

Konvolüsyonel Sinir Ağları ile Sınıflandırma İşlemi

Recep Furkan Koçyiğit

22501048

furkan.kocyigit@std.yildiz.edu.tr

Bilgisayar Mühendisliği Bölümü

Elektrik Elektronik Fakültesi, Yıldız Teknik Üniversitesi

Özet

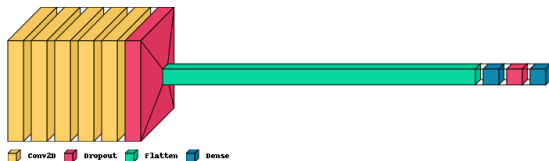
Bu ödevde, içerisinde uçak, otomobil, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon kategorilerinde her kategoride 6.000 adet, toplamda 60.000 adet resim içeren CIFAR-10 veri kümesinde Konvolüsyonel Sinir Ağları (CNN) ile sınıflandırma işlemi yapılmıştır. Ödev Python programlama dili kullanılarak yazılmıştır.

Giriş

Ödev kapsamında aşağıdaki hiper parametreler arasından en iyi başarıyı sağlayan model bulunmaya çalışılmıştır. Eğitim için 40.000, validasyon için 10.000 adet ve test için 10.000 adet veri kullanılmıştır. Veriye bakıldığında düzgün bir dağılım gösterdiği için doğruluk(accuracy) kullanılmıştır.

1. Konvolüsyon Katmanı = [3, 5]
2. Filtre Sayısı = [32, 64]
3. Kernel Boyutu = [3x3, 5x5]
4. Dropout = [Var, Yok]
5. Mini Batch Boyutu = 64
6. Epoch = 10
7. Konvolüsyon Katmanı Aktivasyon Fonksiyonu = ReLU
8. Optimizasyon Algoritması = Adam

En başarılı CNN ağı 5 konvolüsyon katmanı, 64 adet filtre içermekte, kernel boyutu 3x3 ve dropout içeren model olmuştur. Modele Max Pooling işlemi uygulandığında başarının arttığı ve parametre sayısının azalmasından dolayı işlem maliyetinin de azaldığı görülmüştür. Modelin blok diyagramı aşağıdaki gibidir:



Yöntem

Bu sistem aşağıdaki ana modüllerden oluşmaktadır:

1. **Verinin Yüklenmesi:** Verinin Python kütüphanesinden yüklenmesi.
2. **Verinin Ön İşleme Aşamasının Gerçeklenmesi:** 0-255 arasındaki bit değerlerinin normalize edilmesi, çıktıları One-Hot Encode yöntemi uygulanması ve eğitim, validasyon ve test olarak ayrılması.
3. **Model Eğitiminin Görselleştirilmesi:** Eğitilen modelin başarı ve kayıplarının eğitim ve validasyon için gösterilmesi
4. **Modelin Oluşturulması:** CNN i jenerik olarak oluşturmak için metot yazılmıştır. Bu metot girdi olarak katman sayısı, filtre sayısı, kernel boyutu ve dropout bilgisi almaktadır.
5. **Sonuçların yazdırılması:** Eğitimi tamamlanan modellerin karmaşıklık matrislerinin gösterilmesi ve doğruluk(accuracy), f1 skoru, duyarlılık(recall) ve kesinlik(precision) bilgilerini göstermek için yazılmıştır.
6. **Hiper Parametrelere Göre Modellerin Oluşturulması Sonuçların Elde Edilmesi:** Verilen hiper parametrelere göre tüm kombinasyonlarına göre modeller oluşturulur. Modeller eğitilirken ağırlıklar kaydedilir. Modelin eğitim tarihçesi ve sonucu saklanır.
7. **En İyi Modelin Eğitim Kümesi ve Validasyon kümesi ile Eğitilip Başarısının Bulunması:** En iyi başarı oranına sahip modelin hiper parametreleri belirlendikten sonra model oluşturulur ve eğitim ve validasyon kümesinin birleşimi ile tekrar eğitilir. Daha sonra başarısı hesaplanır.

Uygulama

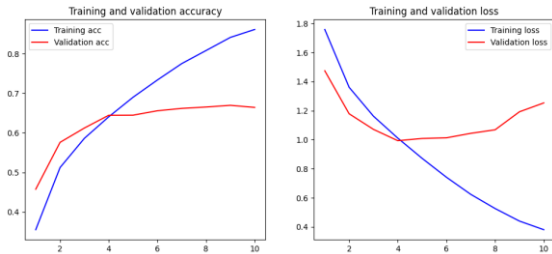
Validasyon verisi ile yapılan testler sonucunda modellerin başarısı aşağıda verilmiştir.

