td></td></tr><tr><td>block4_conv3 (Conv2D)</td><td>(None, 16, 16, 512)</td><td>2359808</td><td></td></tr><tr><td>block4_pool (MaxPooling2D)</td><td>(None, 8, 8, 512)</td><td>0</td><td></td></tr><tr><td>block5_conv1 (Conv2D)</td><td>(None, 8, 8, 512)</td><td>2359808</td><td></td></tr><tr><td>block5_conv2 (Conv2D)</td><td>(None, 8, 8, 512)</td><td>2359808</td><td></td></tr><tr><td>block5_conv3 (Conv2D)</td><td>(None, 8, 8, 512)</td><td>2359808</td><td></td></tr><tr><td>block5_pool (MaxPooling2D)</td><td>(None, 4, 4, 512)</td><td>0</td><td></td></tr><tr><td>global_max_pooling2d_3 (Glob</td><td>(None, 512)</td><td>0</td><td></td></tr><tr><td>dense_6 (Dense)</td><td>(None, 512)</td><td>262656</td><td></td></tr><tr><td>dropout_3 (Dropout)</td><td>(None, 512)</td><td>0</td><td></td></tr><tr><td>dense_7 (Dense)</td><td>(None, 7)</td><td>3591</td><td></td></tr><tr><td colspan="4">Total params: 14,980,935</td></tr><tr><td colspan="4">Trainable params: 7,345,671</td></tr><tr><td colspan="4">Non-trainable params: 7,635,264</td></tr></table>	block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160		block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808		block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808		block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0		block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808		block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808		block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808		block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0		global_max_pooling2d_3 (Glob	(None, 512)	0		dense_6 (Dense)	(None, 512)	262656		dropout_3 (Dropout)	(None, 512)	0		dense_7 (Dense)	(None, 7)	3591		Total params: 14,980,935				Trainable params: 7,345,671				Non-trainable params: 7,635,264			
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160																																																											
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808																																																											
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808																																																											
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0																																																											
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808																																																											
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808																																																											
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808																																																											
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0																																																											
global_max_pooling2d_3 (Glob	(None, 512)	0																																																											
dense_6 (Dense)	(None, 512)	262656																																																											
dropout_3 (Dropout)	(None, 512)	0																																																											
dense_7 (Dense)	(None, 7)	3591																																																											
Total params: 14,980,935																																																													
Trainable params: 7,345,671																																																													
Non-trainable params: 7,635,264																																																													

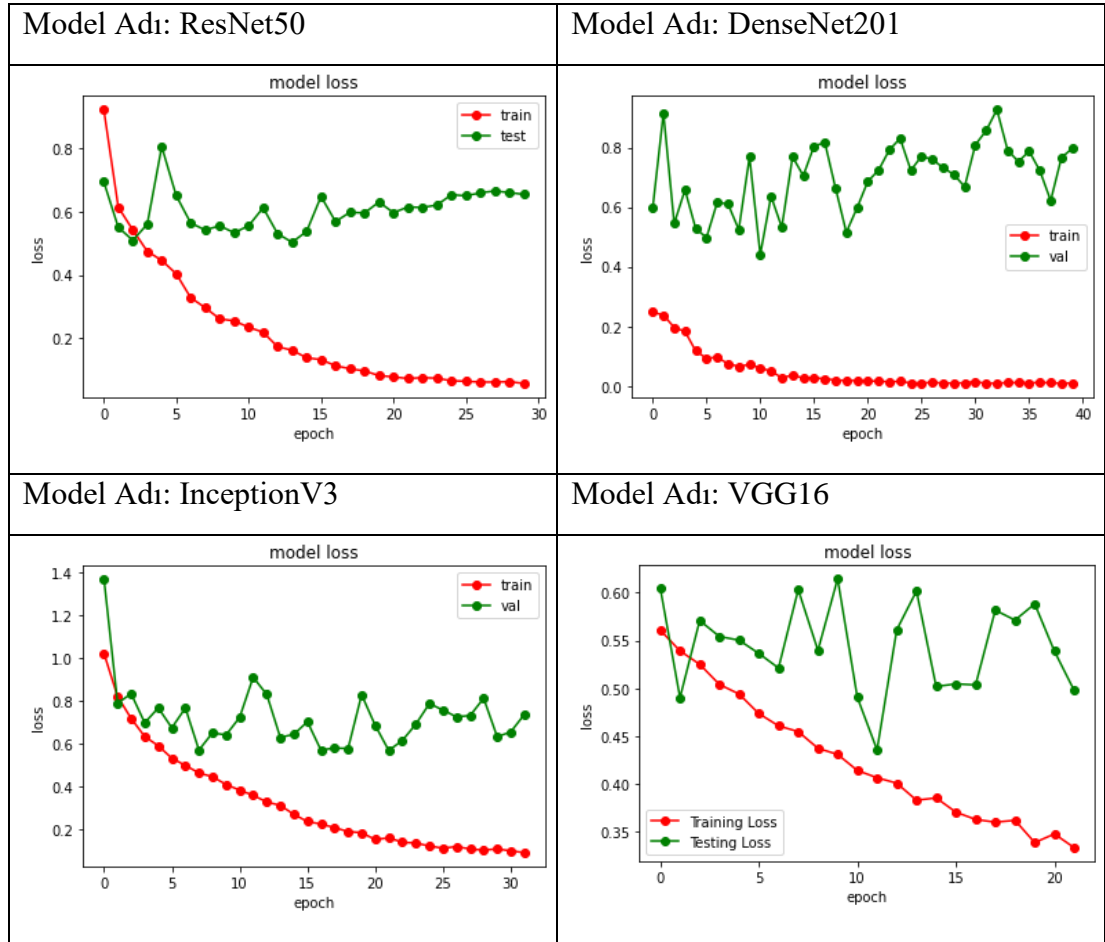
Dönemlere karşı doğruluk grafikleri Çizelge 4.6. 'de gösterilmektedir:

Çizelge 4.6. Hold-out yönteminde kullanılan modellerin Accuracy Grafikleri



Dönemlere karşı kayıp (loss) grafikleri Çizelge 4.7. 'de gösterilmektedir:

Çizelge 4.7. Hold-out yönteminde kullanılan modellerin Loss Grafikleri



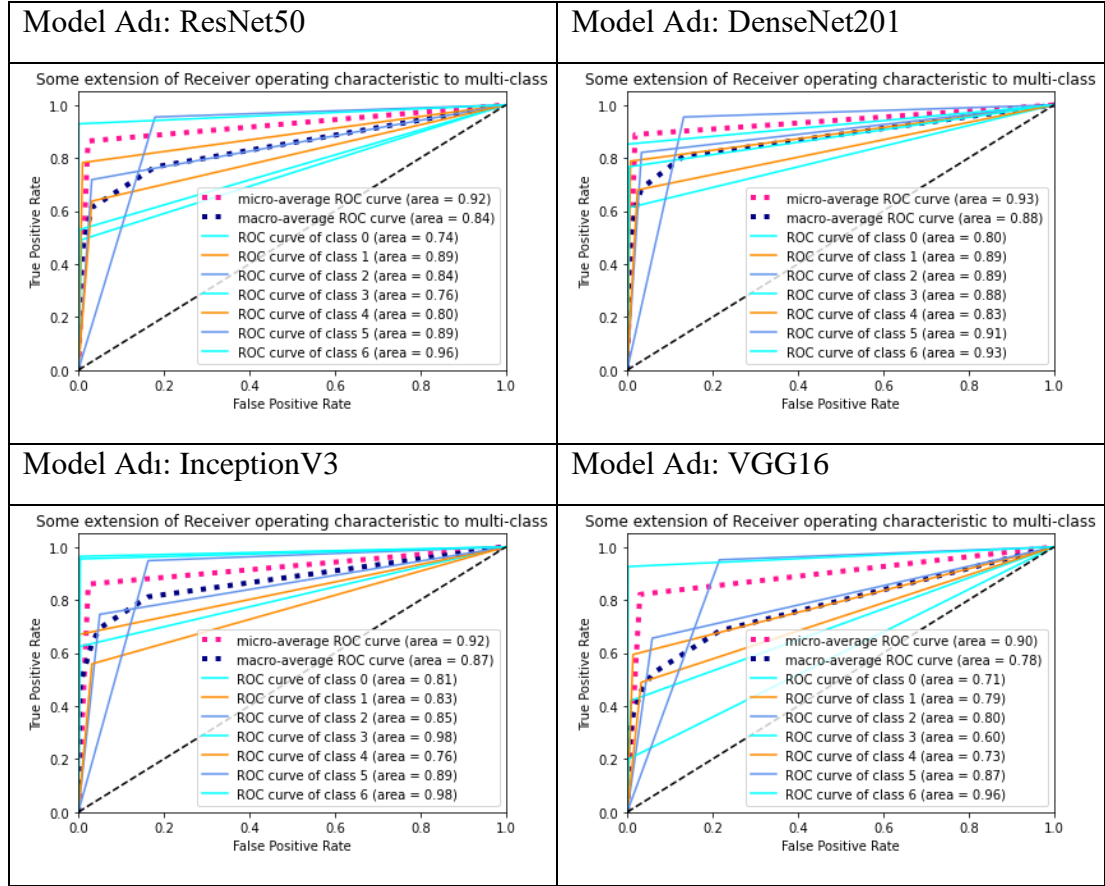
Kullanılan modellerin karışıklık matrisleri Çizelge 4.8. 'de gösterilmektedir:

Çizelge 4.8. Hold-out yönteminde kullanılan modellerin Confusion Matrixleri

Model Ad1: ResNet50								Model Ad1: DenseNet201										
True label	Confusion Matrix							True label	Confusion Matrix							True label		
	Actinic keratoses	29.0	5.0	14.0	0.0	4.0	7.0		0.0	Actinic keratoses	38.0	8.0	12.0	2.0	0.0		2.0	0.0
	Basal cell carcinoma	5.0	79.0	3.0	2.0	2.0	10.0		0.0	Basal cell carcinoma	1.0	78.0	8.0	0.0	3.0		8.0	1.0
	Benign keratosis	5.0	4.0	163.0	3.0	20.0	32.0		0.0	Benign keratosis	3.0	1.0	174.0	1.0	8.0		25.0	0.0
	Dermatofibroma	1.0	3.0	1.0	9.0	0.0	3.0		0.0	Dermatofibroma	1.0	0.0	2.0	23.0	0.0		4.0	0.0
	Melanoma	3.0	3.0	13.0	0.0	151.0	67.0		0.0	Melanoma	4.0	2.0	16.0	4.0	144.0		42.0	0.0
	Melanocytic nevi	1.0	4.0	25.0	0.0	29.0	1273.0		2.0	Melanocytic nevi	2.0	4.0	22.0	2.0	32.0		1299.0	0.0
	Vascular lesions	0.0	0.0	0.0	0.0	1.0	1.0		26.0	Vascular lesions	0.0	0.0	0.0	0.0	0.0		4.0	23.0
Predicted label							Predicted label											
Model Ad1: InceptionV3								Model Ad1: VGG16										
True label	Confusion Matrix							True label	Confusion Matrix							True label		
	Actinic keratoses	40.0	2.0	17.0	1.0	4.0	0.0		0.0	Actinic keratoses	29.0	5.0	22.0	1.0	7.0		5.0	0.0
	Basal cell carcinoma	1.0	67.0	11.0	3.0	5.0	9.0		4.0	Basal cell carcinoma	5.0	60.0	16.0	1.0	3.0		16.0	0.0
	Benign keratosis	4.0	2.0	170.0	1.0	19.0	31.0		1.0	Benign keratosis	6.0	9.0	141.0	3.0	21.0		35.0	0.0
	Dermatofibroma	0.0	0.0	1.0	27.0	0.0	0.0		0.0	Dermatofibroma	1.0	1.0	8.0	5.0	1.0		8.0	1.0
	Melanoma	4.0	0.0	21.0	2.0	119.0	67.0		0.0	Melanoma	2.0	2.0	32.0	0.0	112.0		79.0	2.0
	Melanocytic nevi	0.0	1.0	39.0	2.0	28.0	1278.0		0.0	Melanocytic nevi	2.0	9.0	27.0	1.0	25.0		1272.0	1.0
	Vascular lesions	0.0	0.0	0.0	0.0	1.0	0.0		21.0	Vascular lesions	0.0	0.0	0.0	0.0	1.0		1.0	25.0
Predicted label							Predicted label											

Kullanılan modellerin ROC eğrileri Çizelge 4.9. 'de gösterilmektedir:

Çizelge 4.9. Hold-out yönteminde kullanılan modellerin ROC Eğrileri



ROC eğrisi, 1-spesifikliğe karşı duyarlılık çizilerek hesaplanır ve sınıflandırıcıyı değerlendirmek için kullanılmaktadır. ROC eğrisi köşegenden ne kadar saparsa, sınıflandırıcı o kadar iyi olmaktadır.

ROC eğrisini ve ROC alanını çoklu etiket sınıflandırmasına genişletmek için çıktığı ikili hale getirmek gerekir. Etiket başına bir ROC eğrisi çizilebilir, ancak etiket gösterge matrisinin her bir ögesini ikili bir tahmin (mikro ortalama) olarak ele alarak bir ROC eğrisi de çizilebilir.

Çoklu etiket sınıflandırması için başka bir değerlendirme ölçüsü, her bir etiketin sınıflandırmasına eşit ağırlık veren makro ortalama [27].

ROC eğrisi bize F1 değeri arttıkça AUC altında kalan alanın arttığını göstermiştir.

Çizelge 4.10. Hold-out yönteminde kullanılan modellerin Sınıflandırma Raporları

Model Adı	Classification Report				
ResNet50		precision	recall	f1-score	support
	0-akiec	0.66	0.49	0.56	59
	1-bcc	0.81	0.78	0.79	101
	2-bkl	0.74	0.72	0.73	227
	3-df	0.64	0.53	0.58	17
	4-mel	0.73	0.64	0.68	237
	5-nv	0.91	0.95	0.93	1334
	6-vasc	0.93	0.93	0.93	28
	accuracy			0.86	2003
	macro avg	0.77	0.72	0.74	2003
	weighted avg	0.86	0.86	0.86	2003
DenseNet201		precision	recall	f1-score	support
	0-akiec	0.78	0.61	0.68	62
	1-bcc	0.84	0.79	0.81	99
	2-bkl	0.74	0.82	0.78	212
	3-df	0.72	0.77	0.74	30
	4-mel	0.77	0.68	0.72	212
	5-nv	0.94	0.95	0.95	1361
	6-vasc	0.96	0.85	0.90	27
	accuracy			0.89	2003
	macro avg	0.82	0.78	0.80	2003
	weighted avg	0.89	0.89	0.89	2003
InceptionV3		precision	recall	f1-score	support
	0-akiec	0.82	0.62	0.71	64
	1-bcc	0.93	0.67	0.78	100
	2-bkl	0.66	0.75	0.70	228
	3-df	0.75	0.96	0.84	28
	4-mel	0.68	0.56	0.61	213
	5-nv	0.92	0.95	0.94	1348
	6-vasc	0.81	0.95	0.88	22
	accuracy			0.86	2003
	macro avg	0.79	0.78	0.78	2003
	weighted avg	0.86	0.86	0.86	2003
VGG16		precision	recall	f1-score	support
	0-akiec	0.64	0.42	0.51	69
	1-bcc	0.70	0.59	0.64	101
	2-bkl	0.57	0.66	0.61	215
	3-df	0.45	0.20	0.28	25
	4-mel	0.66	0.49	0.56	229
	5-nv	0.90	0.95	0.92	1337
	6-vasc	0.86	0.93	0.89	27
	accuracy			0.82	2003
	macro avg	0.68	0.61	0.63	2003
	weighted avg	0.81	0.82	0.81	2003

Dengeli dağılmayan veri setlerinde model başarısını sadece Accuracy metriği ile ölçmek yeterli değerlidir ve Precision-Recall metriklerini mutlaka kontrol etmemiz gereklidir [28]. Precision değeri arttıkça Recall değerinin azaldığı görülmektedir. Bu nedenle bu iki değerin harmonik ortalaması olan F1 metriği bize sınıflandırma modelimizin gerçek başarısını göstermektedir.