Model No	Katman Sayısı	Filter Sayısı	Kernel Boyutu	Dropout	Doğruluk
1	5	64	3x3	Var	0.6731
2	5	32	3x3	Var	0.6529
3	3	64	3x3	Var	0.6513
4	3	32	3x3	Var	0.6379
5	3	32	5x5	Var	0.6235
6	3	32	5x5	Yok	0.6194
7	3	32	3x3	Yok	0.6154
8	5	64	3x3	Yok	0.6154
9	3	64	5x5	Var	0.6105
10	3	64	3x3	Yok	0.6053
11	5	32	3x3	Yok	0.6046
12	5	64	5x5	Var	0.6035
13	5	32	5x5	Var	0.5926
14	5	32	5x5	Yok	0.5915
15	3	64	5x5	Yok	0.5792
16	5	64	5x5	Yok	0.5249

Tabloya bakıldığında dropout olan modellerin başarısının olmayanlara göre daha fazla olduğunu söylenebilir. Bunun nedeninin modellerin karmaşık olmasından dolayı aşırı öğrenme(overfit) durumunda olması olarak yorumlanabilir.

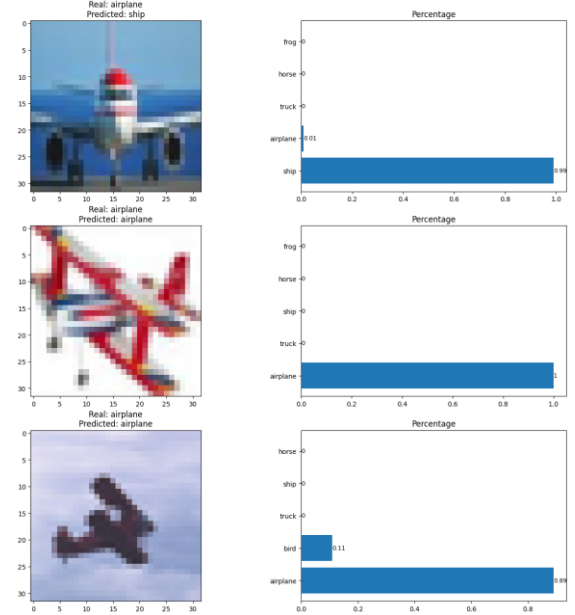
Ayrıca kernel boyutu 5x5 olan modellerin başarısının 3x3 olanlardan düşük olduğunu da gözlemlenmektedir. Burada daha küçük kernel tarafından çıkarılan karmaşık özelliklerin büyük kernel tarafından çıkarılan daha basit özelliklerden etkili olduğu gözlenmektedir.

Aşağıda en başarılı modelin epoch boyunca eğitim ve validasyon doğrulukları ve kayıpları yer almaktadır. 4.Epoch'tan sonra eğitim başarısının arttığını ve validasyon başarısının aynı kaldığı, kayıpta ise eğitim kaybı azalmaya devam ederken validasyonda yükselemeye başladığı görülmektedir. Bu grafiklere bakarak en iyi modelin aşırı öğrendiğini söyleyebiliriz. Dropout kullansak bile regülerizasyon için yeterli olmadığını modelin hala çok karmaşık olduğunu söyleyebiliriz.



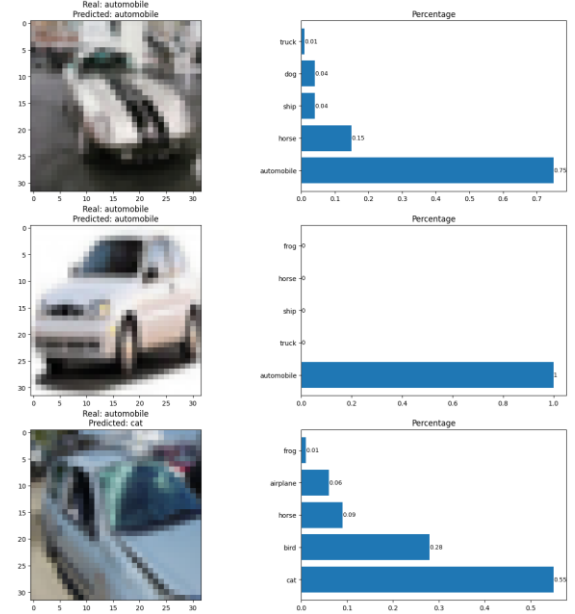
Modelin test verisi için başarısı 0.675 olarak hesaplanmıştır.