Doğruluk (Accuracy) yerine F1 Score değerinin kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktır. Ayrıca sadece False Negative ya da False Positive değil tüm hata maliyetlerini de içerecek bir ölçme metriğine ihtiyaç duyulduğu içinde F1 Score bizim için çok önemlidir.

4.2. 5 katlı çapraz doğrulama (5-CV) Yöntemi ile Değerlendirme

Çapraz doğrulama yönteminde veri seti dengesiz sınıf dağılımlarına sahip olduğundan dolayı sınıflandırma için KFold yerine StratifiedKFold kullanılmıştır. Veriler belirtilen sayıda (5 fold CV) parçaya bölünmektedir. Bu parçalardan 1 tanesi test diğerleri ise eğitim amaçlı kullanılır. Modelin performansını belirlemek için bu işlem her parça için ayrı ayrı belirlenen sayıda yapılarak her aşamada bir başarımlı değeri elde edilmektedir. Bu elde edilen değerlerin ortalaması alınarak oluşturulacak modelin gerçek performans değeri elde edilmiş olur. Çizelge 4.11. 'te kullanılan sınıflandırma algoritmaları ile hangi doğruluk değerlerinin elde edildiği verilmektedir.

ResNet50, DenseNet201, InceptionV3 ve VGG16 modellerinin çok sınıflı sınıflandırma performansı değerlendirilmiştir. Her kat ve modelin ortalama sınıflandırma performansı hesaplanmaktadır. Overlapped (örtüşen) ve her bir ayrı karışıklık matrisi gösterilmektedir. Örtüşen Confusion Matrix oluşturulması tüm kıvrımlarda elde edilen Confusion Matrixlerin toplamını kullanılarak oluşturulmuştur. Böylece genel durum hakkında bir fikir edinilmesi amaçlanmaktadır.

Çizelge 4.11. 5 Fold CV yönteminde kullanılan modellerin hiper parametre değerleri

Model Adı	optimizer	learning_rate (lr)	loss – function	epoch	batch_size
ResNet50	ADAM	0.0001	categorical_crossentropy	15	32
DenseNet201	ADAM	0.0001	categorical_crossentropy	20	64
InceptionV3	ADAM	0.0001	categorical_crossentropy	25	64
VGG16	ADAM	0.0001	categorical_crossentropy	25	32

Çizelge 4.12. 5 Fold CV yönteminde kullanılan modellerin ortalama başarı oranları

Model Adı	5 Fold CV Ortalama Başarı Oranı
ResNet50	%83.69
DenseNet201	%86.66
InceptionV3	%85.77
VGG16	%82.30

5 Fold CV yöntemi üzerinde ortalama başarı oranlarına bakılarak en iyi doğruluk yüzdesini DenseNet201 modeli vermektedir.

Çizelge 4.13. ResNet50 modelin 5 Fold CV başarı sonuçları

Katlama başına puan	Accuracy	Loss
> Fold 1	%84.9226176738739	0.5953788161277771
> Fold 2	%85.62156558036804	0.5996063351631165
> Fold 3	%81.87718391418457	0.6267820596694946
> Fold 4	%82.42636322975159	0.7065300941467285
> Fold 5	%83.62456560134888	0.7421018481254578
> Ortalama	%83.6944591999054	0.6540798306465149

Katlama başına puan ResNet50 modelinde Fold 2 'de en yüksek test başarı oranını elde etmiştir.

Çizelge 4.14. Resnet50 modelin Fold-1 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-1

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Confusion Matrix

21.0	3.0	18.0	3.0	15.0	5.0	0.0
4.0	71.0	9.0	3.0	6.0	9.0	1.0
5.0	4.0	154.0	0.0	21.0	36.0	0.0
0.0	0.0	1.0	18.0	2.0	2.0	0.0
1.0	1.0	21.0	1.0	137.0	62.0	0.0
0.0	5.0	26.0	2.0	33.0	1275.0	0.0
0.0	1.0	0.0	0.0	0.0	2.0	25.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

	precision	recall	f1-score	support
0-akiec	0.68	0.32	0.44	65
1-bcc	0.84	0.69	0.76	103
2-bkl	0.67	0.70	0.69	220
3-df	0.67	0.78	0.72	23
4-mel	0.64	0.61	0.63	223
5-nv	0.92	0.95	0.93	1341
6-vasc	0.96	0.89	0.93	28
accuracy			0.85	2003
macro avg	0.77	0.71	0.73	2003
weighted avg	0.84	0.85	0.84	2003

Çizelge 4.15. Resnet50 modelin Fold-2 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-2

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

33.0

4.0

1.0

1.0

1.0

0.0

0.0

5.0

72.0

1.0

3.0

2.0

0.0

0.0

11.0

12.0

184.0

2.0

32.0

34.0

1.0

3.0

0.0

0.0

14.0

0.0

0.0

0.0

3.0

6.0

12.0

0.0

116.0

35.0

0.0

10.0

9.0

21.0

3.0

72.0

1272.0

3.0

0.0

0.0

1.0

0.0

0.0

0.0

24.0

Predicted Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

1200

1000

800

600

400

200

0

Classification Report

precision

recall

f1-score

support

0-akiec

0.82

0.51

0.63

65

1-bcc

0.87

0.70

0.77

103

2-bkl

0.67

0.84

0.74

220

3-df

0.82

0.61

0.70

23

4-mel

0.67

0.52

0.59

223

5-nv

0.92

0.95

0.93

1341

6-vasc

0.96

0.86

0.91

28

accuracy

0.86

2003

macro avg

0.82

0.71

0.75

2003

weighted avg

0.86

0.86

0.85

2003

Çizelge 4.16. Resnet50 modelin Fold-3 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-3

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Confusion Matrix

39.0	10.0	11.0	1.0	2.0	2.0	0.0
5.0	65.0	7.0	2.0	5.0	19.0	0.0
7.0	8.0	137.0	6.0	12.0	50.0	0.0
0.0	0.0	0.0	14.0	0.0	9.0	0.0
5.0	1.0	30.0	3.0	119.0	64.0	0.0
1.0	7.0	49.0	4.0	37.0	1243.0	0.0
0.0	1.0	0.0	0.0	0.0	5.0	23.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

	precision	recall	f1-score	support
0-akiec	0.68	0.60	0.64	65
1-bcc	0.71	0.63	0.67	103
2-bkl	0.59	0.62	0.60	220
3-df	0.47	0.61	0.53	23
4-mel	0.68	0.54	0.60	222
5-nv	0.89	0.93	0.91	1341
6-vasc	1.00	0.79	0.88	29
accuracy			0.82	2003
macro avg	0.72	0.67	0.69	2003
weighted avg	0.82	0.82	0.82	2003

Çizelge 4.17. Resnet50 modelin Fold-4 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-4

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

23.0

3.0

3.0

0.0

1.0

1.0

0.0

10.0

68.0

8.0

4.0

1.0

7.0

3.0

13.0

9.0

156.0

2.0

41.0

25.0

1.0

2.0

0.0

2.0

14.0

0.0

1.0

0.0

5.0

1.0

5.0

0.0

95.0

21.0

1.0

13.0

21.0

46.0

2.0

83.0

1282.0

11.0

0.0

0.0

0.0

1.0

1.0

4.0

13.0

Predicted Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

1200

1000

800

600

400

200

0

Classification Report

precision

recall

f1-score

support

0-akiec

0.74

0.35

0.47

66

1-bcc

0.67

0.67

0.67

102

2-bkl

0.63

0.71

0.67

220

3-df

0.74

0.61

0.67

23

4-mel

0.74

0.43

0.54

222

5-nv

0.88

0.96

0.92

1341

6-vasc

0.68

0.45

0.54

29

accuracy

0.82

2003

macro avg

0.73

0.60

0.64

2003

weighted avg

0.82

0.82

0.81

2003

Çizelge 4.18. Resnet50 modelin Fold-5 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-5

Confusion Matrix

		Confusion Matrix							<div><div></div><div>1200</div><div>1000</div><div>800</div><div>600</div><div>400</div><div>200</div><div>0</div></div>
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
True Label	Actinic keratoses	21.0	6.0	15.0	6.0	10.0	8.0	0.0	
	Basal cell carcinoma	2.0	59.0	3.0	4.0	5.0	30.0	0.0	
	Benign keratosis	0.0	2.0	149.0	2.0	11.0	55.0	0.0	
	Dermatofibroma	0.0	2.0	0.0	15.0	1.0	5.0	0.0	
	Melanoma	0.0	2.0	10.0	0.0	107.0	104.0	0.0	
	Melanocytic nevi	0.0	2.0	21.0	2.0	15.0	1300.0	1.0	
	Vascular lesions	0.0	0.0	0.0	0.0	0.0	4.0	24.0	
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
		Predicted Label							

Classification Report

	precision	recall	f1-score	support
0-akiec	0.91	0.32	0.47	66
1-bcc	0.81	0.57	0.67	103
2-bkl	0.75	0.68	0.71	219
3-df	0.52	0.65	0.58	23
4-mel	0.72	0.48	0.58	223
5-nv	0.86	0.97	0.91	1341
6-vasc	0.96	0.86	0.91	28
accuracy			0.84	2003
macro avg	0.79	0.65	0.69	2003
weighted avg	0.83	0.84	0.82	2003

Çizelge 4.19. Resnet50 modelin Overlapped 'de karışıklık matrisi ve sınıflandırma metrikleri

Overlapped									
Confusion Matrix	Overlapped Confusion Matrix								
	True Label	Actinic keratoses	137.0	34.0	68.0	15.0	35.0	38.0	0.0
		Basal cell carcinoma	18.0	335.0	40.0	9.0	23.0	88.0	1.0
		Benign keratosis	16.0	23.0	780.0	10.0	61.0	208.0	1.0
		Dermatofibroma	1.0	9.0	5.0	75.0	3.0	21.0	1.0
		Melanoma	8.0	7.0	134.0	4.0	574.0	385.0	1.0
		Melanocytic nevi	2.0	21.0	155.0	9.0	141.0	6372.0	5.0
		Vascular lesions	0.0	5.0	2.0	0.0	1.0	25.0	109.0
			Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions
Predicted Label									
	Precision	Recall	Specificity	F1_Score					
0	0.752747	0.418960	0.995355	0.538310					
1	0.771889	0.651751	0.989580	0.706751					
2	0.658784	0.709736	0.954688	0.683311					
3	0.614754	0.652174	0.995253	0.632911					
4	0.684964	0.515723	0.970344	0.588416					
5	0.892812	0.950336	0.768882	0.920676					
6	0.923729	0.767606	0.999088	0.838462					
Ortalama Precision	%75.70970936824085								
Ortalama Recall	%66.66122469880246								
Ortalama Specificity	%95.33128842957692								
Ortalama F1_Score	%70.12626043287834								

Çizelge 4.20. Resnet50 modelin 5 Fold CV Grafik Sonuçları

ResNet50 Grafik Sonuçları	
Accuracy	<p>Accuracies vs Epochs</p> <p>Legend: Train Accuracy Fold 1 (purple), Train Accuracy Fold 2 (cyan), Train Accuracy Fold 3 (green), Train Accuracy Fold 4 (red), Train Accuracy Fold 5 (black).</p>
Loss	<p>Losses vs Epochs</p> <p>Legend: Train Loss Fold 1 (purple), Train Loss Fold 2 (cyan), Train Loss Fold 3 (green), Train Loss Fold 4 (red), Train Loss Fold 5 (black).</p>
Val_Accuracy	<p>Val Accuracies vs Epochs</p> <p>Legend: Val Accuracy Fold 1 (purple), Val Accuracy Fold 2 (cyan), Val Accuracy Fold 3 (green), Val Accuracy Fold 4 (red), Val Accuracy Fold 5 (black).</p>
Val_Loss	<p>Val Losses vs Epochs</p> <p>Legend: Val Loss Fold 1 (purple), Val Loss Fold 2 (cyan), Val Loss Fold 3 (green), Val Loss Fold 4 (red), Val Loss Fold 5 (black).</p>

Çizelge 4.21. DenseNet201 modelin 5 Fold CV başarı sonuçları

Katlama başına puan	Accuracy	Loss
> Fold 1	%86.32051944732666	0.5409519076347351
> Fold 2	%87.66849637031555	0.5484282970428467
> Fold 3	%84.42336320877075	0.6612337827682495
> Fold 4	%87.16924786567688	0.6074349880218506
> Fold 5	%87.71842122077942	0.6053364872932434
> Ortalama	%86.66000962257385	0.592677092552185

Katlama başına puan DenseNet201 modelinde Fold 5 'de en yüksek test başarı oranını elde etmiştir.

Çizelge 4.22. DenseNet201 modelin Fold-1 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-1										
Confusion Matrix		Confusion Matrix								
		True Label	Actinic keratoses	25.0	4.0	29.0	0.0	5.0	2.0	0.0
			Basal cell carcinoma	2.0	72.0	18.0	0.0	4.0	6.0	1.0
			Benign keratosis	2.0	3.0	190.0	0.0	9.0	16.0	0.0
			Dermatofibroma	1.0	0.0	1.0	18.0	2.0	1.0	0.0
			Melanoma	2.0	1.0	32.0	1.0	145.0	42.0	0.0
			Melanocytic nevi	0.0	6.0	42.0	4.0	35.0	1253.0	1.0
			Vascular lesions	0.0	0.0	0.0	0.0	0.0	2.0	26.0
				Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions
		Predicted Label								
Classification Report										
		precision		recall		f1-score		support		
	0-akiec	0.78		0.38		0.52		65		
	1-bcc	0.84		0.70		0.76		103		
	2-bkl	0.61		0.86		0.71		220		
	3-df	0.78		0.78		0.78		23		
	4-mel	0.72		0.65		0.69		223		
	5-nv	0.95		0.93		0.94		1341		
	6-vasc	0.93		0.93		0.93		28		
	accuracy					0.86		2003		
	macro avg	0.80		0.75		0.76		2003		
	weighted avg	0.87		0.86		0.86		2003		