Uçak kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



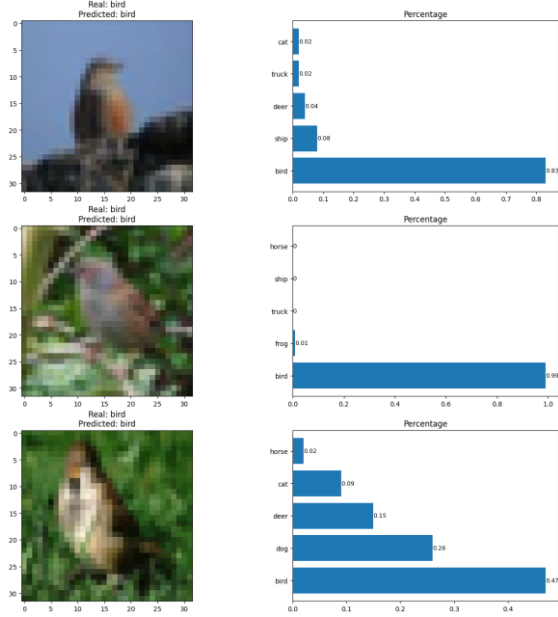
Bu kategorideki üç örnekten ikisini doğru tahmin ettiği ve yanlış bildiği örneği gemi ile karıştırdığı gözlenmektedir.

Otomobil kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



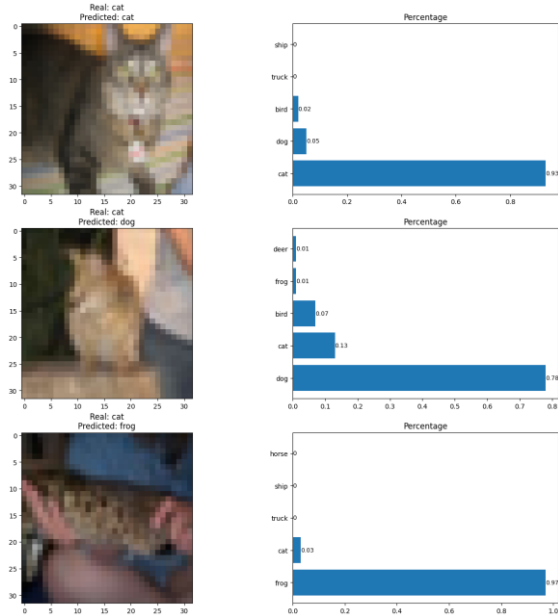
Bu kategoride de üç örnekten ikisini doğru tahmin ettiği ve yanlış bildiği örneği kedi ile karıştırdığı gözlenmektedir. Yanlış bildiği örnek için canlıya ait bir görüntü olarak değerlendirme eğiliminde olduğu gözlenmektedir.

Kuş kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



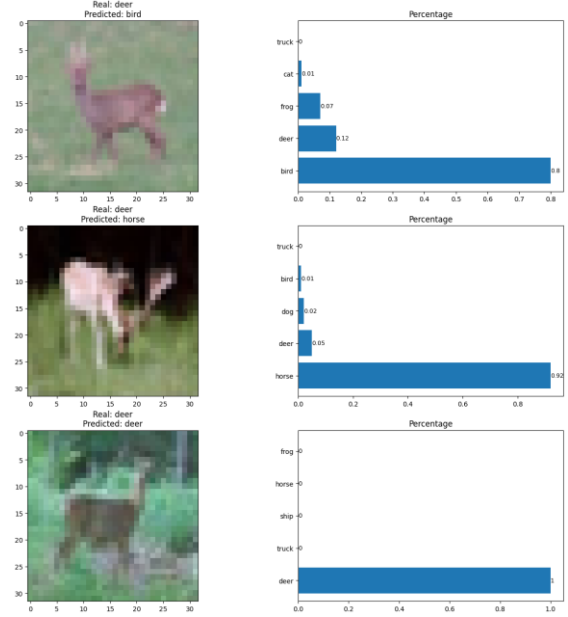
Bu kategoride üç örnekten üçünü de doğru tahmin ettiği gözlenmektedir. Son örnekte doğru tahmin etse bile diğer canlı türlerine ait sınıfların da oranının yüksek olduğu, karıştırma ihtimalinin olduğu gözlenmektedir.