Çizelge 4.23. DenseNet201 modelin Fold-2 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-2								
Confusion Matrix	Confusion Matrix							
	Actinic keratoses	34.0	6.0	11.0	0.0	7.0	7.0	0.0
	Basal cell carcinoma	4.0	76.0	7.0	3.0	3.0	9.0	1.0
	Benign keratosis	4.0	4.0	161.0	0.0	18.0	33.0	0.0
	Dermatofibroma	0.0	2.0	0.0	17.0	1.0	3.0	0.0
	Melanoma	3.0	2.0	11.0	1.0	143.0	63.0	0.0
	Melanocytic nevi	2.0	3.0	8.0	2.0	27.0	1299.0	0.0
	Vascular lesions	0.0	0.0	1.0	0.0	1.0	0.0	26.0
	True Label							
	Predicted Label							
Classification Report								
		precision		recall		f1-score	support	
	0-akiec	0.72		0.52		0.61	65	
	1-bcc	0.82		0.74		0.78	103	
	2-bkl	0.81		0.73		0.77	220	
	3-df	0.74		0.74		0.74	23	
	4-mel	0.71		0.64		0.68	223	
	5-nv	0.92		0.97		0.94	1341	
	6-vasc	0.96		0.93		0.95	28	
	accuracy					0.88	2003	
	macro avg	0.81		0.75		0.78	2003	
	weighted avg	0.87		0.88		0.87	2003	

Çizelge 4.24. DenseNet201 modelin Fold-3 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-3

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

40.0

5.0

8.0

0.0

10.0

2.0

0.0

7.0

64.0

17.0

1.0

5.0

8.0

1.0

4.0

2.0

176.0

4.0

12.0

22.0

0.0

0.0

1.0

0.0

18.0

1.0

3.0

0.0

2.0

1.0

36.0

1.0

138.0

44.0

0.0

1.0

2.0

50.0

3.0

55.0

1230.0

0.0

0.0

0.0

0.0

0.0

0.0

4.0

25.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

0-1200

0

Classification Report

precision

recall

f1-score

support

0-akiec

0.74

0.62

0.67

65

1-bcc

0.85

0.62

0.72

103

2-bkl

0.61

0.80

0.69

220

3-df

0.67

0.78

0.72

23

4-mel

0.62

0.62

0.62

222

5-nv

0.94

0.92

0.93

1341

6-vasc

0.96

0.86

0.91

29

accuracy

0.84

2003

macro avg

0.77

0.75

0.75

2003

weighted avg

0.85

0.84

0.85

2003

Çizelge 4.25. DenseNet201 modelin Fold-4 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-4

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

28.0

5.0

18.0

2.0

9.0

4.0

0.0

5.0

68.0

13.0

1.0

4.0

11.0

0.0

1.0

2.0

168.0

1.0

14.0

34.0

0.0

0.0

0.0

0.0

20.0

1.0

2.0

0.0

0.0

1.0

18.0

0.0

146.0

56.0

1.0

1.0

2.0

18.0

3.0

22.0

1295.0

0.0

0.0

3.0

0.0

0.0

1.0

4.0

21.0

0

200

400

600

800

1000

1200

Classification Report

precision

recall

f1-score

support

0-akiec

0.80

0.42

0.55

66

1-bcc

0.84

0.67

0.74

102

2-bkl

0.71

0.76

0.74

220

3-df

0.74

0.87

0.80

23

4-mel

0.74

0.66

0.70

222

5-nv

0.92

0.97

0.94

1341

6-vasc

0.95

0.72

0.82

29

accuracy

0.87

2003

macro avg

0.82

0.72

0.76

2003

weighted avg

0.87

0.87

0.87

2003

Çizelge 4.26. DenseNet201 modelin Fold-5 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-5

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Confusion Matrix

28.0	9.0	16.0	2.0	8.0	3.0	0.0
4.0	74.0	11.0	0.0	1.0	13.0	0.0
4.0	4.0	178.0	0.0	12.0	21.0	0.0
0.0	2.0	0.0	20.0	0.0	1.0	0.0
1.0	3.0	14.0	2.0	136.0	67.0	0.0
1.0	3.0	22.0	4.0	14.0	1294.0	3.0
0.0	0.0	0.0	0.0	1.0	0.0	27.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

	precision	recall	f1-score	support
0-akiec	0.74	0.42	0.54	66
1-bcc	0.78	0.72	0.75	103
2-bkl	0.74	0.81	0.77	219
3-df	0.71	0.87	0.78	23
4-mel	0.79	0.61	0.69	223
5-nv	0.92	0.96	0.94	1341
6-vasc	0.90	0.96	0.93	28
accuracy			0.88	2003
macro avg	0.80	0.77	0.77	2003
weighted avg	0.87	0.88	0.87	2003

Çizelge 4.27. DenseNet201 modelin Overlapped 'de karışıklık matrisi ve sınıflandırma metrikleri

Overlapped									
Confusion Matrix	Overlapped Confusion Matrix								
	True Label	Actinic keratoses	155.0	29.0	82.0	4.0	39.0	18.0	0.0
		Basal cell carcinoma	22.0	354.0	66.0	5.0	17.0	47.0	3.0
		Benign keratosis	15.0	15.0	873.0	5.0	65.0	126.0	0.0
		Dermatofibroma	1.0	5.0	1.0	93.0	5.0	10.0	0.0
		Melanoma	8.0	8.0	111.0	5.0	708.0	272.0	1.0
		Melanocytic nevi	5.0	16.0	140.0	16.0	153.0	6371.0	4.0
		Vascular lesions	0.0	3.0	1.0	0.0	3.0	10.0	125.0
			Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions
Predicted Label									
	Precision	Recall	Specificity	F1_Score					
0	0.752427	0.474006	0.994736	0.581614					
1	0.823256	0.688716	0.992001	0.750000					
2	0.685243	0.794359	0.955025	0.735777					
3	0.726562	0.808696	0.996465	0.765432					
4	0.715152	0.636119	0.968322	0.673324					
5	0.929530	0.950186	0.854079	0.939745					
6	0.939850	0.880282	0.999190	0.909091					
Ortalama Precision	%79.60028810105609								
Ortalama Recall	%74.74804208551332								
Ortalama Specificity	%96.56879862013329								
Ortalama F1_Score	%76.49975221696935								

Çizelge 4.28. DenseNet201 modelin 5 Fold CV Grafik Sonuçları

DenseNet201 Grafik Sonuçları	
Accuracy	<p>Accuracies vs Epochs</p> <p>Legend: Train Accuracy Fold 1 (magenta), Train Accuracy Fold 2 (cyan), Train Accuracy Fold 3 (green), Train Accuracy Fold 4 (red), Train Accuracy Fold 5 (black).</p>
Loss	<p>Losses vs Epochs</p> <p>Legend: Train Loss Fold 1 (magenta), Train Loss Fold 2 (cyan), Train Loss Fold 3 (green), Train Loss Fold 4 (red), Train Loss Fold 5 (black).</p>
Val_Accuracy	<p>Val Accuracies vs Epochs</p> <p>Legend: Val Accuracy Fold 1 (magenta), Val Accuracy Fold 2 (cyan), Val Accuracy Fold 3 (green), Val Accuracy Fold 4 (red), Val Accuracy Fold 5 (black).</p>
Val_Loss	<p>Val Losses vs Epochs</p> <p>Legend: Val Loss Fold 1 (magenta), Val Loss Fold 2 (cyan), Val Loss Fold 3 (green), Val Loss Fold 4 (red), Val Loss Fold 5 (black).</p>

Çizelge 4.29. InceptionV3 modelin 5 Fold CV başarı sonuçları

Katlama başına puan	Accuracy	Loss
> Fold 1	%85.67149043083191	0.5004254579544067
> Fold 2	%86.8696928024292	0.5192919969558716
> Fold 3	%85.3719413280487	0.6191869974136353
> Fold 4	%85.12231707572937	0.6050950884819031
> Fold 5	%85.82127094268799	0.5546700358390808
> Ortalama	%85.77134251594543	0.5597339153289795

Katlama başına puan InceptionV3 modelinde Fold 2 'de en yüksek test başarı oranını elde etmiştir.

Çizelge 4.30. InceptionV3 modelin Fold-1 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-1

Confusion Matrix

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Confusion Matrix

35.0	3.0	16.0	3.0	4.0	4.0	0.0
2.0	77.0	9.0	4.0	5.0	5.0	1.0
9.0	2.0	164.0	2.0	15.0	28.0	0.0
0.0	1.0	4.0	17.0	0.0	1.0	0.0
3.0	1.0	27.0	1.0	125.0	66.0	0.0
0.0	5.0	26.0	1.0	33.0	1275.0	1.0
0.0	1.0	0.0	0.0	3.0	1.0	23.0

1200

1000

800

600

400

200

0

Classification Report

	precision	recall	f1-score	support
0-akiec	0.71	0.54	0.61	65
1-bcc	0.86	0.75	0.80	103
2-bkl	0.67	0.75	0.70	220
3-df	0.61	0.74	0.67	23
4-mel	0.68	0.56	0.61	223
5-nv	0.92	0.95	0.94	1341
6-vasc	0.92	0.82	0.87	28
accuracy			0.86	2003
macro avg	0.77	0.73	0.74	2003
weighted avg	0.85	0.86	0.85	2003

Çizelge 4.31. InceptionV3 modelin Fold-2 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-2

Confusion Matrix

		Confusion Matrix							<div><div></div><div>1200</div><div>1000</div><div>800</div><div>600</div><div>400</div><div>200</div><div>0</div></div>
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
True Label	Actinic keratoses	37.0	4.0	8.0	2.0	5.0	9.0	0.0	
	Basal cell carcinoma	5.0	72.0	12.0	2.0	4.0	8.0	0.0	
	Benign keratosis	1.0	4.0	166.0	0.0	14.0	35.0	0.0	
	Dermatofibroma	1.0	3.0	1.0	15.0	0.0	3.0	0.0	
	Melanoma	2.0	1.0	17.0	1.0	137.0	63.0	2.0	
	Melanocytic nevi	4.0	3.0	15.0	0.0	31.0	1288.0	0.0	
	Vascular lesions	0.0	0.0	0.0	0.0	2.0	1.0	25.0	
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
		Predicted Label							

Classification Report

	precision	recall	f1-score	support
0-akiec	0.74	0.57	0.64	65
1-bcc	0.83	0.70	0.76	103
2-bkl	0.76	0.75	0.76	220
3-df	0.75	0.65	0.70	23
4-mel	0.71	0.61	0.66	223
5-nv	0.92	0.96	0.94	1341
6-vasc	0.93	0.89	0.91	28
accuracy			0.87	2003
macro avg	0.80	0.73	0.77	2003
weighted avg	0.86	0.87	0.86	2003

Çizelge 4.32. InceptionV3 modelin Fold-3 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-3

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

46.0

7.0

5.0

2.0

8.0

2.0

0.0

8.0

65.0

2.0

0.0

0.0

0.0

0.0

5.0

10.0

168.0

3.0

39.0

34.0

0.0

1.0

2.0

3.0

16.0

1.0

5.0

0.0

4.0

3.0

10.0

1.0

118.0

26.0

1.0

1.0

14.0

31.0

4.0

56.0

1273.0

4.0

0.0

2.0

1.0

0.0

0.0

1.0

24.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

01200

Classification Report

precision

recall

f1-score

support

0-akiec

0.66

0.71

0.68

65

1-bcc

0.87

0.63

0.73

103

2-bkl

0.66

0.76

0.71

220

3-df

0.57

0.70

0.63

23

4-mel

0.72

0.53

0.61

222

5-nv

0.92

0.95

0.93

1341

6-vasc

0.86

0.83

0.84

29

accuracy

0.85

2003

macro avg

0.75

0.73

0.73

2003

weighted avg

0.85

0.85

0.85

2003

Çizelge 4.33. InceptionV3 modelin Fold-4 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-4

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

29.0

5.0

21.0

1.0

5.0

5.0

0.0

1.0

77.0

9.0

1.0

1.0

12.0

1.0

2.0

4.0

170.0

1.0

7.0

36.0

0.0

0.0

1.0

2.0

16.0

1.0

3.0

0.0

2.0

0.0

25.0

1.0

102.0

91.0

1.0

0.0

2.0

27.0

2.0

19.0

1291.0

0.0

0.0

1.0

0.0

0.0

0.0

8.0

20.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

1200

1000

800

600

400

200

0

Predicted Label

Classification Report

precision

recall

f1-score

support

0-akiec

0.85

0.44

0.58

66

1-bcc

0.86

0.75

0.80

102

2-bkl

0.67

0.77

0.72

220

3-df

0.73

0.70

0.71

23

4-mel

0.76

0.46

0.57

222

5-nv

0.89

0.96

0.93

1341

6-vasc

0.91

0.69

0.78

29

accuracy

0.85

2003

macro avg

0.81

0.68

0.73

2003

weighted avg

0.85

0.85

0.84

2003

Çizelge 4.34. InceptionV3 modelin Fold-5 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-5

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

31.0

5.0

17.0

1.0

7.0

5.0

0.0

3.0

74.0

9.0

3.0

4.0

9.0

1.0

2.0

0.0

169.0

0.0

21.0

27.0

0.0

2.0

2.0

2.0

12.0

0.0

5.0

0.0

3.0

1.0

13.0

1.0

141.0

64.0

0.0

2.0

4.0

30.0

3.0

33.0

1267.0

2.0

0.0

0.0

0.0

0.0

0.0

3.0

25.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

precision

recall

f1-score

support

0-akiec

0.72

0.47

0.57

66

1-bcc

0.86

0.72

0.78

103

2-bkl

0.70

0.77

0.74

219

3-df

0.60

0.52

0.56

23

4-mel

0.68

0.63

0.66

223

5-nv

0.92

0.94

0.93

1341

6-vasc

0.89

0.89

0.89

28

accuracy

0.86

2003

macro avg

0.77

0.71

0.73

2003

weighted avg

0.86

0.86

0.86

2003

Çizelge 4.35. InceptionV3 modelin Overlapped 'de karışıklık matrisi ve sınıflandırma metrikleri

Overlapped									
Confusion Matrix	Overlapped Confusion Matrix								
	True Label	Actinic keratoses	178.0	25.0	67.0	8.0	25.0	24.0	0.0
		Basal cell carcinoma	18.0	365.0	49.0	12.0	17.0	48.0	5.0
		Benign keratosis	19.0	12.0	837.0	6.0	67.0	157.0	1.0
		Dermatofibroma	5.0	7.0	9.0	76.0	2.0	16.0	0.0
		Melanoma	18.0	3.0	121.0	5.0	623.0	340.0	3.0
		Melanocytic nevi	8.0	14.0	132.0	11.0	142.0	6394.0	4.0
		Vascular lesions	0.0	2.0	0.0	0.0	6.0	17.0	117.0
		Predicted Label	Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions
	Precision	Recall	Specificity	F1_Score					
0	0.723577	0.544343	0.992981	0.621291					
1	0.852804	0.710117	0.993369	0.774947					
2	0.688889	0.761601	0.957604	0.723423					
3	0.644068	0.660870	0.995758	0.652361					
4	0.706349	0.559748	0.970905	0.624561					
5	0.913951	0.953617	0.818127	0.933363					
6	0.900000	0.823944	0.998683	0.860294					
Ortalama Precision	%77.56625278547926								
Ortalama Recall	%71.63198648351706								
Ortalama Specificity	%96.10610842072257								
Ortalama F1_Score	%74.14627970273103								

Çizelge 4.36. InceptionV3 modelin 5 Fold CV Grafik Sonuçları

InceptionV3 Grafik Sonuçları	
Accuracy	
Loss	
Val_Accuracy	
Val_Loss	

Çizelge 4.37. VGG16 modelin 5 Fold CV başarı sonuçları

Katlama başına puan	Accuracy	Loss
> Fold 1	%82.82576203346252	0.5141875743865967
> Fold 2	%83.52471590042114	0.4474634528160095
> Fold 3	%82.02695846557617	0.5228338837623596
> Fold 4	%82.37643241882324	0.5736400485038757
> Fold 5	%80.77883124351501	0.5533392429351807
> Ortalama	%82.30654001235962	0.5222928404808045

Katlama başına puan VGG16 modelinde Fold 2 'de en yüksek test başarı oranını elde etmiştir.