Kuş kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



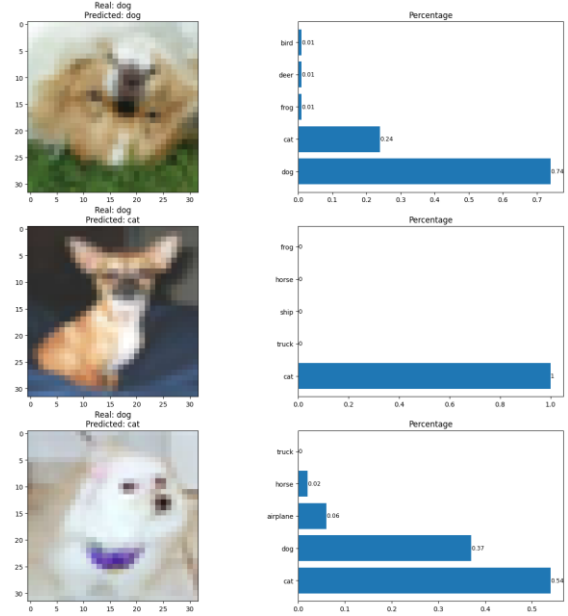
Bu kategoride üç örnekten sadece birini doğru tahmin ettiği gözlenmektedir. İkinci örnekte modelin resim için neredeyse köpek olduğundan emin olduğunu üçüncü resim de ise kurbağa olduğundan emin olduğunu söyleyebiliriz.

Geyik kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



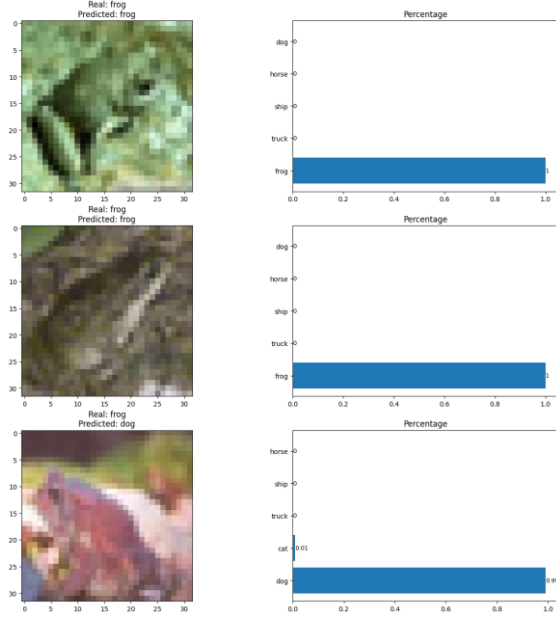
Bu kategoride de üç örnekten sadece birini doğru tahmin ettiği gözlenmektedir. İlk örnekte resmin kuş olduğundan, ikinci resimde ise at olduğundan neredeyse emin olduğunu söyleyebiliriz.

Köpek kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



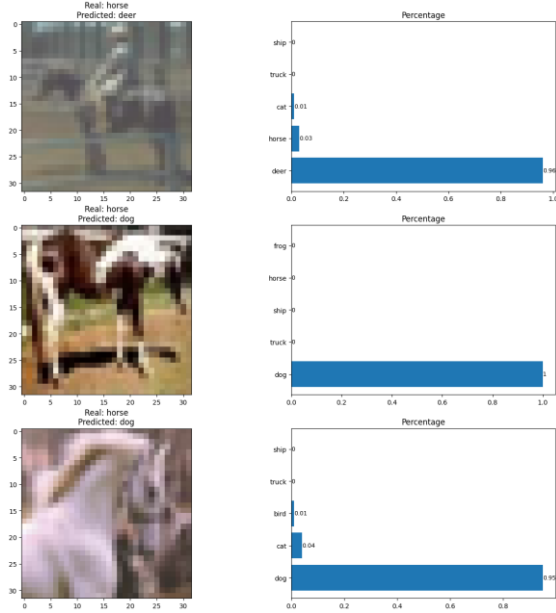
Bu kategoride de üç örnekten sadece birini doğru tahmin ettiği gözlenmektedir. Kedi ile köpeği karıştırma eğiliminde olduğunu söyleyebiliriz.

Kurbağa kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



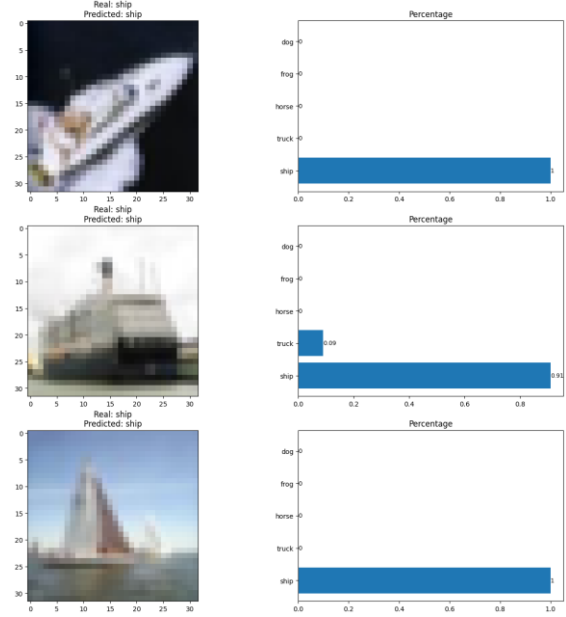
Bu kategoride üç örnekten ikisini doğru tahmin ettiği gözlenmektedir. Karıştırdığı resimde ise ilgili resmin köpek kategorisine ait olduğundan emin olduğunu söyleyebiliriz.

At kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



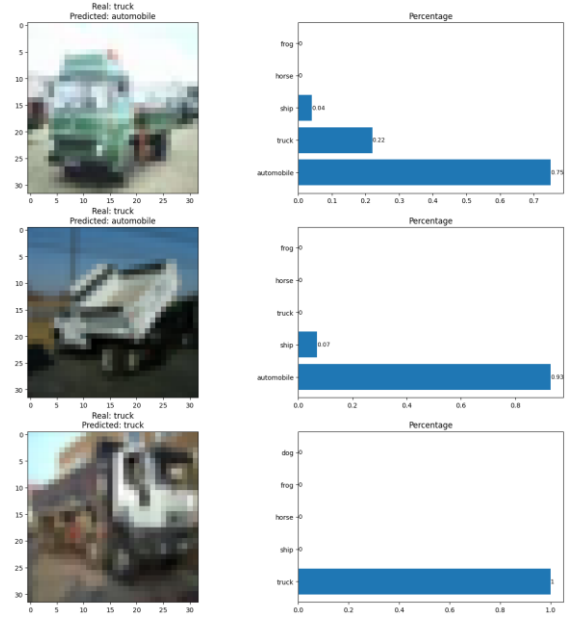
Bu kategoride üç örnekten hiçbirini doğru tahmin edemediği gözlenmektedir. Geayık ve köpek kategorileri ile karıştırdığı görülmektedir.

Gemi kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



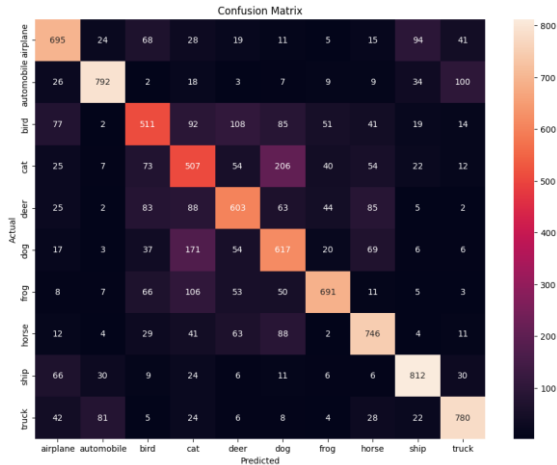
Bu kategoride üç örneğin hepsini doğru tahmin edebildiği görülmektedir.

Kamyon kategorisine ait 3 farklı resim için modelin yaptığı tahminler aşağıdaki gibidir:



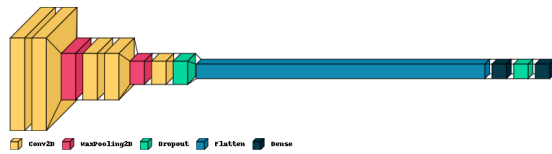
Bu kategoride üç örneğin sadece birini doğru tahmin edebildiği görülmektedir. Yanlış tahmin ettiği iki örnek içinde resimlerin otomobil kategorisine ait olduğunu belirttiği gözlenmektedir.