Çizelge 4.38. VGG16 modelin Fold-1 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-1

Confusion Matrix

Confusion Matrix

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

21.0

6.0

6.0

2.0

1.0

0.0

0.0

6.0

74.0

5.0

0.0

2.0

13.0

2.0

20.0

10.0

171.0

6.0

25.0

54.0

0.0

0.0

1.0

0.0

9.0

0.0

1.0

0.0

14.0

6.0

20.0

5.0

144.0

53.0

1.0

4.0

6.0

18.0

1.0

50.0

1217.0

2.0

0.0

0.0

0.0

0.0

1.0

3.0

23.0

0

200

400

600

800

1000

1200

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

precision

recall

f1-score

support

0-akiec

0.58

0.32

0.42

65

1-bcc

0.73

0.72

0.72

103

2-bkl

0.60

0.78

0.68

220

3-df

0.82

0.39

0.53

23

4-mel

0.59

0.65

0.62

223

5-nv

0.94

0.91

0.92

1341

6-vasc

0.85

0.82

0.84

28

accuracy

0.83

2003

macro avg

0.73

0.65

0.67

2003

weighted avg

0.84

0.83

0.83

2003

Çizelge 4.39. VGG16 modelin Fold-2 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-2

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Confusion Matrix

24.0	5.0	13.0	4.0	7.0	12.0	0.0
1.0	75.0	8.0	4.0	5.0	10.0	0.0
2.0	6.0	157.0	0.0	17.0	38.0	0.0
0.0	2.0	3.0	15.0	1.0	2.0	0.0
2.0	2.0	22.0	2.0	127.0	67.0	1.0
1.0	7.0	40.0	0.0	41.0	1252.0	0.0
0.0	0.0	0.0	0.0	2.0	3.0	23.0

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

Predicted Label

Classification Report

	precision	recall	f1-score	support
0-akiec	0.80	0.37	0.51	65
1-bcc	0.77	0.73	0.75	103
2-bkl	0.65	0.71	0.68	220
3-df	0.60	0.65	0.63	23
4-mel	0.64	0.57	0.60	223
5-nv	0.90	0.93	0.92	1341
6-vasc	0.96	0.82	0.88	28
accuracy			0.84	2003
macro avg	0.76	0.68	0.71	2003
weighted avg	0.83	0.84	0.83	2003

Çizelge 4.40. VGG16 modelin Fold-3 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-3

Confusion Matrix

Confusion Matrix

True Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

28.0

5.0

1.0

1.0

1.0

0.0

0.0

10.0

64.0

3.0

1.0

0.0

8.0

0.0

12.0

16.0

146.0

3.0

25.0

50.0

0.0

0.0

0.0

1.0

9.0

0.0

1.0

0.0

9.0

5.0

25.0

3.0

120.0

33.0

0.0

5.0

12.0

43.0

6.0

74.0

1249.0

2.0

1.0

1.0

1.0

0.0

2.0

0.0

27.0

Predicted Label

Actinic keratoses

Basal cell carcinoma

Benign keratosis

Dermatofibroma

Melanoma

Melanocytic nevi

Vascular lesions

1200

1000

800

600

400

200

0

Classification Report

precision

recall

f1-score

support

0-akiec

0.78

0.43

0.55

65

1-bcc

0.74

0.62

0.68

103

2-bkl

0.58

0.66

0.62

220

3-df

0.82

0.39

0.53

23

4-mel

0.62

0.54

0.58

222

5-nv

0.90

0.93

0.91

1341

6-vasc

0.84

0.93

0.89

29

accuracy

0.82

2003

macro avg

0.75

0.64

0.68

2003

weighted avg

0.82

0.82

0.82

2003

Çizelge 4.41. VGG16 modelin Fold-4 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-4

Confusion Matrix

		Confusion Matrix							<div><div></div><div>1200</div><div>1000</div><div>800</div><div>600</div><div>400</div><div>200</div><div>0</div></div>
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
True Label	Actinic keratoses	19.0	12.0	23.0	0.0	5.0	7.0	0.0	
	Basal cell carcinoma	3.0	79.0	7.0	0.0	3.0	10.0	0.0	
	Benign keratosis	2.0	5.0	145.0	2.0	12.0	54.0	0.0	
	Dermatofibroma	0.0	2.0	1.0	12.0	1.0	7.0	0.0	
	Melanoma	0.0	10.0	31.0	0.0	88.0	92.0	1.0	
	Melanocytic nevi	1.0	16.0	16.0	1.0	23.0	1282.0	2.0	
	Vascular lesions	0.0	2.0	1.0	0.0	1.0	0.0	25.0	
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	
		Predicted Label							

Classification Report

	precision	recall	f1-score	support
0-akiec	0.76	0.29	0.42	66
1-bcc	0.63	0.77	0.69	102
2-bkl	0.65	0.66	0.65	220
3-df	0.80	0.52	0.63	23
4-mel	0.66	0.40	0.50	222
5-nv	0.88	0.96	0.92	1341
6-vasc	0.89	0.86	0.88	29
accuracy			0.82	2003
macro avg	0.75	0.64	0.67	2003
weighted avg	0.81	0.82	0.81	2003

Çizelge 4.42. VGG16 modelin Fold-5 'de karışıklık matrisi ve sınıflandırma metrikleri

FOLD-5

Confusion Matrix

		Confusion Matrix							
True Label	Actinic keratoses	34.0	3.0	25.0	0.0	2.0	2.0	0.0	<div><div></div><div>1000</div><div>800</div><div>600</div><div>400</div><div>200</div><div>0</div></div>
	Basal cell carcinoma	14.0	64.0	10.0	1.0	4.0	10.0	0.0	
	Benign keratosis	13.0	0.0	165.0	0.0	15.0	26.0	0.0	
	Dermatofibroma	6.0	1.0	2.0	11.0	3.0	0.0	0.0	
	Melanoma	9.0	3.0	32.0	1.0	146.0	30.0	2.0	
	Melanocytic nevi	11.0	9.0	45.0	0.0	95.0	1174.0	7.0	
	Vascular lesions	0.0	2.0	0.0	0.0	1.0	1.0	24.0	
	Predicted Label								
		Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions	

Classification Report

	precision	recall	f1-score	support
0-akiec	0.39	0.52	0.44	66
1-bcc	0.78	0.62	0.69	103
2-bkl	0.59	0.75	0.66	219
3-df	0.85	0.48	0.61	23
4-mel	0.55	0.65	0.60	223
5-nv	0.94	0.88	0.91	1341
6-vasc	0.73	0.86	0.79	28
accuracy			0.81	2003
macro avg	0.69	0.68	0.67	2003
weighted avg	0.83	0.81	0.82	2003

Çizelge 4.43. VGG16 modelin Overlapped 'de karışıklık matrisi ve sınıflandırma metrikleri