Aşağıda en iyi başarı oranına sahip modelin test örnekleri için karışıklık matrisi verilmiştir:

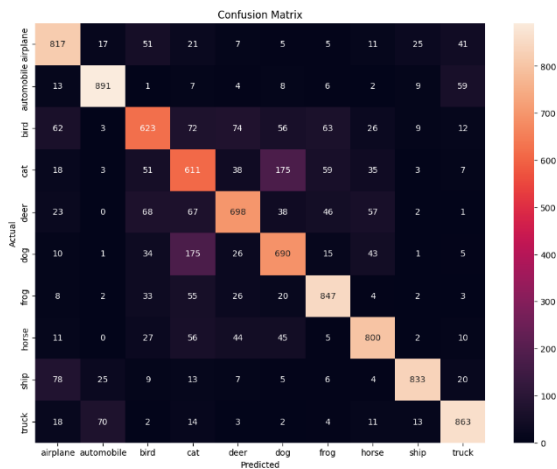


Karmaşıklık matrisine bakıldığında en çok kedi-köpek sınıflarının tahminlerinde yanlış yapıldığı görülmektedir. Hayvan sınıfı tahminlerinde yoğunluk olduğu görülmektedir. Bunun dışında uçak-gemi, otomobil-kamyon sınıflarının tahminlerinde hata yapıldığı görülmektedir.

En iyi CNN modeline her iki konvolüsyon katmanından sonra bir adet Max-Pooling katmanı eklendiğine, başarı oranının 0.675' ten 0.767' ye arttığı görülmüştür. Modelin yapısı aşağıdaki gibidir:

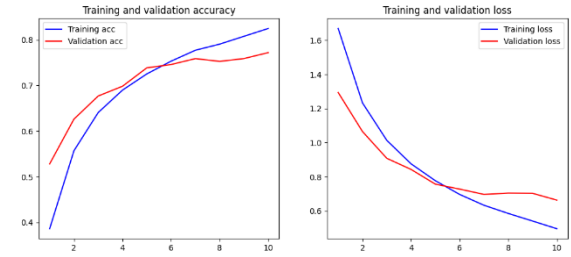


Modelin karmaşıklık matrisi ise aşağıdaki gibidir:



Karışıklık matrisine genel olarak bakıldığında bir önceki karışıklık matrisinden daha az hatalı tahmin olduğu göze çarpmaktadır.

Aşağıda modelin epoch boyunca eğitim ve validasyon doğrulukları ve kayıpları yer almaktadır.



Grafiğe bakıldığında doğruluk ve kayıplarda eğitim ve validasyon için korelasyon olduğu, modelin öğrenmeye devam ettiği görülmektedir.

Sonuç

Hiper parametrelerin başarı üzerindeki etkisine baktığımızda kernel boyutunun 3x3 seçildiğinde ve dropout içeren modellerin daha başarılı sonuçlar ürettiği görülmüştür. Kernel 3x3 seçildiğinde karmaşık özellik çıkarımının etkisinin daha fazla olduğu görülmüştür. Dropout olsa bile modellerin aşırı öğrenme(overfit) durumunda olduğu ve bunun nedeninin ise modellerin fazla karmaşık yapıya sahip olmasından kaynaklandığı görülmüştür.

En iyi başarıyı 5 tane konvolüsyon katmanı içeren ve her katmanda 64 adet filtreye ve 3x3 kernel boyutuna sahip, ağda dropout içeren model olduğu görülmüştür. Başarı oranı ise 0.675 bulunmuştur. Modele Max-Pooling eklendiğinde parametre sayısının 8,5 milyondan 225 bine düşmesine karşın başarı oranının 0.767' ye çıktığı ve 10 epochta overfit durumunda olmadığı öğrenmeye devam ettiği görülmüştür. Max-Pooling yönteminin bunu sağlamasını sağlayan şey ise etkili piksellerin seçilmesini sağlamasıdır. Hem hesaplama maliyetini düşürmüş hem de etkisiz pikselleri ortadan kaldırarak overfit durumunu önlemiştir.