Overlapped									
Confusion Matrix	Overlapped Confusion Matrix								
	True Label	Actinic keratoses	126.0	36.0	93.0	4.0	37.0	30.0	1.0
		Basal cell carcinoma	29.0	356.0	51.0	6.0	23.0	48.0	1.0
		Benign keratosis	24.0	19.0	784.0	3.0	89.0	179.0	1.0
		Dermatofibroma	9.0	6.0	15.0	56.0	13.0	16.0	0.0
		Melanoma	13.0	17.0	135.0	3.0	625.0	313.0	7.0
		Melanocytic nevi	13.0	53.0	205.0	3.0	245.0	6174.0	12.0
		Vascular lesions	0.0	6.0	1.0	0.0	5.0	8.0	122.0
			Actinic keratoses	Basal cell carcinoma	Benign keratosis	Dermatofibroma	Melanoma	Melanocytic nevi	Vascular lesions
	Predicted Label								

Çizelge 4.44. VGG16 modelin 5 Fold CV Grafik Sonuçları

VGG16 Grafik Sonuçları	
Accuracy	<p>Accuracies vs Epochs</p> <p>Legend: Train Accuracy Fold 1 (purple), Train Accuracy Fold 2 (cyan), Train Accuracy Fold 3 (green), Train Accuracy Fold 4 (red), Train Accuracy Fold 5 (black).</p>
Loss	<p>Losses vs Epochs</p> <p>Legend: Train Loss Fold 1 (purple), Train Loss Fold 2 (cyan), Train Loss Fold 3 (green), Train Loss Fold 4 (red), Train Loss Fold 5 (black).</p>
Val_Accuracy	<p>Val Accuracies vs Epochs</p> <p>Legend: Val Accuracy Fold 1 (purple), Val Accuracy Fold 2 (cyan), Val Accuracy Fold 3 (green), Val Accuracy Fold 4 (red), Val Accuracy Fold 5 (black).</p>
Val_Loss	<p>Val Losses vs Epochs</p> <p>Legend: Val Loss Fold 1 (purple), Val Loss Fold 2 (cyan), Val Loss Fold 3 (green), Val Loss Fold 4 (red), Val Loss Fold 5 (black).</p>

Çizelge 4.45 Literatürdeki diğer çalışmalar ile karşılaştırma

Çalışma	Yöntem	Accuracy Değeri
CHATURVEDİ, GUPTA ve PRASAD (2019) [3]	MobileNet	%83.1
ÇEVİK ve ZENGİN (2019) [4]	VGGNET-16	%85.62
ALI (2019) [5]	MobileNet	% 90
FIRILDAK, ÇELİK ve TALU (2020) [2]	AlexNet	% 93,14
RAHMAN ve AMİ (2021) [6]	DenseNet, Xception, ResNet kullanarak üç modelin birleşimi	%85.3
Bizim Çalışma	DenseNet201	%88.82

HAM10000 veri seti üzerinde iki farklı değerlendirme yöntemi kullanılarak sınıflandırıcı modeller oluşturulmuştur. Çizelge 4.45. 'de görüldüğü gibi diğer çalışmaların doğruluk oranları belirtilmiştir.

Derin öğrenme yaklaşımlarında belirli bir yapının olmaması, optimize edilmesi gereken parametrelerin fazla olması, katman sayısı ve katmanlardaki nöron sayılarının değişkenliği sebebiyle model geliştirmek zaman alan bir süreçtir. Literatürde görüntü üzerine MobileNet, VGGNet-16, AlexNet, DenseNet, Xception ve ResNet gibi popüler modeller kullanılarak oldukça yüksek başarımlı sınıflamalar gerçekleştirilse de, bu modellerin daha az kompleks olarak oluşturulması ve modellenmesi günümüzde üzerine yoğunlaşmış en temel araştırma alanlarıdır.

Bizim çalışmada birden fazla model gerçekleştirildiği için ulaşılan en yüksek sonucu veren DenseNet201 ağının hold-out yöntemi üzerinde herhangi bir ön işleme teknikleri olmadan ham veriler kullanılarak başarı oranı %88.82 olarak elde etmiştir.

5. SONUÇLAR VE ÖNERİLER

Özetle, sınıflandırma çalışmalarında karşılaşılan özel durumlar ve model seçimi yapılırken referans alınan metrikler dikkate alınarak sınıflandırma başarımının doğru bir şekilde değerlendirilebilmesi yönünde deneysel çalışmalar yapılmıştır. Doğruluk metriğinin yanı sıra kesinlik, geri çağırma, özgüllük ve f1 skor gibi metriklerinde kullanılması gerektiğinin önemine dikkat çekilmiştir. Çalışmada ayrıca sınıf dengesizliği ve sınıf sayısının fazla olduğu durumlarda modelin başarımının nasıl değerlendirileceği hakkında incelemeler yapılmıştır. Genel olarak incelendiğinde üç farklı durum dikkat çekmiştir.

İlk olarak, sınıflandırma modeli oluşturulurken kullanılan test tekniğinin önemi yapılan çalışmada ön plana çıkmaktadır. Sınıflandırıcının performansının doğru olarak değerlendirilebilmesi için mutlaka tüm verilerin test ve eğitim aşamasında kullanılması gerçeği ortaya çıkmaktadır. Bu nedenle mümkün olduğu sürece Çapraz doğrulama testlerinin uygulanması gerektiği tespit edilmiştir. Hold-Out gibi yöntemler kullanıldığı durumlarda daha yüksek doğruluk değerleri elde edilebilse de, eğer çapraz doğrulama testi veri setine uygulanabiliyor ise buradan elde edilen doğruluk değerinin gerçek sonucu yansıttığı ve bu değer kullanılması gerektiği sonucuna varılmıştır. Sadece çapraz doğrulama testinin uygulanamadığı durumlarda Hold-Out yönteminin kullanılması önerilmektedir.

İkinci olarak, veri setlerindeki sınıf sayısının da sonuçları doğrudan etkilediği görülmektedir. Sınıf sayısının fazla olmasına paralel olarak doğruluk değerinin düştüğü gözlemlenebilmektedir. Ancak, diğer metrikler birlikte değerlendirildiğinde bu düşük başarımın daha az sınıflı veri setlerinden elde edilen yüksek başarımlardan daha olumlu sonuçlar verdiği görülmüştür. Örneğin 7 sınıflı bir veri setinden %80 gibi bir doğruluk değerinin elde edilmesi, 2 sınıflı bir veri setinde %90 gibi bir doğruluk değerinden daha anlamlı ve başarılı olduğu sonucunu ortaya koyabilmektedir.

Son olarak, veri setindeki sınıfların dengesiz olarak dağılımı da doğruluk değerinin değerlendirilmesinde sorunlar ortaya çıkarabildiği sonucuna varılmıştır. Bu durumda, sadece doğruluk metriğinin kullanılması yanlış model seçimine sebep olabilmektedir.

Bu çalışmada yer alan veri seti ile oluşturulan modellerde sadece doğruluk metriğinden faydalanılması rastgele tahminlerde bulunan modellerin daha iyi sonuç verdiği sonucunu yansıtabilmektedir. Bu durumun önüne geçmek için kesinlik, geri çağırma, özgüllük ve f1 skor metriklerinden yararlanılması faydalı olacaktır. Böylece doğru sınıflandırıcın seçilmesi daha kolay hale gelebilecektir.

Sonuç olarak, sınıflandırma çalışmalarında doğru sınıflandırıcının seçilmesinin çok önem arz ettiği ve bu hususta dikkat edilmesi gereken birçok parametrenin olduğu ortaya konulmuştur. Özellikle, veri setinin boyutu, sınıfların dağılımları, sınıf sayısı ve kullanılacak olan test yöntemi ile beraber değerlendirme metriklerinin doğru bir şekilde kullanılmasının, iyi bir sınıflandırma yapılmasında çok önemli parametreler olduğu sonucuna varılmıştır.